

# Das **etoc**-Paket

JEAN-FRAN OIS BURNOL

jfbu (at) free (dot) fr  
Paketversion:  
v1.09b (17.11.2019)



Übersetzt ins Deutsche von Felix Baral-Weber, Jenny Rothkrämer-Vogt, Daniel Büttner, Claudia Dahl, Christian Otto und Christine Römer (FSU Jena). Die letzte Aktualisierung der Übersetzung ist vom 09.04.2015 (Paketversion v1.08d). Neuere Abschnitte bleiben auf English. Hinweise zur Übersetzung an Christine\_Roemer at t-online dot de.

Starting with 1.08h, about 25 code samples (especially the longer ones) are incorporated into the PDF file (thanks to SCOTT PAKIN's `attachfile` package) as *file attachement annotations*. Each file is represented by an icon after the verbatim rendering of the corresponding code. Clicking suitably on the icon will open a dialog to open or save the file contents. This is much better than copy-paste which loses indentation. There is also in some viewer a *Comment pane* which can be opened on the right side of the main window and which provides a convenient clickable list of all these *attachments annotations*.

PDF viewers unable to make these attachments accessible to the user may replace the clickable icons by some blank space.

## INHALTSVERZEICHNIS

<b>Zusammenfassung</b> . . . . .	<b>S. 6</b>
<b>Vorwort</b> . . . . .	<b>1, S. 6</b>
<b>Lizenz</b> . . . . .	<b>2, S. 7</b>
<b>Teil I. Überblick</b>	
<b>Muss ich Computerfreak sein, um etoc benutzen zu können?</b> . . . . .	<b>3, S. 8</b>
Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben.	
<b>Zeilenstile und Stile der Inhaltsangabe (toc display style)</b> . . . . .	<b>4, S. 9</b>
\etocsetstyle für den Zeilenstil – \etocsettocstyle für den Stil der Inhaltsangabe – Kompatibilismus.	
<b>Ein erstes Beispiel</b> . . . . .	<b>5, S. 10</b>
<b>Ein zweites Beispiel</b> . . . . .	<b>6, S. 12</b>
<b>Eine wunderschöne Beispielthese</b> . . . . .	<b>7, S. 14</b>
<b>Verlinkte Liste der Hauptbefehle</b> . . . . .	<b>8, S. 15</b>

Erzeugt von der Quelldatei mit Zeitstempel “Time-stamp: <17-11-2019 at 13:36:39 CET>”.

**Teil II. Beliebig viele TOCs und lokale Versionen davon**

<b>Die \tableofcontents-, \localtableofcontents- und \localtableofcontents-withrelativedepth -Befehle</b>	9, S. 16
<b>Beschriften und anderswo wieder benutzen</b>	10, S. 17
<b>Eine mächtige Funktionsweise von <b>etoc</b>: Die Wiederzuweisung von Levels mit von \etocsetlevel</b>	11, S. 18
<b>Die \etocsettocdepth- und \etocsetnexttocdepth-Befehle</b>	12, S. 19
Die hyperref-Option <i>bookmarksdepth</i> .	
<b>Der Befehl \etocsettocdepth.toc</b>	13, S. 21
Die Befehle \etocbeytoctocdepth und \etocignoretoctocdepth.	
<b>Die Befehle \etocdepthtag.toc und \etocsettagdepth</b>	14, S. 21
Die Befehle \etocbeydepthtags und \etocignoredepthtags.	
<b>Die Befehle \etocglobaldefs und \etoclocaldefs</b>	15, S. 22
<b>Not displayed empty TOCs</b>	16, S. 23
The \etocchecksemptiness command – The \etocnotocifnotoc command – The \etocifwasempty command.	
<b>The \etoclocaltop command</b>	17, S. 24
<b>Befehle zur .toc-Datei hinzufügen</b>	18, S. 25
Die hyperref-Option <i>hidelinks</i> .	

**Teil III. Beispiele**

<b>Überprüfen des Kompatibilitätsmodus</b>	19, S. 27
<b>Ein weiterer Kompatibilitätsmodus</b>	20, S. 28
<b>Emulating the book class</b>	21, S. 30
<b>Ein Inhaltsverzeichnis mit Rahmen</b>	22, S. 34
<b>Ein weiteres TOC mit einer Hintergrundfarbe</b>	23, S. 34
<b>Eine (verrückte) Inline-Darstellung</b>	24, S. 38

**Teil IV. Überraschende Anwendungen von **etoc****

<b>Das TOC für TOCs</b>	25, S. 40
<b>Willkürliche „Listen von ...“, \etoctoccontentsline</b>	26, S. 42
<b>Das TOC als ein Baum</b>	27, S. 43
<b>Das TOC als ein Molekül</b>	28, S. 46

## INHALTSVERZEICHNIS

<b>Das TOC als ein TikZ Mind-Map</b>	29, S. 49
<b>Das TOC als eine Tabelle</b>	30, S. 53
<b>A TOC self-adjusting widths for its typesetting</b>	31, S. 58

### Teil V. Befehle für die TOC-Linienstile

<b>Die \etocsetstyle-, \etocname- und \etocpage-Befehle</b>	32, S. 61
<b>Die \etocskipfirstprefix und \etociffirst Befehle</b>	33, S. 63
<b>Der \etocnumber-Befehl</b>	34, S. 63
<b>Der \etocifnumbered-Schalter</b>	35, S. 63
<b>Die Befehle \etocthename, \etocthenumber sowie \etocthechapter</b>	36, S. 64
<b>Der \etoclink-Befehl</b>	37, S. 64
<b>Die Befehle \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage und \etocthelink</b>	38, S. 65
<b>Die \etocsavedsectiontocline, usw... Befehle.</b>	39, S. 66
<b>Der \etocsetlevel-Befehl</b>	40, S. 66
<b>Umfang der zur .toc-Datei gehörigen Befehle</b>	41, S. 67
Testen des Bereichs – Dies ist eine (blass)rote Subsection zu Demonstrationszwecken.	
<b>Bin ich auch rot?</b>	42, S. 68

### Teil VI. Befehle für den toc-Display-Stil

<b>Festlegung des toc-Display-Stils</b>	43, S. 69
Der Befehl \etocsettocstyle – Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol – Der Befehl \etoc tocstyle – Die Befehle \etocruledstyle, \etocruled und \etoclocalruled – Die Befehle \etocframedstyle, \etocframed und \etoclocalframed – Überschriften, Titel, \etocoldpar, \etocinnertopsep – Der Kompatibilitätsmodus \etocstandarddisplaystyle – Der Befehl \etocinline.	
<b>Starred variants and hooks</b>	44, S. 73

### Teil VII. Eigene Stile von etoc nutzen und anpassen

<b>Überblick über die Grundbefehle zur Stilbearbeitung</b>	45, S. 74
Ein lokales Inhaltsverzeichnis anpassen – Das Aussehen eines Inhaltsverzeichnisses anpassen – Inhaltsverzeichnisse darstellen – Label und Referenzen.	
<b>Der Linienstandard des Pakets: \etocdefaultlines</b>	46, S. 76
<b>etoc individuell anpassen</b>	47, S. 80
Den durch etoc vordefinierten Linienstil individualisieren – Die Darstellung eines Inhaltsverzeichnisses anpassen.	

<b>Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis . . . . .</b>	<b>48, S. 82</b>
--	------------------

## Teil VIII. Tipps

<b>The \etocsetlocaltop.toc command . . . . .</b>	<b>49, S. 85</b>
<b>Hacken „framed parboxes“ . . . . .</b>	<b>50, S. 86</b>
<b>Umstellen der Ebenen . . . . .</b>	<b>51, S. 87</b>
Alle Unterabschnitte in diesem Dokument.	
<b>Statistiken anzeigen . . . . .</b>	<b>52, S. 88</b>
<b>Tiefe Tags nehmen . . . . .</b>	<b>53, S. 90</b>
<b>Das TOC als eine Tabelle setzend (die alte Weise) . . . . .</b>	<b>54, S. 92</b>

## Teil IX. etoc und die Außenwelt

<b>Kompatibilität mit anderen Paketen . . . . .</b>	<b>55, S. 95</b>
Generelle Informationen – Kompatibilität mit beamer – Kompatibilität mit Babel – Kompatibilität mit hyperref – Kompatibilität mit multicol – Kompatibilität mit tocloft – Kompatibilität mit der memoir-Klasse – Kompatibilität mit tocvssec2 – Kompatibilität mit tableof – Kompatibilität mit tocstyle.	
<b>TeX-nische Sachen . . . . .</b>	<b>56, S. 98</b>
<b>Fehler und Katastrophen . . . . .</b>	<b>57, S. 99</b>

## Teil X. Der Code

<b>Zeitstempel . . . . .</b>	<b>58, S. 100</b>
<b>Änderungen . . . . .</b>	<b>59, S. 100</b>
<b>Ausführung . . . . .</b>	<b>60, S. 103</b>

## 1. Vorwort

### Zusammenfassung

Das **etoc**-Paket gibt dem Nutzer lückenlose Möglichkeiten, die Einträge des Inhaltsverzeichnis individuell über *Name*, *Nummer*, und *Seitenzahl* für jede gliedernde Einheit zu setzen. Dieses wird durch die Definition von *Zeilenstilen* (*line-styles*) für jede Unterteilungsstufe des Dokuments realisiert. Das Paket bietet darüber hinaus seinen eigenen Zeilenstile. Einfachere werden als Beispiele in dieser Dokumentation angeboten. Die einfachste Nutzung wird durch die Verwendung der Layout-Hilfsmittel, von Paketen die sich mit Listen-Umgebungen beschäftigen, gewährt.

In Hinsicht auf das *globale Inhaltsverzeichnis* bietet **etoc** vordefinierte Stile, die auf der Mehrspaltenansicht mit optionalen, unterstrichenen Titeln oder eingerahmten Inhalten basieren.

Der \tableofcontents-Befehl kann beliebig mehrfach benutzt werden. Er hat die Variante \localtableofcontents, welche die Inhaltsangabe 'local' zur aktuellen Dokumenten-Sektion ausgibt. Eine Erweiterung der \label\ref-Syntax erlaubt die Reproduktion (mit anderem Layout) einer lokalen Inhaltsangabe, die irgendwo anders im Dokument definiert wurde.

Mit „tiefen Tags“ erhält man noch feinere Kontrolle über jede Inhaltsangabe und darüber, welche unterteilende Einheit angezeigt werden soll, oder nicht.

Mit *Kompatibilismus* wird die Formatierung der Dokumentklasse geerbt (die womöglich von anderen Paketen verändert wurde); und die Formatierung, die von der Dokumentklasse (die womöglich von anderen Paketen verändert wurde) geerbt wird.

Die Zuweisung von Ebenen zu den unterteilenden Einheiten kann zu jeder Zeit geändert werden und **etoc** kann dadurch in einer ziemlich generellen Weise genutzt werden, um selbsterstellte 'Listen von' zu erzeugen. Keine Zusatzdateien werden zusätzlich zu der Standard-.toc-Datei benutzt.

## 1. Vorwort

Andere populäre Pakete, die sich mit Inhaltsangaben befassen, sind **tocloft**, **titletoc** und **minitoc**. Warum also ein anderes? Ich habe **etoc** erst für mich selbst entwickelt und später von den anderen Paketen erfahren.

Wie es auch in der **tocloft**-Dokumentation erklärt wird, ist das Standard-L<sup>A</sup>T<sub>E</sub>X-Layout für Inhaltsangaben in den Klassen-Definitionen begraben. Im Besonderen sind die meisten Längen hart codiert und der einzige Weg für den Benutzer, sie zu ändern, ist, die Klasse in den Dokumenten-Ordner zu kopieren und dann die Längen zu ändern. Die vernünftigere Alternative ist die Benutzung eines auf das Veränderungsziel gerichteten Pakets, wie **tocloft** oder eine andere flexible Dokumentklasse.

Freilich, obwohl jetzt hoffentlich nichts mehr hart codiert ist, muss man noch immer das Paket oder die Klassen-Definitionen durchgehen. Das bedeutet, dass man sich viele (vermutlich große) Menge an Makros merken muss, die nur dieser Aufgabe dienen. Und man wird immer auf die Anpassung eines zu Anfang gegebenen Layouts eingeschränkt sein.

Der Sinn von **etoc** ist ein anderer. Der Nutzer wird sich mit *Name*, *Nummer* und der *Seitenzahl* beschäftigen. Diese korrespondieren mit jeder Dokumenten-Unterteilung (und können in einer Zeile der .toc-Datei gefunden werden). Sie können, wenn gewünscht, durch die Befehle \etocname, \etocnumber und \etocpage abgerufen werden.

**etoc** scheint (mindestens teilweise) mit den Dokument-Klassen **article**, **book**, **report**, **scrartcl**, **scrbook**, **scrreprt** und **memoir** kompatibel zu sein. Seit der Version 1.07k ist es möglich **etoc** gleichzeitig mit **tocloft** zu benutzen.

## 2. Lizenz

```
% Package: etoc
% Version: 1.09b-2019/11/17
% License: LPPL 1.3c
% Copyright (C) 2012-2019 Jean-Francois Burnol <jfbu at free dot fr>
% Copyright (C) 2014-2019 Christine Roemer <Christine_Roemer at t-online dot de>
% and collaborators for the translation into German of the documentation
%
% This Work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, in its
% version 1.3c. This version of this license is in
%     http://www.latex-project.org/lppl/lppl-1-3c.txt
% and the latest version of this license is in
%     http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of
% LaTeX version 2005/12/01 or later.
%
% The Authors of this Work are:
% Jean-Francois Burnol <jfbu at free dot fr>
%     for the source code and English documentation, and
% Christine Roemer <Christine_Roemer at t-online dot de> and collaborators
%     for the translation into German of the documentation.
%
% This Work consists of the main source file etoc.dtx and the derived
% files etoc.sty, etoc.tex, etoc-DE.tex, etoc.pdf, etoc-DE.pdf,
% etoc.dvi, etoc-DE.dvi, README.md.
%
% Running etex (or latex or pdflatex) on etoc.dtx extracts etoc.sty,
% etoc.tex, etoc-DE.tex, and README.md. See README.md for
% further instructions.
```

### 3. Muss ich Computerfreak sein, um **etoc** benutzen zu können?

## Teil I. Überblick

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 6 Abschnitte und 4 Unterabschnitte. Der Name des ersten Abschnitts lautet „Muss ich Computerfreak sein, um **etoc** benutzen zu können?“ und er hat die Gliederungsnummer „3“. Der Name des letzten Abschnitts lautet „Verlinkte Liste der Hauptbefehle“ und er hat die Nummer „8“. Der erste Unterabschnitt heißt „Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben“ und er hat die Nummer „3.1“. Der Name des letzten Unterabschnitts ist „Kompatibilismus“ und dessen Nummer ist „4.3“.

### 3. Muss ich Computerfreak sein, um **etoc** benutzen zu können?

Nicht wirklich. Der einfachste Weg ist die Benutzung von enumerate-Umgebungen mit den Anpassungsmöglichkeiten von Paketen wie `enumitem`,<sup>1</sup> um von **etoc** aus der `.toc`-Datei die abgerufenen Daten anzuzeigen. Die Daten bestehen aus *Name* (`\etocname`), *Gliederungsnummer* (`\etocnumber`) und *Seitenzahl* (`\etocpage`).

Dieses wird [zu Beginn](#) von [Teil V](#) gezeigt. Anspruchsvollere Beispiele würden anspruchsvollere `enumitem`-Optionen benutzen. Man könnte nun sagen, dass der Nutzer sich einige Anpassungen merken muss. Und das ist auch so, aber die Syntax und Optionsbezeichnungen, die es zu behalten gilt, sind in keiner Weise ähnlich.

Die Methode wird [später](#) ([Abschnitt 5](#)) erklärt. Dafür ist ein wenig Wissen über `\leftskip`, `\rightskip` usw. notwendig. Und eine Ausarbeitung, deren Code kann man hier ([Abschnitt 53](#)) finden, die Nachahmung des Standardaussehens einer Inhaltsangabe.

An even closer emulation of the book class design is now included in this documentation as [section 21](#).

Wie Sie beim Lesen dieser Anleitung noch sehen werden, ist **etoc** ziemlich vielfältig (ganz besonders, weil es erlaubt, die Hierarchie jeder unterteilenden Einheit des Dokuments zu re-definieren) und man kann damit überraschende Effekte erreichen: [Teil IV](#) handelt davon. Mehr davon gibt es noch in [Teil VIII](#).

Ein ganz besonders wichtiger Aspekt von **etoc** ist, dass es darauf ausgelegt ist, viele Inhaltsangaben in dem selben Dokument zu erzeugen, *Mit nur einer .toc-Datei*. Die gegenwärtige Dokumentation enthält 42 sichtbare Inhaltsangaben (und ein paar unsichtbare). Sie benutzt dabei nur eine `.toc`-Datei!<sup>2</sup> Darum sollte man zweimal nachdenken, bevor man manuell extra Befehle zu der `.toc`-Datei hinzufügt (siehe [Abschnitt 18](#)).

<sup>1</sup><http://ctan.org/pkg/enumitem>

<sup>2</sup>Das Selbstzählen wurde durch eine Inhaltsangabe erreicht, die in einen gerahmten Paragraphen eingesetzt wurde! ... Die Technik für diese Art von Effekt wird später erklärt.

### 3.1. Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben

#### 3.1. Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben

Es gibt ein paar Einschränkungen für das Nutzen von Listen-Umgebungen beim Setzen der TOCs. Eine davon liegt dem Geltungsbereichsbeschränkungen der Umgebungen inne. Die .toc-Datei könnte neben den Informationen zum Stil einige andere Befehle beinhalten, die Probleme hervorrufen können (L<sup>A</sup>T<sub>E</sub>X-Umgebungen schaffen Gruppen.).

Darum machen die eingebauten „Zeilenstile“ (line styles), wie sie von **etoc** vorgeschlagen werden (die durch die **Hauptinhaltssangabe** in diesem Dokument gezeigt werden) keinen Gebrauch von Umgebungen. Tatsächlich nutzt in dieser Gebrauchsanweisung nur die **Inhaltssangabe** zu Beginn von **Teil V**, die in **Unterabschnitt 51.1** und auch die in **Ab schnitt 38**, **enumerate-** oder **itemize-**Umgebungen, um Zeilenstile anzugeben.

## 4. Zeilenstile und Stile der Inhaltssangabe (toc display style)

Eine Unterscheidung muss zwischen den **Zeilenstilen**, z. B. der Art und Weise wie Namen, Kapitelnummern und Seitenzahlen auf jedem Level angezeigt werden und dem **Inhaltssangabenstil** (mangels einer besseren Bezeichnung) gemacht werden. Dieser sagt, wie die Titel gesetzt werden sollen, ob oder nicht ein Eintrag in die .toc-Datei gemacht werden soll, ob oder nicht der Inhalt mit mehreren Spalten gesetzt werden soll, usw..., der letztere wird von \etocsettocstyle (oder einem höherrangigen Befehl) geregelt. Das erste wird von \etocsetstyle initialisiert.

### 4.1. \etocsetstyle für den Zeilenstil

Das ist der Befehl, der **etoc** sagt, was es mit \etocname, \etocnumber, und \etocpage machen soll. Er hat fünf Pflichtargumente: Das erste ist der Name der unterteilenden Einheit: Im Vorhinein bekannte Namen sind book, part, chapter, section, subsection, subsubsection, paragraph, subparagraph und viele weitere können anhand dem \etocsetlevel-Befehl einem (numerischen) Level zugewiesen werden.<sup>3</sup>

Die vier anderen Argumente von \etocsetstyle spezifizieren: 1) *Was zu tun ist, wenn dieses Level zuerst nach einem höheren auftritt*, 2) & 3) (Zwei Argumente, ein 'Präfix' und ein 'Inhalt') *Was zu tun ist, wenn ein neuer Eintrag dieses Typs gefunden wird*, und 4) *Das letzte Argument ist der Code, der ausgeführt wird, wenn eine unterteilende Einheit von höherer Wichtigkeit wieder von oben getroffen wird*.

### 4.2. \etocsettocstyle für den Stil der Inhaltssangabe

Der \etocsettocstyle-Befehl erlaubt es festzulegen, was vor und nach den Zeileneinträgen eines Inhaltsverzeichnis passieren soll, und im Speziellen wie die Titel ausgegeben werden sollen. Er hat zwei Argumente. Das erste wird vor der Inhaltssangabe ausgeführt (typischerweise wird es Inhalte ausgeben und definiert passende Markierungen für die Seitenüberschriften). Der Zweite wird nach der Inhaltssangabe ausgeführt.

**etoc** bietet vier (anpassbare) höhere Inhaltsabgabenstile: \etocmulticolstyle, \etoc tocstyle, \etocruledstyle und \etocframedstyle. Alle nutzen das multicol-Paket

<sup>3</sup>Mit der memoir-Dokumentklasse kennt **etoc** Appendix als Kapitelüberschrift.

## 5. Ein erstes Beispiel

mit standardmäßig zwei Spalten (Einspaltenmodus kann durch das optionale Argument [1] erreicht werden).

Diese Befehle müssen entweder von `\tableofcontents` oder `\localtableofcontents` gefolgt sein.

### 4.3. Kompatibilismus

Vor den „Zeilenstilen“ und den „Inhaltsangabenstilen“ besteht die Möglichkeit in den Kompabilitätsmodus zu wechseln, der die Standardeinstellungen der Dokumentenklasse nimmt.<sup>4</sup> Dieses wird wie folgt aktiviert:

```
\etocstandardlines      % 'Zeileneinträge' wie ohne \usepackage{etoc}  
\etocstandarddisplaystyle % 'Inhaltsstil' wie ohne \usepackage{etoc}
```

Wenn der Befehl `\etocsetstyle` noch nicht in der Präambel benutzt worden ist, wird das Paket in `\begin{document}` sitzen (in diesem Kompatibilismus). Folglich sollte `\usepackage{etoc}` hoffentlich nichts am Aussehen eines im Vorhinein vorhandenen Dokuments der Klassen `article`, `book`, `report`, `scrartcl`, `scrbook`, `scrreprt` und `memoir` ändern.

Jede Benutzung von `\etocsetstyle` in der (Präambel des Textkörpers) des Dokuments schaltet den Kompabilitätsmodus für Zeilenstile aus, behält ihn aber für den Inhaltsabgabenstil.

Man kann ihn für Zeilenstile anhand von `\etocstandardlines` wieder aktivieren. `\etoctoclines` reaktiviert Zeilenstile wie Sie mit der Hilfe von `\etocsetstyle` definiert wurden, wenn deren Gültigkeitsbereich nicht auf eine Gruppe oder eine Umgebung beschränkt war.

Der Befehl `\etocdefaultlines` setzt den Zeilenstil darauf zurück, wie er intern von `etoc` in [Abschnitt 46](#) beschrieben wurde.

Sogar wenn `\etocsetstyle` benutzt wurde, bleibt der globale Stil zu Beginn, wie er in der Dokumentenklasse definiert wurde (oder von dem `tocloft` Paket). Man muss den Befehl `\etocsettocstyle` oder seine Varianten setzen, um diesem Kompatibilitätsmodus auf dem „Inhaltsangabenstil“-Level zu beenden. Er wird reaktiviert, wenn `\etocstandarddisplaystyle` benutzt wird.

## 5. Ein erstes Beispiel

Hier sehen sie ein erstes Beispiel zur Beschreibung der Zeilenform. Wir fügen die Quelldatei direkt nach dem Start von [Teil I](#) ein:

```
\invisiblelocaltableofcontents \label{toc:overview}
```

Dieses erstellt ein Label `toc:overview`, das wir von jeder Stelle im Dokument benutzen können:

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

Und da wir `\invisible...`,<sup>5</sup> benutzt haben, existiert die lokale Inhaltsangabe nur durch

---

<sup>4</sup>Für den „Inhaltsangabenstil“ überprüft `etoc`, ob es die Klassen kennt, wenn nicht, fällt es zur `article`-Klasse zurück. Es wird auch überprüfen, ob `tocloft` die Inhaltsangaben-Datei verändert hat.

<sup>5</sup>Das ist eine Abkürzung, um die `tocdepth` temporär auf -3 zu setzen. Dieses sagt `etoc`, die Inhaltsangabe und die Überschrift nicht auszugeben.

seine Klone anderswo im Dokument.

---

## My first **etoc**: TOC von Teil I (Überblick)

---

<b>3. Muss ich Computerfreak sein, um etoc benutzen zu können?</b>	8
3.1    Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben	9
<b>4. Zeilenstile und Stile der Inhaltsangabe (toc display style)</b>	9
4.1    \etocsetstyle für den Zeilenstil	9
4.2    \etocsettocstyle für den Stil der Inhaltsangabe	9
4.3    Kompatibilismus	10
<b>5. Ein erstes Beispiel</b>	10
<b>6. Ein zweites Beispiel</b>	12
<b>7. Eine wunderschöne Beispielthese</b>	14
<b>8. Verlinkte Liste der Hauptbefehle</b>	15

Anhand von `\etocdefaultlines` oder dem Stil der Standard-Dokumentklasse, anhand von `\etocstandardlines`, könnten wir auch die Zeilenform benutzen, die von **etoc** definiert wird. Hier waren wir jedoch ein wenig ehrgeiziger und wollten unser eigenen Stil anlegen. Die Technik ist einfach: Jede Überschrift ist an sich ein eigener Paragraph, der sich über mehrere Zeilen erstrecken kann; es ist für seinen `\leftskip` selbst verantwortlich.

Dieses ist ein einfaches Design, das keine Maßnahmen zum Zeilenumbruch bereitstellt. Diese sollten zwischen Abschnitten und Unterabschnitten, etc. unterbunden werden, da wir es nur für die Inhaltsangabe von diesem Abschnitt nutzen. Durch die Abschnitte als höchste Ebene entstand keine Notwendigkeit, einen Stil für den ganzen Teil zu erstellen. (Wir definierten einen Unterabschnitt-Zeilenstil, aber wie sich herausstellt, gibt es keine Unterabschnitte in diesem Teil). Die zwei Befehle, die benutzt wurden, spezifizieren den Zeilen-Stil (`\etocsetstyle`) und den globalen Inhaltsausgabestil (`\etocruledstyle`).

Das `\rightskip` wird von allen angewandt und schafft Platz, wo die Seitenzahlen gedruckt werden. Einzelheiten dieser Technik werden in dem [Abschnitt 6](#) und [Abschnitt 53](#) benutzt. Hier wird eine Inhaltsangabe mit Abschnitten und Paragraphen verwendet. Beide erlauben mehrzeilige Überschriften und nutzen einer Technik, die Seitenzahlen mit dem richtigen Seitenabstand auszugeben. Dieses wurde von der Funktionsweise von Makros aus L<sup>A</sup>T<sub>E</sub>X2e's `@dottedtocline` inspiriert.

Hier der Code:

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

## 6. Ein zweites Beispiel

```
\etocname\nobreak\hfill\nobreak
  \rlap{\makebox[1cm]{\etocpage}}\par}
{ }
\etocsetstyle {subsubsection}
{ }
{\leavevmode\leftskip 1.5cm\relax }
{\mdseries\normalsize\makebox[1cm][l]{\etocnumber}%
  \etocname\nobreak\hfill\nobreak
  \rlap{\makebox[1cm]{\etocpage}}\par}
{ }
\etocruledstyle[1]{\bfseries \Large My first \etoc: TOC von
  \autoref{part:overview} (\nameref{part:overview})}
\tableofcontents \ref{toc:overview}
\endgroup
```

## 6. Ein zweites Beispiel

Dieses zweite Beispiel zeigt nur den Inhalt von [Teil V](#) und [Teil VI](#). Diese Auswahl wird durch die Technik von *depth tags* realisiert, die in [Abschnitt 14](#) und [Abschnitt 53](#) beschrieben wird. Dessen Layout ist in etwa wie das von der [Hauptinhaltsangabe](#), obwohl die Zeilenstile ganz anders programmiert sind.

