

INHALTSVERZEICHNIS

Zusammenfassung	S. 5
Vorwort	1, S. 5
Lizenz	2, S. 6

Teil I. Überblick

Muss ich Computerfreak sein, um etoc benutzen zu können?	3, S. 7
Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben.	
Zeilenstile und Stile der Inhaltsangabe (toc display style)	4, S. 8
<code>\etocsetstyle</code> für den Zeilenstil – <code>\etocsettocstyle</code> für den Stil der Inhaltsangabe – Kompatibilismus.	
Ein erstes Beispiel	5, S. 9
Ein zweites Beispiel	6, S. 11
Ein drittes Beispiel: emulating the book class	7, S. 12
Verlinkte Liste der Hauptbefehle	8, S. 16

Teil II. Beliebig viele TOCs und lokale Versionen davon

Beschriften und anderswo wieder benutzen	9, S. 17
Eine mächtige Funktionsweise von etoc: Die Wiederzuweisung von Levels mit von <code>\etocsetlevel</code>	10, S. 18
Die <code>\etocsettocdepth</code>- und <code>\etocsetnexttocdepth</code>-Befehle	11, S. 19
Die hyperref-Option <i>bookmarksdepth</i> .	
Der Befehl <code>\etocsettocdepth.toc</code>	12, S. 20
Die Befehle <code>\etocbeyondtocdepth</code> und <code>\etocignoretoctocdepth</code> .	
Die Befehle <code>\etocdepthtag.toc</code> und <code>\etocsettagdepth</code>	13, S. 21
Die Befehle <code>\etocbeyonddepthtags</code> und <code>\etocignoredepthtags</code> .	
Befehle zur .toc-Datei hinzufügen	14, S. 22
Die hyperref-Option <i>hidelinks</i> .	
Zwei Beispiele	15, S. 23
Eine wunderschöne Beispielthese – Ein TOC mit einer Hintergrundfarbe.	

Erzeugt von der Quelldatei mit Zeitstempel "Time-stamp: <09-09-2016 at 21:49:18 CEST>".

Übersetzt ins Deutsche von Felix Baral-Weber, Jenny Rothkrämer-Vogt, Daniel Büttner, Claudia Dahl, Christian Otto und Christine Römer (FSU Jena). Hinweise zur Übersetzung an Christine_Roemer at t-online dot de.

Teil III. Überraschende Anwendungen von **etoc**

Das TOC für TOCs	16, S. 27
Willkürliche „Listen von ...“, \etoccontentsline	17, S. 29
Ein TOC mit einem extravaganten Layout	18, S. 30
Ein weiterer Kompatibilitätsmodus	19, S. 31
Das TOC als ein Baum	20, S. 34
Das TOC als ein Molekül	21, S. 37
Das TOC als ein TikZ Mind-Map	22, S. 39
Das TOC als eine Tabelle	23, S. 43
Die Befehle \etocglobaldefs und \etoclocaldefs.	
Not displayed empty TOCs	24, S. 49
The \etoccheckemptiness command – The \etocnotocifnotoc command – The \etocifwasempty command.	
A TOC self-adjusting widths for its typesetting	25, S. 50

Teil IV. Befehle für die TOC-Linienstile

Der \etocsetstyle-Befehl	26, S. 52
Die \etocname- und \etocpage-Befehle – Die \etocskipfirstprefix und \etociffirst Befehle – Der \etocnumber-Befehl – Der \etocifnumbered-Schalter – Die Befehle \etocthenname, \etocthenumber sowie \etocthepage – Der \etoclink-Befehl – Die Befehle \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage und \etocthelink.	
Der \etocsetlevel-Befehl	27, S. 56
Umfang der zur .toc-Datei gehörigen Befehle	28, S. 58
Testen des Bereichs – Dies ist eine (blass)rote Subsection zu Demonstrationszwecken.	
Bin ich auch rot?	29, S. 58

Teil V. Befehle für den toc-Display-Stil

Festlegung des toc-Display-Stils	30, S. 59
Der Befehl \etocsettocstyle – Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol – Der Befehl \etocstocstyle – Die Befehle \etocruledstyle, \etocruled und \etoclocalruled – Die Befehle \etocframedstyle, \etocframed und \etoclocalframed – Überschriften, Titel, \etocoldpar, \etocinnertopsep – Der Kompatibilitätsmodus \etocstandarddisplaystyle – Der Befehl \etocinline.	
Gesternde Varianten des \tableofcontents etc. Befehls	31, S. 63
Das Inhaltsverzeichnis von diesem Teil	32, S. 63
Überprüfen des Kompatibilitätsmodus – Ein Inhaltsverzeichnis mit Rahmen – Eine (verrückte) Inline-Darstellung.	

Teil VI. **etoc** nutzen und anpassen

Überblick über die Grundbefehle zur Stilbearbeitung 33, S. 68

Ein lokales Inhaltsverzeichnis anpassen – Das Aussehen eines Inhaltsverzeichnisses anpassen – Inhaltsverzeichnisse darstellen – Label und Referenzen.

Der Linienstandard des Pakets: `\etocdefaultlines` 34, S. 69

etoc individuell anpassen 35, S. 73

Den durch **etoc** vordefinierten Linienstil individualisieren – Die Darstellung eines Inhaltsverzeichnisses anpassen.

Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis 36, S. 76

Teil VII. Tipps

Hacken „framed parboxes“ 37, S. 79

Umstellen der Ebenen 38, S. 80

Alle Unterabschnitte in diesem Dokument.

Statistiken anzeigen 39, S. 81

Tiefe Tags nehmen 40, S. 83

Das TOC als eine Tabelle setzend (die alte Weise) 41, S. 85

Teil VIII. **etoc** und die Außenwelt

Kompatibilität mit anderen Paketen 42, S. 87

T_EX-nische Sachen 43, S. 89

Fehler und Katastrophen 44, S. 89

Teil IX. Der Code

Zeitstempel 45, S. 90

Änderungen 46, S. 90

Ausführung 47, S. 92

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- \LaTeX -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: 1.08g (2015/08/29 - doc 2016/09/09)
% License: LPPL 1.3c or later
% Copyright (C) 2012-2015 Jean-Francois Burnol <jfbu at free dot fr>
% Copyright (C) 2014-2015 Christine Roemer <Christine_Roemer at t-online dot de>
% and collaborators for the translation into German of the documentation
%
% This Work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either
% version 1.3c of this license or (at your option) any later
% version. This version of this license is in
% http://www.latex-project.org/lppl/lppl-1-3c.txt
% and the latest version of this license is in
% http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of
% LaTeX version 2005/12/01 or later.
%
% The Authors of this Work are:
% Jean-Francois Burnol <jfbu at free dot fr>
% for the source code and English documentation, and
% Christine Roemer <Christine_Roemer at t-online dot de> and collaborators
% for the translation into German of the documentation.
%
% This Work consists of the main source file etoc.dtx and the derived
% files etoc.sty, etoc.ins, etoc.tex, etoc-DE.tex, etoc.pdf, etoc-DE.pdf,
% etoc.dvi, etoc-DE.dvi, README.md.
%
% Running etex (or latex or pdflatex) on etoc.dtx extracts etoc.sty,
% etoc.ins, etoc.tex, etoc-DE.tex, and README.md. See README.md for
% further instructions.
```

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`,¹ 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 IV 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 keinsten 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 40) finden, die Nachahmung des Standardaussehens einer Inhaltsangabe.

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 III handelt davon. Mehr davon gibt es noch in Teil VII.

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 40 sichtbare Inhaltsangaben (und ein paar unsichtbare). Sie benutzt dabei nur eine `.toc`-Datei!² Darum sollte man zweimal nachdenken, bevor man manuell extra Befehle zu der `.toc`-Datei hinzufügt (siehe Abschnitt 14).

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^AT_EX-Umgebungen schaffen Gruppen.).

Darum machen die eingebauten „Zeilenstile“ (line styles), wie sie von **etoc** vorgeschlagen werden (die durch die [Hauptinhaltsangabe](#) in diesem Dokument gezeigt werden) keinen Gebrauch von Umgebungen. Tatsächlich nutzt in dieser Gebrauchsanweisung nur die [Inhaltsangabe](#) zu Beginn von Teil IV und in [Unterabschnitt 38.1](#) `enumerate`-Umgebungen, um Zeilenstile anzugeben.

¹<http://ctan.org/pkg/enumitem>

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

4. Zeilenstile und Stile der Inhaltsangabe (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 *Inhaltsangabenstil* (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.³

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 Inhaltsangabe

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 Inhaltsangabe ausgeführt (typischerweise wird es Inhalte ausgeben und definiert passende Markierungen für die Seitenüberschriften). Der Zweite wird nach der Inhaltsangabe ausgeführt.

etoc bietet vier (anpassbare) höhere Inhaltsabgabestile: `\etocmulticolstyle`, `\etoc-tocstyle`, `\etocruledstyle` und `\etocframedstyle`. Alle nutzen das `multicol`-Paket 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 Kompatibilitätsmodus zu wechseln, der die Standardeinstellungen der Dokumentenklasse nimmt.⁴ Dieses wird wie folgt aktiviert:

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

³Mit der `memoir`-Dokumentenklasse kennt **etoc** Appendix als Kapitelüberschrift.

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

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 Kompatibilitätsmodus für Zeilenstile aus, behält ihn aber für den Inhaltsabgabestil.

Man kann ihn für Zeilenstile anhand von `\etocstandardlines` wieder aktivieren. `\etococlines` 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 34](#) 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 „Inhaltsabgabestil“-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...`⁵ benutzt haben, existiert die lokale Inhaltsangabe nur durch seine Klone anderswo im Dokument.

Anhand von `\etocdefaultlines` oder dem Stil der Standard-Dokumentenklasse, 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.

Mein erstes **etoc**: TOC von [Teil I](#) ([Überblick](#))

3. Muss ich Computerfreak sein, um etoc benutzen zu können?	7
3.1 Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben	7
4. Zeilenstile und Stile der Inhaltsangabe (toc display style)	8
4.1 <code>\etocsetstyle</code> für den Zeilenstil	8

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

5. Ein erstes Beispiel

4.2	<code>\etocsettocstyle</code> für den Stil der Inhaltsangabe	8
4.3	Kompatibilismus	8
5.	Ein erstes Beispiel	9
6.	Ein zweites Beispiel	11
7.	Ein drittes Beispiel: emulating the book class	12
8.	Verlinkte Liste der Hauptbefehle	16

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 ganze Teil zu erstellen. (Wir definierten einen Unterabschnitt-Zeilensstil, 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 40](#) 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^AT_EX2e's `\@dottedtocline` inspiriert.

```
\begingroup\parindent 0pt \parfillskip 0pt \leftskip 0cm \rightskip 1cm
\etocsetstyle {section}
{
  {\leavevmode\leftskip 0cm\relax}
  {\bfseries\normalsize\makebox[.5cm][l]{\etocnumber.}%
   \etocname\nobreak\hfill\nobreak
   \rlap{\makebox[1cm]{\mdseries\etocpage}}\par}
}
\etocsetstyle {subsection}
{
  {\leavevmode\leftskip .5cm\relax }
  {\mdseries\normalsize\makebox[1cm][l]{\etocnumber}%
   \etocname\nobreak\hfill\nobreak
   \rlap{\makebox[1cm]{\etocpage}}\par}
}
\etocsetstyle {subsubsection}
{
  {\leavevmode\leftskip 1.5cm\relax }
  {\mdseries\normalsize\makebox[1cm][l]{\etocnumber}%
   \etocname\nobreak\hfill\nobreak
   \rlap{\makebox[1cm]{\etocpage}}\par}
}
\etocruledstyle[1]{\bfseries \Large My first \etoc: TOC 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 IV](#) und [Teil V](#). Diese Auswahl wird durch die Technik von *depth tags* realisiert, die in [Abschnitt 13](#) und [Abschnitt 40](#) beschrieben wird. Dessen Layout ist in etwa wie das von der [Hauptinhaltsangabe](#), obwohl die Zeilenstile ganz anders programmiert sind.

INHALTSVERZEICHNIS

PART IV – BEFEHLE FÜR DIE TOC-LINIENSTILE

26. DER \ETOCSETSTYLE-BEFEHL.....	52
<i>Die \etocname- und \etocpage-Befehle (26.1, p. 52). Die \etocskipfirstprefix und \etociffirst Befehle (26.2, p. 54). Der \etocnumber-Befehl (26.3, p. 54). Der \etocifnumbered-Schalter (26.4, p. 54). Die Befehle \etocthenname, \etocthenumber sowie \etocthepage (26.5, p. 55). Der \etoclink-Befehl (26.6, p. 55). Die Befehle \etocthelinkedname, \etocthelinkednumber, \etocthelinkedpage und \etocthelink (26.7, p. 55).</i>	
27. DER \ETOCSETLEVEL-BEFEHL.....	56
28. UMFANG DER ZUR .TOC-DATEI GEHÖRIGEN BEFEHLE	58
<i>Testen des Bereichs (28.1, p. 58). Dies ist eine (blass)rote Subsection zu Demonstrationszwecken (28.2, p. 58).</i>	
29. BIN ICH AUCH ROT?	58

PART V – BEFEHLE FÜR DEN TOC-DISPLAY-STIL

30. FESTLEGUNG DES TOC-DISPLAY-STILS.....	59
<i>Der Befehl \etocsettocstyle (30.1, p. 59). Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol (30.2, p. 60). Der Befehl \etococstyle (30.3, p. 60). Die Befehle \etocruledstyle, \etocruled und \etoclocalruled (30.4, p. 61). Die Befehle \etocframedstyle, \etocframed und \etoclocalframed (30.5, p. 61). Überschriften, Titel, \etocoldpar, \etocinnertopsep (30.6, p. 62). Der Kompatibilitätsmodus \etocstandarddisplaystyle (30.7, p. 62). Der Befehl \etocinline (30.8, p. 62).</i>	
31. GESTERNDEN VARIANTEN DES \TABLEOFCONTENTS ETC. BEFEHLS	63
32. DAS INHALTSVERZEICHNIS VON DIESEM TEIL	63
<i>Überprüfen des Kompatibilitätsmodus (32.1, p. 63). Ein Inhaltsverzeichnis mit Rahmen (32.2, p. 64). Eine (verrückte) Inline-Darstellung (32.3, p. 65).</i>	

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}
```

7. Ein drittes Beispiel: emulating the book class

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

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

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

7. Ein drittes Beispiel: 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 42](#)) 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 19](#).

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 T_EX and L^AT_EX 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 25](#).

First we set up some lengths. I use macro registers, not real L^AT_EX 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 25](#) 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\TOCverysmallvskip {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 indicated below which although 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”.