### INHALTSVERZEICHNIS

#### PART V – BEFEHLE FÜR DIE TOC-LINIENSTILE

32. DIE \ETOCSETSTYLE-, \ETOCNAME- UND \ETOCPAGE-BEFEHLE.....	61
33. DIE \ETOCSKIPFIRSTPREFIX UND \ETOCFIRST BEFEHLE .....	63
34. DER \ETOCNUMBER-BEFEHL.....	63
35. DER \ETOCIFNUMBERED-SCHALTER .....	63
36. DIE BEFEHLE \ETOCTHENAME, \ETOCTHENUMBER SOWIE \ETOCTHEPAGE .....	64
37. DER \ETOCLINK-BEFEHL.....	64
38. DIE BEFEHLE \ETOCTHELINKEDNAME, \ETOCTHELINKEDNUMBER, \ETOCTHELINKED-PAGE UND \ETOCTHELINK.....	65
39. DIE \ETOCSAVEDSECTIONTOCLINE, USW... BEFEHLE .....	66
40. DER \ETOCSETLEVEL-BEFEHL.....	66
41. UMFANG DER ZUR .TOC-DATEI GEHÖRIGEN BEFEHLE .....	67
<i>Testen des Bereichs (41.1, p. 67). Dies ist eine (blass)rote Subsection zu Demonstrationszwecken (41.2, p. 67).</i>	
42. BIN ICH AUCH ROT? .....	68

#### PART VI – BEFEHLE FÜR DEN TOC-DISPLAY-STIL

43. FESTLEGUNG DES TOC-DISPLAY-STILS.....	69
---	----

*Der Befehl \etocsettocstyle (43.1, p. 69). Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol (43.2, p. 70). Der Befehl \etoctocstyle (43.3, p. 71). Die*

*Befehle \etocruledstyle, \etocrule und \etoclocalruled (43.4, p. 71). Die Befehle \etocframedstyle, \etocframed und \etoclocalframed (43.5, p. 71). Überschriften, Titel, \etocoldpar, \etocinnertopsep (43.6, p. 72). Der Kompatibilitätsmodus \etocstandarddisplaystyle (43.7, p. 72). Der Befehl \etocinline (43.8, p. 73).*

#### 44. STARRED VARIANTS AND HOOKS..... 73

Der Code:

```
\begingroup
\newcommand*\{DotsAndPage}
{\nobreak\leaders\hbox{\bfseries\normalsize\hbox to .75ex {\hss.\hss}}%
 \hfill\nobreak
 \makebox[\rightskip][r]{\bfseries\normalsize\etocpage}\par}

\etocsetstyle {part}
{\parindent 0pt
 \nobreak
 \etocskipfirstprefix
 {\pagebreak[3]\bigskip
 \large\rmfamily\bfseries\scshape\centering
 \etocifnumbered{Part \etocnumber{} - }{}\etocname\par
 {}}

\etocsetstyle {section}
{\leftskip 0pt \rightskip .75cm \parfillskip-\rightskip
 \nobreak\medskip
 \etocskipfirstprefix
 {\leftskip 0pt \rightskip .75cm \parfillskip-\rightskip
 \pagebreak[1]\smallskip
 \normalsize\rmfamily\bfseries\scshape
 \etocnumber. \etocname\{DotsAndPage \}
 \parfillskip 0pt plus 1fil\relax }

\etocsetstyle {subsection}
{\leftskip1cm\rightskip .75cm \parfillskip 0pt plus 1fil\relax
 \nobreak\smallskip
 {}
 {\footnotesize\sffamily\mdseries\itshape
 \etocname{} (\etocnumber, p. \etocpage). }
 \par\medskip

\etocsettagdepth {preamble} {none}
%\etocsettagdepth {overview} {none}% not needed explicitly, keeps value
%\etocsettagdepth {arbitrarily}{none}
%\etocsettagdepth {examples} {none}
%\etocsettagdepth {surprising} {none}
\etocsettagdepth {linestyles} {subsection}
%\etocsettagdepth {globalcmds} {subsection}
\etocsettagdepth {custom} {none}
%\etocsettagdepth {tips} {none}
%\etocsettagdepth {etocandworld}{none}
\etocsettagdepth {code} {none}

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

## 7. Eine wunderschöne Beispielthese

Hier ist noch ein relativ einfaches Beispiel für die Verwendung der Paket-Funktionsweisen. Lassen Sie uns einige Linienstile einrichten. Wir wählen einen Stil für Abschnitte und Unterabschnitte, welche für die Abschnitte und Unterabschnitte in einem durchschnittlichen Memoirenumfang geeignet wären. Die Linienstilspezifikationen haben eine gewisse Redundanz für die Klarheit, und sie interessieren sich nicht dafür, was bei möglichen Seitenbrüchen zu tun ist. Auch sind sie nicht besorgt über die potentielle mehrspaltige Nutzung.

### \_\_\_\_\_ *My Beautiful Thesis*

<b>Chapter 3</b>	<b>Muss ich Computerfreak sein, um etoc benutzen zu können?</b>	<b>8</b>
3.1	<i>Beschränkungen in der Benutzung von Listen- Umgebungen für Inhaltsangaben</i>	9
<b>Chapter 4</b>	<b>Zeilenstile und Stile der Inhaltsangabe (toc display style)</b>	<b>9</b>
4.1	<i>\etocsetstyle für den Zeilenstil</i>	9
4.2	<i>\etocsettocstyle für den Stil der Inhaltsangabe</i>	9
4.3	<i>Kompatibilismus</i>	10
<b>Chapter 5</b>	<b>Ein erstes Beispiel</b>	<b>10</b>
<b>Chapter 6</b>	<b>Ein zweites Beispiel</b>	<b>12</b>
<b>Chapter 7</b>	<b>Eine wunderschöne Beispielthese</b>	<b>14</b>
<b>Chapter 8</b>	<b>Verlinkte Liste der Hauptbefehle</b>	<b>15</b>

```
\begingroup % wir starten, um eine Gruppe zu behalten
\newlength{\tocleftmargin} \setlength{\tocleftmargin}{5cm}
```

```

\newlength{\tocrightmargin} \setlength{\tocrightmargin}{1cm}

\etocsetstyle{section} % wird vorgegeben, ein Kapitel zu sein
{\addvspace{1ex}\parfillskip0pt
 \leftskip\tocleftmargin % (bereits im Titel getan)
 \rightskip\the\tocrightmargin plus 1fil
 \parindent0pt\color{cyan} % (bereits getan)
 {\bfseries\LARGE\upshape\addvspace{1ex}\leavevmode}
 {\llap{Chapter\hspace{.5em}}\etocnumber\hspace{.75cm}}\etocname
 \hfill\makebox[-\tocrightmargin][1]{\makebox[0pt]{\etocpage}}\par
{}}

\etocsetstyle{subsection} % wird vorgegeben, ein Abschnitt zu sein
{}
{\mdseries\large\addvspace{.5ex}\leavevmode
 {\llap{\etocnumber\hspace{.75cm}}\textit{\etocname}%
 \hfill\makebox[-\tocrightmargin][1]{\makebox[0pt]{\etocpage}}\par
{}}

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

\tableofcontents \ref{toc:overview}
\endgroup

```

## 8. Verlinkte Liste der Hauptbefehle

\etocaftercontentshook	\etoclocalmulticol	\etocsettocstyle
\etocaftertitlehook	\etoclocalruled	\etocskipfirstprefix
\etocaftertochook	\etocmulticolstyle	\etocstandarddisplaystyle
\etocbeforetitlehook	\etocmulticol	\etocstandardlines
\etocdefaultlines	\etocname	\etocthelinkedname
\etocdepthtag.toc	\etocnumber	\etocthelinkednumber
\etocdisplay	\etocbeydepthtags	\etocthelinkedpage
\etocframedstyle	\etocbeytoccdepth	\etoctheblink
\etocframed	\etocpage	\etocthename
\etociffirst	\etocruledstyle	\etocthenumber
\etocifnumbered	\etocruled	\etocthepage
\etocignoredepthtags	\etocsetlevel	\etoccontentsline
\etocignoretoccdepth	\etocsetnexttoccdepth	\etoclines
\etocinline	\etocsetstyle	\localtableofcontents
\etoclink	\etocsettagdepth	\localtableofcontentswithrelativedepth
\etoclocaltop	\etocsettocdepth.toc	\tableofcontents
\etoclocalframed	\etocsettocdepth	

## Teil II.

# Beliebig viele TOCs und lokale Versionen davon

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 10 Abschnitte und 7 Unterabschnitte. Der Name des ersten Abschnitts lautet „Die \tableofcontents-, \localtableofcontents- und \localtableofcontentswithrelativedepth -Befehle“ und er hat die Gliederungsnummer „9“. Der Name des letzten Abschnitts lautet „Befehle zur .toc-Datei hinzufügen“ und er hat die Nummer „18“. Der erste Unterabschnitt heißt „Die hyperref-Option *bookmarksdepth*“ und er hat die Nummer „12.1“. Der Name des letzten Unterabschnitts ist „Die hyperref-Option *hidelinks*“ und dessen Nummer ist „18.1“.

<b>9. Die \tableofcontents-, \localtableofcontents- und \localtableofcontentswithrelativedepth -Befehle</b>	<b>16</b>
<b>10. Beschriften und anderswo wieder benutzen</b>	<b>17</b>
<b>11. Eine mächtige Funktionsweise von <b>etoc</b>: Die Wiederzuweisung von Levels mit von \etocsetlevel</b>	<b>18</b>
<b>12. Die \etocsettocdepth- und \etocsetnexttocdepth-Befehle</b>	<b>19</b>
12.1. Die hyperref-Option <i>bookmarksdepth</i> . . . . .	20
<b>13. Der Befehl \etocsettocdepth.toc</b>	<b>21</b>
13.1. Die Befehle \etocbeytoctocdepth und \etocignoretoctocdepth . . .	21
<b>14. Die Befehle \etocdepthtag.toc und \etocsettagdepth</b>	<b>21</b>
14.1. Die Befehle \etocbeydepthtags und \etocignoredepthtags . . . .	22
<b>15. Die Befehle \etocglobaldefs und \etoclocaldefs</b>	<b>22</b>
<b>16. Not displayed empty TOCs</b>	<b>23</b>
16.1. The \etoccheckemptiness command . . . . .	23
16.2. The \etocnotocifnotoc command . . . . .	23
16.3. The \etocifwasempty command . . . . .	23
<b>17. The \etoclocaltop command</b>	<b>24</b>
<b>18. Befehle zur .toc-Datei hinzufügen</b>	<b>25</b>
18.1. Die hyperref-Option <i>hidelinks</i> . . . . .	25
<b>9. Die \tableofcontents-, \localtableofcontents- und \localtableofcontentswithrelativedepth -Befehle</b>	

**\tableofcontents** **etoc** erlaubt beliebig viele **\tableofcontents**-Befehle in ihrem Dokument. Die Zeilen- und Inhaltsangabenstile können natürlich dazwischen geändert werden.

**\localtableofcontents** Außerdem setzt **\localtableofcontents** lokale Inhaltsangaben: z. B. alle Abschnitte und Unterabschnitte innerhalb eines bestimmten Kapitels oder alle Unterabschnitte und tiefer innerhalb eines gegebenen Unterabschnitt usw.<sup>67</sup>

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

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

will create a local table of contents taking into account the subsections and subsubsections inside this section, independently of what is the value of the tocdepth counter at this position in the document. If the numeric argument had been 3, the local TOC would have displayed also paragraphs. If the section had been a chapter, and again for a relative tocdepth of 2, the taken into account levels would have been sections and subsections.<sup>8</sup>

The macro **\localtableofcontentswithrelativedepth** is **new with 1.09**.<sup>9</sup>

## 10. Beschriften und anderswo wieder benutzen

**etoc** erlaubt das Labeln von Inhaltsangaben, beispielsweise mit **\label{toc:A}**, und wird es woanders **\tableofcontents\ref{toc:A}**, wenn gewollt, wieder anzeigen. Das eigentliche Layout (inklusive Titel) wird lokal bestimmt. Die Zeilen- und Inhaltsangabenstile (inklusive Titel) sind die aktuellen und der Wert von tocdepth leistet Folge. Hier ein Beispiel: Die Inhaltsangabe von **Teil VII** ist innerhalb eines **float**, der auf dieser Seite erscheint: on the following page.

Wir benutzen dies:

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

<sup>6</sup>As is explained in [section 10](#) the syntax allows to create somewhere a local table of contents and to display it at some other location either before or after its origin.

<sup>7</sup>As is explained in [section 11](#) **etoc** allows at anytime to locally redefine the numeric levels associated to named ones, which brings great flexibility to achieve special effects, all done using only a single auxiliary file, the standard **.toc** file.

<sup>8</sup>The situation may be more complex, as the **.toc** file itself may well contain **\setcounter{tocdepth}{...}** commands. This is not recommended practice with **etoc**. See the discussion of **\etocsettocdepth.toc** and **\etocdepthtag.toc** for more on this.

<sup>9</sup>Thanks to Tony ROBERTS for feature request.

## 11. Eine mächtige Funktionsweise von **etoc**: Die Wiederzuweisung von Levels mit von \etocsetlevel

I am from far away

<b>45. Überblick über die Grundbefehle zur Stilbearbeitung</b>	<b>74</b>
45.1. Ein lokales Inhaltsverzeichnis anpassen . . . . .	74
45.2. Das Aussehen eines Inhaltsverzeichnisses anpassen . . . . .	75
45.3. Inhaltsverzeichnisse darstellen . . . . .	75
45.4. Label und Referenzen . . . . .	75
<b>46. Der Linienstandard des Pakets: \etocdefaultlines</b>	<b>76</b>
<b>47. etoc individuell anpassen</b>	<b>80</b>
47.1. Den durch <b>etoc</b> vordefinierten Linienstil individualisieren . . . . .	80
47.2. Die Darstellung eines Inhaltsverzeichnisses anpassen . . . . .	81
<b>48. Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis</b>	<b>82</b>

```
\fboxsep1ex
\etocframedstyle [1]{\fbox{\makebox[.5\linewidth]{\etocfontminusone
    \hyperref[toc:c]{I am from far away}}}}
\tableofcontents \label{toc:d} \ref{toc:c}
\endgroup
\end{figure}
```

Wichtig: Man sollte `\tableofcontents \ref{toc:d}` nicht woanders benutzen. Um wieder zu Klonen, muss man eine Referenz zum Original-Label benutzen: `\tableofcontents \ref{toc:c}`.

## 11. Eine mächtige Funktionsweise von **etoc**: Die Wiederzuweisung von Levels mit von \etocsetlevel

Die intrinsischen Level, die durch **etoc** manipuliert wurden, sind numerisch: von -2 (welches `book` aus der `memoir`-Klasse entspricht) bis hinab zu 5 (`subparagraph`). Jedoch kann die Zuweisung eines numerischen Level zu einem gegebenen Namen zu jeder Zeit mit dem Befehl `\etocsetlevel{<level_name>} {<number>}` abgeändert werden. Daraus und aus der Benutzung des L<sup>A</sup>T<sub>E</sub>X-tocdepth-Zählers resultieren mächtige Anwendungszwecke: `<level_name>` muss sich nicht mit dem eigentlichen Unterteilungsbefehls des Dokuments überschneiden. Außerdem kann **etoc** genutzt werden, um beliebige „Listen von Dingen“ ohne weitere Hilfsdateien, außer der `.toc`-Datei, zu erstellen. Dieses wird in [Teil IV](#) weiter erklärt.

Es wird oft gesagt, dass Unterteilungslevel von `\part` in den Standardklassen, die keinen `\chapter`-Befehl haben, gleich 0 sind, und in denen mit `\chapter`-Befehl gleich -1 ist. Das ist auch *richtig*, wenn man die *automatische Nummerierung*, gesteuert durch den Wert des `secnumdepth`-Zählers, betrachtet. Aber es ist *falsch*, wenn

man den Effekt des `tocdepth`-Zählers sieht. Setzt man `tocdepth` in der `article`-Klasse direkt vor `\tableofcontents` auf -1, verhindert das *nicht*, dass Abschnitte in der Inhaltsangabe aufzutauchen. Man muss es dafür auf -2 stellen, ganz egal ob in der `article`- oder der `book`-Klasse.

Die kanonischen Level, grundsätzlich bekannt aus `etoc`, sind von Relevanz für den `tocdepth`-Zähler in den Standardklassen und werden in der folgenden Tabelle rekapituliert:

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

Mit `etoc` kann der Nutzer innerhalb einem gegebenen Unterabschnitt einfach lokale Inhaltsangaben ausgeben. Darin werden, wenn gewollt, Unterabschnitte in dem Stil von Abschnitten, Paragraphen in dem Stil von Unterabschnitten und Unterparagraphen im Stil von Unter-Unterabschnitten gedruckt. Demnach kann man entscheiden alles auf das Level 6 zu setzen (wird von `etoc` nie angezeigt), außer z. B. Paragraphen, die auf Level 1 befördert werden. So erstellt man eine schöne Inhaltsangabe aller Paragraphen eines Dokuments. (`tocdepth` mindestens 1)<sup>10</sup>

## 12. Die `\etocsettocdepth`- und `\etocsetnexttocdepth`-Befehle

Der `tocdepth`-Zähler hat keinen Einfluss darauf, was in die `.toc`-Datei geschrieben wird; dessen Einfluss ist ausschließlich auf das Setzen der Inhaltsangabe gerichtet.<sup>11</sup> In der Standardklasse ist nur ein `\tableofcontents` möglich, wobei mit `etoc` beliebig viele erlaubt sind. Dadurch könnte man `tocdepth` immer und immer wieder auf einen passenden Wert ändern (der entscheidet, wie tief die Abschnitte angezeigt werden sollen) bis die Inhaltsangabe gesetzt werden muss.

`etoc` bietet `\etocsettocdepth{<level>}` an, wessen verpflichtendes Argument entweder (von -3 bis 5) oder einen Unterteilungstyp wie `subsection` oder `subsubsection` oder alle anderen, die davor in `etoc` mit `\etocsetlevel` (die Schlüsselworte `all` und `none` erkannt werden, allerdings nicht in Zusammenhang mit der Dokumentenunterteilung) deklariert wurden. Das tut das passende `\setcounter{tocdepth}{numeric_level}`.

Wie in dem nächsten Unterabschnitt erklärt wird, wird `tocdepth` von `hyperref` benutzt. Man muss die folgenden Schritte befolgen, um den Einfluss von `hyperref`'s Än-

<sup>10</sup>Normalerweise sollte man diese Inhaltsangabe der Paragraphen nicht im Kompabilitätsmodus ausgeben. Der würde darauf bestehen, einen gigantischen Abstand nach rechts einzufügen.

<sup>11</sup>In the standard classes (at least), it also influences the `\listoftables` and `\listoffigures`, via `\@dottedtocline`.

## 12. Die `\etocsettocdepth`- und `\etocsetnexttocdepth`-Befehle

derungen auf die Lesezeichen zu verhindern. **etoc** hat `\etocsetnexttocdepth{<level>}`. Dessen Einfluss endet direkt nach der nächsten Inhaltsangabe. Daraus folgt, dass `\invisitableofcontents` essenziell für `\etocsetnexttocdepth{none}\tableofcontents` ist.

Der einfachste Weg ist also, eine einfache Instanz des `\etocsettocdepth`-Befehl nach `\begin{document}` und vor dem ersten `\tableofcontents` mit dem tiefsten Level als Argument zu setzen. – Dieses dann lokal zu benutzen, wo es gebraucht wird, anhand von `\etocsetnexttocdepth` vor `\tableofcontents` oder `\localtableofcontents`.

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

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

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

### 12.1. Die hyperref-Option `bookmarksdepth`

Wenn man den Zähler `tocdepth` verändert, um `\tableofcontents` oder `\localtableofcontents` mehrfach zu benutzen, sollte man gewahr sein, dass das Paket `hyperref` standardmäßig den Wert des *aktuellen* Wertes der `tocdepth`-Zählers berücksichtigt, um zu entscheiden, ob die pdf-Datei Lesezeichen, korrespondierend mit den Unterteilungsbefehlen, die in der Quelldatei vorgefunden wurde, anzeigt. Dadurch muss man typischer Weise den `tocdepth`-Zähler auf seinen vorigen Wert zurücksetzen, nachdem man ihn temporär für eine gegebene Inhaltsangabe modifiziert hat.

Alternativ gibt es die `bookmarksdepth=n`-Option des `hyperref`-Pakets. Dabei ist *n* die gewünschte maximale Lesezeichen-Tiefe. Diese kann numerisch sein oder der Name eines Levels, das `hyperref` bekannt ist.

Wir sind in dieser Dokumentation schon an `bookmarksdepth=3` vorbeigekommen. Es war eine Option von `hyperref`. Auch wenn `tocdepth` nach dem Ausgeben einer Inhaltsangabe aus Nachlässigkeit auf 1 stehen gelassen wurde, so wurden der Lesezeichenbaum der pdf-Datei nicht verändert. Da `\etocsetnexttocdepth` jetzt zu dem Paket hinzugefügt wurde, ist keine Verwendung mehr für `bookmarksdepth=3`.

## 13. Der Befehl \etocsettotoctocdepth.toc

Dieser Befehl `\etocsettotoctocdepth.toc` implementiert einige Funktionalitäten von Paket `tocvsec2`<sup>12</sup>. Dieses Paket war inkompatibel mit `etoc`. Es kann nun noch für seine mit `secnumdepth` in Verbindung stehenden Befehle genutzt werden, alle mit `toc` in Verbindung stehenden Aktivitäten werden jedoch durch `etoc` abgebrochen.

Das Handeln von `\etocsettotoctocdepth.toc` ist grundlegend verschieden zu `\etocsettocdepth`. Anstatt den `tocdepth`-Zähler direkt zu verändern, fügt es der `.toc`-Datei eine Linie an, die, wenn ausgeführt, in einer Inhaltsangabe diese Veränderung ausführen wird.

Der Befehl `\etocsettotoctocdepth.toc` akzeptiert, wie `\etocsettotoctocdepth`, Namen und numerische Argumente. In dem Fall eines Namens als Argument ist der tatsächliche numerische Wert noch nicht festgelegt, wenn `.toc`-Datei erstellt wird. Es wird der Wert sein, der aktuell für das benannte Level der Inhaltsangabe (ohne `\etocignoretoc tocdepth` gemacht zu haben) bei jeder Erstellung einer Inhaltsangabe spezifiziert wird.

Der `tocdepth`-Zähler wird niemals auf einen Wert gesetzt, der feiner als der ist, mit dem er zu Beginn der Inhaltsangabe initialisiert wurde. Also ist das Hinzufügen von `\etocsettocdepth.toc` im Dokument ein Weg die Tiefe einer lokalen Inhaltsangabe einzuschränken. Zum Beispiel, um die Einfügung von Unter-unter-Abschnitten oder eines gegebenen Kapitels zu unterbinden.

Dieses wird in allen Inhaltsangaben ausgeführt.

### 13.1. Die Befehle \etocbeytoctocdepth und \etocignoretoc tocdepth

Also wird `\etocignoretoc tocdepth` gesetzt, um den `\etocsettotoctocdepth.toc`-Mechanismus abzubrechen, wenn benötigt; und `\etocbeytoctocdepth` wird es reaktivieren. Das Paket führt `\etocbeytoctocdepth` zu Beginn aus.

## 14. Die Befehle \etocdepthtag.toc und \etocsettagdepth

Wie oben erwähnt, wenn das Argument für `\etocsettotoctocdepth.toc` kein numerisches sondern ein Namen-Level ist, wird durch die tatsächliche Nummer bei der Erstellung der Inhaltsangabe entschieden. Dadurch und anhand der *Schiebe-Mechanismen* von `etoc` kann man einen sehr flexiblen Aufbau erreichen: Man erklärt mit `\etocsetlevel` zuerst ein paar Platzhaltelevel, z. B. `partIII`, und weist ihm das unsichtbare Level 6 zu. Dann, vor dem dritten `\part` Befehl, fügt man `\etocsettotoctocdepth.toc{partIII}` ein. An dem Ort, an dem der Effekt enden soll, setzt man `\etocsettotoctocdepth.toc{all}` ein. Es ist nun möglich, jedes Mal, wenn man eine Inhaltsangabe setzen will, das feinste Level der Einträge aus dem dritten Teil zu spezifizieren. Man muss z. B. nur `\etocsetlevel{partIII}{1}` schreiben. Es wird die Abschnitte limitieren (sofern `tocdepth` mindestens auf diesem Level war, wenn die Inhaltsangabe ausgegeben wurde).

Die Methode hat ein paar Beschränkungen: (i) Das zweite Argument von `\etocsetlevel` muss numerisch sein; (ii) es kann nicht auf -3 gesetzt werden (was nötig wäre, wenn man ganze Bücher aus der Inhaltsangabe streichen wollte in den Klasse `memoir`);

<sup>12</sup><http://ctan.org/pkg/tocvsec2>. Ich danke Denis Broutré meine Aufmerksamkeit auf die Inkompatibilität dieses Pakets mit `etoc` gerichtet zu haben.

## 15. Die Befehle \etocglobaldefs und \etoclocaldefs

(iii) es ist ein bisschen getrickst, seit partIII kein richtiger unterteilender Abschnitt ist sondern nur ein *tag*.

Version 1.07h enthält den Befehl `\etocdepthtag.toc`, der es einfacher macht, diese variable lokale Kontrolle des tocdepth zu erreichen, wenn die Inhaltsangabe erstellt wird.

Es wird als `\etocdepthtag.toc{\<tag_name>}` genutzt, wo der `\<tag_name>` irgendetwas ist, das wird den Tag in die .toc-Datei setzen. Wenn die Inhaltsangabe erstellt wird, initiiert das eine Reihe von Befehlen `\etocsettagdepth{\<tag_name>}{\<level>}`, wo das `\<level>` entweder numerisch (von -3 bis 5) oder ein Name einer unterteilenden Einheit, die **etoc** bekannt ist, oder `none` oder `all`. Der Effekt von den Tags innerhalb der .toc-Datei ist es, den tocdepth-Zählers auf den gewünschten Wert zu setzen (das kann wie schon erwähnt, nicht feiner sein als der initiale Wert des tocdepth-Zählers).

Die hinzugefügte Flexibilität wirkt sich insofern aus, dass `\etocsetlevel` nicht mehr mit dieser getricksten Methode, bei der man Level-Tiefen und `none` und `all` benutzt hat.

Wie immer, wenn die Tag-Tiefen gesetzt wurden, bleiben sie wirkend, bis sie wieder definiert werden oder ihr Wirkungsgebiet beendet wird, indem die umschließende Gruppe geschlossen wird. Für ein Beispiel, siehe hier: [Abschnitt 53](#).

### 14.1. Die Befehle \etocbeydepthtags und \etocignoredepthtags

Nach `\etocignoredepthtags` werden die .toc-Tiefentags ignoriert (nach `\etocdepthtag.toc` funktionieren sie jedoch noch). Das Paket führt zu Beginn `\etocbeydepthtags` aus. Dieses befiehlt **etoc** auf die gefundenen Tags in der .toc-Datei zu reagieren.

## 15. Die Befehle \etocglobaldefs und \etoclocaldefs

In L<sup>A</sup>T<sub>E</sub>X verschwindet die Bedeutung eines Befehls, definiert via `\newcommand\foo{...}` in einer Umgebung (oder Gruppe), vom T<sub>E</sub>X-Speicher beim Austritt aus dieser Umgebung (oder Gruppe). Manchmal ist es jedoch erforderlich, Definitionen mit globaler Reichweite zu geben, für diese hat T<sub>E</sub>X primitive Präfixe `\global`.

Standardmäßig sind die **etoc**-Definitionen von `\etocname` etc. ... lokal, Dies verursacht Probleme in bestimmten Zusammenhängen: bei TOCs als Tabellen ([Abschnitt 30](#), [Abschnitt 54](#)), oder auch mit `enumitem`-*inline*-Varianten der Standardumgebungen, weil der Befehl `\item` dann eine Gruppe schließt (siehe [Abschnitt 38](#)).

Nach `\etocglobaldefs` wird **etoc** seine Definitionen von `\etocname` etc. ... vornehmen, sie haben eine globale Reichweite. Für den normalen Gebrauch ist das nicht notwendig. Es tut nicht weh, beides systematisch zu aktivieren.

Um zur Standardeinstellung zurückzukehren, nehmen Sie `\etoclocaldefs`. Beachten Sie, dass beide, `\etocglobaldefs` und `\etoclocaldefs`, lokal für die Umgebung oder Gruppe sind, in der sie verwendet werden.

## 16. Not displayed empty TOCs

### 16.1. The `\etoccheckemptiness` command

The user needs to issue `\etoccheckemptiness` to tell `etoc` to check whether local tables of contents are empty and in case of emptiness to print nothing at all.<sup>13</sup> This can be useful to authors of L<sup>A</sup>T<sub>E</sub>X classes who for example wish to have a `\chapter` command doing systematically a `\localtableofcontents`, or for people producing files via automatic conversions and some of those might have sectioning commands and others not.

«Emptiness» means that no `\contentsline` command would get executed within the scope of the local table of contents — empty line styles by themselves do not make the TOC empty. `etoc` always executes the `\etocaftertohook` command; and the test for emptiness itself executes everything else found in the `.toc` file. See [section 18](#) in this context.

1. the `\etocifwasempty` command discussed below can be used from inside `\etocaftertohook`.
2. there is no `\etocdoesnotcheckemptiness` but if there was one it would do `\makeatletter\Etoc@checkemptinessfalse\makeatother`.

The suppression of the heading (more precisely of the toc display style elements) may be effective only for the final L<sup>A</sup>T<sub>E</sub>X runs. For example in the situation of a `\tableofcontents\ref{foo}` where the label `foo` is not yet recognized, the heading (but not the contents) is printed and the TOC is declared non-empty. Or, if one adds a `\localtableofcontents` to a document, on the next run, the test for emptiness will in fact apply to the next one, and the last local TOC of the document will have its contents temporarily unknown to `etoc`, hence will be declared non empty, and the heading will be printed.

For a finalized document compiled with initially no auxiliary files, the first L<sup>A</sup>T<sub>E</sub>X run will declare all local TOCs non empty and print for each of them a heading (and no contents naturally). The second L<sup>A</sup>T<sub>E</sub>X run will then correctly decide which local TOC is empty or not.

### 16.2. The `\etocnotocifnotoc` command

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

### 16.3. The `\etocifwasempty` command

The command `\etocifwasempty{\langle YES \rangle}{\langle NO \rangle}` executes `\langle YES \rangle` if the previous TOC was found to be empty and `\langle NO \rangle` if its was not so. This may serve to act appropriately after a truly empty TOC. If `\etoccheckemptiness` has not been issued, this conditional always executes the `\langle NO \rangle` branch.

This command is robust, and `\etocxifwasempty` is its expandable version.

---

<sup>13</sup>Thanks to Paul Gaborit who asked for such a feature.

## 17. The `\etoclocaltop` command

Do not forget the second argument: at least an empty pair of braces must be present.

This conditional may wrongly say that the local TOC is empty or not empty until L<sup>A</sup>T<sub>E</sub>X compilations stabilize. But if it says that a local TOC is empty, this does mean that **etoc** considered the just encountered local table of contents to be empty (for that run) and thus printed nothing (not even a `\par`).

## 17. The `\etoclocaltop` command

New with 1.09 (2019/03/09).

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

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

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

**-3** (-\thr@@)

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

**-1000** (-\@m)

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

**-10000** (-\@M)

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

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

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

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

## 18. Befehle zur .toc-Datei hinzufügen

Wir beschrieben oben die Befehle `\etocsettocdepth.toc` und `\etocdepthtag.toc`, welche beide Befehle in die .toc-Datei einfügen. Ein noch allgemeinerer Weg, um „action tags“ in die .toc-Datei einzufügen, wäre vorstellbar, allerdings würde auch nur indirekt `\addtocontents{toc}{\something}` benutzt werden.

Man sollte vorsichtig sein, wenn man auf diesem Weg Dinge zur .toc-Datei hinzufügt. Zum Beispiel das Einfügen von `\addtocontents{toc}{\string\clearpage}` direkt vor einem `\part` behebt den Fehler, dass Teileinträge in Inhaltsangaben vom Rest getrennt am Ende der Seite auftauchen, kann aber auch Probleme mit mehreren Inhaltsangaben haben: `\clearpage` wird jedes Mal ausgeführt, wenn `etoc` über ein `\tableofcontents` oder `\localtableofcontents` stolpert. Die schlauere Methode wäre es: `\addtocontents{toc}{\string\myclearpage}`, den `\let\myclearpage\relax`, man hätte den Befehl `\let\myclearpage\relax` zu Beginn des Dokuments und wenn benötigt, benutzt man etwas wie:

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

Die `memoir`-Klasse besitzt den Befehl `\settocdepth`, der einen `\changetocdepth`-Befehl in die .toc-Datei setzt. Dieses beeinflusst die Schriftsetzung von `etoc` bei *allen* Inhaltsangaben mit möglichen unvorhersehbaren Ergebnissen: Stellen Sie sich vor, irgendwo im Dokument stünde `\settocdepth{chapter}`, um die Abschnitte von Unterabschnitten nicht in der Hauptinhaltsangabe aufzulisten. In diesem Fall würde eine lokale Inhaltsangabe in einem dieser Kapitel einen Titel ohne jeden Eintrag ausgeben.

Da die `memoir`-Klasse von Haus aus schon mehrfache `\tableofcontents` erlaubt, tritt dieser Fehler schon viel früher, und unabhängig von `etoc`, auf. Siehe Seite 170 der `memoir`-Anleitung.

Für diesen Spezialfall gibt es folgende Befehle: `\etocsettocdepth.toc`, `\etocignoretocdepth`, `\etocbeyetocdepth` oder deren Varianten `\etocdepthtag.toc` und `\etocsettagdepth`.

Nebenbei, jeder `\setcounter{tocdepth}{n}`-Befehl fügt direkt in die .toc-Datei hinzu. Es scheint so, dass seine Wirkung ist (seit der Version 1.07g), beendet den Abschluss jedes Inhaltsverzeichnis zu beenden, automatisch wird ein `\setcounter{tocdepth}{previous_value}` genommen, mit dem Wert, aktiv ein Inhaltsverzeichnis zu nehmen.

### 18.1. Die hyperref-Option `hidelinks`

Die farbigen Links (und auch die Kastenlinks) sind einwenig ärgerlich, wenn in Inhaltsverzeichnissen verwendet, insbesonders, wenn das Dokument `etoc` verwendet und viele von ihnen beinhaltet! Man kann sich wünschen, farbige Links *ausschließlich* für Inhaltsverzeichnisse! In der Tat, warum sollten in TOCs diese Dinge gerahmt oder farbig sein müssen, wenn der Benutzer *bereits erwartet, dass dort Links sind?*

Ich nehme den folgenden Trick: Entweder in der Präambel zu nehmen `\AtBeginDocument` oder danach `\begin{document}`, ich habe den Befehl

```
\addtocontents{toc}{\protect\hypersetup{hidelinks}}
```

## 18. Befehle zur .toc-Datei hinzufügen

Alle TOCs, die von **etoc** gesetzt sind, haben ihre Inhalte in eine Gruppierung getan, (wie in einer Umgebung eingeschlossen). So wird der Befehl `\hypersetup{hidelinks}` ausgeführt von *jedem* TOC, aber sein Effekt wird limitiert zu dem TOC.

Experimentell habe ich herausgefunden, dass die Option `hidelinks` oftmals mit `\hypersetup` eingestellt werden kann (Dies ist nicht bei allen `hyperref`-Optionen der Fall).

# Teil III.

# Beispiele

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 6 Abschnitte und 0 Unterabschnitt. Der Name des ersten Abschnitts lautet „Überprüfen des Kompatibilitätsmodus“ und er hat die Gliederungsnummer „19“. Der Name des letzten Abschnitts lautet „Eine (verrückte) Inline-Darstellung“ und er hat die Nummer „24“.

<b>19. Überprüfen des Kompatibilitätsmodus</b>	<b>27</b>
<b>20. Ein weiterer Kompatibilitätsmodus</b>	<b>28</b>
<b>21. Emulating the book class</b>	<b>30</b>
<b>22. Ein Inhaltsverzeichnis mit Rahmen</b>	<b>34</b>
<b>23. Ein weiteres TOC mit einer Hintergrundfarbe</b>	<b>34</b>
<b>24. Eine (verrückte) Inline-Darstellung</b>	<b>38</b>

We present some additional examples. To understand all code snippets in detail, one will possibly need to have first browsed through [Part V](#) and [Part VI](#).

## 19. Überprüfen des Kompatibilitätsmodus

Als weiteres Beispiel wollen wir nun das Inhaltsverzeichnis dieses Teils setzen. Dazu wollen wir zuerst den Kompatibilitätsmodus testen.<sup>14</sup> Das ursprüngliche Inhaltsverzeichnis ist unsichtbar und wurde am Anfang von [Teil VI](#) definiert, wobei es mit einem Label versehen wurde.