```
\makeatletter
\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}%
\makeatother
```

7. Ein drittes Beispiel: emulating the book class

The code for the part and chapter line styles:

```
\makeatletter
\etocsetstyle{part}
{}
{\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}
\makeatother
```

The remaining divisions share the same code, here abstracted into a macro:

```
\makeatletter
\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
\TOCsetlinestyle {section}      {\TOCnumwidthC}
\TOCsetlinestyle {subsection}   {\TOCnumwidthD}
\TOCsetlinestyle {subsubsection}{\TOCnumwidthE}
\TOCsetlinestyle {paragraph}    {\TOCnumwidthF}
\TOCsetlinestyle {subparagraph} {\TOCnumwidthG}
```

Note 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.)

In practice, typically one will want to use `\localtableofcontents`, hence we need to change a bit the global toc style for using `\section*` rather than `\chapter*` for example. When the time comes for these local “minitocs” one can do something like the following (of course all of that can be stored inside a macro, whose definition can be located in the preamble.)

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.

```
\makeatletter
\etocsettocstyle
  {\if@twocolumn \@restonecoltrue \onecolumn \else \@restonecolfalse \fi
   \setbox\TOCleaderbox\hbox to \TOCleaderboxwidth{\hss.\hss}%
   \parindent\z@
   \dimen@ 2.25em % soon to be \leftskip
   \section *{\kern\dimen@ % \dimen@ works here by sheer luck
    \contentsname
    \@mkboth {\MakeUppercase \contentsname}{\MakeUppercase \contentsname}}%
   \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 }
\makeatother
```

We then need to specify the line styles:

```
\makeatletter
\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}
\makeatother
\TOCsetlinestyle {subsection} {\TOCnumwidthD}
\TOCsetlinestyle {subsubsection}{\TOCnumwidthE}
```

8. Verlinkte Liste der Hauptbefehle

```
\TOCsetlinestyle {paragraph}    {\TOCnumwidthF}  
\TOCsetlinestyle {subparagraph} {\TOCnumwidthG}
```

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

See [section 25](#) for more.

8. Verlinkte Liste der Hauptbefehle

<code>\etocaftercontentshook</code>	<code>\etoclocalmulticol</code>	<code>\etocsettocdepth</code>
<code>\etocaftertitlehook</code>	<code>\etoclocalruled</code>	<code>\etocsettocstyle</code>
<code>\etocaftertochook</code>	<code>\etocmulticolstyle</code>	<code>\etocskipfirstprefix</code>
<code>\etocbeforetitlehook</code>	<code>\etocmulticol</code>	<code>\etocstandarddisplaystyle</code>
<code>\etocdefaultlines</code>	<code>\etocname</code>	<code>\etocstandardlines</code>
<code>\etocdepthtag.toc</code>	<code>\etocnumber</code>	<code>\etocthelinkedname</code>
<code>\etocdisplay</code>	<code>\etocobeydepthtags</code>	<code>\etocthelinkednumber</code>
<code>\etocframedstyle</code>	<code>\etocobeytoctocdepth</code>	<code>\etocthelinkedpage</code>
<code>\etocframed</code>	<code>\etocpage</code>	<code>\etocthelink</code>
<code>\etociffirst</code>	<code>\etocruledstyle</code>	<code>\etocthename</code>
<code>\etocifnumbered</code>	<code>\etocruled</code>	<code>\etocthenumber</code>
<code>\etocignoredepthtags</code>	<code>\etocsetlevel</code>	<code>\etocthepage</code>
<code>\etocignoretoctocdepth</code>	<code>\etocsetnexttocdepth</code>	<code>\etoc toc contents line</code>
<code>\etocinline</code>	<code>\etocsetstyle</code>	<code>\etoc toclines</code>
<code>\etoclink</code>	<code>\etocsettagdepth</code>	<code>\localtableofcontents</code>
<code>\etoclocalframed</code>	<code>\etocsettocdepth.toc</code>	<code>\tableofcontents</code>

Teil II.

Beliebig viele TOCs und lokale Versionen davon

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 7 Abschnitte und 6 Unterabschnitte. Der Name des ersten Abschnitts lautet „Beschriften und anderswo wieder benutzen“ und er hat die Gliederungsnummer „9“. Der Name des letzten Abschnitts lautet „Zwei Beispiele“ und er hat die Nummer „15“. Der erste Unterabschnitt heißt „Die hyperref-Option *bookmarksdepth*“ und er hat die Nummer „11.1“. Der Name des letzten Unterabschnitts ist „Ein TOC mit einer Hintergrundfarbe“ und dessen Nummer ist „15.2“.

9. Beschriften und anderswo wieder benutzen	17
10. Eine mächtige Funktionsweise von <i>etoc</i>: Die Wiederzuweisung von Levels mit von <i>\etocsetlevel</i>	18
11. Die <i>\etocsettocdepth</i>- und <i>\etocsetnexttocdepth</i>-Befehle	19
11.1. Die hyperref-Option <i>bookmarksdepth</i>	20
12. Der Befehl <i>\etocsettocdepth.toc</i>	20
12.1. Die Befehle <i>\etocobeytoctocdepth</i> und <i>\etocignoretoctocdepth</i> . . .	21
13. Die Befehle <i>\etocdepthtag.toc</i> und <i>\etocsettagdepth</i>	21
13.1. Die Befehle <i>\etocobeydepthtags</i> und <i>\etocignoredepthtags</i>	22
14. Befehle zur .toc-Datei hinzufügen	22
14.1. Die hyperref-Option <i>hidelinks</i>	23
15. Zwei Beispiele	23
15.1. Eine wunderschöne Beispielthese	23
15.2. Ein TOC mit einer Hintergrundfarbe	24

9. Beschriften und anderswo wieder benutzen

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

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

10. Eine mächtige Funktionsweise von **etoc**: Die Wiederzuweisung von Levels mit von `\etocsetlevel`

ein Beispiel: Die Inhaltsangabe von **Teil VI** ist innerhalb eines `float`, der auf dieser Seite erscheint: on this page.

Ich bin von weit her

33. Überblick über die Grundbefehle zur Stilbearbeitung	68
33.1. Ein lokales Inhaltsverzeichnis anpassen	68
33.2. Das Aussehen eines Inhaltsverzeichnisses anpassen	68
33.3. Inhaltsverzeichnisse darstellen	69
33.4. Label und Referenzen	69
34. Der Linienstandard des Pakets: <code>\etocdefaultlines</code>	69
35. etoc individuell anpassen	73
35.1. Den durch etoc vordefinierten Linienstil individualisieren	73
35.2. Die Darstellung eines Inhaltsverzeichnisses anpassen	74
36. Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis	76

Wir benutzen dies:

```
\begin{figure}[ht!]  
  \centering  
  \begingroup  
  \etocstandardlines  
  \renewcommand{\etocbkgcolorcmd}{\color{green!5}}  
  \renewcommand{\etocbelowtocskip}{0pt\relax}  
  \fboxseplex  
  \etocframedstyle [1]{\fbox{\makebox[.5\linewidth]{\etocfontminusone  
    \hyperref[toc:c]{I am from far away}}}  
  \tableofcontents \label{toc:d} \ref{toc:c}  
  \endgroup  
\end{figure}
```

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

10. 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 de Befehl `\etocsetlevel{<level_name>}{<number>}` abgeändert werden. Daraus und aus der Benutzung des `LATEX`-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 III](#) 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)⁶

11. 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. 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

⁶Normalerweise sollte man diese Inhaltsangabe der Paragraphen nicht im Kompatibilitätsmodus ausgeben. Der würde darauf bestehen, einen gigantischen Abstand nach rechts einzufügen.

12. Der Befehl `\etocsettocdepth.toc`

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 Änderungen auf die Lesezeichen zu verhindern. **etoc** hat `\etocsetnexttocdepth{<level>}`. Dessen Einfluss endet direkt nach der nächsten Inhaltsangabe. Daraus folgt, dass `\invisibletableofcontents` 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`.

11.1. 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`.

12. Der Befehl `\etocsettocdepth.toc`

Dieser Befehl `\etocsettocdepth.toc` implementiert einige Funktionalitäten von Paket `tocvsec2`⁷. 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 `\etocsettocdepth.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 `\etocsettocdepth.toc` akzeptiert, wie `\etocsettocdepth`, Namen und numerische Argumente. In dem Fall eines Namen als Argument ist der tatsächliche numerische Wert noch nicht festgelegt, wenn `.toc`-Datei erstellt wird. Es wird der Wert sein,

⁷<http://ctan.org/pkg/tocvsec2>. Ich danke D.B. meine Aufmerksamkeit auf die Inkompatibilität dieses Pakets mit **etoc** gerichtet zu haben.

12.1. Die Befehle `\etocbeytoctocdepth` und `\etocignoretoctocdepth`

der aktuell für das benannte Level der Inhaltsangabe (ohne `\etocignoretoctocdepth` 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 einzugrenzen. Zum Beispiel, um die Einfügung von Unter-unter-Abschnitten oder eines gegebenen Kapitels zu unterbinden.

Dieses wird in allen Inhaltsangaben ausgeführt.

12.1. Die Befehle `\etocbeytoctocdepth` und `\etocignoretoctocdepth`

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

13. Die Befehle `\etocdepthtag.toc` und `\etocsettagdepth`

Wie oben erwähnt, wenn das Argument für `\etocsettocdepth.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 `\etocsettocdepth.toc{partIII}` ein. An dem Ort, an dem der Effekt enden soll, setzt man `\etocsettocdepth.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*); (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

14. Befehle zur .toc-Datei hinzufügen

geschlossen wird. Für ein Beispiel, siehe hier: [Abschnitt 40](#).

13.1. Die Befehle `\etocobeydepthtags` und `\etocignoredepthtags`

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

14. 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 schlaunere 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`, `\etocignoretoctocdepth`, `\etocobeytoctocdepth` 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.

14.1. Die hyperref-Option `hidelinks`

Die farbigen Links (und auch die Kastenlinks) sind einwenig ärgerlich, wenn in Inhaltsverzeichnissen verwendet, insbesondere, 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}}
```

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).

15. Zwei Beispiele

15.1. 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 Seitenumbrüchen zu tun ist. Auch sind sie nicht besorgt über die potentielle mehrspaltige Nutzung.

```
\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][l]{\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][l]{\makebox[0pt]{\etocpage}}\par}
```


15. Zwei Beispiele

```
{%
\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
```

—*Meine wunderschöne These*

Kapitel 3	Muss ich Computerfreak sein, um etoc benutzen zu können?	7
3.1	<i>Beschränkungen in der Benutzung von Listen- Umgebungen für Inhaltsangaben</i>	7
Kapitel 4	Zeilenstile und Stile der Inhaltsangabe (toc display style)	8
4.1	<i>\etocsetstyle für den Zeilenstil</i>	8
4.2	<i>\etocsettocstyle für den Stil der Inhaltsangabe</i>	8
4.3	<i>Kompatibilismus</i>	8
Kapitel 5	Ein erstes Beispiel	9
Kapitel 6	Ein zweites Beispiel	11
Kapitel 7	Ein drittes Beispiel: emulating the book class	12
Kapitel 8	Verlinkte Liste der Hauptbefehle	16

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

Inhalt von Teil Eins	
3	Muss ich Computerfreak sein, um etoc benutzen zu können? 7
3.1	Beschränkungen in der Benutzung von Listen- Umgebungen für Inhaltsangaben 7
4	Zeilenstile und Stile der Inhaltsangabe (toc display style) 8
4.1	\etocsetstyle für den Zeilenstil 8
4.2	\etocsettocstyle für den Stil der Inhaltsanga- be 8
4.3	Kompatibilismus 8
5	Ein erstes Beispiel 9
6	Ein zweites Beispiel 11
7	Ein drittes Beispiel: emulating the book class 12
8	Verlinkte Liste der Hauptbefehle 16

`\tableofcontents \ref{toc:overview}` (cf. [Unterabschnitt 15.2](#) and [this other toc](#))

15. Zwei Beispiele

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. Dieses Design (mit anderen Farben) wird auch für [dieses andere Inhaltsverzeichnis](#) benutzt, und der Leser bezieht sich auf seine [Unterabschnitte](#), welche für die Kodierung vorgesehen sind.

Teil III.

Überraschende Anwendungen von **etoc**

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 10 Abschnitte und 4 Unterabschnitte. Der Name des ersten Abschnitts lautet „Das TOC für TOCs“ und er hat die Gliederungsnummer „16“. Der Name des letzten Abschnitts lautet „A TOC self-adjusting widths for its typesetting“ und er hat die Nummer „25“. Der erste Unterabschnitt heißt „Die Befehle `\etocglobaldefs` und `\etoclocaldefs`“ und er hat die Nummer „23.1“. Der Name des letzten Unterabschnitts ist „The `\etocifwasempty` command“ und dessen Nummer ist „24.3“.

16. Das TOC für TOCs	27
17. Willkürliche „Listen von ...“, <code>\etoccontentsline</code>	29
18. Ein TOC mit einem extravagantem Layout	30
19. Ein weiterer Kompatibilitätsmodus	31
20. Das TOC als ein Baum	34
21. Das TOC als ein Molekül	37
22. Das TOC als ein TikZ Mind-Map	39
23. Das TOC als eine Tabelle	43
23.1. Die Befehle <code>\etocglobaldefs</code> und <code>\etoclocaldefs</code>	48
24. Not displayed empty TOCs	49
24.1. The <code>\etoccheckemptiness</code> command	49
24.2. The <code>\etocnotocifnotoc</code> command	49
24.3. The <code>\etocifwasempty</code> command	49
25. A TOC self-adjusting widths for its typesetting	50

16. Das TOC für TOCs

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

⁸Die 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 . . .

16. Das TOC für TOCs

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, 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 `\etoccontentsline{visibletoc}{\thevisibletoc}`. Dieser `etoc`-Befehl `\etoccontentsline{<level_name>}{<name>}` hat die gleiche Wirkung wie:

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

aber sein Nutzen ist es ⁹, 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’).¹⁰

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 Prämbel >
\newcounter{visibletoc}
\renewcommand{\etocaftertitlehook}
```

⁹Mit `\addtocontents` anstatt `\addcontentsline`.

¹⁰ Das Paket bietet eine Stern-Variante `\etoccontentsline*`, 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.

```

    {\stepcounter{visibletoc}\etoccontentsline{visibletoc}{\thevisibletoc}}
\etocsetlevel{visibletoc}{6}
\begin{document}
  < document body >
\subsection{Surprising uses of etoc}
\begingroup
  \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, 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 38](#) gibt eine andere Verwendung von Verschachtelungsstufen an.

17. 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 exercise number will be recoverable via \etocname: v--here--v
\newcommand*{\exercisetotoc}{\etoccontentsline{exercise}{\theexerci}}
\newenvironment{exercise}{\begin{exerci}\exercisetotoc}{\end{exerci}}
\etocsetlevel{exercise}{6}

```

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 $\langle n \rangle$

18. Ein TOC mit einem extravaganten Layout

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\hskip.5em\etocname}\par}
    {\nopagebreak[3]}{}
  \etocruledstyle[4]{\Large\bfseries List of the exercises}
  \setlength{\columnseprule}{.4pt}
  \tableofcontents
\endgroup
```

18. Ein TOC mit einem extravaganten Layout

Hier ist ein Inhaltsverzeichnis, wo die Abschnitte aus einem bestimmten Kapitel durch einen Zahlenbereich dargestellt werden sollen (wie 18–22 bei einem bestimmten Kapitel, 42–49 für ein anderes ... natürlich in das Inhaltsverzeichnis automatisch eingefügt wurden).

Dies ist kein Abbildungseinschluss, das Inhaltsverzeichnis wird von seiner ursprünglichen `tex`-Quelle in dieses Dokument nach dem Austausch des Teils, Kapitels oder Abschnitts mit `Dummy-Teil`, `Dummy-Kapitel` und `Dummy-Abschnitt` (und da ist auch eine `Dummy-Seitenzahl`) produziert. Wir kopierten die Linienstile, wie im Original verwendet, und das Inhaltsverzeichnis zeigt folgende Situation an:

```
\etocsetlevel{dummyspart} {-1} \etocsetlevel{part} {2}
\etocsetlevel{dummyschapter}{0} \etocsetlevel{chapter}{2}
\etocsetlevel{dummyssection}{1} \etocsetlevel{section}{2}
\etocsetnexttocdepth{dummyssection}
```

Jedes Kapitel zeigt die Zahl nur von dem ersten und dem letzten Abschnitt an. Siehe in der Quelldatei `etoc.dtx` für die Details nach, wie das gemacht wird.

Eine andere Technik für diese Art der Dinge erläutert [Abschnitt 39](#).

INHALTSVERZEICHNIS

TEIL I

				KAPITEL.	SEITE.
Einleitung	1—8	5

Kongruenz.

ÜBUNG.

1.	Kongruenz von Subjekt und Verb	...	9—17	7
2.	Kongruenz von Substantiven und Adjektiven		18—22	9
	Kongruenz von Relativen und Antezedenten		23—25	

TEIL II

Government.

3.	Der Akkusativ	17
	Allgemeine Anwendungen	26—30	
	Besondere Anwendungen	31—37	
4.	Verbrektion über zwei Akkusative		...	38—41	23
5.	Der Kausativ	42—49	25
6.	Der Instrumental	27
	Allgemeine Anwendungen	50—54	
	Besondere Anwendungen	55—59	
7.	Der Dativ	33
	Allgemeine Anwendungen	60—65	
	Besondere Anwendungen	66—71	
8.	Der Ablativ	39
	Allgemeine Anwendungen	72—75	
	Besondere Anwendungen	76—86	
9.	Der Lokativ	45
	Allgemeine Anwendungen	87—92	
	Besondere Anwendungen	93—100	

19. 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,

19. Ein weiterer Kompatibilitätsmodus

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 Teilbereiche;

Hier ist beschrieben, wie Sie vorgehen müssen. Man nimmt in die Präambel:

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

Natürlich ist es so, wenn das Dokument nur ein Inhaltsverzeichnis hat, dann gibt es keine Notwendigkeit, die Befehle in ein Makro zu setzen oder sogar in eine Gruppe.¹¹ Mit dem Befehl `etoc` ist ein Inhaltsverzeichnis völlig identisch zu einem von der Dokumentenklasse: `article`, `book`, `scrartcl`, `scrbook`.¹² 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

¹¹Und 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.

¹²Für die `memoir`-Klasse braucht man ein bisschen mehr: Jeder der Befehle `\booknumberline`, `\partnumberline` und `\chapternumberline` müssen in ein `\let` gebettet werden und sie ergeben dann ein:

```
\etocsetstyle{chapter}{}{}{\savedchapterline{\savedchapternumberline
{\etocnumber}\etocname}{\etocpage}}{}
(und analog für part und book).
```


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

3. Muss ich Computerfreak sein, um <code>etoc</code> benutzen zu können?	7
3.1. Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben	7
4. Zeilenstile und Stile der Inhaltsangabe (<code>toc display style</code>)	8
<code>\etocsetstyle</code> für den Zeilenstil (4.1) – <code>\etocsettocstyle</code> für den Stil der Inhaltsangabe (4.2) – <i>Kompatibilismus</i> (4.3).	
5. Ein erstes Beispiel	9
6. Ein zweites Beispiel	11
7. Ein drittes Beispiel: <code>emulating the book class</code>	12
8. Verlinkte Liste der Hauptbefehle	16

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}
  {}{}
  {\savesubsectionline{\numberline{\etocnumber}\etocname}{\etocpage}}
  {}%
\fi}
{\savedsectionline{\numberline{{\color{cyan}\etocthenumber}}\etocname}{\etocpage}}
{}%
```

20. Das TOC als ein Baum

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.

20. Das TOC als ein Baum

Mit `tikz`¹³ und dem Paket `forest`¹⁴ 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 \LaTeX 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.¹⁵

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.¹⁶ Das ist der spätere Job einer `forest`-Umgebung, die den Inhalt von `\treetok` wiedergeben soll.

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

```
\newtoks\treetok
\newtoks\tmptok

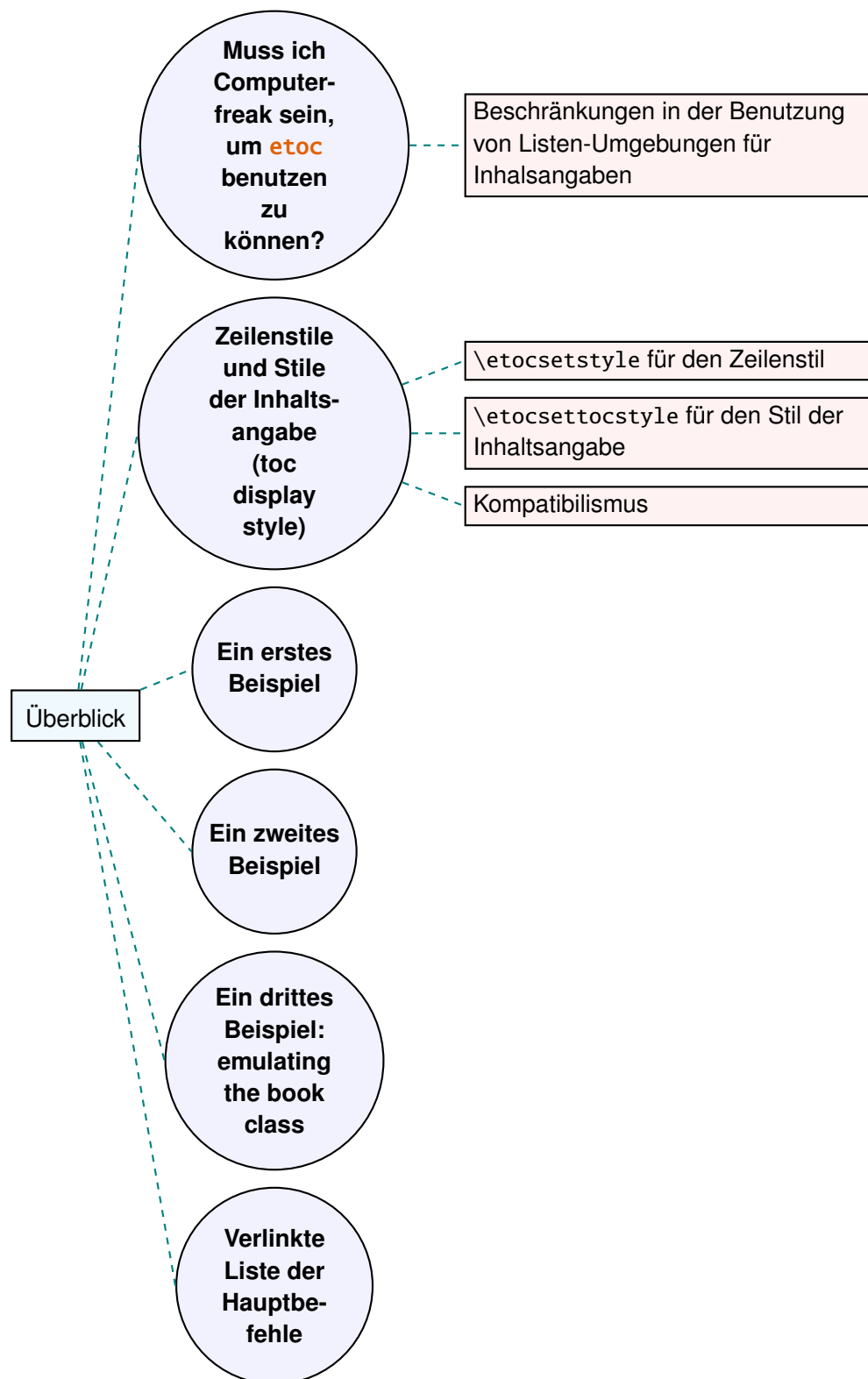
\newcommand*\appendtotok[2]{% #1=toks variable, #2=macro, expands once #2
  #1\expandafter\expandafter\expandafter
```

¹³<http://ctan.org/pkg/pgf>

¹⁴<http://ctan.org/pkg/forest>

¹⁵Dieses Handbuch bis zu der Version vom 14. Mai 2013 verwendete das Paket `tikz-qtrees`, 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).

¹⁶Es 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.



Ein Inhaltsverzeichnis in Form eines forest-Baums.

20. Das TOC als ein Baum

```
{\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[part: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,
    @\the\treetok
  \end{forest}
\end{figure}
```

Warum `\hypersetup{hidelinks}`? Wie in [Unterabschnitt 14.1](#) 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 Satz:

- Für das Drucken der Statistiken zu Beginn der einzelnen Abschnitte (siehe [Abschnitt 39](#)) wurden unbeschädigte Boxen benutzt (das Problem der Darstellung der Links funktioniert nicht).
- Zum Setzen des Inhaltsverzeichnisses als eine Tabelle mit der Methode von [Abschnitt 41](#); 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.

21. 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 $\text{T}_{\text{E}}\text{X}$ 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 `\chaptertok`, 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 % already done earlier
\newtoks\sectiontok \newtoks\subsectiontok \newcommand*\treenode{}}

\newcommand*\appendchildtree}[2]{% token list t1 becomes: t1 child {t2}
  \edef\tmp{\the#1 child {\the#2}}%
  #1\expandafter\tmp}%
}
\newcommand*\preparetreenode{%
  \tmptok\expandafter\etocthelinkednumber}% expanded one time (mandatory)
  \edef\treenode{node {\the\tmptok}}%
}

\etocsetstyle{section}
{\etocskipfirstprefix}
{\appendchildtree\treetok\sectiontok}
```