```
\begingroup % to keep in particular toc=left with local effect
\KOMAoptions{toc=left}
\etocstandarddisplaystyle % wird benötigt, damit toc=left gilt
\etocstandardlines
\tableofcontents \ref{toc:globalcmds}
\endgroup
```

## Inhaltsverzeichnis

<b>43. Festlegung des toc-Display-Stils</b>	<b>69</b>
43.1. Der Befehl \etocsettocstyle . . . . .	69
43.2. Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol	70
43.3. Der Befehl \etoctocstyle . . . . .	71
43.4. Die Befehle \etocruledstyle, \etocruled und \etoclocalrule . . . . .	71

<sup>14</sup>Dieses Dokument wurde als `scrartcl` gesetzt. Wir wollen nun testen, ob der Kompatibilitätsmodus des `etoc`-Pakets die Anpassungen innerhalb der Klassenbefehle umsetzt.

## 20. Ein weiterer Kompatibilitätsmodus

43.5. Die Befehle \etocframedstyle, \etocframed und \etoclocalframed . . .	71
43.6. Überschriften, Titel, \etocoldpar, \etocinnertopsep . . . . .	72
43.7. Der Kompatibilitätsmodus \etocstandarddisplaystyle . . . . .	72
43.8. Der Befehl \etocinline . . . . .	73
<b>44. Starred variants and hooks</b>	<b>73</b>

## 20. Ein weiterer Kompatibilitätsmodus

Wie in [Unterabschnitt 4.3](#) erläutert, sagen die Befehle \etocstandardlines und \etocstandarddisplaystyle **etoc** im Wesentlichen, wie man als Beobachter fungiert. Das Dokumentenklasse-Layout für das Inhaltsverzeichnis wird dann gehorchen. Es gibt keine Möglichkeit, dieses Standard-Layout aus dem Paket anzupassen (ändern von Schriftarten, Rändern, vertikalen Abständen, etc.). Definieren Sie dazu ein Paket von dieser Funktion; weil entweder **etoc** (vorübergehend vielleicht) im Kompatibilitätsmodus ohne Anpassung auf seinen Teil möglich ist oder der Benutzer hat das Layout in \etocsetstyle-Befehlen angegeben (und \etocsettocstyle) ist (angeblich ...) in der kompletten Steuerung.

Nun, es ist eigentlich eine Alternative. Es ist möglich, die Verwendung der \etocsetstyle-Befehle zu nutzen, um einen künstlichen Kompatibilitätsmodus neu zu erstellen, um Effekte zu erreichen, wie die folgenden, alle Dinge gleichen andererseits dem Dokumentenklassen-Standard:

1. Enthält der hyperref-Link nur die Namen, aber nicht die Anzahl der jeweils zu kapselnden Einträge der Inhaltstabellen;
2. verwenden Sie den Dokumentenklassen-Stil für Kapitel und Abschnitte, aber ändern Sie es nur für Teilebereiche;

Hier ist beschrieben, wie Sie vorgehen müssen.

This has become easier since release 1.08k as now **etoc** preserves automatically in \etocsavedsectiontocline, \etocsavedchaptertosline etc... the meaning of the L<sup>A</sup>T<sub>E</sub>X \l@section, \l@chapter, etc... macros as they were at the time the TOC is typeset.

Man nimmt in die Präambel:

```
\makeatletter
\newcommand{\MyStandardTOC}{%
  \begingroup
  % for the book or article classes:
  \%etocsetstyle{part}{}{}
  \% {\etocsavedparttocline{\etocnumber\hspace{1em}\etocname{\etocpage}}}{}%
  % for the scrbook or scrartcl classes:
  \etocsetstyle{part}{}{}
  \%{\etocsavedparttocline{\numberline{\etocnumber}\etocname{\etocpage}}}{}%
  % following is identical in book/article/scrbook/scrartcl classes:
  \etocsetstyle{chapter}{}{} %% only for book and scrbook
  \%{\etocsavedchaptertosline{\numberline{\etocnumber}\etocname{\etocpage}}}{}%
  \etocsetstyle{section}{}{}
  \%{\etocsavedsectiontocline{\numberline{\etocnumber}\etocname{\etocpage}}}{}%
  \etocsetstyle{subsection}{}{}
  \%{\etocsavedsubsectiontocline{\numberline{\etocnumber}\etocname{\etocpage}}}{}%
  % etc... if further sectioning units are used
```

```
\etocstandarddisplaystyle % this is for the title, page-marks, etc...
\tableofcontents
\endgroup
\makeatother
```

One can add to the above arbitrary text formatting commands, for example replace `\etocpage` by `\textcolor{blue}{\etocpage}`.

Wenn das Dokument nur ein Inhaltsverzeichnis hat, dann gibt es keine Notwendigkeit, die Befehle in ein Makro zu setzen oder sogar in eine Gruppe.<sup>15</sup> Mit dem Befehl `etoc` ist ein Inhaltsverzeichnis völlig identisch zu einem von der Dokumentenklasse: `article`, `book`, `scrartcl`, `scrbook`.<sup>16</sup> Die Anzahl und die Namen der einzelnen Einträge sind jeweils separat in einem `hyperref`-Link, wie es immer der Fall bei `etoc` ist, wenn man nicht im Kompatibilitätsmodus ist. Ersetzt man `\etocnumber` mit `\etocthenumber`, dann ergibt es ein Inhaltsverzeichnis, wo die Zahlen nicht mehr verlinkt sind, aber die Namen noch. Oder man kann sich entscheiden, `\etocthename` zu verwenden und eine Hyperlink-Nummer mit `\etocnumber` zu erhalten.

Hier ist ein subtiles Beispiel, wo nur geringfügig die „sections“ modifiziert wurden (hinzufügen von Farbe für die Nummer und das Entfernen des `hyperref`-Links) und die Unterabschnitte, wie in der Standardeinstellung, *mit Ausnahme* von einer bestimmte „section“ im Layout komplett geändert wurde:

## Inhaltsverzeichnis

<b>3. Muss ich Computerfreak sein, um etoc benutzen zu können?</b>	<b>8</b>
3.1. Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben . . . . .	9
<b>4. Zeilenstile und Stile der Inhaltsangabe (toc display style)</b>	<b>9</b>
<code>\etocsetstyle</code> für den Zeilenstil (4.1) – <code>\etocsettocstyle</code> für den Stil der Inhaltsangabe (4.2) – Kompatibilismus (4.3).	
<b>5. Ein erstes Beispiel</b>	<b>10</b>
<b>6. Ein zweites Beispiel</b>	<b>12</b>
<b>7. Eine wunderschöne Beispielthese</b>	<b>14</b>
<b>8. Verlinkte Liste der Hauptbefehle</b>	<b>15</b>

<sup>15</sup> Und wenn darüber hinaus, um das gleiche Layout wie im Standard zu halten, kann man sich die Frage stellen, warum man `etoc` benutzt ... Es gibt *einen* guten Grund: Zahlen und Namen sind separate `hyperref`-Links, nach der Regel gibt es nur eine Verbindung, die sowohl die Nummer und den Namen auf einem toc-Eintrag entsprechen.

<sup>16</sup> For the `memoir` class, one needed prior to `etoc` 1.08k a bit more: each of the command `\booknumberline`, `\partnumberline` and `\chapternumberline` had to be saved. They can now be used directly in the line styles, because their meanings are not modified anymore by `etoc` during its TOC typesetting.

## 21. Emulating the book class

Dieses Beispiel hat nur Abschnitte und Unterabschnitte und der in \MyStandardTOC verwendete Code war:

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

Beachten Sie die Seite mit den Kopf-Zeichen, welche man dem Inhaltsverzeichnis hinzufügen kann. Abschnitte und Unterabschnitte sind genauso wie im Standard (mit Ausnahme der Unterabschnitte, welche von einem bestimmten Benutzer-Abschnitt aus gewählt wird und bis auf die Farbe der Abschnittsnummern), ohne das explizit die Länge, die Schriftart oder andere Formatierungsanweisungen gegeben werden. Aber wir mussten die `scrartcl`-Quellen untersuchen, um festzustellen, was zu nehmen ist für `\leftskip` und `\rightskip` für unsere Nutzer-sections-Einträge.

## 21. Emulating the book class

As explained in [subsection 4.3](#): without explicit use of an `\etocsetstyle` command the package will leave to the document class the hand regarding the “toc line styles”. It is sometimes asked by users (for example those using `etoc` for its `\localtableofcontents`) how to stay close to but not completely identical with the design implemented by the standard classes, such as `book`. I can recommend package `tocloft` for this, as it is compatible with `etoc` (see [section 55](#)) and thus `etoc` will obey the `tocloft` customizations (as long as no use has been made of `\etocsetstyle`). It is also possible to modify only the style for, say, sections and leave the parts, chapters, subsections as in the document class, via the technique from [section 20](#).

But for complete control, here is a translation of the `book` class code into `etoc` lingua. It is then easy to modify the relevant lengths or adjust the used fonts. I thank DENIS BITOUZÉ for prompting me to include this in the `etoc` manual, as it resulted from some conversation we had about this. The code is not 100% faithful to the `book` class, and particularly its rendering of (multi-line) non-numbered units differs (... I think, as I copied pasted as is the code from where I had stored it and did not do much thinking about it again). Some

proficiency in low-level TEX and LATEX macros is needed to understand what the code says, but for modifying fonts or some lengths such in-depth understanding is not needed.

With some extra code one can *automatically adjust the widths* assigned to typesetting sectioning numbers in order to prevent overflows, even with for example XXXVIII; but this is a more advanced feature which I have moved to [section 31](#).

First we set up some lengths. I use macro registers, not real LATEX lengths. When using em's however, this means that one must pay attention to when the actual dimension assignment is made, as this will then depend upon the current font settings. In the code below, at the location where the `\TOCnumwidthB` and `\TOCnumwidthC` will be used, the `1em` from their specification will be matched to the normal medium series font, not the bold font; this is deliberate so that one can compare more readily with the other dimensions; besides, with the `\TOCcomputenumwidths` from [section 31](#) these macros will actually hold a dimension using `pt` as dimensional unit.

```
% it will be easy to globally shift the TOC horizontally if needed
\def\TOCleftmargin      {0pt}
\def\TOCrightmargin    {2.55em}% like LaTeX's \@tocrmarg

% this is for dotted leaders
\newbox\TOCleaderbox
\def\TOCleaderboxwidth {0.7777em}% about like what standard classes do

% vertical spacing
\def\TOCsmalvskip {0pt plus .2pt}
\def\TOCmedvskip   {1em plus 1pt}
\def\TOCbigvskip   {2.25em plus 1pt}

% the "numwidths" for typesetting the numbering of division units.
% I don't recall exactly how (and for which fonts) these figures were chosen.
% They quickly prove too small if using Roman numerals (as do too the book
% class defaults even though they are a bit larger).
\def\TOCnumwidthB {1.5em} % chapter
\def\TOCnumwidthC {2.278em}% section, I think default is 2.3em
\def\TOCnumwidthD {3.056em}% analog in standard class is 3.2em
\def\TOCnumwidthE {3.833em}% analog in standard class is 4.1em
\def\TOCnumwidthF {4.611em}% analog in standard class is 5em
\def\TOCnumwidthG {5.389em}% analog in standard class is 6em

%The code for the "global toc style".

\newcommand*\TOCglobalstyle {%
\etocsettocstyle
{%
\if@twocolumn \restonecoltrue \onecolumn \else \restonecolfalse \fi
\parindent\z@ \leftskip\z@skip \rightskip \z@skip
\setbox\TOCleaderbox\hbox to \TOCleaderboxwidth{\hss.\hss}%
\chapter *{\noindent\kern\TOCleftmargin\relax % uses "pt"...
\contentsname
\@mkboth {\MakeUppercase \contentsname}{\MakeUppercase \contentsname}%
\rightskip \TOCrightmargin\relax
\parfillskip -\rightskip % or a smaller value if desired
\leftskip \TOCleftmargin\relax }
{\if@restonecol \twocolumn \fi\cleardoublepage}%
}%
\etocsetstyle{part}{}%
```

## 21. Emulating the book class

```

{\addpenalty {-\@highpenalty}%
\addvspace \TOCbigvskip
\leavevmode
{\large \bfseries % use a group to limit font change
\interlinepenalty\@M
\etocifnumbered{\etocnumber\hspace{1em}}{}%
\etocname
\nobreak\hfil\makebox[-\parfillskip][r]{\etocpage}\par
\nobreak
}
}
}
}%
%
\etocsetstyle{chapter}
{\advance\leftskip\TOCnumwidthB\relax
\addpenalty {-\@highpenalty }%
\vskip \TOCmedvskip\relax
\leavevmode
\interlinepenalty\@M
\etocifnumbered
{\llap{\makebox[\TOCnumwidthB][l]{\bfseries\etocnumber}}}
{\advance\leftskip-\TOCnumwidthB\relax}%
\bfseries\etocname
\nobreak\hfil\makebox[-\parfillskip][r]{\etocpage}\par }%
\penalty \@highpenalty
}
}
}%
{\advance\leftskip-\TOCnumwidthB\relax}%
%
\TOCsetlinestyle {section} {\TOCnumwidthC}%
\TOCsetlinestyle {subsection} {\TOCnumwidthD}%
\TOCsetlinestyle {subsubsection}{\TOCnumwidthE}%
\TOCsetlinestyle {paragraph} {\TOCnumwidthF}%
\TOCsetlinestyle { subparagraph} {\TOCnumwidthG}%
}% end of \TOCglobalstyle

% The common code for line styles is abstracted into a macro:

\newcommand\TOCsetlinestyle [2]{% #1= unit, #2= numwidth as macro
\etocsetstyle{#1}
{\advance\leftskip#2\relax
\vskip \TOCverysmallvskip\relax
\leavevmode
\interlinepenalty\@M
\etocifnumbered
{\llap{\makebox[#2][l]{\etocnumber}}}{\advance\leftskip-#2\relax}%
\etocname
\nobreak\leaders \copy\TOCleaderbox
\hfil\makebox[-\parfillskip][r]{\etocpage}%
\par }%
}
}%
{\advance\leftskip-#2\relax}%
}
\makeatother

```

Nota Bene: the code deliberately handles the non-numbered sectioning units unlike the

way of the standard document classes (particularly regarding the alignment of multi-line headings.)

The whole thing was encapsulated in \TOCglobalstyle, because we also want a \TOClocalstyle for local tables of contents which typically will want to use \section\* rather than \chapter\* and not insert page marks in the headers. The \TOClocalstyle is to be issued once, after the main document TOC, or rather before using \localtableofcontents. If one wants a full TOC at end of document one will naturally have to issue again \TOCglobalstyle there.

```
\makeatletter
\newcommand*\TOClocalstyle {%
\etocsettocstyle
{\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
\setbox\TOCleaderbox\hbox to \TOCleaderboxwidth{\hss.\hss}%
\parindent\z@
\dimen@ 2.25em % for left indenting
\section *{\kern\dimen@ % use of \dimen@ works here by sheer luck
\contentsname
% un-comment this if marks are wanted:
% \mkboth {\MakeUppercase \contentsname}{\MakeUppercase \contentsname}%
}% end of \section
\parskip \z@skip
\vspace{-1.25\baselineskip}% somewhat ad hoc
\leftskip 2.25em
\rightskip 4.5em
\advance\rightskip-\TOCrightmargin\relax
\leavevmode\leaders\hrule\@height\p@\hfill\kern\z@\par
\rightskip 4.5em
\parfillskip -\TOCrightmargin\relax }
\nobreak\vskip-.5\baselineskip
\leavevmode\leaders\hrule\@height\p@\hfill\kern\z@\par
\bigskip
\if@restonecol \twocolumn \fi }%
%
\etocsetstyle{section}
{\advance\leftskip\TOCnumwidthC\relax}
{\addpenalty \secpenalty
\etociffirst{}{\addvspace{\TOCmedvskip}}%
\leavevmode
{\interlinepenalty\@M
\bfseries\etocifnumbered
{\llap{\makebox[\TOCnumwidthC][l]{\etocnumber}}}
\advance\leftskip-\TOCnumwidthC}%
\etocname\nobreak\hfil\makebox[-\parfillskip][r]{\etocpage}\par }%
\penalty \highpenalty }
{}%
\advance\leftskip-\TOCnumwidthC\relax}%
% the rest is identical with code for global tocs:
\TOCsetlinestyle {subsection} {\TOCnumwidthD}%
\TOCsetlinestyle {subsubsection}{\TOCnumwidthE}%
\TOCsetlinestyle {paragraph} {\TOCnumwidthF}%
\TOCsetlinestyle { subparagraph}{\TOCnumwidthG}%
}% end of \TOClocalstyle
\makeatother
```

## 23. Ein weiteres TOC mit einer Hintergrundfarbe

As mentioned previously, this handles non-numbered (multi-line) sectioning units somewhat differently from what happens in the standard document classes.

For some reason this code has some hard-coded 2.25em and 4.5em which were not abstracted into macros or lengths. The code inserts horizontal rules above and below the TOC contents in a non-separable by pagebreak way.

See [section 31](#) for more.

## 22. Ein Inhaltsverzeichnis mit Rahmen

Als nächstes wollen wir ein gerahmtes Inhaltsverzeichnis erzeugen, wobei wir den vom Paket vorgegebenen Linienstil nutzen und ein paar Farben hinzufügen wollen. (Es wurde in eine Float-Umgebung eingefügt, die man on the next page findet.)

```
\etocdefaultlines
\begin{group}
\renewcommand{\etoccolumnsep}{2em}
\renewcommand{\etocinnerleftsep}{1.5em}
\renewcommand{\etocinnersrightsep}{1.5em}
% Befehle für die Hintergrundfarbe des Inhaltsverzeichnisses
\renewcommand{\etocbkgcolorcmd}{\color{yellow!10}}
% Befehle für die Rechtecke vor und nach dem Inhalt des Verzeichnisses
\renewcommand{\etocoprule}{\hrule height 1pt}
\renewcommand{\etocoprulecolorcmd}{\color{red!25}}
\renewcommand{\etocbottomrule}{\hrule height 1pt}
\renewcommand{\etocbottomrulecolorcmd}{\color{red!25}}
% Befehle für die Rechtecke links und rechts des Inhaltsverzeichnisses
\renewcommand{\etocleftrule}{\vrule width 5pt}
\renewcommand{\etocrightrule}{\vrule width 5pt}
\renewcommand{\etocleftrulecolorcmd}{\color{red!25}}
\renewcommand{\etocrightrulecolorcmd}{\color{red!25}}
% mit \fcolorbox legen wir einen farbigen Rahmen für den Titel an
\fboxrule1pt
\renewcommand{\etocbelowtocskip}{0pt\relax}
\etocframedstyle {\normalsize\rmfamily\itshape
\fcolorbox{red}{white}{\parbox{.8\linewidth}{\centering
Dies ist das nach \etoc gestaltete Inhaltsverzeichnis, welches
jedoch nur die Abschnitte und Unterabschnitte dieses Teils
enthält. Da es gerahmt ist, darf es nicht größer sein als eine Seite.
Es hat das Label \label{toc:b}}}}
\begin{figure}[ht!]
\centering
\tableofcontents \label{toc:b} \ref{toc:globalcmds}
\end{figure}
\end{group}
```

## 23. Ein weiteres TOC mit einer Hintergrundfarbe

Lassen Sie uns nun einige anspruchsvollere Linienstile versuchen. Das Display wird den Paket-Befehl \etocframedstyle verwenden, damit das erzeugte Inhaltsverzeichnis auf

Dies ist das nach **etoc** gestaltete Inhaltsverzeichnis, welches jedoch nur die Abschnitte und Unterabschnitte dieses Teils enthält. Da es gerahmt ist, darf es nicht größer sein als eine Seite. Es hat das Label **toc:b**.

<b>Festlegung des toc-Display-Stils</b>		Die Befehle \etocframestyle, \etocframed
.....	43, S. 69	.....
Der Befehl \etocsettocstyle	.....	.....
.....	43.1, S. 69	43.5, S. 71
Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol	.....	Überschriften, Titel, \etocoldpar, \etocinnertopsep
.....	43.2, S. 70	43.6, S. 72
Der Befehl \etoctocstyle	.....	Der Kompatibilitätsmodus \etocstandarddisplaystyle
.....	43.3, S. 71	.....
Der Befehl \etocininline	.....	43.7, S. 72
Die Befehle \etocruledstyle, \etocruled und \etoclocalruled	.....	Der Befehl \etocininline
.....	43.4, S. 71	.....
		<b>Starred variants and hooks</b>
		.....
		44, S. 73

eine einzige Seite passt. Wir wickeln es in eine [Figur-Umgebung](#), damit es auf der aktuellen Seite angezeigt wird. [on page 37](#).

Das aktuelle Design ist nicht in **etoc** vorgefertigt; sondern nutzt seine ‘eingerahmten’ Stile mit einer Hintergrundfarbe. Die Rahmengrenzen wurden so gesetzt, dass die gleiche Farbe, wie die Hintergrundfarbe, für die gesamte Sache dienen soll.

The details of the line styles used here are a bit involved, they were written by the author at some early stage of this documentation and have only been slightly revised to use more L<sup>A</sup>T<sub>E</sub>X-commands and less T<sub>E</sub>X-primitives. Similar code is used also for [this other toc](#).

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

\renewcommand{\etocbkgcolorcmd}{\color{tocbackground}}
\renewcommand{\etoclefrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocrightrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etocbottomrulecolorcmd}{\color{tocbackground}}
\renewcommand{\etoctoprulecolorcmd}{\color{tocbackground}}

\renewcommand{\etoclefrule}{\vrule width 3cm}
\renewcommand{\etocrightrule}{\vrule width 1cm}
\renewcommand{\etocbottomrule}{\hrule height 12pt}
\renewcommand{\etoctoprule}{\hrule height 12pt}

\renewcommand{\etocinnertopsep}{0pt}
\renewcommand{\etocinnerbottomsep}{0pt}
\renewcommand{\etocinnerleftsep}{0pt}
\renewcommand{\etocinnersrightsep}{0pt}

\newcommand\shiftedwhiterule[2]{%
  \hbox to \linewidth{\color{white}%

```

## 23. Ein weiteres TOC mit einer Hintergrundfarbe

```
\hskip#1\leaders\vrule height1pt\hfil}\nointerlineskip  
\vskip#2}

\etocsetstyle{subsubsection}
{\etocskipfirstprefix}
{\shiftedwhiterule{\leftskip}{6pt}}
{\sffamily\footnotesize
 \leftskip2.3cm\hangindent1cm\rightskip.5cm\relax
 \makebox[1cm][1]{\color{subsecnum}\etocnumber}%
 \color{black}\etocname
 \nobreak\leaders\hbox to .2cm{\hss.}\hfill
 \rlap{\makebox[.5cm][r]{\etocpage\hspace{.1cm}}}\par
 \nointerlineskip\vskip3pt
{}}

\etocsetstyle{subsection}
{\etocskipfirstprefix}
{\shiftedwhiterule{1.5cm}{6pt}}
{\sffamily\small
 \leftskip1.5cm\hangindent.8cm\rightskip.5cm\relax
 \makebox[.75cm][1]{\color{subsecnum}\etocnumber}%
 \color{black}\etocname
 \nobreak\leaders\hbox to .2cm{\hss.}\hfill
 \rlap{\makebox[.5cm][r]{\etocpage\hspace{.1cm}}}\par
 \nointerlineskip\vskip3pt
{}}

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

\etocsetstyle{section}
{\coloredstuff{blue}{\hfil \bfseries\large Contents of Part One\hfil}}
{\vskip3pt\sffamily\small}
{\coloredstuff{blue}
 {\leftskip1.5cm\rightskip.5cm\parfillskip-\rightskip
 \makebox[0pt][r]{\makebox[.5cm][1]{\etocnumber}}%
 \etocname\nobreak\hfill\makebox[.5cm][r]{\etocpage\hspace{.1cm}}}%
 \vskip6pt
{}}

\etocframedstyle[1]{}
\tableofcontents \label{toc:floating} \ref{toc:overview}
\vspace{-\baselineskip}
\centeredline{|\tableofcontents \ref{toc:overview}|}
(\emph{cf.} \hyperref[toc:clone]{this other toc})
\end{figure}
```

## Contents of Part One

3	Muss ich Computerfreak sein, um <b>etoc</b> benutzen zu können?	8
3.1	Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben .....	9
4	Zeilenstile und Stile der Inhaltsangabe (toc display style)	9
4.1	\etocsetstyle für den Zeilenstil .....	9
4.2	\etocsettocstyle für den Stil der Inhaltsangabe .....	9
4.3	Kompatibilismus .....	10
5	Ein erstes Beispiel	10
6	Ein zweites Beispiel	12
7	Eine wunderschöne Beispielthese	14
8	Verlinkte Liste der Hauptbefehle	15

\tableofcontents \ref{toc:overview} (cf. [this other toc](#))

## 24. Eine (verrückte) Inline-Darstellung

Nun wollen wir eine etwas verrückte Inline-Darstellung erzeugen, womit das Inhaltsverzeichnis des gesamten Dokuments gesetzt werden soll. Hier werden die Unterabschnitte als Fußnoten dargestellt. Diese Art der Darstellung bietet sich eher für verlinkte Dokumente an, weniger für gedruckte! (Mir hingegen gefällt diese Art der Darstellung, obwohl mein Geschmack scheinbar oft nicht geteilt wird.)

Hier ist das inline dargestellte Inhaltsverzeichnis. *Zusammenfassung, Vorwort, Lizenz, Überblick: Muss ich Computerfreak sein, um etoc benutzen zu können?*<sup>17</sup>, Zeilenstile und Stile der Inhaltsangabe (toc display style)<sup>18</sup>, Ein erstes Beispiel, Ein zweites Beispiel, Eine wunderschöne Beispielthese, Verlinkte Liste der Hauptbefehle. Beliebig viele TOCs und lokale Versionen davon: Die \tableofcontents-, \localtableofcontents- und \localtableofcontentswithrelativedepth -Befehle, Beschriften und anderswo wieder benutzen, Eine mächtige Funktionsweise von etoc: Die Wiederzuweisung von Levels mit von \etocsetlevel, Die \etocsettocdepth- und \etocsetnexttocdepth-Befehle<sup>19</sup>, Der Befehl \etocsettocdepth.toc<sup>20</sup>, Die Befehle \etocdepthtag.toc und \etocsettagdepth<sup>21</sup>, Die Befehle \etocglobaldefs und \etoclocaldefs, Not displayed empty TOCs<sup>22</sup>, The \etoclocaltop command, Befehle zur .toc-Datei hinzufügen<sup>23</sup>. Beispiele: Überprüfen des Kompatibilitätsmodus, Ein weiterer Kompatibilitätsmodus, Emulating the book class, Ein Inhaltsverzeichnis mit Rahmen, Ein weiteres TOC mit einer Hintergrundfarbe, Eine (verrückte) Inline-Darstellung. Überraschende Anwendungen von etoc: Das TOC für TOCs, Willkürliche „Listen von …“, \etoccontentline, Das TOC als ein Baum, Das TOC als ein Molekül, Das TOC als ein TikZ Mind-Map, Das TOC als eine Tabelle, A TOC self-adjusting widths for its typesetting. Befehle für die TOC-Linienstile: Die \etocsetstyle-, \etocname- und \etocpage-Befehle, Die \etocskipfirstprefix und \etociffirst Befehle, Der \etocnumber-Befehl, Der \etocifnumbered-Schalter, Die Befehle \etocthename, \etocthenumber sowie \etocthepage, Der \etoclink-Befehl, Die Befehle \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage und \etocthealink, Die \etocsavedsectiontocline, usw... Befehle., Der \etocsetlevel-Befehl, Umfang der zur .toc-Datei gehörigen Befehle<sup>24</sup>, Bin ich auch rot?. Befehle für den toc-Display-Stil: Festlegung des toc-Display-Stils<sup>25</sup>, Starred variants and hooks. Eigene Stile von etoc nutzen und anpassen: Über-

---

<sup>17</sup>Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben.

<sup>18</sup>\etocsetstyle für den Zeilenstil; \etocsettocstyle für den Stil der Inhaltsangabe; Kompatibilismus.

<sup>19</sup>Die hyperref-Option bookmarksdepth.

<sup>20</sup>Die Befehle \etocbeytoctocdepth und \etocignoretoctocdepth.

<sup>21</sup>Die Befehle \etocbeydepthtags und \etocignoredepthtags.

<sup>22</sup>The \etoccheckemptiness command; The \etocnotocifnotoc command; The \etocifwasempty command.

<sup>23</sup>Die hyperref-Option hidelinks.

<sup>24</sup>Testen des Bereichs; Dies ist eine (blass)rote Subsection zu Demonstrationszwecken.

<sup>25</sup>Der Befehl \etocsettocstyle; Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol; Der Befehl \etocstyle; Die Befehle \etocruled-style, \etocrule und \etoclocalrule; Die Befehle \etocframedstyle, \etocframed und \etoclocalframed; Überschriften, Titel, \etocoldpar, \etocinnertop-sep; Der Kompatibilitätsmodus \etocstandarddisplaystyle; Der Befehl \etocin-line.

blick über die Grundbefehle zur Stilbearbeitung<sup>26</sup>, Der Linienstandard des Pakets: \etoc-defaultlines, **etoc** individuell anpassen<sup>27</sup>, Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis. **Tipps:** The \etocsetlocaltop.toc command, Hacken „framed parboxes“, Umstellen der Ebenen<sup>28</sup>, Statistiken anzeigen, Tiefe Tags nehmen, Das TOC als eine Tabelle setzend (die alte Weise). **etoc und die Außenwelt:** Kompatibilität mit anderen Paketen<sup>29</sup>, *TEX-nische Sachen, Fehler und Katastrophen*. **Der Code:** Zeitstempel, Änderungen, Ausführung.

Zur Erzeugung dieses Inhaltsverzeichnisses wurde folgender Code genutzt:

```
\begingroup
\newsavebox{\forsubsections}
\etocsetstyle{part}{\upshape. \etocskipfirstprefix}
    {. \upshape}
    {\bfseries\etocname:~~}
    {}
\etocsetstyle{section}{\itshape\etocskipfirstprefix}
    {,}
    {\mdseries\etocname}
    {}
\etocsetstyle{subsection}
    {\begin{lrbox}{\forsubsections}\upshape\etocskipfirstprefix}
    {; }
    {\etocname}
    {\end{lrbox}\footnote{\unhbox\forsubsections.}}
\etocsetstyle{subsubsection}
    {(\itshape\etocskipfirstprefix}
    {,}
    {\etocname}
    {\upshape)}
\etocsettocstyle{Hier ist das inline dargestellte Inhaltsverzeichnis. }{.\par}
\tableofcontents
\endgroup
```

---

<sup>26</sup>Ein lokales Inhaltsverzeichnis anpassen; Das Aussehen eines Inhaltsverzeichnisses anpassen; Inhaltsverzeichnisse darstellen; Label und Referenzen.

<sup>27</sup>Den durch **etoc** vordefinierten Linienstil individualisieren; Die Darstellung eines Inhaltsverzeichnisses anpassen.

<sup>28</sup>Alle Unterabschnitte in diesem Dokument.

<sup>29</sup>Generelle Informationen; Kompatibilität mit beamer; Kompatibilität mit Babel; Kompatibilität mit hyperref; Kompatibilität mit multicol; Kompatibilität mit tocloft; Kompatibilität mit der memoir-Klasse; Kompatibilität mit tocvssec2; Kompatibilität mit tableof; Kompatibilität mit tocstyle.

## Teil IV.

# Überraschende Anwendungen von etoc

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 7 Abschnitte und 0 Unterabschnitt. Der Name des ersten Abschnitts lautet „Das TOC für TOCs“ und er hat die Gliederungsnummer „25“. Der Name des letzten Abschnitts lautet „A TOC self-adjusting widths for its typesetting“ und er hat die Nummer „31“.

<b>25. Das TOC für TOCs</b>	<b>40</b>
<b>26. Willkürliche „Listen von ...“, \etoctoccontentsline</b>	<b>42</b>
<b>27. Das TOC als ein Baum</b>	<b>43</b>
<b>28. Das TOC als ein Molekül</b>	<b>46</b>
<b>29. Das TOC als ein TikZ Mind-Map</b>	<b>49</b>
<b>30. Das TOC als eine Tabelle</b>	<b>53</b>
<b>31. A TOC self-adjusting widths for its typesetting</b>	<b>58</b>

## 25. Das TOC für TOCs

Hier ist die nummerierte und verknüpfte Liste aller Tabellen von Inhalten, die in diesem Dokument dargestellt sind<sup>30</sup> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42. Und hier erhalten wir, was wir gerade geschrieben haben:

Hier ist die nummerierte und verknüpfte Liste aller Tabellen von Inhalten, die in diesem Dokument dargestellt sind: \tableofcontents.

Die vorausgehenden Arbeiten waren wie folgt. Zuerst definierten wir einen Zähler `visibletoc`, dessen Berufung es ist, bei jedem angezeigten Inhaltsverzeichnis erhöht zu werden. `etoc` hat einen eigenen Zähler, aber es zählen alle Inhaltsverzeichnisse, auch wenn diese nicht angezeigt werden, da die `tocdepth` Werte -2 oder -3 waren.

Wir könnten manuell `\refstepcounter{visibletoc}` und `\label` an allen geeigneten Stellen im Dokument hinzufügen, und wir würden dann `\ref-` Befehle verwenden, aber dies stellt eine hohe manuelle Bearbeitung der Quelle dar.

Es gibt eine viel bessere Möglichkeit: Es gibt einen Aufhänger `\etocaftertitlehook` und wir sagten, es erhöht sich der `visibletoc`-Zähler und wir schreiben eine Anlage zu der `.toc`-Datei, in analoger Weise zu dem Gliederungsbefehl, zum Beispiel Kapitel,

---

<sup>30</sup>Die Inhaltsverzeichnisse können „Schwimmer“ setzen, um die Reihenfolge zu ändern: die Zahlen werden in der Reihenfolge der Inhaltsverzeichnisse in dem Dokument aufgeführt; aber die Nummerierung selbst ist von der Reihenfolge der Inhaltsverzeichnisse in der Quelle dieses Dokumentes ...

Abschnitt oder Unterabschnitt. Wie **etoc** seinen eigenen Zähler erhöht, noch bevor der Titel des Inhaltsverzeichnisses gesetzt wird, bietet diese (die meiste Zeit) einen besseren Link-Zielpunkt als jeder Zähler von \etocaftertitlehook (der Link wird nach dem Titel platziert). Anstatt mit \refstepcounter{visibletoc}, innerhalb von \etocaftertitlehook, formulieren wir \stepcounter{visibletoc}, dann folgt der Befehl \etoc toccontentsline{visibletoc}{\thevisibletoc}. Dieser **etoc**-Befehl \etoc toccontentsline{\langle level\\_name \rangle}{\langle name \rangle} hat die gleiche Wirkung wie:

```
\addcontentsline{toc}{\langle level_name \rangle}{\langle name \rangle}
```

aber sein Nutzen ist es<sup>31</sup>, die Reparatur der automatischen Erstellung von Lesezeichen zu verhindern mit \addcontentsline bei dem hyperref-Paket, wie die pdf-Lesezeichen machen sie nicht viel Sinn hier (und würde eine Beschwerde bezüglich von hyperref eruiieren, denn das Lesezeichen-Niveau ist ‘unbekannt’).<sup>32</sup>

Schließlich war in der Präambel des Dokuments \etocsetlevel{visibletoc}{6}. Das Niveau 6 (oder etwas mit einer höheren Zahl) wird ignoriert, auch wenn tocdepth zum Beispiel den Wert 10 hat; das ist unabhängig davon, ob **etoc** die Dokumentenklasse als Standardlinien oder als eigene Linienstile nutzt oder die, die vom Benutzer mit \etocsetstyle Befehl definiert wurden. Es gibt also keinen Grund zur Sorge, dass etwas schief gehen könnte.

Dann haben wir hier nur \etocsetlevel{visibletoc}{0} gesetzt. Und nur diese Art von Einträgen weisen wir vorübergehend eine Teil- und Kapitel-Ebene 1 zu (oder etwas höher als null) und setzen tocdepth auf den Wert 0. Wir haben auch \etocsetstyle{visibletoc}{\etocskipfirstprefix}{, }{\etocname}{}, welche eine innere Anzeige mit dem Komma als Trennzeichen definiert. Wie **etoc** Aspekte \par automatisch als Standard das Inhaltsverzeichnis setzt, verwendeten wir den Befehl \etocinline (auch als \etocnopar bekannt), welches die Reaktion ausmacht .

Hier sind die Details der Implementierung:

```
< in der Präambel >
\newcounter{visibletoc}
\renewcommand{\etocaftertitlehook}
  {\stepcounter{visibletoc}\etoc toccontentsline{visibletoc}{\thevisibletoc}}
\etocsetlevel{visibletoc}{6}
\begin{document}
  < document body >
\subsection{Surprising uses of etoc}
\begin{group}
  \etocinline
  \etocsetlevel{part}{1}
  % \etocsetlevel{chapter}{1} % (keine Kapitel in scrartcl Klasse)
  \etocsetlevel{visibletoc}{0}
  \etocsetstyle{visibletoc}
    {\etocskipfirstprefix}{, }{\color{niceone}\etocname}{}
  \etocsettocstyle{}{} % keine Titel, Regeln oder Rahmen festgelegt!
  \etocsetnexttocdepth{visibletoc} % zeigt nur die 'visibletoc' Einträge an aus .toc
```

Hier ist die nummerierte und verknüpfte Liste aller Tabellen von Inhalten,

---

<sup>31</sup>Mit \addtocontents anstatt \addcontentsline.

<sup>32</sup>Das Paket bietet eine Stern-Variante \etoc toccontentsline\*, die zulässt, dass die Erstellung von Lesezeichen und das Haben eines dritten obligatorischen Arguments, welches das Lesezeichen-Niveau darstellt, funktioniert; abhängig vom Kontext kann die Hauptrolle und die Nicht-Stern-Variante profitabel durch \phantomsection sein.

## 26. Willkürliche „Listen von ...“, \etoccontentsline

die in diesem Dokument angezeigt werden: \tableofcontents.  
\endgroup

Nach \etocsetstyle{visibletoc}{...}{...}{...}{...}, wird für alle zukünftigen Inhaltsverzeichnisse (nicht im Kompatibilitätsmodus) der definierte Stil für das Niveau 0 verwendet (was normalerweise der Pegel für ein Kapitel ist). Um diese Änderung auf die einfachste Art und Weise zu halten, muss man alles in eine Gruppe setzen.

Der Abschnitt 51 gibt eine andere Verwendung von Verschachtelungsstufen an.

## 26. Willkürliche „Listen von ...“, \etoccontentsline

Die Idee der Internabwehr ist leistungsfähig und ermöglicht etoc -Anzeigelisten, um beliebige Dinge in dem Dokument enthalten zu lassen. Und all das mit noch nichts anderem als der .toc Datei! Beispiel: Stellen Sie sich ein Dokument mit Dutzenden von Übungen vor, vielleicht als \newtheorem{exercise}{}[section]. Lassen Sie uns erklären, wie man etoc anweist, eine verlinkte Liste all dieser Übungen anzuzeigen. Dafür haben wir in der Präambel geschrieben:

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

Auf diese Weise wird \etocname die Übungsnummer vergeben (aber \etocnumber wird leer sein). Hätten wir stattdessen

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

verwendet, wäre die Übungsnummer via \etocnumber verfügbar und \etocname wäre leer gewesen. Es spielt keine Rolle, welches der beiden Verfahren verwendet wird. Der etoc Befehl \etoccontentsline{...}{...} wird als Ersatz für \addcontentsline{toc}{...}{...} verwendet: Das ist die, die durch das Reparieren von hyperref zu \addcontentsline verwendet wird, um in ihrem Prozess die Erstellung von Lesezeichen zu vermeiden. Wenn man hyperref ermächtigt, Lesezeichen auf einer bestimmten Ebene  $\langle n \rangle$  zu erschaffen, kann einer (hier mit  $\langle n \rangle = 2$ ) die Stern-Variante \etoccontentsline\* benutzen, welche ein zusätzliches Argument hat:

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

Der Zähler exerci wird bereits durch das exerci-Theoremumfeld erhöht und sorgt für den richtigen Ort für den Link zusätzlich zum Paket hyperref. Der Befehl \exercisetotoc fügt für jede Übung eine Zeile in die .toc-Datei, entsprechend einer fiktiven Dokumenteinheit mit dem Namen ‘Übung’ ein. Eine Vier-Säulen-Liste, einschließlich der Abschnitte, kann dann erreichen mit

```
\etocsetnexttocdepth{2}      % Abschnitte sind auf Stufe 1 und werden anzeigen
\begingroup
\etocsetlevel{exercise}{2}    % aber:
```

```

\etocsetlevel{chapter}{3}      %    keine Kapitel
\etocsetlevel{subsection}{3}   %    keine Unterabschnitte
\etocsetlevel{part}{3}         %    keine 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\hspace{.5em}\etocname}\par
  {\nopagebreak[3]}{%
\etocruledstyle[4]{\Large\bfseries List of the exercises}
\setlength{\columnseprule}{.4pt}
\tableofcontents
\endgroup

```

## 27. Das TOC als ein Baum

Mit `tikz`<sup>33</sup> und dem Paket `forest`<sup>34</sup> wird das Inhaltsverzeichnis als Baum angezeigt. Die Technik ist, den von `etoc` geänderten Befehl `\tableofcontents` nicht für den Satz zu verwenden, sondern ein Makro, oder besser hier eine *symbolische Liste*-Variable vorzubereiten, mit allen Anweisungen, die später ausgeführt werden. LESLIE LAMPORT's Buch erwähnt überhaupt nicht die symbolische Liste und L<sup>A</sup>T<sub>E</sub>X vermittelt den Eindruck, nicht wirklich darauf zu warten, dass die Mehrheit der Nutzer immer über sie (oder begrenzte Makros) etwas hört; die ganze Sache und der folgende Abschnitt sind daher für fortgeschrittene Anwender.

Setzt man die `\etocnumber`- und `\etocname`-Befehle in `\treetok`, dann würde das nichts nützen: Auf welche Zahl oder Namen würde sich das dann beziehen, in einer verzögerten Ausführung?

Wir brauchen die Speicherung der Makro-Namen, aber nicht die Makro-Inhalte. Und wir wollen auch die korrekten `hyperref`-Hyper-Links erhalten.<sup>35</sup>

Die Befehle `\etocname`, etc... sind robust. Es ist einfacher mit `\etocthelinkednumber`, `\etocthelinkedname` und `\etocthelinkedpage`, das die gleichen Informationen in einer einfacher zugänglichen Form enthält.

Aus diesem `forest`-Baum haben wir ganz besondere `etoc` -Stile für Abschnitte und Unterabschnitte konzipiert. Sie verwenden ein symbolisches Listenregister mit dem Namen `\treetok` und ein Makro `\appendtotok`, deren Aufgabe es ist, zu einer bestimmten symbolischen Listenvariablen den Inhalt eines Makros als zweites Argument anzuhängen. All dies wird in Reaktion auf einen Befehl `\tableofcontents` passieren, aber noch ist *nichts* in deren Prozess.<sup>36</sup> Das ist der spätere Job einer `forest`-Umgebung, die den Inhalt von `\treetok` wiedergeben soll.

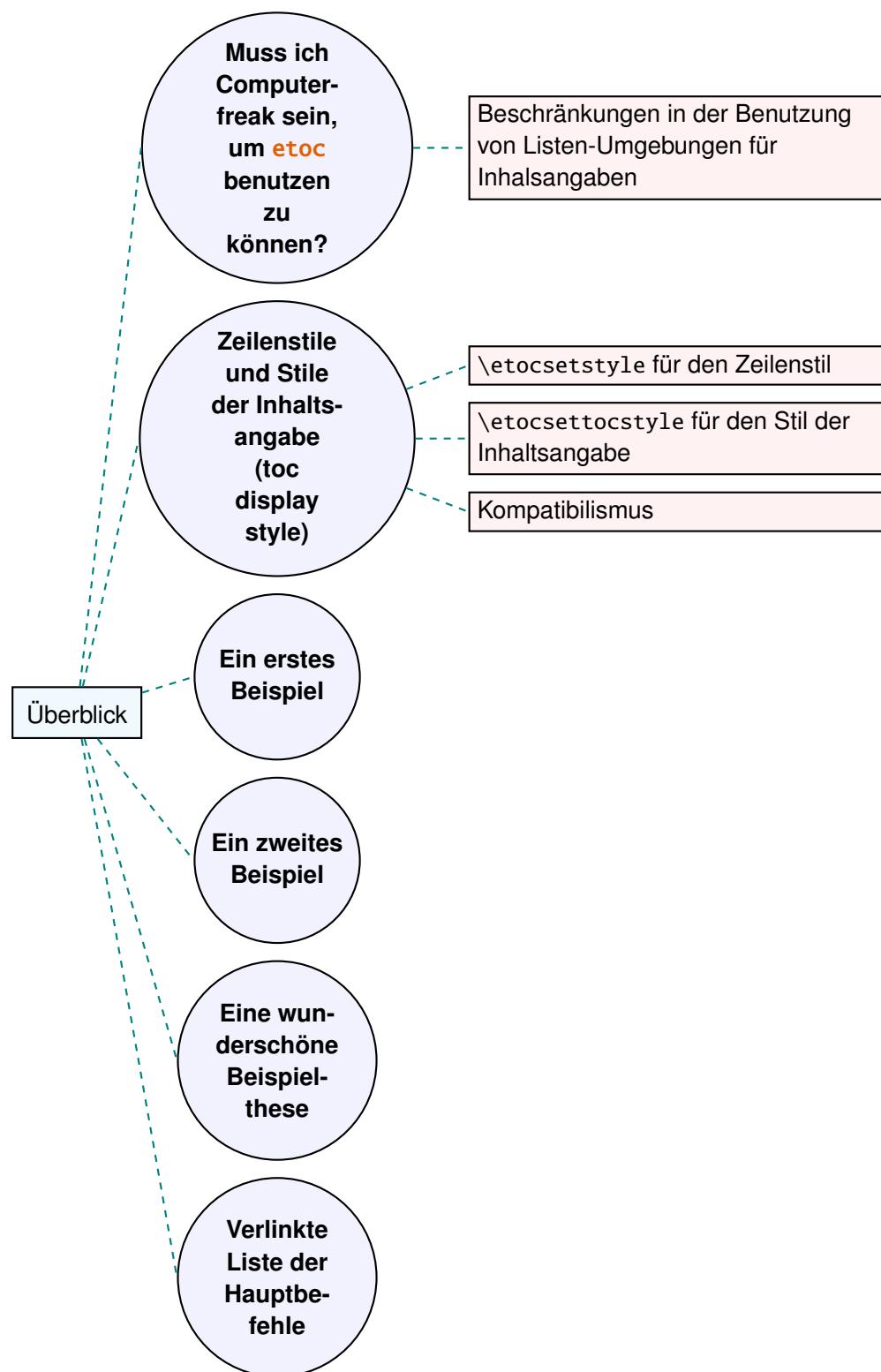
---

<sup>33</sup><http://ctan.org/pkg/pgf>

<sup>34</sup><http://ctan.org/pkg/forest>

<sup>35</sup>Dieses Handbuch bis zu der Version vom 14. Mai 2013 verwendete das Paket `tikz-qtree`, aber es gab Probleme mit den Hyper-Links. Kein solches Problem entsteht mit `forest` oder mit der nativen Syntax `tikz` für Bäume (die im nächsten Abschnitt dargestellt sind).

<sup>36</sup>Es gibt immer ein `\par`, welches kein Problem ist, es kann aber unterdrückt werden, wenn der Bedarf über den Befehl `\etocinline` oder dessen Synonym `\etocnopar` besteht.



Der resultierende Baum wird in einem **Puffer** angelegt, welcher on the preceding page erscheint. Hier ist der zur Herstellung notwendige Code:

```
% \newtoks\treetok % move this (uncommented) to preamble
% \newtoks\tmptok

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

\newcommand*\PrepareSectionNode{%
  \tmptok {\centering\bfseries}%
  \appendtotok\tmptok\etocthelinkedname
  \edef\foresttreenode{ [\noexpand\parbox{2cm}{\the\tmptok}]}%
}

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

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

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

\etocsettocstyle
{\treetok{{\hyperref[toc:overview]{Überblick}}}}
{\global\appendtotok\treetok{ }}

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

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

## 28. Das TOC als ein Molekül

```
@\the\treetok  
\end{forest}  
\end{figure}
```

Warum `\hypersetup{hidelinks}`? Wie in [Unterabschnitt 18.1](#) erläutert ist, werden die Links in Inhaltsverzeichnissen nicht eingefärbt, noch gerahmt, so dass dieses Dokument den Befehl `\hypersetup{hidelinks}` hinzufügt. Aber immer werden die `\treetok`-Inhalte in die `\hyperlink`-Ursprungsbefehle in `\etocthelinkedname` ausgepackt, etc. .... Sie werden in der normalen Umgebung für Links (die in diesem Dokument zu kolorieren sind) ausgeführt. Der `etoc`-Code versucht zu erraten, was der aktuelle „Stil“ für Links ist (ein Begriff, der nicht wirklich von `hyperref` vorausgesetzt zu sein scheint) und speichert es in `\etocthelinkedname`, etc. .... Ich entschied mich für die einfachere Lösung, es dem Benutzer zu überlassen, ob er es neu erstellt, was auch immer für Bedingungen erwünscht sind. So, hier ist es notwendig, das `\hypersetup{hidelinks}` in dem `figure`-Umgebung neu zu erstellen.

Es gibt einige andere Beispiele in dieser Dokumentation, wobei `\tableofcontents` verwendet wird, um Material zu haben für den späteren Schriftsatz:

- Für das Drucken der Statistiken zu Beginn der einzelnen Abschnitte (siehe [Abschnitt 52](#)) wurden unbeschädigte Boxen benutzt (das Problem der Darstellung der Links funktioniert nicht).
- Zum Setzen des Inhaltsverzeichnisses als eine Tabelle mit der Methode von [Abschnitt 54](#); auch dort werden wir `\hypersetup{hidelinks}` setzen, nachdem wir die Namen, Nummern und Seitennummern in einem symbolischen Listenregister gesetzt haben.
- Und zwei weitere Baum-Beispiele sind im nächsten Abschnitt.

## 28. Das TOC als ein Molekül

Es ist auch möglich, einen Inhaltsverzeichnisbaum nach der TikZ-Syntax für Bäume zu konstruieren: Aber das ist eine kompliziertere Aufgabe für die `etoc` Linienstile, welche die Klammern in einer Art und Weise von `TeX` behandeln. (Sie brauchen für das Auffüllen der symbolischen Liste im Zusammenhang mit jedem Schritt immer eine Balance, sonst müssen immer kompliziertere Tricks angewandt werden.).

Die einfachste Strategie ist es, eine symbolische Liste zuzuweisen (oder Sie verwenden ein Makro) für jede Ebene: Wir brauchen ein `\parttok`, ein `\chaptortok`, ein `\sectiontok` und ein `\subsectiontok`, um in der Aufgabe die `\treetok` zu befüllen. Wir interessieren uns hier für die Tabelle der Inhalte von diesem (oder einem anderen) Dokumententeil, nur ein `\sectiontok` und ein `\subsectiontok` wird dafür gebraucht.

```
% \newtoks\treetok % put this uncommented in preamble  
% \newtoks\sectiontok  
% \newtoks\subsectiontok  
  
\newcommand*{\treenode}{}  
  
\newcommand*{\appendchildtree}[2]{% token list t1 becomes: t1 child {t2}
```

```

\edef\tmp{\the#1 child {\the#2}%
#1\expandafter{\tmp}%
}
\newcommand*\preparerenode{%
\tmptok\expandafter{\etocthelinkednumber}% expanded one time (mandatory)
\edef\treenode{node {\the\tmpTok}%
}

\etocsetstyle{section}
{\etocskipfirstprefix
{\appendchildtree\treetok\sectionTok}
{\preparerenode \sectionTok\expandafter{\treenode}}
{\appendchildtree\treetok\sectionTok}

\etocsetstyle{subsection}
{\etocskipfirstprefix
{\appendchildtree\sectionTok\subsectionTok}
{\preparerenode \subsectionTok\expandafter{\treenode}}
{\appendchildtree\sectionTok\subsectionTok}

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

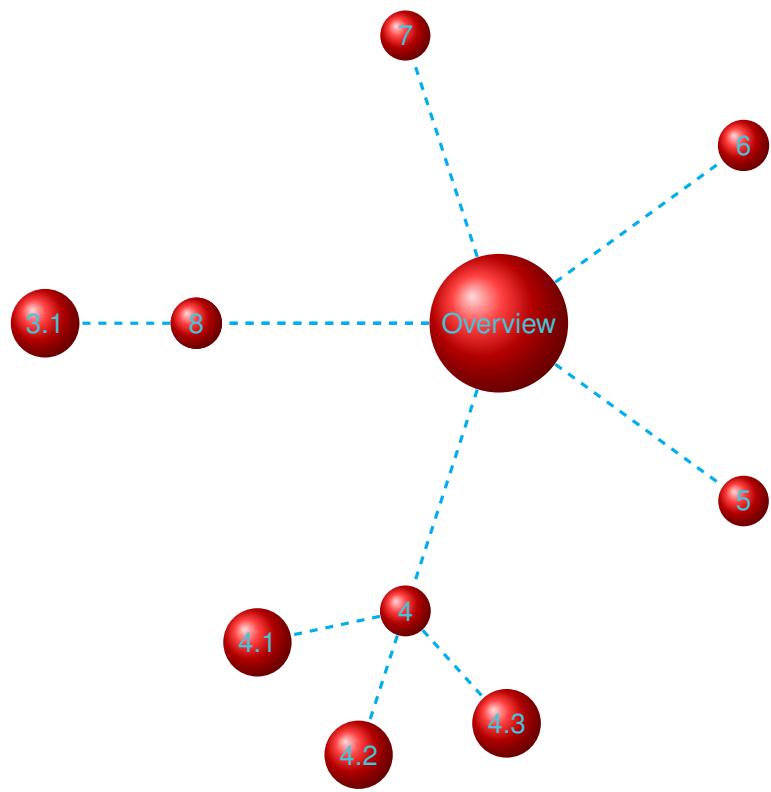
\begin{figure}[thp!]\centering
\etocsetnexttocdepth{subsection}
\tableofcontents \label{toc:molecule}\ref{toc:overview}
\hypersetup{hidelinks}%
\begin{tikzpicture}
[grow cyclic,
 level 1/.style={level distance=4cm,sibling angle=72},
 level 2/.style={level distance=2cm,sibling angle=60},
 every node/.style={ball color=red,circle,text=SkyBlue},
 edge from parent path={[dashed,very thick,color=cyan]
 (\tikzparentnode) -(\tikzchildnode)}]
 \the\treetok
\end{tikzpicture}
\centeredline{Ein Inhaltsverzeichnis als ein "Molek\"ul".}
\end{figure}

```

Der `\tableofcontents`-Befehl erscheint direkt über der Umgebung in dem `tikzpicture`-Umfeld und ist eine Gleitfigur (über Inhaltslabel sicherlich auf die gleiche Seite zu beziehen, wie die in dem das Bild tatsächlich gedruckt werden soll). Wir erhalten somit das Inhaltsverzeichnis als „[Molek\"ul](#)“, welches oben erscheint on the following page.

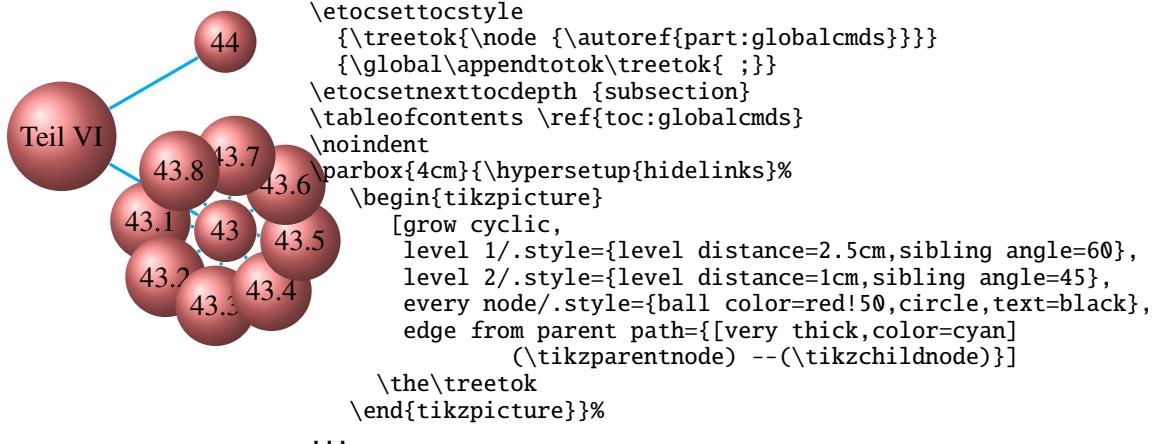
Dieses [TikZ-TOC](#) ist komplett verlinkt, wie das vorherige [Forest-TOC](#) auch.

28. Das TOC als ein Molekül



Ein Inhaltsverzeichnis als ein „Molekül“.

Seitlich ist das (völlig mit Hyper-Links versehene) Inhaltsverzeichnis von [Teil VI](#).



## 29. Das TOC als ein TikZ Mind-Map

Das ist in dem gleichen Sinn wie das „Molekül“-Beispiel. Die Verwendung des  $\epsilon$ - $\text{\TeX}$ -Primitivs- $\backslash\text{\unexpanded}$  vereinfacht den Code.<sup>37</sup>

Es ist schwierig, alles auf eine Seite einzupassen. Da kommt  $\text{\resizebox}$  zur Hilfe, das auch Hyperlinks bewahrt. Dennoch habe ich für dieses Beispiel einige Abschnitte von der Anzeige ausgeschlossen, dafür nehmend die Technik von [etoc depth tags](#). Das vollverlinkte TOC erscheint hier on the next page.