21. Das TOC als ein Molekül

```

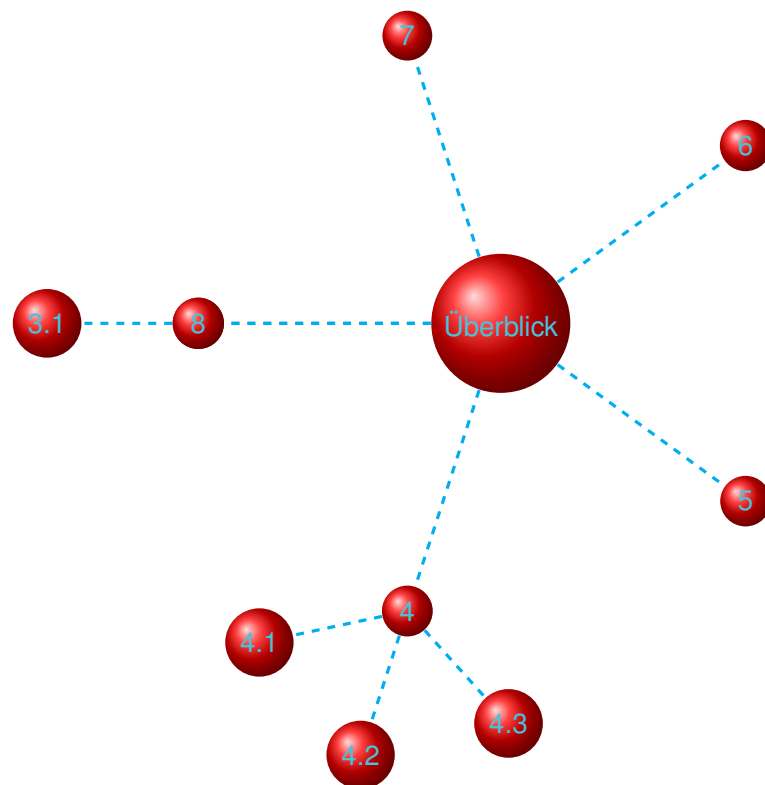
{\preparetreenode \sectiontok\expandafter{\treenode}}
{\appendchildtree\treetok\sectiontok}

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

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

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

```



Ein Inhaltsverzeichnis als ein „Molekül“.

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 „Molekül“, welches oben erscheint above. Hier ist der aktuelle, tatsächlich produzierte Code:

```

\begin{figure}[thp!]\centering
\etocsetnexttocdepth{subsection}

```

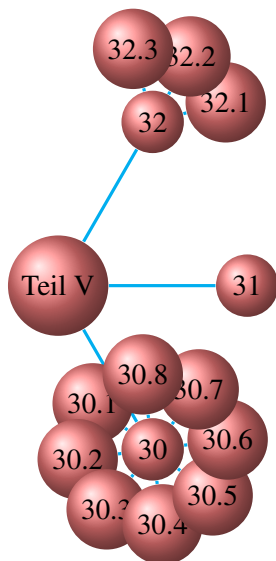
```

\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}
\end{figure}

```

Dieses **TikZ-TOC** ist komplett verlinkt, wie das vorherige **Forest-TOC** auch.



Seitlich ist das (völlig mit Hyper-Links versehene) Inhaltsverzeichnis von **Teil V**.

```

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

    \the\treetok
\end{tikzpicture}}%
...

```

22. Das TOC als ein TikZ Mind-Map

Das ist in dem gleichen Sinn wie das „Molekül“-Beispiel. Die Verwendung des ε -TeX-Primitivs-`\unexpanded` vereinfacht den Code.¹⁷

Es ist schwierig, alles auf eine Seite einzupassen. Da kommt `\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 following page.

```

\begin{group}
\newtoks\treetok
\newtoks\parttok

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

```

¹⁷das „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](#).




```

\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 }%
}

\newcounter{partco}

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

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

\etocsetstyle{section}
{
  {}
  {}
  % define the section node

```

22. Das TOC als ein TikZ Mind-Map

```
\edef\childnode{child {node {\unexpanded\expandafter{\etocthelinkednumber}
\unexpanded\expandafter{\etocthelinkedname}}}%
% append it to the current \parttok
\appendtotok\parttok\childnode
}
{}

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

\etocsetnexttocdepth{section}
% use of depth tags to cut out some sections.
\etocsettagdepth {preamble} {none}
\etocsettagdepth {overview} {part}
\etocsettagdepth {arbitrarily}{part}
\etocsettagdepth {surprising} {section}
\etocsettagdepth {linestyles} {part}
\etocsettagdepth {globalcmds} {part}
\etocsettagdepth {custom} {section}
\etocsettagdepth {tips} {part}
\etocsettagdepth {etocandworld}{part}
\etocsettagdepth {code} {section}

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

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

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 Tech-

nik, 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.toc tree
\immediate\write\TOCasTree{\the\treetok}%
```

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

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

23. 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`.

INHALTSVERZEICHNIS		
	Zusammenfassung	5
1	Vorwort	5
2	Lizenz	6
I	Überblick	6
3	Muss ich Computerfreak sein, um etoc benutzen zu können?	7
	3.1 <i>Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben</i>	7
4	Zeilenstile und Stile der Inhaltsangabe (toc display style)	8

23. Das TOC als eine Tabelle

	4.1 <code>\etocsetstyle</code> für den Zeilenstil	8
	4.2 <code>\etocsettocstyle</code> für den Stil der Inhaltsangabe	8
	4.3 <i>Kompatibilismus</i>	8
5	Ein erstes Beispiel	9
6	Ein zweites Beispiel	11
7	Ein drittes Beispiel: emulating the book class	12
8	Verlinkte Liste der Hauptbefehle	16
II	Beliebig viele TOCs und lokale Versionen davon	17
9	Beschriften und anderswo wieder benutzen	17
10	Eine mächtige Funktionsweise von etoc : Die Wiederzuweisung von Levels mit von <code>\etocsetlevel</code>	18
11	Die <code>\etocsettocdepth-</code> und <code>\etocsetnexttocdepth-</code> Befehle	19
	11.1 <i>Die hyperref-Option</i> <code>bookmarksdepth</code>	20
12	Der Befehl <code>\etocsettocdepth.toc</code>	20
	12.1 <i>Die Befehle</i> <code>\etocobeytoctocdepth</code> und <code>\etocignoretoctocdepth</code>	21
13	Die Befehle <code>\etocdepthtag.toc</code> und <code>\etocsettagdepth</code>	21
	13.1 <i>Die Befehle</i> <code>\etocobeydepthtags</code> und <code>\etocignoredepthtags</code>	22
14	Befehle zur .toc-Datei hinzufügen	22
	14.1 <i>Die hyperref-Option</i> <code>hidelinks</code>	23
15	Zwei Beispiele	23
	15.1 <i>Eine wunderschöne Beispielthese</i>	23
	15.2 <i>Ein TOC mit einer Hintergrundfarbe</i>	24
III	Überraschende Anwendungen von etoc	27
16	Das TOC für TOCs	27
17	Willkürliche „Listen von ...“, <code>\etoccontentsline</code>	29
18	Ein TOC mit einem extravagantem Layout	30
19	Ein weiterer Kompatibilitätsmodus	31
20	Das TOC als ein Baum	34
21	Das TOC als ein Molekül	37
22	Das TOC als ein TikZ Mind-Map	39
23	Das TOC als eine Tabelle	43
	23.1 <i>Die Befehle</i> <code>\etocglobaldefs</code> und <code>\etoclocaldefs</code>	48
24	Not displayed empty TOCs	49
	24.1 <i>The</i> <code>\etoccheckemptiness command</code>	49
	24.2 <i>The</i> <code>\etocnotocifnotoc command</code>	49
	24.3 <i>The</i> <code>\etocifwasempty command</code>	49

25	A TOC self-adjusting widths for its typesetting	50
IV	Befehle für die TOC-Linienstile	52
26	Der <code>\etocsetstyle</code> -Befehl	52
	26.1 Die <code>\etocname</code> - und <code>\etocpage</code> -Befehle	52
	26.2 Die <code>\etocskipfirstprefix</code> und <code>\etociffirst</code> Befehle	54
	26.3 Der <code>\etocnumber</code> -Befehl	54
	26.4 Der <code>\etocifnumbered</code> -Schalter	54
	26.5 Die Befehle <code>\etocthenname</code> , <code>\etocthenumber</code> sowie <code>\etocthepage</code>	55
	26.6 Der <code>\etoclink</code> -Befehl	55
	26.7 Die Befehle <code>\etocthelinkedname</code> , <code>\etocthelinkednumber</code> , <code>\etocthelinkedpage</code> und <code>\etocthelink</code>	55
27	Der <code>\etocsetlevel</code> -Befehl	56
28	Umfang der zur <code>.toc</code> -Datei gehörigen Befehle	58
	28.1 Testen des Bereichs	58
	28.2 Dies ist eine (blass)rote Subsection zu Demonstrationszwecken	58
29	Bin ich auch rot?	58
V	Befehle für den toc-Display-Stil	59
30	Festlegung des toc-Display-Stils	59
	30.1 Der Befehl <code>\etocsettocstyle</code>	59
	30.2 Die Befehle <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> und <code>\etoclocalmulticol</code>	60
	30.3 Der Befehl <code>\etococstyle</code>	60
	30.4 Die Befehle <code>\etocruledstyle</code> , <code>\etocruled</code> und <code>\etoclocalruled</code>	61
	30.5 Die Befehle <code>\etocframedstyle</code> , <code>\etocframed</code> und <code>\etoclocalframed</code>	61
	30.6 Überschriften, Titel, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	62
	30.7 Der Kompatibilitätsmodus <code>\etocstandarddisplaystyle</code>	62
	30.8 Der Befehl <code>\etocinline</code>	62
31	Gesternde Varianten des <code>\tableofcontents</code> etc. Befehls	63
32	Das Inhaltsverzeichnis von diesem Teil	63
	32.1 Überprüfen des Kompatibilitätsmodus	63
	32.2 Ein Inhaltsverzeichnis mit Rahmen	64

23. Das TOC als eine Tabelle

	32.3 Eine (verrückte) Inline-Darstellung	65
VI	etoc nutzen und anpassen	68
33	Überblick über die Grundbefehle zur Stilbearbeitung	68
	33.1 Ein lokales Inhaltsverzeichnis anpassen	68
	33.2 Das Aussehen eines Inhaltsverzeichnisses anpassen	68
	33.3 Inhaltsverzeichnisse darstellen	69
	33.4 Label und Referenzen	69
34	Der Linienstandard des Pakets: \etocdefaultlines	69
35	etoc individuell anpassen	73
	35.1 Den durch etoc vordefinierten Linienstil individualisieren	73
	35.2 Die Darstellung eines Inhaltsverzeichnisses anpassen	74
36	Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis	76
VII	Tipps	79
37	Hacken „framed parboxes“	79
38	Umstellen der Ebenen	80
	38.1 Alle Unterabschnitte in diesem Dokument	80
39	Statistiken anzeigen	81
40	Tiefe Tags nehmen	83
41	Das TOC als eine Tabelle setzend (die alte Weise)	85
VIII	etoc und die Außenwelt	87
42	Kompatibilität mit anderen Paketen	87
43	T _E X-nische Sachen	89
44	Fehler und Katastrophen	89
IX	Der Code	90
45	Zeitstempel	90
46	Änderungen	90
47	Ausführung	92

```

\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}}{\hline}}
{\etocnumber&\etocname &\etocpage }
{}

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

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

\etocsetnexttocdepth {subsection}

\tableofcontents
\endgroup

```

Beim Untersuchen des Codes wird sich der Leser fragen, warum die `\hline` 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 `\hline`, 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 `\hline` 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 `\hline`. 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 Titel ein nicht nummeriertes Abstrakt folgte, mit dem `\etocifnumbered`-Schalter kam die Rettung.

Eine letzte technische Anmerkung: Weil wir `\hline` in die Äste setzen setzen, bestand keine

23. Das TOC als eine Tabelle

Notwendigkeit, die erweiterten Varianten `\etociffirst` und `\etocifnumbered` einzustellen.

Für die Hardliner ist der alte Weg zur Erlangung des exakt gleichen Resultats hier zu sehen [Abschnitt 41](#).

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

Abschnitt	Nummer	Seite
Muss ich Computerfreak sein, um etoc benutzen zu können?	3	7
Zeilenstile und Stile der Inhaltsangabe (toc display style)	4	8
Ein erstes Beispiel	5	9
Ein zweites Beispiel	6	11
Ein drittes Beispiel: emulating the book class	7	12
Verlinkte Liste der Hauptbefehle	8	16

```
\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}
```

23.1. Die Befehle `\etocglobaldefs` und `\etoclocaldefs`

In \LaTeX verschwindet die Bedeutung eines Befehls, definiert via `\newcommand\foo{...}` in einer Umgebung (oder Gruppe), vom \TeX -Speicher beim Austritt aus dieser Umgebung (oder Gruppe). Manchmal ist es jedoch erforderlich, Definitionen mit globaler Reichweite zu geben, für diese hat \TeX primitive Präfixe `\global`.

Standardmäßig sind die **etoc**-Definitionen von `\etocname` etc. ... lokal, Dies verursacht Probleme in bestimmten Zusammenhängen, nicht nur wie hier bei TOCs als Tabellen, sondern auch mit `enumitem-inline`-Varianten der Standardumgebungen, weil der Befehl `\item` dann eine Gruppe schließt.

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.

24. Not displayed empty TOCs

24.1. 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.¹⁸ This can be useful to authors of \LaTeX classes who for example wish to have a `\chapter` command doing systematically a `\localtableofcontents`, or for people producing files via automatic conversions and some of those might have sectioning commands and others not.

«Emptiness» means that no `\contentsline` command would get executed within the scope of the local table of contents — empty line styles by themselves do not make the TOC empty. `etoc` always executes the `\etocaftertochook` command; and the test for emptiness itself executes everything else found in the `.toc` file. See [section 14](#) in this context.

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

For a finalized document compiled with initially no auxiliary files, the first \LaTeX run will declare all local TOCs non empty and print for each of them a heading (and no contents naturally). The second \LaTeX run will then correctly decide which local TOC is empty or not.

24.2. 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.

24.3. The `\etocifwasempty` command

The command `\etocifwasempty{<YES>}{<NO>}` executes `<YES>` if the previous TOC was found to be empty and `<NO>` if it 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 `<NO>` branch.

¹⁸Thanks to Paul Gaborit who asked for such a feature.

25. A TOC self-adjusting widths for its typesetting

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

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

This conditional may wrongly say that the local TOC is empty or not empty until \LaTeX compilations stabilize. But if it says that a local TOC is empty, this does mean that `etoc` considered the just encountered local table of contents to be empty (for that run) and thus printed nothing (not even a `\par`).

25. A TOC self-adjusting widths for its typesetting

This is a continuation of [section 7](#). The goal is to adjust automatically the “numwidths” used for typesetting the unit numbers in the (local) tables of contents.

```
\newcommand*\TOCcomputenumwidths [1][\% #1=empty or "local"]
\begingroup
\def\TOCnumwidthB {0pt}%
\def\TOCnumwidthC {0pt}%
\def\TOCnumwidthD {0pt}%
\def\TOCnumwidthE {0pt}%
\def\TOCnumwidthF {0pt}%
\def\TOCnumwidthG {0pt}%
\etocsetstyle{part}{}{}{}{}{}%
\etocsetstyle{chapter}{}
  {\setbox0\hbox{\bfseries\etocthenumber\kern0.5em}}
  {\ifdim\wd0>\TOCnumwidthB\edef\TOCnumwidthB{\the\wd0}\fi}{}%
\etocsetstyle{section}{}
  {\setbox0\hbox{\bfseries\etocthenumber\kern0.5em}}
  {\ifdim\wd0>\TOCnumwidthC\edef\TOCnumwidthC{\the\wd0}\fi}{}%
\etocsetstyle{subsection}{}
  {\setbox0\hbox{\etocthenumber\kern0.5em}}
  {\ifdim\wd0>\TOCnumwidthD\edef\TOCnumwidthD{\the\wd0}\fi}{}%
\etocsetstyle{subsubsection}{}
  {\setbox0\hbox{\etocthenumber\kern0.5em}}
  {\ifdim\wd0>\TOCnumwidthE\edef\TOCnumwidthE{\the\wd0}\fi}{}%
\etocsetstyle{paragraph}{}
  {\setbox0\hbox{\etocthenumber\kern0.5em}}
  {\ifdim\wd0>\TOCnumwidthF\edef\TOCnumwidthF{\the\wd0}\fi}{}%
\etocsetstyle{subparagraph}{}
  {\setbox0\hbox{\etocthenumber\kern0.5em}}
  {\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
}%

```

One invokes either only once `\TOCcomputenumwidths` (with a prior `\etocsetnexttocdepth{all}`) or, alternatively multiple times, using the optional argument:

```

\begin{group}
\etocsettocdepth{paragraph}% for example
\TOCcomputenumwidths [local]%
\localtableofcontents
\end{group}

```

Then each local table of contents will compute the various “numwidths” to match exactly (with an extra 0.5em), for each sectioning unit (apart from `\part` as it has no associated “numwidth” in the table of contents code of the book class), the widest number found. As the code makes global assignments, it is either to be used only once, or for each local table of contents. Notice that the bold font serves for both chapter and section styles, but the code from [section 7](#) does not use bold for sections in the global TOC, 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 `\TOCcomputenumwidth` should insert the `\bfseries` in the box only after testing that optional parameter `local` was used, but well.

Teil IV.

Befehle für die TOC-Linienstile

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 4 Abschnitte und 9 Unterabschnitte. Der Name des ersten Abschnitts lautet „Der `\etocsetstyle`-Befehl“ und er hat die Gliederungsnummer „26“. Der Name des letzten Abschnitts lautet „Bin ich auch rot?“ und er hat die Nummer „29“. Der erste Unterabschnitt heißt „Die `\etocname`- und `\etocpage`-Befehle“ und er hat die Nummer „26.1“. Der Name des letzten Unterabschnitts ist „Dies ist eine (blass)rote Subsection zu Demonstrationszwecken“ und dessen Nummer ist „28.2“.

Inhalt von Teil IV

26 Der `\etocsetstyle`-Befehl (Seite 52)

- 26.1 Die `\etocname`- und `\etocpage`-Befehle (S. 52)
- 26.2 Die `\etocskipfirstprefix` und `\etociffirst` Befehle (S. 54)
- 26.3 Der `\etocnumber`-Befehl (S. 54)
- 26.4 Der `\etocifnumbered`-Schalter (S. 54)
- 26.5 Die Befehle `\etocthenname`, `\etocthenumber` sowie `\etocthepage` (S. 55)
- 26.6 Der `\etoclink`-Befehl (S. 55)
- 26.7 Die Befehle `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` und `\etocthelink` (S. 55)

27 Der `\etocsetlevel`-Befehl (Seite 56)

28 Umfang der zur `.toc`-Datei gehörigen Befehle (Seite 58)

- 28.1 Testen des Bereichs (S. 58)
- 28.2 Dies ist eine (blass)rote Subsection zu Demonstrationszwecken (S. 58)

29 Bin ich auch rot? (Seite 58)

26. Der `\etocsetstyle`-Befehl

26.1. Die `\etocname`- und `\etocpage`-Befehle

Hier soll erläutert werden, wie `etoc` dazu verwendet wurde, das Inhaltsverzeichnis am Beginn des zweiten Teils zu erstellen. Es handelt sich dabei um ein lokales Inhaltsverzeichnis, weshalb der Befehl `\localtableofcontents` verwendet wurde.

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:

```

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

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

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

```

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 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

¹⁹Das 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.

26. Der `\etocsetstyle`-Befehl

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).

26.2. Die `\etocskipfirstprefix` und `\etociffirst` Befehle

Der gewählte `subsubsection`-Stil machte dies, angewiesen von dem Befehl `\etocskipfirstprefix`, der `etoc` befiehlt, für den ersten Punkt *nicht* den angepassten `<prefix>`-Code zu nehmen.²⁰

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.

26.3. 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.

26.4. Der `\etocifnumbered`-Schalter

Die `\fbox` würde ein unästhetisches Ergebnis bringen im Fall eines nicht nummerierten Abschnitts (welches beispielsweise im Inhaltsverzeichnis gelandet ist mittels eines `\addcontentsline` Befehls).²¹

Der `\etocifnumbered{<A>}{}`-Befehl führt `<A>` aus, wenn die Nummer existiert, und `` 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
```

²⁰Mit der Version vor 1.08 musste es das letzte Zeichen im `<start>`-Code sein. Es kann dort nun überall erscheinen.

²¹Wie man sieht, wird `\fcolorbox` verwendet wird statt `\fbox`. Aufgrund einer Neudefinition – durchgeführt vom `xcolor`-Paket – haben wir `\fbox` verwendet (und nicht `hyperref`), sonst hätten wir `\protect\fbox` benötigt.

26.5. Die Befehle `\etocthenname`, `\etocthenumber` sowie `\etocthepage`

```
{\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 `<contents>`-Teil des `section`-Levels hätte das behoben: `{... (page \etocpage)\par}`.

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

26.5. Die Befehle `\etocthenname`, `\etocthenumber` sowie `\etocthepage`

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 Formatierung zu ändern. Diese Funktionalität wird von `\etocthenname`, `\etocthenumber` und `\etocthepage` bereitgestellt.

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

26.6. Der `\etoclink`-Befehl

Der Befehl `\etoclink{<linkname>}` 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{\etocthenname}` das ursprüngliche `\etocname`. Es ist darauf zu achten, dass wenn `hyperref` nicht damit beauftragt wurde, die Seitenzahl mit einem Link zu versehen (mittels seiner Option `linktoc=all`), dann hat `etoc`’s `\etocpage` auch keinen Link, `\etoclink{\etocthepage}` allerdings schon.

Der Befehl `\etoclink` ist robust.

26.7. Die Befehle `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` und `\etocthelink`

Das Folgende ist für fortgeschrittene Anwendungen von fortgeschrittenen Benutzern. Das `\etocthenname`-Makro wurde bereits schon einmal erwähnt; verwendet man es in Anweisungen wie `\global\let\lastone\etocthenname` in einem `etoc` line-Stil, definiert das Macro `\lastone` so, dass es sich ausbreitet bis zum letzten Namen auf dem korrespondierenden Level. Jedoch bestand bisher keine Möglichkeit, dass Gleiche auch für die Link-Daten zu tun.

27. Der `\etocsetlevel`-Befehl

Das in diesem Paket enthaltene `\etocthelinkedname` stellt dies nun zur Verfügung, inklusive der Link-Daten. Es wurde in dieser Dokumentation benutzt bei Erstellung der [Beispiele mit Bäumen](#).

Ebenfalls bereit gestellt mit ähnlichen Funktionen: `\etocthelinkednumber`, `\etocthelinkedpage` (welches nur einen Link enthält, wenn `hyperref` einen zu dieser Seitenzahl hinzugefügt hat) und `\etocthelink{<linkname>}`, welches es erlaubt einen Link mit einem beliebigen Name zu erstellen.

All diese Befehle sind *nicht* robust. In der Tat wurde mit 1.08a `\etocname` als robuste Variante von `\etocthelinkedname` gestartet.

Veralteter Beispiel (ein Inhaltsverzeichnis als eine Inline-enumitem-Umgebung): **30.** Festlegung des toc-Display-Stils (S. 59); **31.** Gesternde Varianten des `\tableofcontents` etc. Befehls (S. 63), und **32.** Das Inhaltsverzeichnis von diesem Teil (S. 63).

```
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 zum gleichen Resultat:

```
Beispiel\begin{group}\etocglobaldefs % <-- NEU mit etoc 1.08.
\etocsetstyle {section}
{
\begin{itemize*}[itemjoin={{; }}, itemjoin*={{, und }}}
{}
{\item [{\bfseries\etocnumber.}
\etocname\ (\emph{S. \etocpage })}
\end{itemize*}.}%
\etocsetnexttocdepth {section}%
\etocsettocstyle {(ein Inhaltsverzeichnis als eine Inline-
\texttt{enumitem}-Umgebung): }{}%
\etocinline\tableofcontents \ref{toc:globalcmds}
\end{group}
```

Beispiel(ein Inhaltsverzeichnis als eine Inline-enumitem-Umgebung): **30.** Festlegung des toc-Display-Stils (S. 59); **31.** Gesternde Varianten des `\tableofcontents` etc. Befehls (S. 63), und **32.** Das Inhaltsverzeichnis von diesem Teil (S. 63).

27. Der `\etocsetlevel`-Befehl

Wie bereits in [Teil III](#) 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 III](#) 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 \etoclines could have been used, too.
\etocmarkboth\contentsname
\etocmulticolstyle[1] % one-column display
  {\pdfbookmark[1]{Table of contents}{MAINTOC}% create a bookmark in the pdf
   \noindent\bfseries\Large
   \leaders\hrule height1pt\hfill
   \MakeUppercase{Table of Contents}}
\begingroup % use a group to limit the scope of the
  \etocsetlevel{subsection}{3} % subsection level change.
  \etocsetlevel{subsubsection}{4} % anything > tocdepth=3.
  \tableofcontents \label{toc:main}
\endgroup

```

Auf diese Art und Weise benutzen die „subsections“ die Stile, die ursprünglich für „sub-subsections“ vorgesehen waren, während diese gar nicht gedruckt werden. Ohne diese Modifikation wäre das Erscheinungsbild ein anderes gewesen: die Paketlinienstile sind dafür

29. Bin ich auch rot?

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^AT_EX-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.

28. Umfang der zur .toc-Datei gehörigen Befehle

28.1. 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
```

28.2. 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.

29. 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 V.

Befehle für den toc-Display-Stil

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 3 Abschnitte und 11 Unterabschnitte. Der Name des ersten Abschnitts lautet „Festlegung des toc-Display-Stils“ und er hat die Gliederungsnummer „30“. Der Name des letzten Abschnitts lautet „Das Inhaltsverzeichnis von diesem Teil“ und er hat die Nummer „32“. Der erste Unterabschnitt heißt „Der Befehl `\etocsettocstyle`“ und er hat die Nummer „30.1“. Der Name des letzten Unterabschnitts ist „Eine (verrückte) Inline-Darstellung“ und dessen Nummer ist „32.3“.

30. Festlegung des toc-Display-Stils	59
30.1. Der Befehl <code>\etocsettocstyle</code>	59
30.2. Die Befehle <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> und <code>\etoclocalmulticol</code>	60
30.3. Der Befehl <code>\etocstyle</code>	60
30.4. Die Befehle <code>\etocruledstyle</code> , <code>\etocruled</code> und <code>\etoclocalruled</code> . . .	61
30.5. Die Befehle <code>\etocframedstyle</code> , <code>\etocframed</code> und <code>\etoclocalframed</code> .	61
30.6. Überschriften, Titel, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	62
30.7. Der Kompatibilitätsmodus <code>\etocstandarddisplaystyle</code>	62
30.8. Der Befehl <code>\etocinline</code>	62
31. Gesternde Varianten des <code>\tableofcontents</code> etc. Befehls	63
32. Das Inhaltsverzeichnis von diesem Teil	63
32.1. Überprüfen des Kompatibilitätsmodus	63
32.2. Ein Inhaltsverzeichnis mit Rahmen	64
32.3. Eine (verrückte) Inline-Darstellung	65

30. 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})
```

30.1. Der Befehl `\etocsettocstyle`

Dies ist ein Befehl mit zwei notwendigen Argumenten:

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

30. Festlegung des toc-Display-Stils

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 \TeX 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-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.