```
\begingroup
% \newtoks\treetok % in preamble
% \newtoks\parttok % in preamble

\newcommand*\partnode {} % just to check we don't overwrite something
\newcommand*\childnode {}

\newcommand*\tmprotate {} % just to check we don't overwrite something
\newcommand*\tmpoption {} % just to check we don't overwrite something
\newcommand*\tmpstuff {} % just to check we don't overwrite something

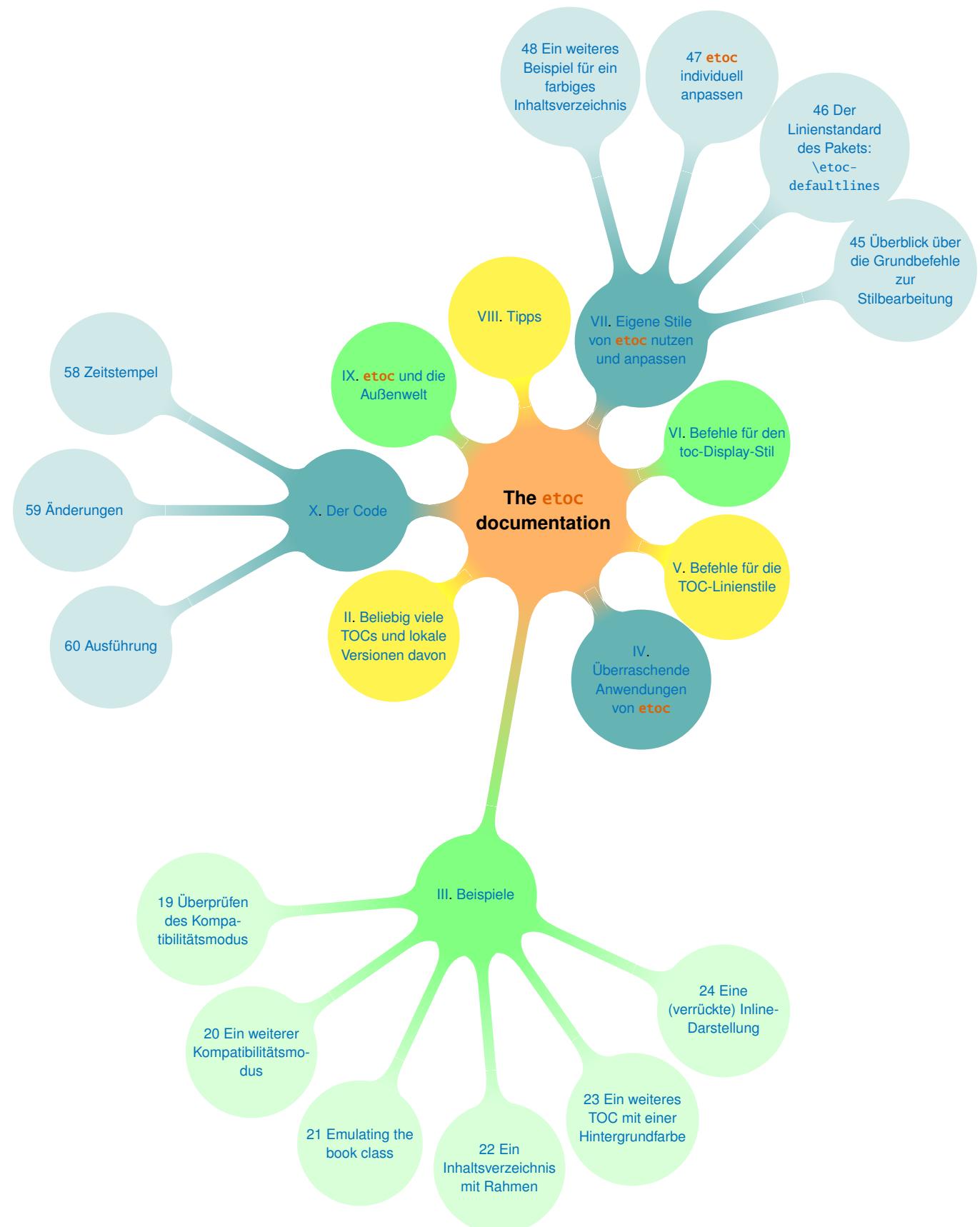
\newcommand*\appendtotok[2]{% #1=toks variable, #2=macro, expands once #2
  #1\expandafter\expandafter\expandafter{\expandafter\the\expandafter #1#2}%

\newcommand*\appendchildtree[3]{%
% this is to construct "t1 child [#3]{t2}" from #1=t1 and #2=t2
% t1 and t2 are two toks variable (not macros)
% #3 = for example teal!60
  \edef\tmpstuff {\the#1 child [#3]{\the#2}}%
  #1\expandafter {\tmpstuff }%
}
```

---

<sup>37</sup>das „Molekül“-Beispiel wurde am 2015/03/11 in diese Dokumentation aufgenommen . Das „Mindmap“-Beispiel wurde motiviert von <http://tex.stackexchange.com/a/232584/4686>. Weitere Hilfe kam damals via <http://tex.stackexchange.com/q/232816/4686> und dies führte zu dem weiteren Beispiel [title page](#).

## 29. Das TOC als ein TikZ Mind-Map



```

\newcounter{partco}

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

\etocsetstyle{part}
  {\etocskipfirstprefix}
% This updates the global tree with the data from the previous
% part and all its children sections. Moved here because for some parts the
% sections are not displayed due to depth tags.
{\ifnum\value{partco}=3
  \appendchildtree\treetok\parttok {branch color= green!50,level distance=10cm}%
\else
\ifcase\pseudomodthree{\value{partco}}%
  \or \appendchildtree\treetok\parttok {branch color= teal!60}%
  first
  \or \appendchildtree\treetok\parttok {branch color= yellow!80}%
  second
  \else\appendchildtree\treetok\parttok {branch color= green!50}%
  third and next ...
\fi\fi
}
{\stepcounter{partco}%
% customize manually some TikZ set-up (should be done inside the TikZ thing I guess)
\def\tmoption {}%
\def\tmrotate {}%
\ifnum\value{partco}=5 \def\tmrotate {[counterclockwise from =-40]}\fi
\ifnum\value{partco}=8 \def\tmrotate {[counterclockwise from =-50]}\fi
% define the part node
\edef\partnode{\node \tmoption
  {\unexpanded\expandafter{\etoctheboxednumber}.
   \unexpanded\expandafter{\etoctheboxedname}}\tmrotate }%
% this is a starting point which will be filled it by the section children
\parttok\expandafter{\partnode}%
\ifcase\pseudomodthree{\value{partco}}%
  \or \appendchildtree\treetok\parttok {branch color= teal!60}%
  first
  \or \appendchildtree\treetok\parttok {branch color= yellow!80}%
  second
  \else\appendchildtree\treetok\parttok {branch color= green!50}%
  third and next ...
\fi
}

\etocsetstyle{section}
{%
  %
  %
  % define the section node
  \edef\childnode{\child {node {\unexpanded\expandafter{\etoctheboxednumber}
    \unexpanded\expandafter{\etoctheboxedname}}}}%
  % append it to the current \parttok
  \appendtotok\parttok\childnode
  %
  %
}

\etocsettocstyle
{\setcounter{partco}{0}%
 \treetok{\node [root concept]{\textbf{The \etoc documentation}}}%
 \global\appendtotok\treetok{ ;}%
% The \global above is mandatory because etoc always typesets TOC inside a group

\etocsetnexttocdepth{section}
% use of depth tags to cut out some sections.

```

## 29. Das TOC als ein TikZ Mind-Map

```
\etocsettagdepth {preamble}    {none}
\etocsettagdepth {overview}   {part}
\etocsettagdepth {arbitrarily}{part}
\etocsettagdepth {examples}   {section}
\etocsettagdepth {surprising} {part}
\etocsettagdepth {linestyles} {part}
\etocsettagdepth {globalcmds} {part}
\etocsettagdepth {custom}     {section}
\etocsettagdepth {tips}       {part}
\etocsettagdepth {etocandworld}{part}
\etocsettagdepth {code}       {section}

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

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

Eine interessante Alternative zur Verwendung von **etoc** ist, das gesamte TOC in einen TikZ-Baum zu konvertieren (vielleicht mit Ausnahme einiger Teile) und in eine Datei zu drucken, von der es zurückgewonnen und durch den Autor direkt bearbeitet werden kann. Dinge, stehend in der .log-Datei, verschaffen zerbrochene Zeilen. Hier gibt es eine Technik, um zu nicht zerbrechenden Zeilen zu kommen. Sobald \treetok von **etoc** berechnet wurde (wie im **molecule**-Beispiel oder dem aktuellen Beispiel), wird dieses folgende Code-Schnipsel ausgegeben zur Dateierweiterung

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

Der Autor kann sie von dort aus kopieren und manuell anpassen, um ein geignetes TikZ-Bild zu erhalten. Siehe auch

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

## 30. Das TOC als eine Tabelle

Mit der Veröffentlichung-1.08 ist es einfacher geworden, das TOC als Tabelle zu setzen. Es ist nun möglich, eine Tabelle im Titel-Teil des TOC zu eröffnen (das erste Argument zu `\etocsettocstyle`) und zu schließen nach dem Inhalt (zweites Argument von `\etocsettocstyle`) und es ist auch möglich, Linienstile wie in Tabularen & und Tabellen am Ende der Zeilen \\ zu nehmen. Es gibt einige Bedingungen und ein paar Einschränkungen:

1. Es ist notwendig, `\etocglobaldefs` auszugeben für `etoc`-Definitionen, um einen globalen Skopus zu haben.
2. Es ist unmöglich, mit einer `<start>`-, `<prefix>`-, `<contents>`- oder `<finish>`-Spezifikation mit einer einzigen `\hline` zu beginnen, d. h. eine nicht mit einem \\ vorausgehende (es ist aber möglich zu setzen \\ am Ende eines `<prefix>` und von `\hline` am Beginn von `<contents>`).
3. So ist als nächstes zu erläutern, dass es empfehlenswert ist, \\ am Beginn der `<prefix>`- oder `<contents>`-Spezifikationen zu verwenden, um die *vorherige* Reihe abzuschließen, nicht um die *aktuelle* Reihe zu beenden; wenn das TOC ein partielles (ein `\localtableofcontents`) ist, wird dieser Ratschlag zur Pflicht (in fast allen Situationen).

Hier ein Beispiel eines TOCs als ein longtable.

TABLE OF CONTENTS		
1	Zusammenfassung	6
2	Vorwort	6
2	Lizenz	7
I	<b>Überblick</b>	8
3	Muss ich Computerfreak sein, um <code>etoc</code> benutzen zu können?	8
	3.1 <i>Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben</i>	9
4	Zeilenstile und Stile der Inhaltsangabe (toc display style)	9
	4.1 <i>\etocsetstyle für den Zeilenstil</i>	9
	4.2 <i>\etocsettocstyle für den Stil der Inhaltsangabe</i>	9
	4.3 <i>Kompatibilismus</i>	10
5	Ein erstes Beispiel	10
6	Ein zweites Beispiel	12
7	Eine wunderschöne Beispielthese	14
8	Verlinkte Liste der Hauptbefehle	15
II	<b>Beliebig viele TOCs und lokale Versionen davon</b>	16

### 30. Das TOC als eine Tabelle

<b>9</b>	Die \tableofcontents-, \localtableofcontents- und \localtableofcontents-withrelativedepth -Befehle	16
<b>10</b>	Beschriften und anderswo wieder benutzen	17
<b>11</b>	Eine mächtige Funktionsweise von <b>etoc</b> : Die Wiederzuweisung von Levels mit von \etoc-setlevel	18
<b>12</b>	Die \etocsettocdepth- und \etocsetnexttocdepth-Befehle 12.1 <i>Die hyperref-Option bookmarksdepth</i>	19
<b>13</b>	Der Befehl \etocsettocdepth.toc 13.1 <i>Die Befehle \etocbeytocdepth und \etocignoretoctocdepth</i>	21
<b>14</b>	Die Befehle \etocdepthtag.toc und \etocsettagdepth 14.1 <i>Die Befehle \etocbeydepthtags und \etocignoredepthtags</i>	21
<b>15</b>	Die Befehle \etocglobaldefs und \etoclocaldefs	22
<b>16</b>	Not displayed empty TOCs 16.1 <i>The \etocchecksemptiness command</i> 16.2 <i>The \etocnotocifnotoc command</i> 16.3 <i>The \etocifwasempty command</i>	23
<b>17</b>	The \etoclocaltop command	24
<b>18</b>	Befehle zur .toc-Datei hinzufügen 18.1 <i>Die hyperref-Option hidelinks</i>	25
<b>III</b>	<b>Beispiele</b>	27
<b>19</b>	Überprüfen des Kompatibilitätsmodus	27
<b>20</b>	Ein weiterer Kompatibilitätsmodus	28
<b>21</b>	Emulating the book class	30
<b>22</b>	Ein Inhaltsverzeichnis mit Rahmen	34
<b>23</b>	Ein weiteres TOC mit einer Hintergrundfarbe	34
<b>24</b>	Eine (verrückte) Inline-Darstellung	38
<b>IV</b>	<b>Überraschende Anwendungen von etoc</b>	40
<b>25</b>	Das TOC für TOCs	40
<b>26</b>	Willkürliche „Listen von ...“, \etoc toc contentsline	42
<b>27</b>	Das TOC als ein Baum	43
<b>28</b>	Das TOC als ein Molekül	46
<b>29</b>	Das TOC als ein TikZ Mind-Map	49
<b>30</b>	Das TOC als eine Tabelle	53
<b>31</b>	A TOC self-adjusting widths for its typesetting	58
<b>V</b>	<b>Befehle für die TOC-Linienstile</b>	61

<b>32</b>	Die \etocsetstyle-, \etocname- und \etocpage-Befehle	61
<b>33</b>	Die \etocskipfirstprefix und \etociffirst Befehle	63
<b>34</b>	Der \etocnumber-Befehl	63
<b>35</b>	Der \etocifnumbered-Schalter	63
<b>36</b>	Die Befehle \etocthename, \etocthenumber sowie \etocthechapter	64
<b>37</b>	Der \etoclink-Befehl	64
<b>38</b>	Die Befehle \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage und \etocthelink	65
<b>39</b>	Die \etocsavedsectiontocline, usw... Befehle.	66
<b>40</b>	Der \etocsetlevel-Befehl	66
<b>41</b>	Umfang der zur .toc-Datei gehörigen Befehle 41.1 <i>Testen des Bereichs</i> 41.2 <i>Dies ist eine (blass)rote Subsection zu Demonstrationszwecken</i>	67
<b>42</b>	Bin ich auch rot?	68
<b>VI</b>	<b>Befehle für den toc-Display-Stil</b>	69
<b>43</b>	Festlegung des toc-Display-Stils 43.1 <i>Der Befehl \etocsettocstyle</i> 43.2 <i>Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol</i> 43.3 <i>Der Befehl \etoc tocstyle</i> 43.4 <i>Die Befehle \etocruledstyle, \etocruled und \etoclocalruled</i> 43.5 <i>Die Befehle \etocframedstyle, \etocframed und \etoclocalframed</i> 43.6 <i>Überschriften, Titel, \etocoldpar, \etocinnertopsep</i> 43.7 <i>Der Kompatibilitätsmodus \etocstandarddisplaystyle</i> 43.8 <i>Der Befehl \etoc inline</i>	69
<b>44</b>	Starred variants and hooks	73
<b>VII</b>	<b>Eigene Stile von etoc nutzen und anpassen</b>	74
<b>45</b>	Überblick über die Grundbefehle zur Stilbearbeitung 45.1 <i>Ein lokales Inhaltsverzeichnis anpassen</i> 45.2 <i>Das Aussehen eines Inhaltsverzeichnisses anpassen</i>	74

### 30. Das TOC als eine Tabelle

	45.3 <i>Inhaltsverzeichnisse darstellen</i>	75
	45.4 <i>Label und Referenzen</i>	75
<b>46</b>	Der Linienstandard des Pakets: \etoc-defaultlines	76
<b>47</b>	<b>etoc</b> individuell anpassen	80
	47.1 <i>Den durch <b>etoc</b> vordefinierten Linienstil individualisieren</i>	80
	47.2 <i>Die Darstellung eines Inhaltsverzeichnisses anpassen</i>	81
<b>48</b>	Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis	82
<b>VIII</b>	<b>Tipps</b>	85
<b>49</b>	The \etocsetlocaltop.toc command	85
<b>50</b>	Hacken „framed parboxes“	86
<b>51</b>	Umstellen der Ebenen	87
	51.1 <i>Alle Unterabschnitte in diesem Dokument</i>	87
<b>52</b>	Statistiken anzeigen	88
<b>53</b>	Tiefe Tags nehmen	90
<b>54</b>	Das TOC als eine Tabelle setzend (die alte Weise)	92
<b>IX</b>	<b>etoc und die Außenwelt</b>	95
<b>55</b>	Kompatibilität mit anderen Paketen	95
	55.1 <i>Generelle Informationen</i>	95
	55.2 <i>Kompatibilität mit beamer</i>	95
	55.3 <i>Kompatibilität mit Babel</i>	96
	55.4 <i>Kompatibilität mit hyperref</i>	96
	55.5 <i>Kompatibilität mit multicols</i>	96
	55.6 <i>Kompatibilität mit tocloft</i>	96
	55.7 <i>Kompatibilität mit der memoir-Klasse</i>	96
	55.8 <i>Kompatibilität mit tocvsec2</i>	97
	55.9 <i>Kompatibilität mit tableof</i>	98
	55.10 <i>Kompatibilität mit tocstyle</i>	98
<b>56</b>	TeX-nische Sachen	98
<b>57</b>	Fehler und Katastrophen	99
<b>X</b>	<b>Der Code</b>	100
<b>58</b>	Zeitstempel	100
<b>59</b>	Änderungen	100
<b>60</b>	Ausführung	103

```
\begingroup
\etocglobaldefs % notwendig für \etocname etc... um zu überleben &
% Beachten Sie die Positionierung der \\
\etocsetstyle{part}
{ }
{\hline}
{\strut\etocnumber &\bfseries\etocname}
```

```

&\etocpage }

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

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

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

\etocsetnexttocdepth {subsection}

\tableofcontents
\endgroup

```

Beim Untersuchen des Codes wird sich der Leser fragen, warum die \\ immer zuerst in *<prefix+contents>* und nicht, wie es intuitiver ist, ziemlich später gegeben wird. In vielen Fällen kann man in der Tat die TOC-als-Tabellen mit \\, die am Ende der Stilspezifikationen lokalisiert sind (aber fast nie für ein \localtableofcontents) konstruieren. Das Problem bei dem vorhergehenden Beispiel war die Positionierung der \hline's.

Aufgrund der technischen Aspekte, wie TEX die Ausrichtung jeder Definition oder Festsetzung definiert, nach einem \\ beginnt eine neue Reihe und so erzeugt \hline ein illegales Token (das sieht aus wie ein misplaced \noalign Fehler). etoc hat nicht nur solche Definitionen um \etocname etc. ... zu konstruieren, sondern es ist auch der Fall, dass einige Pakete Dinge in die .toc-Datei tun und in der Folge gibt es keine Garantie dafür, dass zwischen zwei \contentsline kein solches Token wie \relax ist, wie in den Kontexten von „alignments forces“ macht es TEX unmöglich, um eine Zelle zu beginnen, dann setzen Sie eine \hline.

Der sicherste Weg ist, jede Linienspezifikation zu beginnen mit einem \\. Wir hatten ein kleines Problem mit dem Fakt, dass wir Teile wollten, die nicht nur eine Regel darüber haben (einfach, sie machen \\\\hline), sondern auch darunter: Nach jedem Teil ist ein Abschnitt, und es sind diese Abschnitte, die verwendet werden, um die fehlende \hline einzufügen (dies wird unter Vorbehalt mit Hilfe der \etociffirst-Bedingung getan).

Der aufmerksame Leser wird eine letzte Feinheit bemerken: Dem Titel, so wurde vermutet, sollten nicht eine sondern zwei \hline's folgen. Um dieses Problem zu lösen, könnten wir ein einmaliges Makro verwenden, das sich redefiniert, aber wir bemerkten, dass dem

### 31. A TOC self-adjusting widths for its typesetting

Titel ein nicht nummeriertes Abstrakt folgte, mit dem `\etocifnumbered`-Schalter kam die Rettung.

Eine letzte technische Anmerkung: Weil wir `\\"` in die Äste setzen setzen, bestand keine Notwendigkeit, die erweiterten Varianten `\etocxifffirst` und `\etocxifnumbered` einzustellen.

Für die Hardliner ist der alte Weg zur Erlangung des exakt gleichen Resultats hier zu sehen [Abschnitt 54](#).

Hier ist auch ein viel einfacheres Beispiel. Es ist ein lokales Inhaltsverzeichnis.

Abschnitt	Nummer	Seite
Muss ich Computerfreak sein, um <code>etoc</code> benutzen zu können?	3	8
Zeilenstile und Stile der Inhaltsangabe (toc display style)	4	9
Ein erstes Beispiel	5	10
Ein zweites Beispiel	6	12
Eine wunderschöne Beispiethese	7	14
Verlinkte Liste der Hauptbefehle	8	15

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

\etocsettocstyle
  {\hypersetup{hidelinks}\begin{tabular}{|p{4cm}|c|c|}\hline
   \multicolumn{1}{|c|}{\bfseries Abschnitt}&
   \bfseries Nummer&
   \bfseries Seite\\
   \hline\end{tabular}}
\etocglobaldefs % VERPFLICHTEND !!
\etocsetnexttocdepth{1}

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

## 31. A TOC self-adjusting widths for its typesetting

This is a continuation of [section 21](#). The goal is to adjust automatically the “numwidths” used for typesetting the unit numbers in the (local) tables of contents.

```

\makeatletter
\newcommand*\TOCcompute@numwidths [2]{% #1=empty/"local", #2=minimal indent
  \begingroup
    \def\TOCnumwidthB {0pt}%
    \def\TOCnumwidthC {0pt}%
    \def\TOCnumwidthD {0pt}%
    \def\TOCnumwidthE {0pt}%
    \def\TOCnumwidthF {0pt}%
    \def\TOCnumwidthG {0pt}%
    \etocsetstyle{part}{}{}{}{}{%
      {\setbox0\hbox{\bfseries\etocthenumber\kern#2}}
      {\ifdim\wd0>\TOCnumwidthB\edef\TOCnumwidthB{\the\wd0}\fi}{}%
    }
    \etocsetstyle{chapter}{}{%
      {\setbox0\hbox{\bfseries\etocthenumber\kern#2}}
      {\ifdim\wd0>\TOCnumwidthC\edef\TOCnumwidthC{\the\wd0}\fi}{}%
    }
    \etocsetstyle{section}{}{%
      {\setbox0\hbox{\bfseries\etocthenumber\kern#2}}
      {\ifdim\wd0>\TOCnumwidthD\edef\TOCnumwidthD{\the\wd0}\fi}{}%
    }
    \etocsetstyle{subsection}{}{%
      {\setbox0\hbox{\etocthenumber\kern#2}}
      {\ifdim\wd0>\TOCnumwidthE\edef\TOCnumwidthE{\the\wd0}\fi}{}%
    }
    \etocsetstyle{subsubsection}{}{%
      {\setbox0\hbox{\etocthenumber\kern#2}}
      {\ifdim\wd0>\TOCnumwidthF\edef\TOCnumwidthF{\the\wd0}\fi}{}%
    }
    \etocsetstyle{paragraph}{}{%
      {\setbox0\hbox{\etocthenumber\kern#2}}
      {\ifdim\wd0>\TOCnumwidthG\edef\TOCnumwidthG{\the\wd0}\fi}{}%
    }
    %
    \etocsettocstyle{}{%
      \global\let\TOCnumwidthB\TOCnumwidthB
      \global\let\TOCnumwidthC\TOCnumwidthC
      \global\let\TOCnumwidthD\TOCnumwidthD
      \global\let\TOCnumwidthE\TOCnumwidthE
      \global\let\TOCnumwidthF\TOCnumwidthF
      \global\let\TOCnumwidthG\TOCnumwidthG}%
    \etocnopar
    \csname #1tableofcontents\endcsname
  \typeout{Next TOCs will use \TOCnumwidthB\space for chapter number width}%
  \typeout{Next TOCs will use \TOCnumwidthC\space for section number width}%
  \typeout{Next TOCs will use \TOCnumwidthD\space for subsection number width}%
  \typeout{Next TOCs will use \TOCnumwidthE\space for subsubsection number width}%
  \typeout{Next TOCs will use \TOCnumwidthF\space for paragraph number width}%
  \typeout{Next TOCs will use \TOCnumwidthG\space for subparagraph number width}%
  \endgroup
}%
\newcommand*\TOCcomputenumwidths [1][0.5em]{%
  \TOCcompute@numwidths {}{#1}%
}%
\newcommand*\TOCcomputelocalnumwidths [1][0.5em]{%
  \TOCcompute@numwidths {local}{#1}%
}%
\makeatother

```

The optional parameter to `\TOCcomputenumwidths` specifies the minimal indent. In case nothing is numbered you may wish a higher value than `0.5em`. For each local ta-

### 31. A TOC self-adjusting widths for its typesetting

ble of contents to have its own width computations, the macro `\TOCcomputelocalnumwidths` is provided. As the code makes global assignments, either use (once) `\TOCcomputenumwidths` or do `\TOCcomputelocalnumwidths` for each local table of contents.

```
\TOCcomputelocalnumwidths % may use optional argument to replace 0.5em  
\localtableofcontents
```

Notes:

1. naturally these are only suggestions. For example one could put everything in single macros `\TOCtoc` and `\TOClocaltoc` to simultaneously compute the numwidths and then typeset the (local) table of contents.
2. if you want to adjust the `tocdepth` recall from [subsection 12.1](#) that it influences `hyperref` hence you may need to use a group `\begingroup... \endgroup`. Or, one can use `\etocsetnexttocdepth{\langle level \rangle}` but (with the code as here) this must then be issued twice, once for `\TOCcomputelocalnumwidths`, once for `\localtableofcontents`.
3. the bold font serves above for both chapter and section numwidth computations, but the code from [section 21](#) uses `\bfseries` only in local TOCs. Thus the `\TOCcomputenumwidth` will set the parameter `\TOCnumwidthC` to a value slightly larger than needed in the main TOC. Hence the section style in `\TOCcompute@numwidths` should possibly insert the `\bfseries` in the box only after testing for the optional parameter `local`.

# Teil V.

## Befehle für die TOC-Liniestile

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 11 Abschnitte und 2 Unterabschnitte. Der Name des ersten Abschnitts lautet „Die \etocsetstyle-, \etocname- und \etocpage-Befehle“ und er hat die Gliederungsnummer „32“. Der Name des letzten Abschnitts lautet „Bin ich auch rot?“ und er hat die Nummer „42“. Der erste Unterabschnitt heißt „Testen des Bereichs“ und er hat die Nummer „41.1“. Der Name des letzten Unterabschnitts ist „Dies ist eine (blass)rote Subsection zu Demonstrationszwecken“ und dessen Nummer ist „41.2“.

### Inhalt von Teil V

- 32** Die \etocsetstyle-, \etocname- und \etocpage-Befehle (Seite 61)
- 33** Die \etocskipfirstprefix und \etociffirst Befehle (Seite 63)
- 34** Der \etocnumber-Befehl (Seite 63)
- 35** Der \etocifnumbered-Schalter (Seite 63)
- 36** Die Befehle \etocthename, \etocthenumber sowie \etoctheplace (Seite 64)
- 37** Der \etoclink-Befehl (Seite 64)
- 38** Die Befehle \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage und \etocthelink (Seite 65)
- 39** Die \etocsavedsectiontocline, usw... Befehle. (Seite 66)
- 40** Der \etocsetlevel-Befehl (Seite 66)
- 41** Umfang der zur .toc-Datei gehörigen Befehle (Seite 67)
  - 41.1 Testen des Bereichs (S. 67)
  - 41.2 Dies ist eine (blass)rote Subsection zu Demonstrationszwecken (S. 67)
- 42** Bin ich auch rot? (Seite 68)

### 32. Die \etocsetstyle-, \etocname- und \etocpage-Befehle

Hier soll erläutert werden, wie **etoc** dazu verwendet wurde, das Inhaltsverzeichnis am Beginn dieses **Teil V**s zu erstellen. Es handelt sich dabei um ein lokales Inhaltsverzeichnis, weshalb der Befehl \localtableofcontents verwendet wurde.

## 32. Die \etocsetstyle-, \etocname- und \etocpage-Befehle

Wir müssen hierbei unterscheiden zwischen den Linienstilen (*line styles*) und dem Ausgabestil (*toc display style*). Die *line styles* erhält man grundsätzlich auf die folgende Art und Weise:

[38](#)

```
\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]}
```

Diese provisorischen Stil-Definitionen beruhen auf der automatischen Nummerierung, die von der `enumerate`-Umgebung generiert wird, es ist aber wesentlich besser den Befehl `\etocnumber` innerhalb des Item-labels zu verwenden, denn das gibt das richtige Ergebnis. Die verbesserten Definitionen werden dahingehend später erklärt.

Mit diesem Stil müsste man so einfallsreich sein, etwas zu designen, was nicht mit Abschnitten oder Unterabschnitten zu tun hat, vielleicht so etwas wie Superscripte. Nun, normalerweise braucht man Abschnitte und Unterabschnitte nicht nummeriert und aufgelistet in der TOC, also hat unser vermeintlicher Benutzer ein Design gewählt, welches dafür nichts bereitstellt und fügte noch hinzu:

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

Das ist der gleiche Sachverhalt wie mit den Standardlinienstilen!

Jeder `\etocsetstyle`-Befehl hat fünf notwendige Argumente:

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

Die `\<levelname\>`'s sind die Abschnittseinteilungen der Standard-Dokument-Klassen: Von `part` (oder `book`, welches von der `memoir`-Klasse benutzt wird) herunter bis zu `subparagraph`.

Der `\<start\>`-Code wird ausgeführt, wenn ein toc-Eintrag dieses Levels gefunden wurde und der vorherige ein höheres Level hatte. Der `\<finish\>`-Code wird ausgeführt, sobald man wieder einen Eintrag mit einem höheren toc-Level findet. In der Zwischenzeit werden alle

---

<sup>38</sup>Das vorliegende Dokument hat `\renewcommand{\familydefault}{\sfdefault}` in seiner Präambel, deshalb schaltet der Befehl `\normalfont` zur sans Schriftart; aus diesem Grund habe ich im Abschnitt `line-style \rmfamily` geschrieben.

Einträge für dieses Level gesetzt, indem zuerst der `<prefix>`-Code und dann der `<contents>`-Code ausgeführt wird.

Die (robusten) Befehle `\etocname`, `\etocnumber` und `\etocpage` sind für die Benutzung innerhalb der `<prefix>` und `<contents>`-Teile aus der `\etocsetstyle`-Spezifikation. Diese enthalten den Namen, die Nummerierung und die Seitennummer des entsprechenden toc-Eintrags. Ist das Paket `hyperref` im Dokument eingebunden, und es existieren Verlinkungen zu den TOC-Daten, dann werden diese Links in den Befehlen `\etocname`, `\etocnumber` und `\etocpage` gehalten (Der letzte hat nur einen Link, wenn `hyperref` die Option `linktoc=all` übergeben wurde).

### 33. Die `\etocskipfirstprefix` und `\etociffirst` Befehle

Der gewählte subsubsection-Stil machte dies, angewiesen von dem Befehl `\etocskipfirstprefix`, der `etoc` befiebt, für den ersten Punkt *nicht* den angepassten `<prefix>`-Code zu nehmen.<sup>39</sup>

Der Befehl `\etociffirst{<YES CODE>}{{<NO CODE>}}` (neu mit 1.08) ist eine flexiblere Möglichkeit, die `<prefix>`- (und `<contents>`)-Spezifikationen anzupassen. Er führt die `<YES CODE>`-Verzweigung aus, wenn dies die erste Einheit auf dieser Ebene (innerhalb einer Unterstufe) ist und `<NO CODE>`, wenn nicht. Dies ist ein robuster Befehl, der Erweiterungen (beispielsweise in einem `enumitem`-Label) überlebt.

Die Variante `\etocxiffirst` macht das Gleiche, aber sie ist erweiterbar.

### 34. Der `\etocnumber`-Befehl

Bis hierhin würde unsere Spezifikation die Nummerierung verwenden, die von der `enumerate`-Umgebung generiert wird, jedoch möchten wir natürlich die in der `.toc`-Datei deklarierten Nummern benutzen. Das kann man mit Hilfe des `\etocnumber`-Befehls erreichen. Um die Bezeichnungen in die `enumerate`-Auflistung zu bekommen, fährt man mit der Syntax `label=` des `enumitem`-Pakets fort:

```
\etocsetstyle{section}
{\begin{enumerate}[label=\etocnumber]}
{\normalsize\bfseries\rmfamily\item}
{\etocname{} (page \etocpage)}
{\end{enumerate}}
```

Lieber als nur `\etocnumber` haben wir dann so etwas wie `\fbox{\etocnumber}` benutzt. Dabei ist festzuhalten, dass `\etocnumber` ein robuster Befehl ist, was erklärt, warum er in der Label-Spezifikation benutzt werden kann, ohne `\protect` hinzuzufügen.

### 35. Der `\etocifnumbered`-Schalter

Die `\fbox` würde ein unästhetisches Ergebnis bringen im Fall eines nicht nummerierten Abschnitts (welches beispielsweise im Inhaltsverzeichnis gelandet ist mittels eines `\add-`

---

<sup>39</sup>Mit der Version vor 1.08 musste es das letzte Zeichen im `<start>`-Code sein. Es kann dort nun überall erscheinen.

## 37. Der \etoclink-Befehl

contentsline Befehls).<sup>40</sup>

Der \etocifnumbered{\langle A \rangle}{\langle B \rangle}-Befehl führt \langle A \rangle aus, wenn die Nummer existiert, und \langle B \rangle falls nicht. Also benutzen wir es in dem Code, welcher letztendlich für das section-Level gewählt wurde:

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

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

Wenn wir nur das section-Level geändert hätten und nicht das subsection-Level, dann wäre ein Kompilierfehler aufgetreten, denn der Paket-Stil für die „subsections“ erwartet dann ‘im Vertikal-Modus’ zu starten. Ein zusätzliches \par-Token im \langle contents \rangle-Teil des section-Levels hätte das behoben: \dots (page \etocpage) \par}.

Der Befehl \etocifnumbered ist robust; \etocxifnumbered (neu für 1.08) hat den gleichen Effekt, aber er ist expandierbar.

## 36. Die Befehle \etocname, \etocnumber sowie \etocpage

Es ist manchmal wünschenswert, Zugriff auf Name, Nummer und Seitenzahl zu haben, ohne die hyperref-Daten: Ähnlich wie die \*-Variante des \ref-Befehls, wenn das hyperref-Paket verwendet wird. Beispielsweise könnte man den Wunsch haben, eine Einheit oder Seitenzahl in einer Art numerischem Kontext verwenden zu wollen oder seine Formatisierung zu ändern. Diese Funktionalität wird von \etocname, \etocnumber und \etocpage bereitgestellt.

Diese Befehle sind aber nicht geschützt, also sollten sie in veränderlichen Kontexten (zum Beispiel Label-Beschreibungen mit vorangestelltem \protect) verwendet werden.

## 37. Der \etoclink-Befehl

Der Befehl \etoclink{\langle linkname \rangle} kann in Linienstil-Beschreibungen benutzt werden, in ähnlicher Art und Weise wie die argumentlosen Befehle \etocname, \etocnumber und \etocpage. Er erzeugt einen Link (falls solch ein von hyperref zur .toc-Datei hinzugefügt wurde), dessen Ziel der entsprechende Dokumentteil ist und dessen Name das angegebene Argument ist. Daher ist \etoclink{\etocname} das ursprüngliche \etocname. Es ist darauf zu achten, dass wenn hyperref nicht damit beauftragt wurde, die Seitenzahl

---

<sup>40</sup>Wie man sieht, wird \fcolorbox verwendet statt \fbox. Aufgrund einer Neudefinition – durchgeführt vom xcolor-Paket – haben wir \fbox verwendet (und nicht hyperref), sonst hätten wir \protect\fbox benötigt.

mit einem Link zu versehen (mittels seiner Option `linktoc=all`), dann hat **etoc**'s `\etoc-page` auch keinen Link, `\etoclink{\etocthePage}` allerdings schon.