30.2. 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_coloumns⟩]` können von 1 bis 10 gehen (default: 2; ist der Wert 1, so wird natürlich keine multicol-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⟩}.`

30.3. Der Befehl `\etocstyle`

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

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

30.3.1. Der Befehl `\etocstylewithmarks`

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

30.4. Die Befehle `\etocruledstyle`, `\etocruled` und `\etoclocalruled`

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.

30.3.1.1. Möchten wir wirklich paragraph-Einträge im Inhaltsverzeichnis?

30.3.1.2. Wirklich?

30.4. 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 *TeX*, ein Dokument-Vorbereitung-System.): wenn es nicht in eine Zeile passt, dann sollte es in eine `\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\fboxseplex  
  \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⟩}
```

30.5. 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).

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 35.2](#)).

Eine `minipage` wird benutzt, von daher ist das produzierte Inhaltsverzeichnis nicht kompatibel mit Seitenumbrüchen. Um diese zu erlauben, wird empfohlen, die Befehle `mdframed`²² oder `tcolorbox`²³ in den Argumenten von `\etocsettocstyle` zu verwenden.

Beispiele in diesem Dokument findet man auf den Seiten [18](#), [25](#), [65](#) und [78](#).

²²<http://ctan.org/pkg/mdframed>

²³<http://ctan.org/pkg/tcolorbox>

30.6. 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 multicol-Umgebung ein und eben dieses `\etocoldpar`²⁴ 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 *TeX*, *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`.

30.7. 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 Änderung von innerhalb der Klasse sollten berücksichtigt werden.

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

30.8. Der Befehl `\etocinline`

Mit `\etocinline` 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.

²⁴Dieser 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.

31. Gesternde Varianten des `\tableofcontents` etc. Befehls

Die Befehle `\tableofcontents`, `\localtableofcontents`, `\etocmulticol` und all ihre Cousins und Varianten haben Stern-Varianten (der Stern muss vor den anderen Argumenten kommen). Für alle Klassen außer `memoir` verhalten diese sich wie die Originale. Für `memoir` jedoch, schreibt das Original einen Eintrag in die `.toc`-Datei, genau so wie die Verwendung des echten `\tableofcontents` Befehls in dieser Klasse, während aber die Stern-Variante genau das nicht tut, wie es in `memoir` die Regel ist.

Sobald man damit beginnt, lokale Inhaltsverzeichnisse zu benutzen merkt man, dass der `memoir`-Standard, welcher dazu gedacht ist, einen `chapter`-Eintrag für jedes Inhaltsverzeichnis zu erzeugen, nicht praktisch ist. Der Befehl `\etocmemoirtoc\tocfmt{<Typ>}{<name>}` ändert das Format (`<Typ>` ist wieder `chapter`, `section`, `subsection` etc. und `<name>` kann bspw. `\contentsname` sein.) Die Standardwerte für diese Größen sind `chapter` und `\contentsname`.

Das Format der tatsächlichen Überschrift des Inhaltsverzeichnisses sollte ebenfalls ordnungsgemäß gesetzt werden (z. B. mit `\etocstyle`), um einfach die selbe Einheitsgröße zu nutzen wie im ersten Argument von `\etocmemoirtoc\tocfmt`.

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 (wenn der Standard nicht geändert wurde, wie oben angedeutet), 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.

Unabhängig von der Situation hat man bei der `memoir`-Klasse immer ein Hook-Makro namens `\etocaftertitlehook`, welches man unterdrückt, indem man die Stern-Varianten der Darstellungsbefehle nutzt. Außer für die `memoir`-Klasse ist dieser Hook dazu gedacht, überhaupt nichts zu tun.

Außerdem gibt es auch `\etocaftercontentshook`, `\etocbeforetitlehook` und `\etoc-aftertochook` all drei ähnlich definiert um nichts zu tun und die für einige spezielle Effekte genutzt werden können.

32. Das Inhaltsverzeichnis von diesem Teil

32.1. Überprüfen des Kompatibilitätsmodus

Als drittes Beispiel wollen wir nun das Inhaltsverzeichnis dieses Teils setzen. Dazu wollen wir zuerst den Kompatibilitätsmodus testen.²⁵ Das ursprüngliche Inhaltsverzeichnis ist unsichtbar und wurde am Anfang von [Teil V](#) definiert, wobei es mit einem Label versehen wurde.

```
\KOMAoptions{toc=left}
\etocstandarddisplaystyle % wird benötigt, damit toc=left gilt
\etocstandardlines
\tableofcontents \ref{toc:globalcmds}
```

²⁵ Dieses Dokument wurde als `scrartcl` gesetzt. Wir wollen nun testen, ob der Kompatibilitätsmodus des `etoc`-Pakets die Anpassungen innerhalb der Klassenbefehle umsetzt.

Inhaltsverzeichnis

30. Festlegung des toc-Display-Stils	59
30.1. Der Befehl <code>\etocsettocstyle</code>	59
30.2. Die Befehle <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> und <code>\etoclocalmulticol</code>	60
30.3. Der Befehl <code>\etoclocalmulticolstyle</code>	60
30.4. Die Befehle <code>\etocruledstyle</code> , <code>\etocruled</code> und <code>\etoclocalruled</code>	61
30.5. Die Befehle <code>\etocframedstyle</code> , <code>\etocframed</code> und <code>\etoclocalframed</code>	61
30.6. Überschriften, Titel, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	62
30.7. Der Kompatibilitätsmodus <code>\etocstandarddisplaystyle</code>	62
30.8. Der Befehl <code>\etocinline</code>	62
31. Gesternde Varianten des <code>\tableofcontents</code> etc. Befehls	63
32. Das Inhaltsverzeichnis von diesem Teil	63
32.1. Überprüfen des Kompatibilitätsmodus	63
32.2. Ein Inhaltsverzeichnis mit Rahmen	64
32.3. Eine (verrückte) Inline-Darstellung	65

32.2. 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{\etocinnerrightsep}{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{\etocprehrule}{\hrule height 1pt}
\renewcommand{\etocprehrulecolorcmd}{\color{red!25}}
\renewcommand{\etocposthrule}{\hrule height 1pt}
\renewcommand{\etocposthrulecolorcmd}{\color{red!25}}
% Befehle für die Rechtecke links und rechts des Inhaltsverzeichnisses
\renewcommand{\etocleftvrule}{\vrule width 5pt}
\renewcommand{\etocrightvrule}{\vrule width 5pt}
\renewcommand{\etocleftvrulecolorcmd}{\color{red!25}}
\renewcommand{\etocrightvrulecolorcmd}{\color{red!25}}
% mit \fcolorbox legen wir einen farbigen Rahmen für den Titel an
\fbxrule1pt
\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 |toc:b|.}}}
\begin{figure}[ht!]

```



```

\centering
\tableofcontents \label{toc:b} \ref{toc:globalcmds}
\end{figure}
\endgroup

```

*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`.*

Festlegung des toc-Display-Stils

30, S. 59

Der Befehl `\etocsettocstyle` 30.1, S. 59

Die Befehle `\etocmulticolstyle`, `\etocmulticol` und `\etoclocal-`

`multicol` 30.2, S. 60

Der Befehl `\etoctocstyle` 30.3, S. 60

Die Befehle `\etocruledstyle`, `\etoc-`

`ruled` und `\etoclocalruled` 30.4, S. 61

Die Befehle `\etocframedstyle`, `\etoc-`

`framed` und `\etoclocalframed` 30.5, S. 61

Überschriften, Titel, `\etocoldpar`, `\etocinnertopsep` 30.6, S. 62

Der Kompatibilitätsmodus `\etoc-`

`standarddisplaystyle` 30.7, S. 62

Der Befehl `\etocinline` 30.8, S. 62

30.8, S. 62

Gesternde Varianten des `\tableof-` `contents` etc. Befehls 31, S. 63

Das Inhaltsverzeichnis von die- sem Teil 32, S. 63

Überprüfen des Kompatibilitätsmodus

. 32.1, S. 63

Ein Inhaltsverzeichnis mit Rahmen

. 32.2, S. 64

Eine (verrückte) Inline-Darstellung

. 32.3, S. 65

32.3. 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?*²⁶, *Zeilenstile und Stile der Inhaltsangabe (toc display style)*²⁷, *Ein erstes Beispiel, Ein zweites Beispiel, Ein drittes Beispiel: emulating the book class, Verlinkte Liste der Hauptbefehle.* **Beliebig viele TOCs und lokale Versionen davon:** *Beschriften und anderswo wieder benutzen, Eine mächtige Funktionsweise von **etoc**: Die Wiederzuweisung von Levels mit von `\etoc-setlevel`, Die `\etocsettocdepth`- und `\etocsetnexttocdepth`-Befehle*²⁸, *Der Befehl*

²⁶Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben.

²⁷`\etocsetstyle` für den Zeilenstil; `\etocsettocstyle` für den Stil der Inhaltsangabe; Kompatibilismus.

²⁸Die `hyperref`-Option `bookmarksdepth`.

`\etocsettocdepth.toc`²⁹, Die Befehle `\etocdepthtag.toc` und `\etocsettagdepth`³⁰, Befehle zur `.toc`-Datei hinzufügen³¹, Zwei Beispiele³². **Überraschende Anwendungen von `etoc`:** Das TOC für TOCs, Willkürliche „Listen von ...“, `\etoccontentsline`, Ein TOC mit einem extravaganten Layout, Ein weiterer Kompatibilitätsmodus, Das TOC als ein Baum, Das TOC als ein Molekül, Das TOC als ein TikZ Mind-Map, Das TOC als eine Tabelle³³, Not displayed empty TOCs³⁴, A TOC self-adjusting widths for its typesetting. **Befehle für die TOC-Linienstile:** Der `\etocsetstyle`-Befehl³⁵, Der `\etocsetlevel`-Befehl, Umfang der zur `.toc`-Datei gehörigen Befehle³⁶, Bin ich auch rot?. **Befehle für den `toc-Display`-Stil:** Festlegung des `toc-Display`-Stils³⁷, Gesternde Varianten des `\tableofcontents` etc. Befehls, Das Inhaltsverzeichnis von diesem Teil³⁸. **`etoc` nutzen und anpassen:** Überblick über die Grundbefehle zur Stilbearbeitung³⁹, Der Linienstandard des Pakets: `\etocdefaultlines`, `etoc` individuell anpassen⁴⁰, Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis. **Tipps:** Hacken „framed parboxes“, Umstellen der Ebenen⁴¹, Statistiken anzeigen, Tiefe Tags nehmen, Das TOC als eine Tabelle setzend (die alte Weise). **`etoc` und die Außenwelt:** Kompatibilität mit anderen Paketen, \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.}}
```

²⁹Die Befehle `\etocobeytocdepth` und `\etocignoretocdepth`.

³⁰Die Befehle `\etocobeydepthtags` und `\etocignoredepthtags`.

³¹Die `hyperref`-Option `hidelinks`.

³²Eine wunderschöne Beispielthese; Ein TOC mit einer Hintergrundfarbe.

³³Die Befehle `\etocglobaldefs` und `\etoclocaldefs`.

³⁴The `\etoccheckemptiness` command; The `\etocnotocifnotoc` command; The `\etocifwasempty` command.

³⁵Die `\etocname`- und `\etocpage`-Befehle; Die `\etocskipfirstprefix` und `\etociffirst` Befehle; Der `\etocnumber`-Befehl; Der `\etocifnumbered`-Schalter; Die Befehle `\etocthenname`, `\etocthenumber` sowie `\etocthepage`; Der `\etoclink`-Befehl; Die Befehle `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` und `\etocthelink`.

³⁶Testen des Bereichs; Dies ist eine (blass)rote Subsection zu Demonstrationszwecken.

³⁷Der Befehl `\etocsettocstyle`; Die Befehle `\etocmulticolstyle`, `\etocmulticol` und `\etoclocalmulticol`; Der Befehl `\etocdocstyle`; Die Befehle `\etocruledstyle`, `\etocruled` und `\etoclocalruled`; Die Befehle `\etocframedstyle`, `\etocframed` und `\etoclocalframed`; Überschriften, Titel, `\etocoldpar`, `\etocinnertopsep`; Der Kompatibilitätsmodus `\etocstandarddisplaystyle`; Der Befehl `\etocinline`.

³⁸Überprüfen des Kompatibilitätsmodus; Ein Inhaltsverzeichnis mit Rahmen; Eine (verrückte) Inline-Darstellung.

³⁹Ein lokales Inhaltsverzeichnis anpassen; Das Aussehen eines Inhaltsverzeichnisses anpassen; Inhaltsverzeichnisse darstellen; Label und Referenzen.

⁴⁰Den durch `etoc` vordefinierten Linienstil individualisieren; Die Darstellung eines Inhaltsverzeichnisses anpassen.

⁴¹Alle Unterabschnitte in diesem Dokument.

32.3. Eine (verrückte) Inline-Darstellung

```
\etocsetstyle{subsubsection}
{ (\itshape\etocskipfirstprefix}{, }{\etocname}{\/\upshape)}
\etocsettocstyle{Hier ist das inline dargestellte Inhaltsverzeichnis. }{.\par}
\tableofcontents
\endgroup
```

Teil VI.

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 „33“. Der Name des letzten Abschnitts lautet „Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis“ und er hat die Nummer „36“. Der erste Unterabschnitt heißt „Ein lokales Inhaltsverzeichnis anpassen“ und er hat die Nummer „33.1“. Der Name des letzten Unterabschnitts ist „Die Darstellung eines Inhaltsverzeichnisses anpassen“ und dessen Nummer ist „35.2“.

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

Überblick über die Grundbefehle zur Stilbearbeitung	33, S. 68	Der Linienstandard des Pakets: \etoc-defaultlines	34, S. 69
Ein lokales Inhaltsverzeichnis anpassen	33.1, S. 68	etoc individuell anpassen	35, S. 73
Das Aussehen eines Inhaltsverzeichnisses anpassen	33.2, S. 68	Den durch etoc vordefinierten Linienstil individualisieren	35.1, S. 73
Inhaltsverzeichnisse darstellen	33.3, S. 69	Die Darstellung eines Inhaltsverzeichnisses anpassen	35.2, S. 74
Label und Referenzen	33.4, S. 69	Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis	36, S. 76

33. Überblick über die Grundbefehle zur Stilbearbeitung

33.1. Ein lokales Inhaltsverzeichnis anpassen

```
\etocsetstyle{<Name der Ebene>}{<Start>}{<Präfix>}{<Inhalt>}{<Abschluss>}
\etocname, \etocnumber, \etocpage, \etocifnumbered{<A>}{<B>}
\etocthenname, \etocthenumber, \etocthe page, \etoclink{<Name des Links>}
```

33.2. Das Aussehen eines Inhaltsverzeichnisses anpassen

```
\etocmulticolstyle[<Anzahl der Spalten>]{<Überschrift>}
\etoc tocstyle[<Ebene>]{<Anzahl der Spalten>}{<Titel>}
\etoc tocstylewithmarks[<Ebene>]{<Anzahl der Spalten>}{<Titel>}{<Symbol>}
\etoc tocstylewithmarksnouc[<Ebene>]{<Anzahl der Spalten>}{<Titel>}{<Symbol>}
\etocruledstyle[<Anzahl der Spalten>]{<Titel>}
\etocframedstyle[<Anzahl der Spalten>]{<Titel>}
\etocsettocstyle{<vor dem Inhaltsverzeichnis>}{<nach dem Inhaltsverzeichnis>}
```

33.3. 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
```

33.4. 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.

34. Der Linienstandard des Pakets: `\etocdefaultlines`

Die Linienstile stammen bereits aus einer frühen Entwicklungsphase des Pakets, weswegen sie sich noch sehr an \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 Linienstils verändert werden können.

Wurden die vorgegebenen Linienstile nicht durch `\etocsetstyle` verändert, so kann man sie mit den Befehlen `\etocdefaultlines` bzw. `\etococlines` einschalten. Außer, 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 `\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 \LaTeX erzeugt wird.⁴² Verwendet man hingegen ein einspaltiges Layout und enthält der Teil nur wenige Unter-Unterabschnitte, so wird das Inhaltsverzeichnis im Vergleich zur mit \LaTeX gesetzten Variante etwas in die Breite gezogen. Weiterhin wirkt diese Darstellung weniger hierarchisch als der \LaTeX -Standard, da sich die Darstellung von Abschnitten und Unterabschnitten kaum unterscheidet. (Dies gilt besonders, wenn die Dokumentklasse den Befehl `\chapter` erlaubt.)

Da wir für das [zentrale Inhaltsverzeichnis](#) `\etocsetlevel{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 `\chapter` erlaubt. Daher werden die Abschnitte im Folgenden als Kapitel und die Unterabschnitte als Abschnitte gedruckt, was wir u. a. mit Hilfe des Befehls `\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 `\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. }}}
\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.
\etocsetnexttocdepth{1}
\begingroup
\etocsetlevel{section}{0}
\etocsetlevel{subsection}{1}
\renewcommand*{\etocfontone}{\normalfont \normalsize}
\renewcommand*{\etoclineleaders}{
  {\hbox{\normalfont\normalsize\hbox to 1ex {\hss.\hss}}}}
\sloppy
\tableofcontents
\endgroup
```

⁴²Außerdem werden dadurch einzelne Teil- bzw. Kapiteleinträge am Ende einer Seite oder einer Spalte vermieden, solange diese weiter untergliedert sind.

*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. 7

4. Zeilenstile und Stile der Inhalts- angabe (toc display style)

`\etocsetstyle` für den Zeilenstil
.....4.1, S. 8

`\etocsettocstyle` für den Stil der In-
haltsangabe4.2, S. 8

Kompatibilismus4.3, S. 8

5. Ein erstes Beispiel

6. Ein zweites Beispiel

7. Ein drittes Beispiel: emulating the book class

8. Verlinkte Liste der Hauptbefeh- le

Teil II. Beliebige viele TOCs und lokale Versionen davon

9. Beschriften und anderswo wie- der benutzen

10. Eine mächtige Funktionswei- se von **etoc**: Die Wiederzuweisung von Levels mit von `\etocsetlevel`

11. Die `\etocsettocdepth-` und `\etoc- setnexttocdepth`-Befehle

Die hyperref-Option `bookmarksdepth` ..
.....11.1, S. 20

12. Der Befehl `\etocsettocdep- th.toc`

Die Befehle `\etocobeytocdepth` und
`\etocignoretocdepth` ..12.1, S. 21

13. Die Befehle `\etocdepthtag.toc` und `\etocsettagdepth`

Die Befehle `\etocbeydepthtags` und
`\etocignoredepthtags`13.1, S. 22

14. Befehle zur `.toc`-Datei hinzu- fügen

Die hyperref-Option `hidelinks`
.....14.1, S. 23

15. Zwei Beispiele

Eine wunderschöne Beispielthese
.....15.1, S. 23

Ein TOC mit einer Hintergrundfarbe ...
.....15.2, S. 24

Teil III. Überraschende An- wendungen von **etoc**

16. Das TOC für TOCs

17. Willkürliche „Listen von ...“, `\etoccontentsline`

18. Ein TOC mit einem extravan- tem Layout

19. Ein weiterer Kompatibilitäts- modus

20. Das TOC als ein Baum

21. Das TOC als ein Molekül

22. Das TOC als ein TikZ Mind-Map

23. Das TOC als eine Tabelle

Die Befehle `\etocglobaldefs` und `\etoclocaldefs`23.1, S. 48

24. Not displayed empty TOCs

The `\etoccheckemptiness` command24.1, S. 49

The `\etocnotocifnotoc` command24.2, S. 49

The `\etocifwasempty` command24.3, S. 49

25. A TOC self-adjusting widths for its typesetting

Teil IV. Befehle für die TOC-Linienstile

26. Der `\etocsetstyle`-Befehl

Die `\etocname`- und `\etocpage`-Befehle26.1, S. 52

Die `\etocskipfirstprefix` und `\etociffirst` Befehle26.2, S. 54

Der `\etocnumber`-Befehl ...26.3, S. 54

Der `\etocifnumbered`-Schalter26.4, S. 54

Die Befehle `\etocthenname`, `\etocthenumber` sowie `\etocthepage`26.5, S. 55

Der `\etoclink`-Befehl26.6, S. 55

Die Befehle `\etocthelinkedname`, `\etocthelinkednumber`, `\etocthelinkedpage` und `\etocthelink`26.7, S. 55

27. Der `\etocsetlevel`-Befehl

28. Umfang der zur `.toc`-Datei gehörigen Befehle

Testen des Bereichs28.1, S. 58

Dies ist eine (blass)rote Subsection zu Demonstrationszwecken ...28.2, S. 58

29. Bin ich auch rot?

Teil V. Befehle für den toc-Display-Stil

30. Festlegung des toc-Display-Stils

Der Befehl `\etocsettocstyle`30.1, S. 59

Die Befehle `\etocmulticolstyle`, `\etocmulticol` und `\etoclocalmulticol`30.2, S. 60

Der Befehl `\etocctocstyle` .30.3, S. 60

Die Befehle `\etocruledstyle`, `\etocruled` und `\etoclocalruled`30.4, S. 61

Die Befehle `\etocframedstyle`, `\etocframed` und `\etoclocalframed`30.5, S. 61

Überschriften, Titel, `\etocoldpar`, `\etocinnertopsep`30.6, S. 62

Der Kompatibilitätsmodus `\etocstandarddisplaystyle`30.7, S. 62

Der Befehl `\etocinline` ...30.8, S. 62

31. Gesternde Varianten des `\tableofcontents` etc. Befehls

32. Das Inhaltsverzeichnis von diesem Teil

Überprüfen des Kompatibilitätsmodus32.1, S. 63

Ein Inhaltsverzeichnis mit Rahmen32.2, S. 64

Eine (verrückte) Inline-Darstellung
.....32.3, S. 65

Teil VI. **etoc** nutzen und anpassen

33. Überblick über die Grundbefehle zur Stilbearbeitung

Ein lokales Inhaltsverzeichnis anpassen
.....33.1, S. 68
Das Aussehen eines Inhaltsverzeichnisses anpassen33.2, S. 68
Inhaltsverzeichnisse darstellen
.....33.3, S. 69
Label und Referenzen33.4, S. 69

34. Der Linienstandard des Pakets: `\etocdefaultlines`

35. **etoc** individuell anpassen

Den durch **etoc** vordefinierten Linienstil individualisieren35.1, S. 73
Die Darstellung eines Inhaltsverzeichnisses anpassen35.2, S. 74

36. Ein weiteres Beispiel für ein farbiges Inhaltsverzeichnis

Teil VII. Tipps

37. Hacken „framed parboxes“

38. Umstellen der Ebenen

Alle Unterabschnitte in diesem Dokument38.1, S. 80

39. Statistiken anzeigen

40. Tiefe Tags nehmen

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

Teil VIII. **etoc** und die Außenwelt

42. Kompatibilität mit anderen Paketen

43. T_EX-nische Sachen

44. Fehler und Katastrophen

Teil IX. Der Code

45. Zeitstempel

46. Änderungen

47. Ausführung

35. **etoc** individuell anpassen

35.1. 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}
\newcommand*\etocfonttwo{\normalfont \normalsize}
\newcommand*\etocfontthree{\normalfont \footnotesize}
```

35. *etoc* individuell anpassen

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

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

\newcommand*\etocbaselinespreadminustwo{1}
\newcommand*\etocbaselinespreadminusone{1}
\newcommand*\etocbaselinespreadzero{1}
\newcommand*\etocbaselinespreadone{1}
\newcommand*\etocbaselinespreadtwo{1}
\newcommand*\etocbaselinespreadthree{.9}
\newcommand*\etocclineleaders
  {\hbox{\normalfont\normalsize\hbox to 2ex {\hss.\hss}}}
\newcommand*\etocabbrevpagename{p.~} % 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 agiert lediglich auf globaler Ebene, sodass die Einträge des Inhaltsverzeichnisses (hoffentlich) so dargestellt werden, als ob das Paket *etoc* fehlen würde.⁴³

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 \LaTeX 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.

35.2. 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.

⁴³Innerhalb der KOMA-script-Klassen muss der Befehl `\etocstandarddisplaystyle` genutzt werden, damit die Option `toc=left` auch für die Linieneinträge gilt.

35.2. Die Darstellung eines Inhaltsverzeichnisses anpassen

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

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

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

\newcommand*\etoclefttrule{\vrule}
\newcommand*\etocrighttrule{\vrule}
\newcommand*\etocbottomrule{\hrule}
\newcommand*\etoclefttrulecolorcmd{\relax}
\newcommand*\etocrighttrulecolorcmd{\relax}
\newcommand*\etocbottomrulecolorcmd{\relax}

\newcommand*\etocbkgcolorcmd{\relax}

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

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 `\etocbkgcolorcmd` - 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{\etocbottomrule}{\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 \TeX 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 \TeX -Befehlen, mit deren Hilfe man Rechtecke erzeugt, genutzt werden.