Der Befehl `\etoclink` ist robust.

Since `etoc 1.08j` it contains the link destination in an already expanded form, so for example can be used even after a & in a tabular construction, if `\etocglobaldefs` was issued.

## 38. Die Befehle `\etoctheLinkedName`, `\etoctheLinkedNumber`, `\etoctheLinkedPage` und `\etoctheLink`

These macros are fragile and contain the link destination in already expanded form. So their meaning can be stored for delayed usage. For example this is done here when doing the [examples with trees](#).

Notice though that for example `\etoctheLinkedNumber` is really a link only if `hyperref` was configured to let the numbered heading be linked (which is the default). Similarly for the others.

`\etoctheLink` will fetch the link destination if available from name or page number and allows via `\etoctheLink{foo}` to link an arbitrary text to the same destination inside the produced PDF.

The `\etocname`, `\etocnumber`, `\etocpage` and `\etoclink` are simply their robust variants. They too contain the hyperlink destination in already expanded form since `1.08a` for the first three, since `1.08j` for `\etoclink`.

Veralteter Beispiel (ein Inhaltsverzeichnis als eine Inline-enumitem-Umgebung): **43.** Festlegung des toc-Display-Stils (S. 69), und **44.** Starred variants and hooks (S. 73).

```
Veralteter Beispiel\begin{group}
\etocsetstyle {section}
  {\begin{itemize*}[itemjoin={{; }, itemjoin*={{, und }}}]
    \global\let\TmpEtocNumber\etoctheLinkedNumber
    \global\let\TmpEtocName\etoctheLinkedName
    \global\let\TmpEtocPage\etoctheLinkedPage }
  % Obiges ist erforderlich, da \item, hier in itemize*, eine Gruppe schließt.
  {\item [{\bfseries\TmpEtocNumber}]\TmpEtocName\
    (\emph{S. \TmpEtocPage})}
  \end{itemize*} }%
\etocsetnexttocdepth {section}%
\etocsettocstyle {(ein Inhaltsverzeichnis als eine Inline-%
  \texttt{enumitem}-Umgebung): }{}%
\etocinline\tableofcontents \ref{toc:globalcmds}
\end{group}
```

Mit **etoc** 1.08 gibt es ein einfacher Weg mit Hilfe von `\etocglobaldefs` zum gleichen Resultat:

```
Beispiel\begin{group}\etocglobaldefs % <-- NEU mit etoc 1.08.
\etocsetstyle {section}
  {\begin{itemize*}[itemjoin={{; }, itemjoin*={{, und }}}]
    {}
    {\item [{\bfseries\etocnumber}]}
  \end{itemize*} }
```

#### 40. Der `\etocsetlevel`-Befehl

```
\etocname\ (\emph{S. \etocpage })}  
\end{itemize*}.}%  
\etocsetnexttocdepth {section}%  
\etocsettocstyle {(ein Inhaltsverzeichnis als eine Inline-%  
    \texttt{enumitem}-Umgebung): }{}%  
\etocinline\tableofcontents \ref{toc:globalcmds}  
\endgroup
```

Beispiel(ein Inhaltsverzeichnis als eine Inline-`enumitem`-Umgebung): **43.** Festlegung des toc-Display-Stils (S. 69), und **44.** Starred variants and hooks (S. 73).

### 39. Die `\etocsavedsectiontocline`, usw... Befehle.

Since 1.08k `etoc` preserves automatically in `\etocsavedsectiontocline`, `\etocsavedchaptertoscline` etc... the meaning of the L<sup>A</sup>T<sub>E</sub>X `\l@section`, `\l@chapter`, etc... in force at the time each `\tableofcontents` or `\localtableofcontents` macro is executed.

### 40. Der `\etocsetlevel`-Befehl

Wie bereits in [Teil IV](#) erklärt, kann man `etoc` mitteilen, welches Level er mit einem bestimmten Abschnitts-Befehl assoziieren soll, nämlich mit `\etocsetlevel`. Zum Beispiel:

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

Im Kompatibilitäts-Modus wird davon ausgegangen, dass die Makros `\l@cell`, `\l@molecule`, ... entweder durch den Benutzer oder von einer anderen Klasse definiert wurden: Nur `\etocsetlevel` ist nicht genug, um auf dem korrespondierenden Pegel „out-of-the-box“ zu arbeiten.

Jedoch, ist im Kompatibilitätsmodus kein Inhaltsverzeichnis gesetzt, dann ist nur wichtig, dass die verschiedenen Linienstile gesetzt sind. Wenn beispielsweise `section` auf Level 1 ist, dann muss man nicht mehr `\etocsetstyle{molecule}{...}{...}{...}` nach `\etocsetstyle{molecule}{1}` aufrufen, da `\etocsetstyle{section}{...}{...}{...}` `{...}` bereits wirksam ist (und es wurde im Paket selbst in der Definition seines Linienstils gesetzt).

Die akzeptierten Level laufen von -2 bis einschließlich 6. Alles andere wird auf 6 abgebildet, welches ein Dummy-Level ist und nie dargestellt wird. Das Paket tut Folgendes:

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

Die benutzerdefinierten Stile von **etoc** werden von \etocdefaultlines aktiviert. Sie werden vom Haupt-Inhaltsverzeichnis dieses Dokuments dargestellt. Die Level-Zuweisungen können jederzeit modifiziert werden: In [Teil IV](#) sieht man verschiedene Anwendungen dieser Technik.

Als ein weiteres Beispiel soll hier angeführt werden, wie das [Haupt-Inhaltsverzeichnis](#) dieses Dokuments gesetzt wurde, nämlich mit den folgenden Instruktionen:

```
\etocsettocdepth {subsubsection} % set the initial tocdepth
\etocdefaultlines % use the package default line styles. At this early stage in
                  % the document they had not yet been modified by \etocsetstyle
                  % commands, so \etoc toc lines could have been used, too.

\etocmarkboth\contentsname
\etocmulticolstyle[1]           % one-column display
  {\pdfbookmark[1]{Table of contents}{MAINTOC}}% create a bookmark in the pdf
    \noindent\bfseries\Large
    \leaders\hrule height1pt\hfill
    \MakeUppercase{Table of Contents}
\begin{group}                   % use a group to limit the scope of the
  \etocsetlevel{subsection}{3}    % subsection level change.
  \etocsetlevel{subsubsection}{4}  % anything > tocdepth=3.
  \tableofcontents \label{toc:main}
\end{group}
```

Auf diese Art und Weise benutzen die „subsections“ die Stile, die ursprünglich für „subsubsections“ vorgesehen waren, während diese gar nicht gedruckt werden. Ohne diese Modifikation wäre das Erscheinungsbild ein anderes gewesen: die Paketlinienstile sind dafür gedacht gewesen, in Dokumenten mit vielen vielen „subsubsections“ verarbeitet zu werden, in einem Zwei-Spalten-Layout, um diesen ein kompakteres Ergebnis zu liefern als die Standard-L<sup>A</sup>T<sub>E</sub>X-Inhaltsverzeichnisse. Hier haben wir jedoch noch ein paar „subsubsections“, daher ist es interessanter, diese zu ignorieren und dafür „sections“, und „subsections“ visuell unterschiedlich zu drucken.

## 41. Umfang der zur .toc-Datei gehörigen Befehle

### 41.1. Testen des Bereichs

Lasst uns zur Farbe Rot wechseln und außerdem diesen Befehl zur .toc-Datei hinzufügen:

```
\color{red!50}                                % Ändern der Textfarbe
\addtocontents{toc}{\string\color{red!50}} % das Gleiche in der .toc-Datei
```

### 41.2. Dies ist eine (blass)rote Subsection zu Demonstrationszwecken

Gut, der Titel hier wurde schwarz dargestellt, was daran liegt, wie die `scrartcl`-Klasse funktioniert (in der `article`-Klasse wäre er rot gewesen), aber wir sind ohnehin mehr daran interessiert, wie er im Inhaltsverzeichnis aussieht: Und tatsächlich ist er rot im [Haupt-Inhaltsverzeichnis](#) dieses Dokuments, genau so wie im [Inhaltsverzeichnis für diesen Abschnitt](#). Beide Befehle gehorchen wie erwartet dem `\color{red!50}`-Befehl, der in die .toc-Datei geschrieben wurde.

Aber lasst uns jetzt diese „subsection“ beenden und eine „section“ starten.

42. Bin ich auch rot?

## 42. Bin ich auch rot?

Die Frage ist, ob es im Inhaltsverzeichnis erscheint: Die Antwort ist ja, es ist rot im Haupt-Inhaltsverzeichnis und nein es ist nicht rot im Inhaltsverzeichnis für diesen Abschnitt. Grund dafür ist, dass der `<finish>`-Code für dieses „subsection“-Level eine Gruppe geschlossen hat, da es `\end{enumerate}` verwendet hat.

Das zeigt noch einmal die Diskussion aus Unterabschnitt 3.1.

Die standardmäßigen Paket-line-Stile beinhalten keine Gruppen-Öffnung- und Schliess-Befehle: Die Gültigkeit eines Befehls in der .toc-Datei wird also solange bestehen, bis er explizit durch einen weiteren Befehl wieder rückgängig gemacht wird.

```
\normalcolor  
\addtocontents{toc}{\string\normalcolor}
```

Zurück zu Schwarz. Man beachte das diese Bereichsprobleme auch in mehrsprachigen Dokumenten auftauchen, da das babel-Paket auftretende Sprachänderungen ebenfalls in die .toc-Datei schreibt.

# Teil VI.

# Befehle für den toc-Display-Stil

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 2 Abschnitte und 8 Unterabschnitte. Der Name des ersten Abschnitts lautet „Festlegung des toc-Display-Stils“ und er hat die Gliederungsnummer „43“. Der Name des letzten Abschnitts lautet „Starred variants and hooks“ und er hat die Nummer „44“. Der erste Unterabschnitt heißt „Der Befehl \etocsettocstyle“ und er hat die Nummer „43.1“. Der Name des letzten Unterabschnitts ist „Der Befehl \etocinline“ und dessen Nummer ist „43.8“.

<b>43. Festlegung des toc-Display-Stils</b>	<b>69</b>
43.1. Der Befehl \etocsettocstyle . . . . .	69
43.2. Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol . . . . .	70
43.3. Der Befehl \etoctocstyle . . . . .	71
43.4. Die Befehle \etocruledstyle, \etocruled und \etoclocalruled . . . . .	71
43.5. Die Befehle \etocframedstyle, \etocframed und \etoclocalframed . . . . .	71
43.6. Überschriften, Titel, \etocoldpar, \etocinnertopsep . . . . .	72
43.7. Der Kompatibilitätsmodus \etocstandarddisplaystyle . . . . .	72
43.8. Der Befehl \etocinline . . . . .	73
<b>44. Starred variants and hooks</b>	<b>73</b>

## 43. Festlegung des toc-Display-Stils

Der *toc display*-Stil gibt an, ob das Inhaltsverzeichnis mit mehreren Spalten erscheint oder nur mit einer, ob der Titel gesetzt ist wie in der *article*- oder *book*-Klasse, oder ob er über den Einträgen zentriert sein soll, mit Regeln an der Seite, oder ob das gesamte Inhaltsverzeichnis in einen Rahmen gepackt werden soll. Entscheidet man sich bspw. für eine einheitliche Überschrift und ein Ein-Spalten-Layout, so verwendet man dafür folgende Befehle:

```
\etocruledstyle[1]{Title} \tableofcontents      (or \localtableofcontents)
shortcuts:           \etocruled[1]{Title}   (or \etoclocalruled[1]{Title})
```

### 43.1. Der Befehl \etocsettocstyle

Dies ist ein Befehl mit zwei notwendigen Argumenten:

```
\etocsettocstyle{\{before_toc\}}{\{after_toc\}}
```

Der {\{before\_toc\}}-Teil ist verantwortlich für den Schriftsatz der Überschriften, so kann er beispielsweise etwas sein wie \section\*{\contentsname}.

Generell gesprochen, sollte diese Überschrift *T<sub>E</sub>X* im „Vertikal-Modus“ lassen, wenn der eigentliche Schriftsatz des Inhalts beginnt: die Linienstile (entweder von den Standardklassen oder den Paketlinienstilen) erwarten alle im ‘Vertikal-Modus’ zu starten.

Das erste Argument von \etocsettocstyle kann auch Instruktionen beinhalten, um die Seitenüberschriften zu markieren. Oder es kann prüfen (*book*-Klasse), ob der Zwei-

### 43. Festlegung des toc-Display-Stils

Spalten-Modus eingeschaltet ist, um dann zum Ein-Spalten-Modus zu wechseln. Der `<after_toc>`-Teil würde dann wieder den Ein-Spalten-Modus aktivieren.

Die im Folgenden beschriebenen Befehle `\etocmulticolstyle`, `\etocruledstyle` und `\etocframedstyle` rufen alle `\etocsettocstyle` als Subroutine auf, um eine Mehrspalten-Umgebung mit Hilfe von `{<before_toc>}` zu starten, welche mit `{<after_toc>}` wieder geschlossen wird.

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

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

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

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

### 43.2. Die Befehle `\etocmulticolstyle`, `\etocmulticol` und `\etoclocalmulticol`

Dies ist ein Befehl mit einem optionalen und einem notwendigen Argument:

```
\etocmulticolstyle[<number_of_columns>]{<heading>}
```

Die `[<number_of_columns>]` können von 1 bis 10 gehen (default: 2; ist der Wert 1, so wird natürlich keine `multicols`-Umgebung angelegt). Das `<heading>` wird typischerweise ‘vertikaler’ Art sein, also wie: `<heading> = \section*{<title>}`, aber man kann auch horizontale Dinge realisieren wie `\fbox{Hallo Welt}` (`etoc` fügt automatisch ein `\par` ans Ende des „heading“-Arguments von `\etocmulticolstyle`). Hier ist einmal exemplarisch dargestellt, wie das Haupt-Inhaltsverzeichnis dieses Dokuments konfiguriert wurde:

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

Nach `\etocmulticolstyle` haben alle zukünftigen `\tableofcontents` den spezifizierten Stil, falls es zwischendurch nicht mehr geändert wird. Eine kürzere Variante für nur ein Inhaltsverzeichnis ohne Einfluss auf spätere ist:

```
\etocmulticol[<number_of_columns>]{<heading>}
```

oder

```
\etoclocalmulticol[<number_of_columns>]{<heading>}.
```

### 43.3. Der Befehl \etocstyle

```
\etocstyle[<kind>]{<number_of_columns>}{<title>}
=\etocmulticolumnstyle[number_of_columns]{\kind*{title}}
```

Wobei Typ gewählt wird aus chapter, section usw. Standardwert ist chapter oder section, abhängig von der Dokumentklasse.

#### 43.3.1. Der Befehl \etocstylewithmarks

```
\etocstylewithmarks[<kind>]{<number_of_columns>}{<title>}{<mark>}
=\etocmulticolumnstyle[number_of_columns]{\kind*{title \markboth{\MakeUppercase{mark}}}}
```

Wobei Typ gewählt wird aus chapter, section usw. Die wirkliche Anzeige der Markierungen („marks“) hängt vom Seitenstil ab. Es gibt die Variante `\etocstylewithmarksnouc`, welche Großschreibung unterdrückt.

##### 43.3.1.1. Möchten wir wirklich paragraph-Einträge im Inhaltsverzeichnis?

##### 43.3.1.2. Wirklich?

### 43.4. Die Befehle \etocruledstyle, \etocruled und \etoclocalruled

Das generelle Format von \etocruledstyle ist:

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

Der Titel ist horizontales Material (der LR-Modus von *L<sup>A</sup>T<sub>E</sub>X, ein Dokument-Vorbereitungssystem*,): wenn es nicht in eine Zeile passt, dann sollte es in einer \parbox mit einer gegebenen Weite geschrieben werden. Der grüne Rahmen für die Überschrift des Inhaltsverzeichnisses zu [Beginn des zweiten Teils dieses Dokuments](#) wurde erzeugt durch:

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

Eine kürzerer Weg den Stil zu setzen mit \etocruledstyle und dann einen \tableofcontents zu erstellen, alles zusammen in einer Gruppe damit zukünftige Inhaltsverzeichnisse nicht betroffen sind, ist:

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

### 43.5. Die Befehle \etocframedstyle, \etocframed und \etoclocalframed

Die gleiche Arbeitsweise:

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

und die passende Abkürzung:

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

Diese wird verwendet, wenn man nicht den Stil zukünftiger Inhaltsverzeichnisse modifizieren will (eine andere Möglichkeit wäre es, das Ganze in Klammern zu packen oder innerhalb eines \begingroup... \endgroup; es gibt auch noch einen \etoclocalframed für ein lokales Inhaltsverzeichnis).

## 43. Festlegung des toc-Display-Stils

Das gesamte Inhaltsverzeichnis ist eingerahmt. Der Titel selbst aber nicht: Möchte man das erreichen, dann muss man das im `<title>`-Argument mit `\etocframedstyle` oder `\etocframed` festhalten. Die Farben für den Hintergrund und der anderen Komponenten (oben, links, rechts, unten) des Rahmens sind festgelegt mittels geeigneter `\renewcommand`'s (siehe [Unterabschnitt 47.2](#)).

Eine `minipage` wird benutzt, von daher ist das produzierte Inhaltsverzeichnis nicht kompatibel mit Seitenumbrüchen. Um diese zu erlauben, wird empfohlen, die Befehle `mdframed`<sup>41</sup> oder `tcolorbox`<sup>42</sup> in den Argumenten von `\etocsettocstyle` zu verwenden.

Beispiele in diesem Dokument findet man auf den Seiten [18](#), [37](#), [35](#) und [83](#).

### 43.6. Überschriften, Titel, `\etocoldpar`, `\etocinnertopsep`

Das für `\etocmulticolstyle` notwendige `<heading>`-Argument kann entweder Vertikal-Modus-Material sein wie `\section*{\emph{Inhaltsverzeichnis}}` oder die horizontale Variante wie das einfache `\etocmulticolstyle{Hallo Welt}`.

Es kann kein explizites `\par` oder eine Leerzeile in das notwendige Argument von `\etocmulticolstyle` eingefügt werden, aber `etoc` stellt `\etocoldpar` als Ersatz bereit: Es fügt `\let\etocoldpar\par` vor der `multcols`-Umgebung ein und eben dieses `\etocoldpar`<sup>43</sup> am Ende der Überschrift ein, und macht dann noch einen vertikalen Sprung der Größe `\etocinnertopsep`.

Auf der anderen Seite erwarten die Befehle `\etocruledstyle` und `\etocframedstyle` ein Argument im „LR-Modus“ (um die Terminologie von *L<sup>A</sup>T<sub>E</sub>X, a document preparation system* zu nutzen). Das bedeutet, dass mehrzeilige Titel nur möglich sind, wenn man sie in etwas wie eine `\parbox` einschließt.

Eine wichtige Größe verwendet von sowohl `\etocmulticolstyle`, `\etocruledstyle` als auch `\etocframedstyle` ist `\etocinnertopsep`. Es gibt an, wieviel Platz zwischen den Überschriften und dem Start des Inhalts liegen soll. Der Default-Wert ist `2ex` und man kann es mittels `\renewcommand{\etocinnertopsep}{<neuer_Wert>}` ändern, nicht mit `\setlength`.

### 43.7. Der Kompatibilitätsmodus `\etocstandarddisplaystyle`

`etoc` wird dann das emulieren, für was eigentlich die Dokument-Klasse verantwortlich ist, in Bezug auf den globalen Display-Stil des Inhaltsverzeichnisses. Allen Änderungen innerhalb der Klasse sollten berücksichtigt werden.

#### 43.7.1. Die Befehle `\etocarticlestyle`, `\etocbookstyle`, ...

Dies sind Befehle, die abhängig von der Dokumentenklasse intern von `etoc` im Kompatibilitätsmodus verwendet werden. Beispielsweise nimmt `\etocarticlestyle \section*{\contentsname}` (und setzt Markierungen in die Seite), während `\etocbookstyle \chapter*{\contentsname}` nimmt. Es kann nützlich sein, `\etocarticlestyle` für ein

---

<sup>41</sup><http://ctan.org/pkg/mdframed>

<sup>42</sup><http://ctan.org/pkg/tcolorbox>

<sup>43</sup>Dieser Befehl `\etocoldpar` (= funktionierendes `\par` im Argument von `\etocmulticolstyle`) ist nicht verwandt mit dem Schalter `\etocinline`, dessen Zweck es ist `etoc` mitzuteilen, kein `\par` vor dem Inhaltsverzeichnis auszuführen.

\localtableofcontents in der book-Klasse und im Kompatibilitätsmodus für den globalen TOC-Anzeige-Stil zu nehmen.

### 43.8. Der Befehl \etocininline

Mit `\etocininline` oder seinem Synonym `\etocnopar`, von dem `\tableofcontents`-Befehl und seinen Varianten werden *nicht* zuerst ein `\par` erstellt, um den vorherigen `paragraph` zu schließen. Von daher kann man das Inhaltsverzeichnis in einem Inline-Stil darstellen; oder, wenn nur gebraucht um eine Token-Liste bzw. ein Makro vorzubereiten, wird es bei der Ausführung nichts im Token-Stream hinterlassen.

Mit `\etocdisplay` werden (nochmal) von dem `\tableofcontents`-Befehl und seinen Varianten zuerst ein `\par` erstellt.

## 44. Starred variants and hooks

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

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

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

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

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

With this set-up issuing `\localtableofcontents *` will ignore the `\etocaftertitlehook` hence not send the local toc title to the .toc file. This mimicks the `memoir` class behavior, and can also be used with it. For more on `memoir` class with `etoc`, see subsection 55.7.

There are further hook macros: `\etocaftercontentshook`, `\etocbeforetitlehook` and `\etocaftertochook` which are initially defined to do nothing and can be used for some special effects. They are executed whether or not the table of contents command was starred.<sup>44</sup>

---

<sup>44</sup>Hackers can test within these hooks if the context is starred or not via an `\ifx\Etoc@aftertitlehook\@empty` conditional. Perhaps a future release will provide `\etociftocisstarred` wrapper.

## Teil VII.

# Eigene Stile von **etoc** nutzen und anpassen

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 4 Abschnitte und 6 Unterabschnitte. Der Name des ersten Abschnitts lautet „Überblick über die Grundbefehle zur Stilbearbeitung“ und er hat die Gliederungsnummer „45“. Der Name des letzten Abschnitts lautet „Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis“ und er hat die Nummer „48“. Der erste Unterabschnitt heißt „Ein lokales Inhaltsverzeichnis anpassen“ und er hat die Nummer „45.1“. Der Name des letzten Unterabschnitts ist „Die Darstellung eines Inhaltsverzeichnisses anpassen“ und dessen Nummer ist „47.2“.

Dies ist das Inhaltsverzeichnis mit den Abschnitten und Unterabschnitten von diesem Teil. Es hat das Label `toc:c`.

<b>Überblick über die Grundbefehle zur Stilbearbeitung</b>	45, S. 74	<b>Der Linienstandard des Pakets: \etoc-defaultlines</b>	46, S. 76
Ein lokales Inhaltsverzeichnis anpassen	45.1, S. 74	<b>etoc individuell anpassen</b>	47, S. 80
Das Aussehen eines Inhaltsverzeichnisses anpassen	45.2, S. 75	Den durch <b>etoc</b> vordefinierten Linienstil individualisieren	47.1, S. 80
Inhaltsverzeichnisse darstellen	45.3, S. 75	Die Darstellung eines Inhaltsverzeichnisses anpassen	47.2, S. 81
Label und Referenzen	45.4, S. 75	<b>Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis</b>	48, S. 82

## 45. Überblick über die Grundbefehle zur Stilbearbeitung

### 45.1. Ein lokales Inhaltsverzeichnis anpassen

```
\etocsetstyle{\(Name der Ebene)}{\(Start)}{\(Präfix)}{\(Inhalt)}{\(Abschluss)}  
\etocname, \etocnumber, \etocpage, \etocifnumbered{\(A)}{\(B)}  
\etocthename, \etocthenumber, \etocthepage, \etoclink{\(Name des Links)}
```

## 45.2. Das Aussehen eines Inhaltsverzeichnisses anpassen

```
\etocmulticolstyle[<Anzahl der Spalten>]{<Überschrift>}
\etoctocstyle[<Ebene>]{<Anzahl der Spalten>}{<Titel>}
\etoctocstylewithmarks[<Ebene>]{<Anzahl der Spalten>}{<Titel>}{<Symbol>}
\etoctocstylewithmarksnouc[<Ebene>]{<Anzahl der Spalten>}{<Titel>}{<Symbol>}
\etocruledstyle[<Anzahl der Spalten>]{<Titel>}
\etocframedstyle[<Anzahl der Spalten>]{<Titel>}
\etocsettocstyle{<vor dem Inhaltsverzeichnis>}{<nach dem Inhaltsverzeichnis>}
```

## 45.3. Inhaltsverzeichnisse darstellen

```
\tableofcontents
\localtableofcontents
\etocmulticol[<Anzahl der Spalten>]{<Überschrift>}
\etoclocalmulticol[<Anzahl der Spalten>]{<Überschrift>}
\etocruled[<Anzahl der Spalten>]{<Titel>}
\etoclocalruled[<Anzahl der Spalten>]{<Titel>}
\etocframed[<Anzahl der Spalten>]{<Titel>}
\etoclocalframed[<Anzahl der Spalten>]{<Titel>}
und die mit Sternchen versehenen Varianten
```

## 45.4. Label und Referenzen

Optional dürfen die Befehle für Label oder Referenzen auf die Befehle, die die Darstellung des Inhaltsverzeichnisses realisieren, folgen. Hierbei ist unwichtig, ob letztere mit Sternchen versehen sind oder nicht.

```
\tableofcontents \label{toc:here}
\tableofcontents \ref{toc:far}
\tableofcontents \label{toc:here} \ref{toc:far}
\localtableofcontents \label{toc:here}
\localtableofcontents \ref{toc:far}
\localtableofcontents \label{toc:here} \ref{toc:far}
\etocmulticol, etc. funktionieren ähnlich
```

Neu mit 1.08e: `\localtableofcontents \ref{toc:far}` als Synonym für `\tableofcontents \ref{toc:far}`.

Wird ein anderes Inhaltsverzeichnis erneut dargestellt, so wird nur dessen Inhalt angepasst. Sowohl das Aussehen des Inhaltsverzeichnisses als auch die Darstellung seiner Linien orientieren sich an dem, das gerade definiert wurde und nicht an dem, von dem der Inhalt übernommen wurde.

## 46. Der Linienstandard des Pakets: \etocdefaultlines

Die Linienstile stammen bereits aus einer frühen Entwicklungsphase des Pakets, weswegen sie sich noch sehr an  $\text{\TeX}$  orientieren und dadurch gruselig aussehen. Daher bieten sie keinen guten Einstiegspunkt. Oft muss man, will man an dem Stil etwas ändern, diesen aus dem Quellcode kopieren und direkt bearbeiten. Trotzdem soll in diesem Abschnitt erläutert werden, wie die Schriftarten, die vertikalen Abstände, etc. des gewählten Liniestils verändert werden können.

Wurden die vorgegebenen Linienstile nicht durch  $\text{\etocsetstyle}$  verändert, so kann man sie mit den Befehlen  $\text{\etocdefaultlines}$  bzw.  $\text{\etocclines}$  einschalten. Außerdem, dass der Einzug eines Unterabschnitts etwas kleiner ist als der eines Abschnitts, gleichen sich beide Gliederungsebenen. (Wie dieses Dokument zeigt, werden Abschnitte in Dokumentklassen, denen der Befehl  $\text{\chapter}$  fehlt, stets mit einer fettgedruckten Type gesetzt.) Unter-Unterabschnitte werden inline, im selben Paragraphen und ohne Nummer oder Seitenzahl dargestellt. Dieser Stil wurde für Dokumente, die viele Unter-Unterabschnitte enthalten, entwickelt und anschließend mit solchen getestet. Für derartige Dokumente sollte ein zweispaltiges Layout verwendet werden, da diese Darstellung kompakter ist als jene, die standardmäßig durch  $\text{\LaTeX}$  erzeugt wird.<sup>45</sup> Verwendet man hingegen ein einspaltiges Layout und enthält der Teil nur wenige Unter-Unterabschnitte, so wird das Inhaltsverzeichnis im Vergleich zur mit  $\text{\LaTeX}$  gesetzten Variante etwas in die Breite gezogen. Weiterhin wirkt diese Darstellung weniger hieratisch als der  $\text{\LaTeX}$ -Standard, da sich die Darstellung von Abschnitten und Unterabschnitten kaum unterscheidet. (Dies gilt besonders, wenn die Dokumentklasse den Befehl  $\text{\chapter}$  erlaubt.)

Da wir für das zentrale Inhaltsverzeichnis  $\text{\etocsetlevel}{\text{\subsection}}{3}$  gesetzt haben, werden die Unterabschnitte und die Unter-Unterabschnitte inline dargestellt.

Im Gegensatz dazu wollen wir nun das zentrale Inhaltsverzeichnis so darstellen, als ob wir eine Dokumentklasse genutzt hätten, die den Befehl  $\text{\chapter}$  erlaubt. Daher werden die Abschnitte im Folgenden als Kapitel und die Unterabschnitte als Abschnitte gedruckt, was wir u. a. mit Hilfe des Befehls  $\text{\etocsetlevel}$  erreichen. Zusätzlich nutzen wir für Abschnitte, welche in Wirklichkeit unsere Unterabschnitte sind, statt der fetten Serie einer Schriftart die halbfette Variante. Hierfür modifizieren wir den Befehl  $\text{\etocfontone}$ . Weiterhin nutzen wir einen gepunkteten Einzug, da dieser weniger Platz einnimmt als der Paketstandard.

```
\etocruledstyle[2]{\normalfont\normalsize\rmfamily\itshape
\fbox{\parbox{.6\linewidth}{%
\leftskip 0pt plus .5fil
\rightskip 0pt plus -.5fil
\parfillskip 0pt plus 1fil
}}}
Dies ist das zentrale zweispaltige Inhaltsverzeichnis, das den
Standardlinienstil von \etoc nutzt, wobei die Abschnitte wie Kapitel und die
Unterabschnitte wie Abschnitte gesetzt wurden. }}}
```

$\text{\etocdefaultlines}$

```
\renewcommand{\etocpartname}{\partname}%
% <1.08b hat \partname als Standard, aber dies arbeitet nicht gut mit frenchb
% daher ist der Standard jetzt nun einfach Teil. Für die deutsche Sprache
% können wir \partname oder Teil nehmen.
```

---

<sup>45</sup>Außerdem werden dadurch einzelne Teil- bzw. Kapitaleinträge am Ende einer Seite oder einer Spalte vermieden, solange diese weiter untergliedert sind.

```

\etocsetnexttocdepth{1}
\begingroup
\etocsetlevel{section}{0}
\etocsetlevel{subsection}{1}
\renewcommand*\etocfontone{\normalfont \normalsize}
\renewcommand*\etocclineleaders{
    \hbox{\normalfont\normalsize\hbox to 1ex {\hss.\hss}}}
\sloppy
\tableofcontents
\endgroup

```

*Dies ist das zentrale zweispaltige Inhaltsverzeichnis, das den Standardlinienstil von **etoc** nutzt, wobei die Abschnitte wie Kapitel und die Unterabschnitte wie Abschnitte gesetzt wurden.*

## Zusammenfassung

### 1. Vorwort

### 2. Lizenz

### Teil I. Überblick

#### 3. Muss ich Computerfreak sein, um **etoc** benutzen zu können?

Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben ..... 3.1, S. 9

#### 4. Zeilenstile und Stile der Inhaltsangabe (toc display style)

\etocsetstyle für den Zeilenstil ..... 4.1, S. 9

\etocsettocstyle für den Stil der Inhaltsangabe ..... 4.2, S. 9

Kompatibilismus ..... 4.3, S. 10

### 5. Ein erstes Beispiel

### 6. Ein zweites Beispiel

### 7. Eine wunderschöne Beispieldatei

## 8. Verlinkte Liste der Hauptbefehle

### Teil II. Beliebig viele TOCs und lokale Versionen davon

#### 9. Die \tableofcontents-, \localtableofcontents- und \localtableofcontentswithrelativedepth - Befehle

#### 10. Beschriften und anderswo wieder benutzen

#### 11. Eine mächtige Funktionsweise von **etoc**: Die Wiederzuweisung von Levels mit von \etocsetlevel

#### 12. Die \etocsettocdepth- und \etocsetnexttocdepth-Befehle

Die hyperref-Option *bookmarksdepth* ..... 12.1, S. 20

#### 13. Der Befehl \etocsettocdepth.toc

Die Befehle \etocbeytotoctocdepth und \etocignoretocdepth ..... 13.1, S. 21

## 46. Der Linienstandard des Pakets: \etocdefaultlines

### 14. Die Befehle \etocdepthtag.toc und \etocsettagdepth

Die Befehle \etocbeydepthtags und \etocignoredepthtags ....14.1, S. 22

### 15. Die Befehle \etocglobaldefs und \etoclocaldefs

### 16. Not displayed empty TOCs

The \etoccheckemptiness command .....16.1, S. 23

The \etocnotocifnotoc command .....16.2, S. 23

The \etocifwasempty command .....16.3, S. 23

### 17. The \etoclocaltop command

### 18. Befehle zur .toc-Datei hinzufügen

Die hyperref-Option *hidelinks* .....18.1, S. 25

## Teil III. Beispiele

### 19. Überprüfen des Kompatibilitätsmodus

### 20. Ein weiterer Kompatibilitätsmodus

### 21. Emulating the book class

### 22. Ein Inhaltsverzeichnis mit Rahmen

### 23. Ein weiteres TOC mit einer Hintergrundfarbe

### 24. Eine (verrückte) Inline-Darstellung

## Teil IV. Überraschende Anwendungen von etoc

### 25. Das TOC für TOCs

### 26. Willkürliche „Listen von ...“, \etoccontentsline

### 27. Das TOC als ein Baum

### 28. Das TOC als ein Molekül

### 29. Das TOC als ein TikZ Mind-Map

### 30. Das TOC als eine Tabelle

### 31. A TOC self-adjusting widths for its typesetting

## Teil V. Befehle für die TOC-Linienstile

### 32. Die \etocsetstyle-, \etocname- und \etocpage-Befehle

### 33. Die \etocskipfirstprefix und \etociffirst Befehle

### 34. Der \etocnumber-Befehl

### 35. Der \etocifnumbered-Schalter

### 36. Die Befehle \etocthename, \etocthenumber sowie \etocthechapter

### 37. Der \etoclink-Befehl

### 38. Die Befehle \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage und \etocthelink

### 39. Die \etocsavedsectiontocline, usw... Befehle.

### 40. Der \etocsetlevel-Befehl

### 41. Umfang der zur .toc-Datei gehörigen Befehle

Testen des Bereichs .....41.1, S. 67

Dies ist eine (blass)rote Subsection zu Demonstrationszwecken ...41.2, S. 67

## 42. Bin ich auch rot?

### Teil VI. Befehle für den toc-Display-Stil

#### 43. Festlegung des toc-Display-Stils

Der Befehl \etocsettocstyle .....	43.1, S. 69
Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol ..	43.2, S. 70
Der Befehl \etoctocstyle ..	43.3, S. 71
Die Befehle \etocruledstyle, \etocruled und \etoclocalruled ..	43.4, S. 71
Die Befehle \etocframedstyle, \etocframed und \etoclocalframed ..	43.5, S. 71
Überschriften, Titel, \etocoldpar, \etocinnertopsep .....	43.6, S. 72
Der Kompatibilitätsmodus \etocstandarddisplaystyle .....	43.7, S. 72
Der Befehl \etocinline ..	43.8, S. 73

#### 44. Starred variants and hooks

### Teil VII. Eigene Stile von etoc nutzen und anpassen

#### 45. Überblick über die Grundbefehle zur Stilbearbeitung

Ein lokales Inhaltsverzeichnis anpassen .....	45.1, S. 74
Das Aussehen eines Inhaltsverzeichnisses anpassen .....	45.2, S. 75
Inhaltsverzeichnisse darstellen .....	45.3, S. 75
Label und Referenzen .....	45.4, S. 75

### 46. Der Linienstandard des Pakets: \etocdefaultlines

#### 47. etoc individuell anpassen

Den durch etoc vordefinierten Linienstil individualisieren .....	47.1, S. 80
Die Darstellung eines Inhaltsverzeichnisses anpassen .....	47.2, S. 81

#### 48. Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis

### Teil VIII. Tipps

#### 49. The \etocsetlocaltop.toc command

#### 50. Hacken „framed parboxes“

#### 51. Umstellen der Ebenen

Alle Unterabschnitte in diesem Dokument .....	51.1, S. 87
---	-------------

#### 52. Statistiken anzeigen

#### 53. Tiefe Tags nehmen

#### 54. Das TOC als eine Tabelle setzend (die alte Weise)

### Teil IX. etoc und die Außenwelt

#### 55. Kompatibilität mit anderen Paketen

Generelle Informationen .....	55.1, S. 95
Kompatibilität mit beamer .....	55.2, S. 95
Kompatibilität mit Babel .....	55.3, S. 96
Kompatibilität mit hyperref .....	55.4, S. 96
Kompatibilität mit multicol .....	55.5, S. 96
Kompatibilität mit tocloft .....	55.6, S. 96
Kompatibilität mit der memoir-Klasse ..	55.7, S. 96

## 47. **etoc** individuell anpassen

Kompatibilität mit tocvsec2 ..55.8, S. 97	<b>Teil X. Der Code</b>
Kompatibilität mit tableof ..55.9, S. 98	
Kompatibilität mit tocstyle .....	<b>58. Zeitstempel</b>
..... 55.10, S. 98	
<b>56. <math>\text{\TeX}</math>-nische Sachen</b>	<b>59. Änderungen</b>
<b>57. Fehler und Katastrophen</b>	<b>60. Ausführung</b>

## 47. **etoc** individuell anpassen

### 47.1. Den durch **etoc** vordefinierten Linienstil individualisieren

Im Folgenden sollen einfach die wichtigsten Befehle des Pakets aufgelistet werden. Man kann diese bequem mittels \renewcommand anpassen:

```
\newcommand*\etocfontminustwo{\normalfont \LARGE \bfseries}
\newcommand*\etocfontminusone{\normalfont \large \bfseries}
\newcommand*\etocfontzero{\normalfont \large \bfseries}
\newcommand*\etocfontone{\normalfont \normalsize \bfseries}
% (in classes with chapter, \etocfontone does not do \bfseries)
\newcommand*\etocfonttwo{\normalfont \normalsize}
\newcommand*\etocfontthree{\normalfont \footnotesize}

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

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

\newcommand*\etocbaselinespreadminustwo{1}
\newcommand*\etocbaselinespreadminusone{1}
\newcommand*\etocbaselinespreadzero{1}
\newcommand*\etocbaselinespreadone{1}
\newcommand*\etocbaselinespreadtwo{1}
\newcommand*\etocbaselinespreadthree{.9}
\newcommand*\etoctoclineleaders
    {\hbox{\normalfont\normalsize\hbox to 2ex {\hss.\hss}}}
\newcommand*\etocabbrevpagename{p.~} % muss der Sprache angepasst werden
                                % deutsche Nutzer: \renewcommand*\etocabbrevpagename{S.~}
\newcommand*\etocpartname{Part} % früher, vor v1.08b, war \partname
% aber dies macht keinen Sinn, z.B. mit babel+frenchb deren \frenchpartname
% nimmt in den Account den Wert des Teilzählers.
\newcommand*\etocbookname{Book} % muss der Sprache angepasst werden
```

Der Standardlinienstil kann innerhalb der **etoc** -Umgebung nicht individualisiert werden. Wie bereits erwähnt, wird das Paket nach Aufruf des Befehls kaum verwendet und

## 47.2. Die Darstellung eines Inhaltsverzeichnisses anpassen

agiert lediglich auf globaler Ebene, sodass die Einträge des Inhaltsverzeichnisses (hoffentlich) so dargestellt werden, als ob das Paket **etoc** fehlen würde.<sup>46</sup>

Wurde mittels \etocsetlevel die Tiefe der Gliederung soweit herabgesetzt, dass **etoc** die Befehle für bestimmte Gliederungsebenen kennt, so wirkt der Kompatibilitätsmodus, also \etocstandardlines, auch auf diese. Dies gilt natürlich nur für Gliederungsbefehle, die dem Standard von L<sup>A</sup>T<sub>E</sub>X entsprechen und die alle Definitionen liefern, welche zum Darstellen von Einträgen im Inhaltsverzeichnis benötigt werden. (Es muss ein Makro der Art \l@irgendwas existieren.)

Sobald der Befehl \etocsetstyle aufgerufen wird, wird zusätzlich der Kompatibilitätsmodus ausgeschaltet. Es spielt hierbei keine Rolle, ob der Befehl in der Präambel oder im Hauptteil des Dokuments steht.

### 47.2. Die Darstellung eines Inhaltsverzeichnisses anpassen

Erneut wollen wir uns darauf beschränken die relevanten Makros aufzulisten, da deren Funktion aus ihrem Namen ableitbar ist. Soll die Länge eines Befehls verändert werden, so muss der entsprechende Befehl mittels \renewcommand und nicht durch \setlength verändert werden. Weiterhin erlauben Befehle, welche die Farbe verändern, nicht nur die Standardfarben. Will man ihr Verhalten ändern, so muss man in der neuen Definition den Befehl \color aufrufen. Um ihre Wirkung zu beenden, muss man sie so neu definieren, dass sie danach nur den Befehl \relax enthalten.

```
\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*\etocdefaultncol{2}
\newcommand*\etocinnertopsep{2ex}
\newcommand*\etoctoprule{\hrule}
\newcommand*\etoctoprulecolor{\relax}

% nur für gerahmte Inhaltsverzeichnisse:
\newcommand*\etocinnerleftsep{2em}
\newcommand*\etocinnersrightsep{2em}
\newcommand*\etocinnerbottomsep{3.5ex}

\newcommand*\etocleftrule{\vrule}
\newcommand*\etocrightrule{\vrule}
\newcommand*\etocbottomrule{\hrule}
\newcommand*\etocleftrulecolor{\relax}
\newcommand*\etocrightrulecolor{\relax}
\newcommand*\etocbottomrulecolor{\relax}

\newcommand*\etocbkgcolor{\relax}

% Haken
\newcommand\etocframedphook{\relax}
```

---

<sup>46</sup> Innerhalb der KOMA-script-Klassen muss der Befehl \etocstandarddisplaystyle genutzt werden, damit die Option toc=left auch für die Linieneinträge gilt.

## 48. Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis

Der Befehl \etocframedmphook ist direkt hinter dem Anfang einer Minipage, innerhalb der der Inhalt eines gerahmten Verzeichnisses gesetzt wird, verankert.

Wurden die Farbbefehle, also \...colorcmd, nicht verändert, so expandieren sie lediglich zu \relax. (Daher muss man für ihre Verwendung nicht die Pakete color oder xcolor einbinden.) Hat man jedoch einen solchen Befehl - wie zum Beispiel \etocbkg-colorcmd - so verändert, dass er nun tatsächlich eine Farbe anzeigt, so kann dies nur mittels \renewcommand{\etocbkgcolorcmd}{\relax} aufgehoben werden und nicht durch \let\etocbkgcolorcmd\relax.

Wie man an dem Beispiel

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

erkennen kann, werden die Größen des oberen Rechtecks in den Maßen ex und em angegeben.

Sobald der Titel während der Befehlsausführung von \etocruledtoc oder \etocframedtoc gesetzt wurde, nutzt der Paketcode dessen Schriftgröße, um die Bedeutung von „1ex“ zu ermitteln. Natürlich dürfen auch die anderen Befehle zur Erzeugung eines Rechtecks Maße annehmen, deren Wert sich nur durch eine zugehörige Schriftgröße ergibt. Hierbei dient jedoch die Schrift vor dem Inhaltsverzeichnis als Referenz.

Wie in T<sub>E</sub>X üblich, werden sowohl das Rechteck vor als auch das nach dem Inhalt des Verzeichnisses durch einfache Einzüge (einer festgelegten Höhe) realisiert. Es muss sich hierbei also nicht um echte Rechtecke handeln. Die Rechtecke links und rechts vom Inhalt werden hingegen nicht durch (horizontale) Einzüge verwirklicht. Sie stellen vielmehr Objekte einer bestimmten Weite dar. Daher darf hier *nur* die Syntax von T<sub>E</sub>X-Befehlen, mit deren Hilfe man Rechtecke erzeugt, genutzt werden.

## 48. Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis

Beim Befehl \etocframedstyle wird der Titel in der Mitte des oberen Rechtecks positioniert. Dies würde im folgenden Beispiel jedoch sehr unschön aussehen, weswegen wir den Titel innerhalb der *<start>*-Umgebung eines Abschnittsbefehls platzieren. Hierdurch wird er innerhalb des Rahmens gesetzt.

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

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

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

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

```

\renewcommand{\etocinnersrightsep}{0pt}

\newcommand\shiftedwhiterule[2]{%
  \hbox to \linewidth{\color{white}%
    \hspace{\skip#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 % Auch ohne Rahmen nutzt \colorbox den Befehl \fboxsep!
  \noindent\colorbox{secbackground}
  {\parbox{\linewidth}{%
    \vskip5pt
    {\noindent\color{#1}#2\par}\nointerlineskip
    \vskip3pt}}%
  \par\nointerlineskip
}

\etocsetstyle{section}
{\coloredstuff{white}
  {\hfil \hyperref[toc:b]{\bfseries\large Ich bin der Zwilling von
    diesem Verzeichnis (Klick mich an!)}\hfil}}
{\vskip3pt\sffamily\small
{\coloredstuff{white}
  {\leftskip1.5cm\rightskip.5cm\parfillskip-\rightskip
    \makebox[0pt][r]{\makebox[.5cm][r]{\etocnumber\hskip.2cm}}\%
    \etocname\hfill\makebox[.5cm][r]{\etocpage\hskip.2cm}}\%
  \vskip6pt }
{}}

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

```

Ich bin der Zwilling von diesem Verzeichnis (Klick mich an!)	
<b>43 Festlegung des toc-Display-Stils</b>	<b>69</b>
<b>43.1 Der Befehl \etocsettocstyle . . . . .</b>	<b>69</b>
<b>43.2 Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol . . . . .</b>	<b>70</b>
<b>43.3 Der Befehl \etoctocstyle . . . . .</b>	<b>71</b>
<b>43.4 Die Befehle \etocruledstyle, \etocruled und \etoclocalruled . . . . .</b>	<b>71</b>
<b>43.5 Die Befehle \etocframedstyle, \etocframed und \etoclocalframed . . . . .</b>	<b>71</b>
<b>43.6 Überschriften, Titel, \etocoldpar, \etocinnertopsep . . . . .</b>	<b>72</b>
<b>43.7 Der Kompatibilitätsmodus \etocstandarddisplaystyle . . . . .</b>	<b>72</b>
<b>43.8 Der Befehl \etocinline . . . . .</b>	<b>73</b>
<b>44 Starred variants and hooks</b>	<b>73</b>

Das Inhaltsverzeichnis befindet sich in einer `float`-Umgebung, die man on the preceding page findet. Da diese Art von Code keine zusätzlichen Pakete verlangt, ist er etwas komplexer.<sup>47</sup> Außerdem wurde er in einem früheren Stadium entwickelt und seither nicht überarbeitet.

Da Rahmen (welche hier genutzt wurden, um die Hintergrundfarbe zu setzen) nur die Hintergrundfarbe innerhalb einer Seite verändern, würde es sich anbieten, weitere Pakete einzubinden, mit deren Hilfe man die Hintergrundfarbe über mehrere Seiten hinweg verändern kann.

Alle Befehle, welche die Farbe verändern, sind in `etoc` zu Beginn so definiert, dass sie nichts verändern. Daher kann der Nutzer selbst entscheiden, ob er Farben nutzen will oder nicht.

---

<sup>47</sup>Zudem zeigt er die Vorliebe des Autors für die `\TeX`-Syntax.

# Teil VIII.

## Tipps

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 6 Abschnitte und 1 Unterabschnitt. Der Name des ersten Abschnitts lautet „The \etocsetlocaltop.toc command“ und er hat die Gliederungsnummer „49“. Der Name des letzten Abschnitts lautet „Das TOC als eine Tabelle setzend (die alte Weise)“ und er hat die Nummer „54“. Der erste Unterabschnitt heißt „Alle Unterabschnitte in diesem Dokument“ und er hat die Nummer „51.1“. Der Name des letzten Unterabschnitts ist „Alle Unterabschnitte in diesem Dokument“ und dessen Nummer ist „51.1“.

<b>49. The \etocsetlocaltop.toc command</b>	<b>85</b>
<b>50. Hacken „framed parboxes“</b>	<b>86</b>
<b>51. Umstellen der Ebenen</b>	<b>87</b>
51.1. Alle Unterabschnitte in diesem Dokument . . . . .	87
<b>52. Statistiken anzeigen</b>	<b>88</b>
<b>53. Tiefe Tags nehmen</b>	<b>90</b>
<b>54. Das TOC als eine Tabelle setzend (die alte Weise)</b>	<b>92</b>

### 49. The \etocsetlocaltop.toc command

It is important to understand that `\localtableofcontents` works entirely from data *in the .toc file*. If the document, say with `article` class, contains starred sectioning commands, which are not accompanied by suitable `\addcontentsline`, then these units are completely transparent to `\localtableofcontents`:

- If it is issued before `\section*{Foo}`, say locally to a `\section`, then the local TOC will include not only the subsections between the `\section` and the `\section*{Foo}` but also those following, and be stopped uniquely by a later `\section` or `\part` from the document’s body.
- If the command is issued right after `\section*{Foo}` and the later was itself subsequent to a (numbered) `\subsection`, then `etoc` will think it must display a TOC local to that `subsection` level.

Before concluding that `etoc` is buggy, imagine how the global TOC of the document would look like in such circumstances: the `\localtableofcontents` does exactly the job of extracting from that global TOC various local TOCs; it works only from material in that (virtual) global TOC.

Since release 1.08k, `etoc` provides the one-argument command `\etocsetlocaltop.toc` to insert into the `.toc` file a kind of “ghost” of a given sectioning unit. Here is an example:

## 50. Hacken „framed parboxes“

```
\part*{Extra unnumbered part}
\etocsetlocaltop.toc{part}
\localtableofcontents
```

Then in this example, `\localtableofcontents` will know be stopped by the next `\part` (naturally a numbered one or a `\part*` followed by suitable `\addcontentsline`) or also by a later `\etocsetlocaltop.toc{part}`.

This is all fine but keep in mind again that `\localtableofcontents` can only display units which have been pushed to the `.toc` file, so that `\part*` must be followed by numbered sections or `\section*/\addcontentsline` pairs if `\localtableofcontents` is to display anything useful. One can not use `etoc` to typeset data not actually there in the `.toc` file!

Thus, in the situation above, one may wonder why the document does not use `\addcontentsline` after the `\part*` as it must do so for sub-sectioning units if they are to appear in the local TOC.

One may imagine some document where all Parts are non-numbered (and are not pushed to the `.toc` file) but nevertheless the user wants its `\localtableofcontents` for each one: then indeed `\etocsetlocaltop.toc` will help in that case.

There are at least two more methods to handle this situation:

- the first one is simply to use the `\addcontentsline{toc}{part}{...}` commands, and to set the `part` style to do nothing, so that the main TOC will not display them,
- the second one does not add the `\addcontentsline{toc}{part}{...}` but uses “depth tags” ([section 53](#)) and multiple `\tableofcontents` (not `\localtableofcontents`), one for each desired local TOC.

There is a variant of the first method which would set the level of `part` temporarily to 6 for the typesetting of the main TOC. But then the `chapter` toc line style will execute inside the main TOC only once its `start` and `finish` parts (see [section 32](#)). This is actually also a problem of the second method.

But not of the first method which uses

```
\etocsetstyle{part}{}{}{}{}{}
```

nor of the approach with `\etocsetlocaltop.toc`: the `start` and `finish` portions of the `chapter` toc line style will get executed at the appropriate times in the main, global, TOC.

Be aware however that this use of `\etocsetlocaltop.toc{part}` will also cause the main global TOC to execute (once) the `start` and `finish` portions of the `part` toc line style (assuming here again it is not typeset in compatibility mode).

Final remark: usage of `\etocsetlocaltop.toc` interferes with `\etoccheckemptiness` only to the extent that it modifies (as explained in this section) the selection made by `\localtableofcontents`, hence the decision whether the local TOC is empty or not.

## 50. Hacken „framed parboxes“

```
\etocdefaultlines
\renewcommand\etocoprulerule{\hrule height 2pt depth 2pt}
\etocruled{\color{green}\fboxrule2pt\fboxsep1ex
           \fbox{\raisebox{-\fontdimen22\textfont2}{\color{blue}\parbox{.5\linewidth}{}}}
```

```
{\normalfont Dieser Text ist vertikal perfekt zentriert
unter Berücksichtigung der \emph{surrounding horizontal rules.}}}}
\ref{toc:globalcmds}
```

Dieser Text ist vertikal perfekt zentriert unter Berücksichtigung der *surrounding horizontal rules*.

<b>Festlegung des toc-Display-Stils . . . . .</b>	<b>43, S. 69</b>	Die Befehle \etocframedstyle, \etocframed und \etoclocalframed . . . . .
Der Befehl \etocsettocstyle . . . . .	43.1, S. 69	43.5, S. 71
Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol . . . . .	43.2, S. 70	Überschriften, Titel, \etocoldpar, \etocinnertopsep . . . . . 43.6, S. 72
Der Befehl \etoctocstyle 43.3, S. 71		Der Kompatibilitätsmodus \etocstandarddisplaystyle . . . . . 43.7, S. 72
Die Befehle \etocruledstyle, \etocrule und \etoclocalrule . . . . .	43.4, S. 71	Der Befehl \etocinline . 43.8, S. 73
		<b>Starred variants and hooks . . . . .</b>
		44, S. 73

## 51. Umstellen der Ebenen

Lasst uns alle Unterabschnitte zeigen und aufzählen, die in diesem Dokument vorkommen (Siehe [Teil IV](#) für andere Verwendungen dieser Technik.):

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

### 51.1. Alle Unterabschnitte in diesem Dokument

- |   |  |  |
|---|--|--|
| <ul style="list-style-type: none"> <li>i. Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben (3.1, S. 9)</li> <li>ii. \etocsetstyle für den Zeilenstil (4.1, S. 9)</li> <li>iii. \etocsettocstyle für den Stil der Inhaltsangabe</li> </ul> | <ul style="list-style-type: none"> <li>(4.2, S. 9)</li> <li>iv. Kompatibilismus (4.3, S. 10)</li> <li>v. Die hyperref-Option <i>bookmarksdepth</i> (12.1, S. 20)</li> <li>vi. Die Befehle \etocobeytoctocdepth und \etocignoretoctocdepth (13.1, S. 21)</li> </ul> | <ul style="list-style-type: none"> <li>cignoretoctocdepth (13.1, S. 21)</li> <li>vii. Die Befehle \etocobeydepthtags und \etocignoredepthtags (14.1, S. 22)</li> <li>viii. The \etocchecksemptiness command</li> </ul> |
|---|--|--|

## 52. Statistiken anzeigen

(16.1, S. 23)	und <code>\etoclocalframed</code> (43.5, S. 71)	<b>xxviii.</b> Alle Unterabschnitte in diesem Dokument (51.1, S. 87)
<b>ix.</b> The <code>\etocnotocif-notoc</code> command (16.2, S. 23)	<b>xix.</b> Überschriften, Titel, <code>\etocoldpar</code> , <code>\etocinertopsep</code> (43.6, S. 72)	<b>xxix.</b> Generelle Informationen (55.1, S. 95)
<b>x.</b> The <code>\etocifwasempty</code> command (16.3, S. 23)	<b>xx.</b> Der Kompatibilitätsmodus <code>\etocstandarddisplaystyle</code> (43.7, S. 72)	<b>xxx.</b> Kompatibilität mit beamer (55.2, S. 95)
<b>xi.</b> Die hyperref-Option <code>hidelinks</code> (18.1, S. 25)	<b>xxi.</b> Der Befehl <code>\etocin-line</code> (43.8, S. 73)	<b>xxxi.</b> Kompatibilität mit Babel (55.3, S. 96)
<b>xii.</b> Testen des Bereichs (41.1, S. 67)	<b>xxii.</b> Ein lokales Inhaltsverzeichnis anpassen (45.1, S. 74)	<b>xxxii.</b> Kompatibilität mit hyperref (55.4, S. 96)
<b>xiii.</b> Dies ist eine (blass)rote Subsection zu Demonstrationszwecken (41.2, S. 67)	<b>xxiii.</b> Das Aussehen eines Inhaltsverzeichnisses anpassen (45.2, S. 75)	<b>xxxiii.</b> Kompatibilität mit <code>multicol</code> (55.5, S. 96)
<b>xiv.</b> Der Befehl <code>\etocset-tocstyle</code> (43.1, S. 69)	<b>xxiv.</b> Inhaltsverzeichnisse darstellen (45.3, S. 75)	<b>xxxiv.</b> Kompatibilität mit <code>tocloft</code> (55.6, S. 96)
<b>xv.</b> Die Befehle <code>\etocmul-ticolsstyle</code> , <code>\etocmulti-col</code> und <code>\etoclocalmulticol</code> (43.2, S. 70)	<b>xxv.</b> Label und Referenzen (45.4, S. 75)	<b>xxxv.</b> Kompatibilität mit der <code>memoir</code> -Klasse (55.7, S. 96)
<b>xvi.</b> Der Befehl <code>\etoctoc-style</code> (43.3, S. 71)	<b>xxvi.</b> Den durch <b>etoc</b> vordefinierten Linienstil individualisieren (47.1, S. 80)	<b>xxxvi.</b> Kompatibilität mit <code>tocvsec2</code> (55.8, S. 97)
<b>xvii.</b> Die Befehle <code>\etocruledstyle</code> , <code>\etocruled</code> und <code>\etoclocalruled</code> (43.4, S. 71)	<b>xxvii.</b> Die Darstellung eines Inhaltsverzeichnisses anpassen (47.2, S. 81)	<b>xxxvii.</b> Kompatibilität mit <code>tableof</code> (55.9, S. 98)
<b>xviii.</b> Die Befehle <code>\etoc-framedstyle</code> , <code>\etocframed</code>		<b>xxxviii.</b> Kompatibilität mit <code>tocstyle</code> (55.10, S. 98)

## 52. Statistiken anzeigen

Jeder Teil dieses Dokuments beginnt mit einem Paragraphen, der sagt, wieviele Kapitel und Unterkapitel er hat. Nun, jeder von diesen Paragraphen ist ein Inhaltsverzeichnis! Wir haben ein Makro `\thispartstats` geschaffen, um dies zu tun. Es nutzt „Speicher“-Boxen, um die Informationen über die ersten und späteren Kapitel und Unterkapitel zu halten. Boxen zu nehmen, ist die einfachste Art und Weise hyperref-Links für spätere Verwendungen zu kapseln (ob es einer ist oder keiner). Allerdings kann man dann nichts an der Schriftart oder Farbe ändern. Wenn eine solche Notwendigkeit entsteht, muss man von der Boxenverwendung zum Makro nehmen wechseln und die hyperref-Daten für später speichern, wie im oben im Code von [Abschnitt 28](#) vorgestellt. Wir präsentieren auch diese zweite Methode.

Aber als Erstes der Code von `\thispartstats`:

```
\newsavebox\firstnamei \newsavebox\firstnumberi
\newsavebox\lastnamei \newsavebox\lastnumberi
\newsavebox\firstnameii \newsavebox\firstnumberii
\newsavebox\lastnameii \newsavebox\lastnumberii
\newcounter{mycounti} \newcounter{mycountii}
\newcommand*\thispartstatsauxi{} \newcommand*\thispartstatsauxii{}
\newcommand*\oldtocdepth{}{}
```

```

\newcommand*\thispartstats{%
  \setcounter{mycounti}{0}%
  \setcounter{mycountii}{0}%
  \def\thispartstatsauxi{%
    \sbox{\firstnamei}{\etocname}%
    \sbox{\firstnumberi}{\etocnumber}%
    \def\thispartstatsauxi{}}
  \def\thispartstatsauxii{%
    \sbox{\firstnameii}{\etocname}%
    \sbox{\firstnumberii}{\etocnumber}%
    \def\thispartstatsauxii{}}
  \begingroup
  \etocsetstyle{subsection} {} {}
    {\thispartstatsauxii
      \stepcounter{mycountii}%
      \sbox{\lastnameii}{\etocname}%
      \sbox{\lastnumberii}{\etocnumber}} {}%
  \etocsetstyle{section} {}
  {}
  {\thispartstatsauxi
    \stepcounter{mycounti}%
    \sbox{\lastnamei}{\etocname}%
    \sbox{\lastnumberi}{\etocnumber}}
% Hier wird ein Befehl definiert, der fuer jedes Teil ein paar Fakten sammelt
% und ausgibt.
  {Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält
    \arabic{mycounti} Abschnitt\ifnum\value{mycounti}>1 e\fi{} und
    \arabic{mycountii} Unterabschnitt\ifnum\value{mycountii}>1 e\fi. Der Name des ersten
    Abschnitts lautet
    \glqq\unhbox\firstnamei{}\grqq~und er hat die Gliederungsnummer
    \glqq\unhbox\firstnumberi\grqq. Der Name des letzten Abschnitts lautet
    \glqq\unhbox\lastnamei{}\grqq~und er hat die Nummer
    \glqq\unhbox\lastnumberi\grqq.
    \ifnum\value{mycountii}>0 Der erste
    Unterabschnitt heißt
    \glqq\unhbox\firstnameii{}\grqq~und er hat die Nummer
    \glqq\unhbox\firstnumberii\grqq. Der Name des letzten Unterabschnitts ist
    \glqq\unhbox\lastnameii{}\grqq~und dessen Nummer ist
    \glqq\unhbox\lastnumberii\grqq.\fi }%
  \etocinline
  \etocsettocstyle {}{}%
  \etocsetnexttocdepth{2}%
  \localtableofcontents % to be used at the top level of a Part.
  \endgroup
}

```

Und nun, die Variante mit Makros anstelle von den mit Boxen (diese Variante steht hier für die Verwendung in einem Kapitel).

```

\makeatletter
\newcommand*\firstsubname {} \newcommand*\lastsubname {}
\newcommand*\firstsubnumber {} \newcommand*\lastsubnumber {}
\newcommand*\thisspecialstatsaux{}
\newcommand*{\thisspecialstats}{%
  \setcounter{mycounti}{0}%

```

### 53. Tiefe Tags nehmen

```
\def\thisspecialstatsaux{%
    \let\firstsubname\etocthelinkedname
    \let\firstsubnumber\etocthelinkednumber
    \def\thisspecialstatsaux{}}
\begingroup
\etocsetstyle{subsection}
{}
{}
{\thisspecialstatsaux
    \stepcounter{mycounti}%
    \let\lastsubname\etocthelinkedname
    \let\lastsubnumber\etocthelinkednumber }
{Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält
\arabic{mycounti} Unterabschnitt\ifnum\value{mycounti}>1 e\fi{}. Der Name des ersten
Abschnitts lautet\emph{\firstsubname{}} und er hat die Gliederungsnummer
{\firstsubnumber}. Der Name des letzten Unterabschnitts lautet
\emph{\lastsubname{}} und er hat die Nummer{\lastsubnumber}.}
\etocsettocstyle {}{%
\etocinline
\etocsetnexttocdepth {1}%
\localtableofcontents % to be used within a section
\endgroup
}
\makeatother
```

## 53. Tiefe Tags nehmen

Wir wollen ein TOC, das für jeden `\part` eine Überschrift hat (außer der letzte Teil mit dem Quellcode, wofür beschließen, ihn nicht einzubeziehen), und wir wollen [Teil VI](#) öffnen für die Paragraphen. Um dies zu erreichen, fügen wir zu dieser Quelle verschiedene `\etocdepthtag.toc`-Befehle und es bleibt jetzt, zu setzen die Levels für jeden Tag mit `\etocsettagdepth` (dies wurde oben in diesem Dokument verwendet in [Abschnitt 6](#), in einer Gruppe, damit es keine Auswirkungen auf andere Inhaltsverzeichnisse hat).

Allerdings sind die Linienstile des Pakets nicht in der Lage Paragraphen anzuzeigen und die Standardlinienstile der Dokumentklasse geben zu viel vertikalen Abstand (in diesem Zusammenhang) bei der Anzeige einer Teilüberschrift im TOC aus. Deshalb können wir unseren eigenen, schnell entwickelten Linienstil, in der Art von [Abschnitt 5](#) (aber auf dem Weg Seitenzahlen von rechts, was mehr wie eine Methode von L<sup>A</sup>T<sub>E</sub>X2e's `\@dottedtocline` ist; und mehrzeilige Überschriften lassen nun den Bereich unter den Zahlen leer, im Gegensatz zu dem Code von [Abschnitt 5](#)). Und nach ein paar Minuten für die Wahl der Längen (jetzt ist das erstmal getan, es kann leicht recycelt) erhalten wir:

### A TOC using depth tags

I Überblick	8
II Beliebig viele TOCs und lokale Versionen davon	16
III Beispiele	27

<b>IV Überraschende Anwendungen von etoc</b>	40
<b>V Befehle für die TOC-Linienstile</b>	61
<b>VI Befehle für den toc-Display-Stil</b>	69
<b>43 Festlegung des toc-Display-Stils .....</b>	69
43.1 Der Befehl \etocsettocstyle .....	69
43.2 Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol.....	70
43.3 Der Befehl \etoctocstyle .....	71
43.3.1 Der Befehl \etoctocstylewithmarks .....	71
43.3.1.1 Möchten wir wirklich paragraph-Einträge im Inhaltsverzeichnis?.....	71
43.3.1.2 Wirklich?.....	71
43.4 Die Befehle \etocruledstyle, \etocruled und \etoclocalruled.....	71
43.5 Die Befehle \etocframedstyle, \etocframed und \etoclocalframed .....	71
43.6 Überschriften, Titel, \etocoldpar, \etocinnertopsep .....	72
43.7 Der Kompatibilitätsmodus \etocstandarddisplaystyle .....	72
43.7.1 Die Befehle \etocarticlestyle, \etocbookstyle, .... .	72
43.8 Der Befehl \etocinline .....	73
<b>44 Starred variants and hooks .....</b>	73
<b>VII Eigene Stile von etoc nutzen und anpassen</b>	74
<b>VIII Tipps</b>	85
<b>IX etoc und die Außenwelt</b>	95

---

```
\etocsetnexttocdepth {all}
\begin{group}
\parindent 0pt \leftskip 0cm \rightskip .75cm \parfillskip -\rightskip
\newcommand*{\EndParWithPagenoInMargin}{%
    \nobreak\hfill
    \makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
    \par}
\renewcommand*{\etoctoclineleaders}{%
    {\hbox{\normalfont\normalsize\hbox to .75ex {\hss.\hss}}}}
\newcommand*{\EndParWithPagenoInMarginAndLeaders}{%
    \nobreak\leaders\etoctoclineleaders\hfill
    \makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
    \par}
\etocsetstyle {part}{%
    {}%
    {\leavevmode\leftskip 1cm\relax}%
    {\bfseries\large\llap{\makebox[1cm][r]{\etocnumber\ \ }}}%
    \etocname\EndParWithPagenoInMargin\smallskip}%
    {}%
\etocsetstyle {section}{%
    {}%
    {\leavevmode\leftskip 1.75cm\relax}%
    {\bfseries\normalsize\llap{\makebox[.75cm][l]{\etocnumber}}}%
    {}%
}
```

#### 54. Das TOC als eine Tabelle setzend (die alte Weise)

```
\etocname\EndParWithPagenoInMarginAndLeaders}
  {}
\etocsetstyle {subsection}
  {}
{\leavevmode\leftskip 2.75cm\relax }
{\mdseries\normalsize\llap{\makebox[1cm][l]{\etocnumber}}%
 \etocname\EndParWithPagenoInMarginAndLeaders}
  {}
\etocsetstyle {subsubsection}
  {}
{\leavevmode\leftskip 4cm\relax }
{\mdseries\normalsize\llap{\makebox[1.25cm][l]{\etocnumber}}%
 \etocname\EndParWithPagenoInMarginAndLeaders}
  {}
\etocsetstyle {paragraph}
  {}
{\leavevmode\leftskip 5.5cm\relax }
{\mdseries\normalsize\llap{\makebox[1.5cm][l]{\etocnumber}}%
 \etocname\EndParWithPagenoInMarginAndLeaders}
  {}
\etocsettagdepth {preamble}    {none}
\etocsettagdepth {overview}   {part}
% \etocsettagdepth {arbitrarily}{part} % not needed explicitely, keeps value
\etocsettagdepth {examples}   {part}
\etocsettagdepth {surprising} {part}
\etocsettagdepth {linestyles} {part}
\etocsettagdepth {globalcmds} {paragraph}
\etocsettagdepth {custom}     {part}
\etocsettagdepth {tips}       {part}
\etocsettagdepth {etocandworld}{part}
\etocsettagdepth {code}       {none}
\renewcommand\etocoprule {\hrule height 3pt\relax }
\renewcommand\etocoprulecolorcmd {\color{blue}}
\renewcommand\etocaftercontentshook
  {\medskip\begin{group}\color{blue}\hrule height 3pt \endgroup}
\etocruledstyle [1]{\Large\bfseries
  \fbox{\makebox[8cm]{A TOC using depth tags}}}
\slippery
\tableofcontents
\endgroup
```

#### 54. Das TOC als eine Tabelle setzend (die alte Weise)

Aufgrund, unter anderem, der Tatsache, dass die Ausrichtung Zellen kreiert und Gruppen schließt und dass durch standardmäßige Definitionen von `\etocname`, `\etocnumber` und `\etocpage` durch `etoc` lokal sind, war es nicht leicht, ein TOC als Tabelle mit `etoc` zu setzen, vor der Freigabe von Version 1.08.

Nicht nur `\etocname` etc. ... verursachte ein Problem, sondern auch die grundlegenden Redefinition von `\contentsline`, die von `etoc` erst nach dem ersten Argument von `\etocsettocstyle` gemacht wurde, ist beseitigt worden; folglich, wenn dieses Argument in einer Tabelle geöffnet wurde, würde die `etoc`-Redefinition von `\contentsline` in der

erten Zelle der ersten Reihe durchgeführt werden und danach verloren gehen.

So musste man auf die Technik zurückgreifen, die in [Abschnitt 27](#) erklärt wurde, auf die Verwendung der Festlegung von \tableofcontents als eine Möglichkeit, Daten zu speichern, die später gezeigt wurde.

Für das Protokoll, dies ist, wie das TOC in [Abschnitt 30](#) in der Vergangenheit codiert wurde. Hier haben wir nicht das Problem mit der Positionierung der \hlines; wir sind konfrontiert mit dem neueren Verfahren. Andererseits müssen wir Zeichenregister manipulieren, die den meisten L<sup>A</sup>T<sub>E</sub>X-Nutzern nicht bekannt sind (Makros könnten verwendet werden, doch das würde umständlich sein, außer vielleicht mit Hilfe von ε-T<sub>E</sub>X \nicht expandiert).

Die Methode hier ist die stärkste, weil sie aus der .toc-Datei nur die Daten herausfiltert, die wir wollen (die anderen Dinge werden nicht ignoriert, sie werden ausgeführt, jedoch ohne hoffentlich Chaos zu schaffen; typisch sind dafür die Sprache von Anleitungen zu ändern, etc. . . ). Und wir sind weniger anfällig, potentielle Opfer von verschiedenen Fallen bei externen Makros innerhalb der .toc-Datei aus anderen Paketen stammend.

Hinweis: Anstatt \toks-Register wäre es hier einfacher, ε-T<sub>E</sub>X \unexpanded Primitive zu verwenden. Sehen Sie dazu zum Beispiel [Abschnitt 29](#).

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

\newcommand*\PreparePart{%
  \toks0 \expandafter{\etocthelinkednumber}%
  \toks2 \expandafter{\etocthelinkedname}%
  \toks4 \expandafter{\etocthelinkedpage}%
  \edef\toctablepiece {\noexpand\hline
    \noexpand\strut\the\toks0 &\noexpand\bfseries\the\toks2
    &\the\toks4 \noexpand\\ \noexpand\hline}%
}
\newcommand*\PrepareSection{%
  \toks0 \expandafter{\etocthelinkednumber}%
  \toks2 \expandafter{\etocthelinkedname}%
  \toks4 \expandafter{\etocthelinkedpage}%
  \edef\toctablepiece {\the\toks0 &\the\toks2 &\the\toks4 \noexpand\\}%
}
% new version for 1.07k, 2014/03/06
\newcommand*\PrepareSubsection{%
  \toks0 \expandafter{\etocthelinkednumber}%
  \toks2 \expandafter{\expandafter\itshape\etocthelinkedname\strut}%
  \toks4 \expandafter{\expandafter\itshape\etocthelinkedpage}%
  \edef\toctablepiece{\&\noexpand\makebox[1cm][c]{\the\toks0}%
    \noexpand\parbox[t]{\dimexpr6cm-\tabcolsep\relax}%
    {\noexpand\sloppy\the\toks2}%
    &\the\toks4 \noexpand\\}%
}

\begingroup
\etocsetstyle{part}{}{}{\PreparePart \appendtotok\toctabletok\toctablepiece{}}
\etocsetstyle{section}{}{}{\PrepareSection \appendtotok\toctabletok\toctablepiece{}}
\etocsetstyle{subsection}{}{}{\PrepareSubsection\appendtotok\toctabletok\toctablepiece{}}
```

#### 54. Das TOC als eine Tabelle setzend (die alte Weise)

```
\etocsettocstyle
  {\toctabletok{\hypersetup{hidelinks}%
    \begin{longtable}{|>{\bfseries c|}p{7cm}|r|}\hline
      \multicolumn{3}{|c|}{\Large\bfseries\strut TABLE OF CONTENTS}%
    \\\hline\hline}
  {\global\toctabletok\expandafter{\the\toctabletok\hline\end{longtable}}}}
\etocsettocdepth {subsection}
\tableofcontents
\the\toctabletok
\endgroup
```

# Teil IX.

## etoc und die Außenwelt

### 55. Kompatibilität mit anderen Paketen

#### 55.1. Generelle Informationen

Die Inhalte der .toc-Datei (falls bereits vorhanden) werden einmal in den Speicher von **etoc** getan, bei `\begin{document}`.<sup>48</sup>

Die .toc-Datei bleibt für andere Pakete nutzbar für Lesevorgänge bis zum Ort des ersten Inhaltsverzeichnisses, zu diesem Zeitpunkt wird ein Schreib-stream durch **etoc** geöffnet und von diesem Punkt aus wird die Datei gelöscht bis ihr Inhalt wieder auf die Platte von LATEX am Ende der Kompilierung geschrieben wurde.

**etoc** erwartet die Sektionierungseinheiten des Dokuments, um ihre Daten in eine Datei mit der Endung .toc zu schreiben, in Form von Linien, die enthalten den `\contentsline`-Befehl und dessen Argumente.

**etoc** erfordert von der .toc-Datei, das `\contentsline`-Makro zu verwenden (möglicherweise von anderen Paketen belegt).

**etoc** kann nicht wirklich mit Paketen zusammensein, die `\tableofcontents`-Befehle ändern: Eine Art Waffenstilstand kann erreicht werden, wenn **etoc** zuletzt geladen wird, daher ist es der Gewinner.

Ändern Sie nicht den `\tableofcontents`-Befehl, wie diesen folgenden:

```
\let\oldtableofcontents\tableofcontents,  
\renewcommand\tableofcontents{\oldtableofcontents\mystuff},  
da dies den \label/\ref-Mechanismus unmöglich macht. Sie können vielmehr  
\renewcommand\etocaftertochook{\mystuff}  
ausführen und es ergibt auch \etocaftercontentshook, das führt ein bisschen eher  
49 aus, gerade vor dem Schließen des Teils vom toc-Anzeigestil (und damit innerhalb  
einer Gruppe).
```

Don't use `\if<condition> stuff \tableofcontents\fi`, but:

```
\if<condition> stuff \expandafter\tableofcontents\fi
```

Also a `\else` immediately following `\tableofcontents` or `\localtableofcontents` requires a previous `\expandafter`.

#### 55.2. Kompatibilität mit beamer

**etoc** steht inkompatibel mit der `beamer`-Klasse. Aber, wenn `beamer` im article-Modus verwendet wird, d.h. die `article`-Klasse in Verbindung mit dem `beamerarticle`-Paket, dann sollte **etoc** arbeiten.

<sup>48</sup>Neu mit 1.07m. Früherer Versionen benutzten die .toc-Datei bei `\usepackage{etoc}`. Danke an Denis Bitouzé für ein Babel-etoc Problem entdecken zu haben.

<sup>49</sup>Im Gegensatz zu `\etocaftertochook`, `\etocaftercontentshook` wird nicht ausgeführt, wenn `tocdepth` das Drucken des TOC nicht ermöglicht.

## 55. Kompatibilität mit anderen Paketen

### 55.3. Kompatibilität mit Babel

Man sollte **etoc** nach babel laden.

### 55.4. Kompatibilität mit hyperref

Please inform the author in case of issues: **etoc** was from the start designed to be 100% compatible with package **hyperref**.

Es ist **hyperref** bekannt und hoffentlich kompatibel mit ihm: die Macros `\etocname`, `\etocnumber` und `\etocpage` enthalten die **hyperref**-Links, wenn vorhanden (Beachten Sie, dass die `linktoc=all`-Option von **hyperref** mitteilt, dass ein Link zur Seitenzahl zu setzen ist, für einen einen toc-Eintrag.). Beispielsweise sind die Inhaltsverzeichnisse des vorliegenden Dokuments komplett verlinkt. Es ist kein Thema, ob **etoc** vor oder nach **hyperref** geladen wird.

### 55.5. Kompatibilität mit **multicol**

**etoc** lädt das Paket **multicol**.

### 55.6. Kompatibilität mit **tocloft**

Die Release 1.07k fügt Kompatibilität mit dem Paket **tocloft** hinzu:<sup>50</sup> Schritte wurden getan, um eine Neudeinition von `\tableofcontents` zu vermeiden, getan von **tocloft** an `\begin{document}`.

Solange **etoc** ungeschickt ist im Kompatibilitätsmodus, wird von **tocloft** die Anpassung sowohl für den Liniestil als auch den TOC-Titel getan. Man kann immer noch von den *tiefen Tags*-Management mit **etoc** profitieren, von seinem `\localtableofcontents`, von seinem `\label+\ref`-Mechanismus. Man kann `\etocsetstyle` nehmen, um mit **etoc** das Layout des TOC zu definieren und dann eher **tocloft** für etwas anderes, wenn auf `\tableofcontents` folgen `\etocstandardlines` und `\etocstandarddisplaystyle`. In diesem Kompatibilitätsmodus `\etocsetlevel{division unit}{6}` wird die gewählte Bereichsebene gerendert, aber der Austausch von Ebenen ist andererseits nicht möglich.

Man sollte **etoc** nach **tocloft** laden. Sonst wird eine Warnung ausgegeben.

Es ist möglich (unter Verwendung des Pakets **tocloft** beispielsweise), im gesamten Dokument die Makros `\l@section`, `\l@subsection` ... zu benutzen und der Effekt wird im nächsten Inhaltsverzeichnis durch den Kompatibilitätsmodus von **etoc** sichtbar werden.

### 55.7. Kompatibilität mit der **memoir**-Klasse

Die Release 1.071 hat ebenfalls die Kompatibilität mit der **memoir**-Klasse verbessert: Seine `appendix`-Ebene kann mit **etoc** gemacht werden. Es ist die gleiche Ebene wie bei `chapter`, also der Kapitelebenstil kann eventuell ein Test für einige boolesche-Ausdrücke

<sup>50</sup><http://ctan.org/pkg/tocloft>

sein, die aktiviert sein können via einer Instruktion an die .toc-Datei aufgenommen werden, wenn man zwei Arten von Unterteilungen unterscheidet.

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

Eine ungewöhnliche Situation tritt auf, wenn man zwei aufeinanderfolgende `\localtableofcontents` hat (offensichtlich ist das eine nicht sehr wahrscheinliche Situation), direkt nach einem `\part` zum Beispiel. Der erste würde dann eine Kapitelüberschrift erzeugen, welche in die .toc-Datei geschrieben werden würde. Das zweite Inhaltsverzeichnis denkt dann, es ist ein lokales zu diesem Kapitel gehörig und würde aus diesem Grund nichts darstellen. Die Lösung für dieses Problem ist es, das zweite als Klon des ersten zu definieren.

Actually, that `\localtableofcontents` by default uses like `\tableofcontents` a chapter style is often inconvenient. One can use the `memoir` macros to modify this: according to `memoir` documentation, `\tocheadstart` and `\printtoctitle` seem to be the most relevant. If you do that you will probably want the added line in .toc file to use `{section}` not `{chapter}`. `etoc` provides to this effect `\etocmemoirtotocfmt{<kind>}{<name>}` whose first argument `<kind>` is `chapter`, `section`, `subsection`, etc... and the second argument stands for the title and thus `\contentsname` is appropriate. Hence

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

seems the natural choice in this context as an addition to the customizing the `memoir` toc related macros to achieve a section like rendering (for more details on the latter, please refer to `memoir` manual).

A completely different approach, arguably simpler, is to overrule the `etoc` default under `memoir` (after the main document `\tableofcontents` which should still use a chapter styling) via these simple commands:

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

exactly as one would do with book class for example. Don't use any `\phantomsection` in there, `hyperref` hyperlinks will work just fine (and the anchor will be slightly above toc title, not below).

Notice that prior to `etoc 1.09b`, the `\etocaftertitlehook` macro was used by `etoc` itself under the `memoir` class. Hence `\localtableofcontents` obeyed the `memoir` mechanism relative to starred or not starred, even after issuing `\etocarticlestyle`, as long as the user did not redefine `\etocaftertitlehook`. Starting with 1.09b (which syncs with a `memoir` change at v3.7i) it is needed in a case like here where the user issues `\etocarticlestyle` to also configure explicitly the `\etocaftertitlehook` to achieve the "insert title in .toc" effect (if so wished), because `etoc` uses another mechanism and does not pre-empt usage of this user macro anymore.

To recover the `etoc` default under `memoir` class one can issue `\etocmemoirstyle` and then `\etocmemoirtotocfmt{chapter}{\contentsname}` (but attention that a custom `\etocaftertitlehook` will still be obeyed; for info: it will get executed before the `\cfttocbeforelisthook`).

## 55.8. Kompatibilität mit tocvsec2

## 56. *T<sub>E</sub>X-nische Sachen*

**etoc** deaktiviert die Modifikationen des \tableofcontents durch `tocvsec2`<sup>51</sup> und es kanzelt auch andere toc-bezogene Macros, aber es reimplantiert partiell ihre Funktionalität mit `\etocsettocdepth.toc`. Zwei L<sup>A</sup>T<sub>E</sub>X-Läufe sind für das erneute Nehmen des Befehls in einem Dokument notwendig, um einen Effekt im Inhaltsverzeichnis zu haben.

### 55.9. Kompatibilität mit `tableof`

Es ist möglich, **etoc** und `tableof`<sup>52</sup> gleichzeitig zu verwenden. Die Veröffentlichung 1.08 von **etoc** benötigt mindestens die Version 1.4a von `tableof`. Wenn man `\etocglobaldefs` in die Präambel aufgenommen hat, muss dies nach dem Laden des Pakets `tableof` sein. Der `tableof`-Befehl `\nexttocwithtags` sollte, wie erwartet, funktionieren.

Die `tableof`-Befehle `\tableof`, `\tablenotof`, ... werden das Inhaltsverzeichnis (a priori global) nach den Standards der Dokumentklasse setzen, gehorchend den **etoc**-Tiefenauszeichnungen, wie erklärt in der `tableof`-Dokumentation, sie setzen keinen TOC-Titel. Sie sollten *nicht* verwendet werden im Fall, wenn `\etocglobaldefs` vorher ausgegeben wurde; es sei denn, ihr Skopus wurde da beendet oder `\etoclocaldefs` hat seine Einflussnahme abgebrochen.

### 55.10. Kompatibilität mit `tocstyle`

Nicht bekannt. (not tested a.t.t.o.w.).

## 56. *T<sub>E</sub>X-nische Sachen*

Die `\etocname`, `\etocnumber`, `\etocpage` Befehle sind gegen vorzeitige Expansion geschützt. Sie enthalten geeignete hyperref-Links, wenn das Paket hyperref geladen und aktiviert für das TOC ist. Die Befehle `\etoclink` und `\etocifnumbered` sind auch gegen vorzeige Expansion geschützt. Auch `\etociffirst` und `\etoccontentsline`.

Andererseits `\etocthename`, `\etocthenumber`, `\etoctheplace`, haben keine Anbindung an hyperref-Links und sind nicht gegen Expansionen geschützt. Und `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` sind auch nicht gegen Expansionen geschützt.

Befehle wie `\etocsetstyle`, `\etocsetlevel`, `\etocsettocstyle`, `\etocmulticolstyle`, `\etocruledstyle`, `\etocframedstyle` folgen Gruppierungen von L<sup>A</sup>T<sub>E</sub>X. Alle TOCs werden in Gruppierungen gesetzt.

Wenn ein `\localtableofcontents` durch den Nutzer in das Dokument aufgenommen wird, eine Zeile, in einen inneren **etoc** -Befehl und eine Identifikationszahl in die .toc-Datei aufgenommen wird. Das korrekte Inhaltsverzeichnis wird nur nach dem nächsten L<sup>A</sup>T<sub>E</sub>X-Lauf angezeigt werden.

---

<sup>51</sup><http://ctan.org/pkg/tocvsec2>

<sup>52</sup><http://ctan.org/pkg/tableof>

## 57. Fehler und Katastrophen

Nach der Verwendung von `\etocsetstyle` für eine Ebene nehmen die restlichen, nicht angepassten Ebenen die Standardstile von **etoc** (diejenigen, die aktiviert werden von `\etocdefaultlines`). Man muss sicherstellen, dass alle Ebenen, die für das nächste Inhaltsverzeichnis gebraucht werden, miteinander kompatibel sind: Insbesonders, dass die Standardlinienstile jeweils im „vertikalen Modus“ gestartet werden.

Wenn man mehrere `\tableofcontents`-Befehle in einem Dokument verwendet, sollte man sich hüten, mehrere Satzanweisungen in der `.toc`-Datei zu addieren, so sie mit `\tableofcontents` ausgeführt für alle TOCs ausgeführt werden: Auch für `\localtableofcontents` spielt es keine Rolle, ob die Anweisungen Material außerhalb des Anwendungsbereichs betreffen, sie werden sie trotzdem erhalten. Falls unbedingt erforderlich (aber es das sollte es nie) müssen diese Instruktionen so sein, dass einfach aktiviert oder deaktiviert werden von der Dokumentquelle, wie es sein muss.

So wie es üblich ist bei TOCs und Labels, muss nach jeder Änderung L<sup>A</sup>T<sub>E</sub>X eine bestimmte Anzahl von Durchführungen haben, um für das Dokument das endgültige Aussehen zu erhalten (mindestens zweimal durchlaufen).

# Teil X.

## Der Code

<b>Zeitstempel</b>	<b>58, S. 100</b>
<b>Änderungen</b>	<b>59, S. 100</b>
<b>Ausführung</b>	<b>60, S. 103</b>

### 58. Zeitstempel

Die deutsche Übersetzung wurden zum letzten Mal am 09.04.2015 (Paketversion v1.08d) geändert (abgesehen von den organisatorischen Veränderungen). Einige neuere Abschnitte bleiben auf Englisch.

Dokumentation erzeugt von der Quelldatei mit dem Stempel Time-stamp: <17-11-2019 at 13:36:39 CET>. Die dokumentierte Paketversion ist v1.09b von 17.11.2019.

### 59. Änderungen

v1.09a, v1.09b [2019/11/17]

Sync with memoir v3.7i which has a better location of the TOC hyperref anchor. The \etocaftertitlehook can now freely be used also with memoir class (formerly its usage in case of memoir class was preempted by etoc itself). For more details refer to the section "Compatibility with the memoir class".

v1.09 [2019/03/09]

New features: \etoclocaletop, \localtableofcontentswithrelativedepth. Thanks to Tony Roberts for feature request.

Note to hackers: internal control sequence \Etoc@localtop is gone.

etoc now requires e-TeX (\numexpr, \unless).

v1.08p [2018/07/04]

Fixed bug surfacing in case of linktoc=page option of hyperref. Thanks to Denis Bitouzé for report (cf. <https://github.com/ho-tex/hyperref/issues/65>, <https://github.com/dbitouze/yathesis/issues/61>).

v1.08o [2018/06/15]

Fixed bug showing up if an unnumbered TOC entry starts with a brace, and document uses hyperref. Caused by a typo in a macro name at previous release.

v1.08n [2018/02/23]

Refactoring of core macros detecting \numberline and its variants.

v1.08m [2018/02/07]

Fix to 1.08k's introduced incompatibility with KOMA-script and tocbasic's \nonumberline.

v1.08l [2017/10/23]

Workaround an issue with Emacs/AUCTeX wrongly reporting about actually non-existent LaTeX errors, which was triggered by some strings written (indirectly) to log file by etoc under some circumstances.

v1.08k [2017/09/28]

Adds \etocsetlocaletop.toc. See corresponding manual section for details.

Adds \etocsavedparttocline, \etocsavedchaptertosline, \etocsavedsectiontocline, ... They can be used in the context of the technique explained in section "Another compatibility mode".

Formerly, etoc redefined for the duration of the TOC the memoir macro \chapternumberline and its likes to have same meaning as \numberline (of course, not when executed in compatibility mode), for the sake of extraction of \etocnumber.

New method detects presence of any `\<foo>numberline` macro without any change to originals; they can thus be used as is when applying the approach of "Another compatibility mode" section from manual.

v1.08j [2017/09/21]

Since 1.08a-2015/03/13 `\etocname`, `\etocnumber`, `\etocpage` contain, if `hyperref` is present and configured for using hyperlinks in the TOC, the link destination in already expanded form. This means one can use them even if the style closes a group (for example from a & in a tabular), if `\etocglobaldefs` was issued; also one can save their meaning for delayed usage (with for example `\LetLtxMacro` as they are robust).

But for some legacy reason `\etoclink`, contrarily to `\etocthealink`, was handled differently. Now, `\etoclink` also contains the link destination in already expanded form, and can thus be used even if the line style issues a &, as long as `\etocglobaldefs` is issued.

Also, bugs dating back to the early days of the package, but surfacing only under relatively rare conditions such as usage of `hyperref` with its option "linktoc=page" got fixed.

v1.08i [2016/09/29]

This fixes an issue dating back to 1.08e-2015/04/17: under `\etoccheckemptiness` regime, some circumstances (such as adding to an already compiled document a `\localtableofcontents` before the main `\tableofcontents`) created an "Undefined control sequence \Etoc@localtop" error. Thanks to Denis Bitouzé for reporting the problem.

On this occasion, `\etocdoesnotcheckemptiness` has been added to unset the flag.

A rather more exotic issue was fixed: the emptiness check for local tocs could get confused if the tocdepth counter was varying in some specific ways from inside the toc file.

After adding to a document a `\localtableofcontents`, two LaTeX passes are needed for etoc to get a chance to print the correct local contents. Formerly, etoc issued a Warning on the first pass; it now also induces LaTeX into announcing "There were undefined references", as this is nearer to the end of the log file and console output.

v1.08h [2016/09/25]

New functioning of `\etocsetnexttocdepth`: the tocdepth counter is modified only at the time of the table of contents, not before. This fixes an issue which arose when `\etocsetnexttocdepth` was used multiple times with no intervening table of contents. Thanks to Denis Bitouzé for reporting the problem.

The PDF documentation includes about 25 LaTeX code snippets also as file attachment annotations, additionally to their verbatim typesetting. The ordering of the documentation contents has been slightly re-organized.

A previous documentation-only update on 2016/09/09 added a new section with the (approximate) translation into etoc lingua of the book class toc style, for easy customizability.

v1.08g [2015/08/29]

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

v1.08f [2015/04/28]

Minor changes to the documentation. `\etoc-setlevel` more economical.

v1.08e [2015/04/17]

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

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

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

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

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

v1.08d [2015/04/09]

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

Thanks to Christine Römer!

v1.08c [2015/03/30]

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

v1.08b [2015/03/18]

Bug fixes:

## 59. Änderungen

- extra space token removed from '\localtableofcontents' (showed only for inline TOCs.)
- \etocpartname (a macro used by the package own default line styles) was defined to be '\partname', but this is not compatible at least with babel+french context. Now simply expands to Part.
- some problems fixed in the German documentation.
- [2015/03/28] some more problems fixed in the documentation. Added mention of \etocarticlestyle and \etocbookstyle.

v1.08a [2015/03/13]

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

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

v1.08 [2015/03/10]

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

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

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

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

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

v1.07n [2015/03/05]

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

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

v1.07m [2015/01/23]

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

Improved global toc display emulation under KOMA-script classes.

New command \etocbeforetitlehook. New command \etocdisplay.

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

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

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

v1.07l [2014/04/22]

Fixes a bug with the 1.07k compatibility layer with tocloft which had broken the 1.07k (sic) compatibility with memoir (yes, memoir class 1.07k testing had been done before adding the tocloft thing to the source code . . . ). Also, etoc when detecting tocsec2 now checks if this is under the memoir class, as then nothing special needs to be done to rescue \tableofcontents, contrarily to the situation with the native tocsec2.

v1.07k [2014/03/06]

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

v1.07j [2013/12/03]

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

v1.07i [2013/10/21]

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

v1.07h [2013/10/16]

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

v1.07g [2013/10/13]

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

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

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

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

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

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

v1.07f [2013/03/07]

New macros `\etoctheLINKEDNAME`, `\etoctheLINKEDNUMBER`, `\etoctheLINKEDPAGE`, and `\etoctheLINK`.

v1.07e [2013/03/01]

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

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

More such examples added 2013/03/03.

v1.07d [2013/02/24]

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

v1.07b [2013/02/02]

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

Additional examples in the documentation.

v1.07 [2013/01/29]

New commands:

`\etoctheNAME`, `\etoctheNUMBER`, `\etocthePAGE`, `\etocLINK`,  
`\etocTOCcontentsLINE`, `\etocTOCcontentsLINE*`  
`\etocNOPAR`, `\etocaftercontentsHOOKE`

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.

## 60. Ausführung

Diese Dokumentation wurde ohne Einbeziehung des Quellcodes erstellt.  
Um die deutsche Dokumentation mit Einbeziehung des Quellcodes zu erstellen, ist zu kompilieren etex mit `etoc.dtx`, um `etoc-DE.tex` zu erzeugen, dann ist `\NoSourceCode` in `etoc-DE.tex` auf 0 zu setzen, dann ist pdflatex dreimal mit `etoc-DE.tex` laufen zu lassen.