36. 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{\etocbottomrulecolorcmd}{\color{tocbackground}}

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

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

\newcommand\shiftedwhiterule[2]{%
  \hbox to \linewidth{\color{white}%
    \hskip#1\leaders\vrule height1pt\hfil}\nointerlineskip\vskip#2}

\etocsetstyle{subsubsection}{\etocskipfirstprefix}
{\shiftedwhiterule{\leftskip}{6pt}}
{\sffamily\footnotesize
  \leftskip2.5cm\hangindent1cm\rightskip1cm\noindent
  \hbox to 1cm{\color{subsecnum}\etocnumber\hss}%
  \color{black}\etocname\leaders\hbox to .2cm{\hss.}\hfill
  \rlap{\hbox to 1cm{\hss\etocpage\hskip.2cm}}\par
  \nointerlineskip\vskip3pt}
{}

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

\newcommand{\coloredstuff}[2]{%
  \leftskip0pt\rightskip0pt\parskip0pt
  \fboxsep0pt % 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\hspace{.2cm}}}%
    \etocname\hfill\makebox[.5cm][r]{\etocpage\hspace{.2cm}}}%
  \vskip6pt }
{}}

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

```

Das Inhaltsverzeichnis befindet sich in einer [float](#)-Umgebung, die man on the following page findet. Da diese Art von Code keine zusätzlichen Pakete verlangt, ist er etwas komplexer.⁴⁴ 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.

⁴⁴Zudem zeigt er die Vorliebe des Autors für die T_EX-Syntax.

Ich bin der Zwilling von diesem Verzeichnis (Klick mich an!)	
30	Festlegung des toc-Display-Stils 59
30.1	Der Befehl <code>\etocsettocstyle</code> 59
30.2	Die Befehle <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> und <code>\etoclocalmulticol</code> 60
30.3	Der Befehl <code>\etoclocalmulticol</code> 60
30.4	Die Befehle <code>\etocruledstyle</code> , <code>\etocruled</code> und <code>\etoclocalruled</code> 61
30.5	Die Befehle <code>\etocframedstyle</code> , <code>\etocframed</code> und <code>\etoclocalframed</code> 61
30.6	Überschriften, Titel, <code>\etocoldpar</code> , <code>\etocinnertopsep</code> ... 62
30.7	Der Kompatibilitätsmodus <code>\etocstandarddisplaystyle</code> . 62
30.8	Der Befehl <code>\etocinline</code> 62
31	Gesternde Varianten des <code>\tableofcontents</code> etc. Befehls 63
32	Das Inhaltsverzeichnis von diesem Teil 63
32.1	Überprüfen des Kompatibilitätsmodus 63
32.2	Ein Inhaltsverzeichnis mit Rahmen 64
32.3	Eine (verrückte) Inline-Darstellung 65

Teil VII.

Tipps

Zu Beginn wollen wir ein paar Fakten zu diesem Teil nennen. Er enthält 5 Abschnitte und 1 Unterabschnitt. Der Name des ersten Abschnitts lautet „Hacken „framed parboxes““ und er hat die Gliederungsnummer „37“. Der Name des letzten Abschnitts lautet „Das TOC als eine Tabelle setzend (die alte Weise)“ und er hat die Nummer „41“. Der erste Unterabschnitt heißt „Alle Unterabschnitte in diesem Dokument“ und er hat die Nummer „38.1“. Der Name des letzten Unterabschnitts ist „Alle Unterabschnitte in diesem Dokument“ und dessen Nummer ist „38.1“.

37. Hacken „framed parboxes“	79
38. Umstellen der Ebenen	80
38.1. Alle Unterabschnitte in diesem Dokument	80
39. Statistiken anzeigen	81
40. Tiefe Tags nehmen	83
41. Das TOC als eine Tabelle setzend (die alte Weise)	85

37. Hacken „framed parboxes“

```
\etocdefaultlines
\renewcommand\etocparbox{\hrule height 2pt depth 2pt}
\etocruled{\color{green}\fboxrule2pt\fboxseplex
\fbx{\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*.

Festlegung des toc-Display-Stils	30, S. 59	Die Befehle \etocruledstyle, \etocruled und \etoclocalruled	30.4, S. 61
Der Befehl \etocsettocstyle	30.1, S. 59	Die Befehle \etocframedstyle, \etocframed und \etoclocalframed	30.5, S. 61
Die Befehle \etocmulticolstyle, \etocmulticol und \etoclocalmulticol	30.2, S. 60	Überschriften, Titel, \etocoldpar, \etocinnertopsep	30.6, S. 62
Der Befehl \etocstyle	30.3, S. 60		

38. Umstellen der Ebenen

Der Kompatibilitätsmodus <code>\etocstandarddisplaystyle</code> . . .	30.7, S. 62	Überprüfen des Kompatibilitätsmodus	32.1, S. 63
Der Befehl <code>\etocinline</code> .	30.8, S. 62	Ein Inhaltsverzeichnis mit Rahmen	32.2, S. 64
Gesternde Varianten des <code>\tableofcontents</code> etc. Befehls . .	31, S. 63	Eine (verrückte) Inline-Darstellung	32.3, S. 65
Das Inhaltsverzeichnis von diesem Teil	32, S. 63		

38. Umstellen der Ebenen

Lasst uns alle Unterabschnitte zeigen und aufzählen, die in diesem Dokument vorkommen (Siehe [Teil III](#) 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
```

38.1. Alle Unterabschnitte in diesem Dokument

i. Beschränkungen in der Benutzung von Listen-Umgebungen für Inhaltsangaben (3.1, S. 7)	ignoredepthtags (13.1, S. 22)	<code>\etocpage</code> -Befehle (26.1, S. 52)
ii. <code>\etocsetstyle</code> für den Zeilenstil (4.1, S. 8)	viii. Die <code>hyperref</code> -Option <code>hidelinks</code> (14.1, S. 23)	xvi. Die <code>\etocskipfirstprefix</code> und <code>\etociffirst</code> Befehle (26.2, S. 54)
iii. <code>\etocsettocstyle</code> für den Stil der Inhaltsangabe (4.2, S. 8)	ix. Eine wunderschöne Beispielthese (15.1, S. 23)	xvii. Der <code>\etocnumber</code> -Befehl (26.3, S. 54)
iv. Kompatibilismus (4.3, S. 8)	x. Ein TOC mit einer Hintergrundfarbe (15.2, S. 24)	xviii. Der <code>\etocifnumbered</code> -Schalter (26.4, S. 54)
v. Die <code>hyperref</code> -Option <code>bookmarksdepth</code> (11.1, S. 20)	xi. Die Befehle <code>\etocglobaldefs</code> und <code>\etoclocaldefs</code> (23.1, S. 48)	xix. Die Befehle <code>\etocthenname</code> , <code>\etocthenumber</code> sowie <code>\etocthepage</code> (26.5, S. 55)
vi. Die Befehle <code>\etocobeytoctocdepth</code> und <code>\etocignoretoctocdepth</code> (12.1, S. 21)	xii. The <code>\etoccheckemptiness</code> command (24.1, S. 49)	xx. Der <code>\etoclink</code> -Befehl (26.6, S. 55)
vii. Die Befehle <code>\etocobeydepthtags</code> und <code>\etoc-</code>	xiii. The <code>\etocnotocifnotoc</code> command (24.2, S. 49)	xxi. Die Befehle <code>\etocthelinkedname</code> , <code>\etocthelinkednumber</code> , <code>\etocthelinkedpage</code> und <code>\etocthelink</code> (26.7, S. 55)
	xiv. The <code>\etocifwasempty</code> command (24.3, S. 49)	
	xv. Die <code>\etocname-</code> und	

xxii. Testen des Bereichs (28.1, S. 58)

xxiii. Dies ist eine (blass)rote Subsection zu Demonstrationzwecken (28.2, S. 58)

xxiv. Der Befehl `\etocsettocstyle` (30.1, S. 59)

xxv. Die Befehle `\etocmulticolstyle`, `\etocmulticol` und `\etoclocalmulticol` (30.2, S. 60)

xxvi. Der Befehl `\etoc tocstyle` (30.3, S. 60)

xxvii. Die Befehle `\etocruledstyle`, `\etocruled` und `\etoclocalruled` (30.4, S. 61)

xxviii. Die Befehle `\etocframedstyle`, `\etocframed` und `\etoclocalframed`

(30.5, S. 61)

xxix. Überschriften, Titel, `\etocoldpar`, `\etocinnertopsep` (30.6, S. 62)

xxx. Der Kompatibilitätsmodus `\etocstandarddisplaystyle` (30.7, S. 62)

xxxi. Der Befehl `\etocinline` (30.8, S. 62)

xxxii. Überprüfen des Kompatibilitätsmodus (32.1, S. 63)

xxxiii. Ein Inhaltsverzeichnis mit Rahmen (32.2, S. 64)

xxxiv. Eine (verrückte) Inline-Darstellung (32.3, S. 65)

xxxv. Ein lokales Inhaltsverzeichnis anpassen (33.1,

S. 68)

xxxvi. Das Aussehen eines Inhaltsverzeichnisses anpassen (33.2, S. 68)

xxxvii. Inhaltsverzeichnisse darstellen (33.3, S. 69)

xxxviii. Label und Referenzen (33.4, S. 69)

xxxix. Den durch **etoc** vordefinierten Linienstil individualisieren (35.1, S. 73)

xl. Die Darstellung eines Inhaltsverzeichnisses anpassen (35.2, S. 74)

xli. Alle Unterabschnitte in diesem Dokument (38.1, S. 80)

39. 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 21](#) 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{%
```

39. Statistiken anzeigen

```

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

```

40. 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 V](#) 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 kochen 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^AT_EX₂ε'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:

Ein TOC – nehmend tiefe Tags

I Überblick	6
II Beliebig viele TOCs und lokale Versionen davon	17
III Überraschende Anwendungen von etoc	27
IV Befehle für die TOC-Linienstile	52
V Befehle für den toc-Display-Stil	59
30 Festlegung des toc-Display-Stils	59
30.1 Der Befehl <code>\etocsettocstyle</code>	59
30.2 Die Befehle <code>\etocmulticolstyle</code> , <code>\etocmulticol</code> und <code>\etocalmulticol</code>	60
30.3 Der Befehl <code>\etocstyle</code>	60
30.3.1 Der Befehl <code>\etocstylewithmarks</code>	60

40. Tiefe Tags nehmen

30.3.1.1	Möchten wir wirklich paragraph-Einträge im Inhaltsverzeichnis?.....	61
30.3.1.2	Wirklich?.....	61
30.4	Die Befehle <code>\etocruledstyle</code> , <code>\etocruled</code> und <code>\etoclocalruled</code>	61
30.5	Die Befehle <code>\etocframedstyle</code> , <code>\etocframed</code> und <code>\etoclocalframed</code>	61
30.6	Überschriften, Titel, <code>\etocoldpar</code> , <code>\etocinnertopsep</code>	62
30.7	Der Kompatibilitätsmodus <code>\etocstandarddisplaystyle</code>	62
30.7.1	Die Befehle <code>\etocarticlestyle</code> , <code>\etocbookstyle</code> ,	62
30.8	Der Befehl <code>\etocinline</code>	62
31	Gesternde Varianten des <code>\tableofcontents</code> etc. Befehls	63
32	Das Inhaltsverzeichnis von diesem Teil	63
32.1	Überprüfen des Kompatibilitätsmodus	63
32.2	Ein Inhaltsverzeichnis mit Rahmen.....	64
32.3	Eine (verrückte) Inline-Darstellung.....	65
VI	<code>etoc</code> nutzen und anpassen	68
VII	Tipps	79
VIII	<code>etoc</code> und die Außenwelt	87

```

\etocsetnexttodepth {all}
\begingroup
\parindent 0pt \leftskip 0cm \rightskip .75cm \parfillskip -\rightskip
\newcommand*{\EndParWithPagenoInMargin}
{
\nobreak\hfill
\nobreak\makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
\par}
\renewcommand*\etoclineleaders
{
\hbox{\normalfont\normalsize\hbox to .75ex {\hss.\hss}}}
\newcommand*{\EndParWithPagenoInMarginAndLeaders}
{
\nobreak\leaders\etoclineleaders\hfill
\nobreak\makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
\par}
\etocsetstyle {part}
{
{}
{\leavevmode\leftskip 1cm\relax}
{\bfseries\large\llap{\makebox[1cm][r]{\etocnumber\ }}%
\etocname\EndParWithPagenoInMargin\smallskip}
{}
}
\etocsetstyle {section}
{
{}
{\leavevmode\leftskip 1.75cm\relax}
{\bfseries\normalsize\llap{\makebox[.75cm][l]{\etocnumber}}%
\etocname\EndParWithPagenoInMarginAndLeaders}
{}
}
\etocsetstyle {subsection}
{
{}
{\leavevmode\leftskip 2.75cm\relax}
{\mdseries\normalsize\llap{\makebox[1cm][l]{\etocnumber}}%
\etocname\EndParWithPagenoInMarginAndLeaders}
}

```

```

{}
\etocsetstyle {subsubsection}
{}
{\leavevmode\leftskip 4cm\relax }
{\mdseries\normalsize\llap{\makebox[1.25cm][l]{\etocnumber}}}%
\etocname\EndParWithPagenoInMarginAndLeaders}
{}
\etocsetstyle {paragraph}
{}
{\leavevmode\leftskip 5.5cm\relax }
{\mdseries\normalsize\llap{\makebox[1.5cm][l]{\etocnumber}}}%
\etocname\EndParWithPagenoInMarginAndLeaders}
{}
\etocsettagdepth {preamble} {none}
\etocsettagdepth {overview} {part}
\etocsettagdepth {arbitrarily}{part}
\etocsettagdepth {surprising} {part}
\etocsettagdepth {linestyles} {part}
\etocsettagdepth {globalcmds} {paragraph}
\etocsettagdepth {custom} {part}
\etocsettagdepth {tips} {part}
\etocsettagdepth {etocandworld}{part}
\etocsettagdepth {code} {none}
\renewcommand\etocoprul { \hrule height 3pt\relax }
\renewcommand\etocoprulcolorcmd {\color{blue}}
\renewcommand\etocaftercontentshook
{\medskip\begin{group} \color{blue}\hrule height 3pt \end{group} }
\etocruledstyle [1]{\Large\bfseries
\fbbox{\makebox[8cm]{A TOC using depth tags}}}
\sloppy
\tableofcontents
\endgroup

```

41. 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 grundlegende 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 ersten Zelle der ersten Reihe durchgeführt werden und danach verloren gehen.

So musste man auf die Technik zurückgreifen, die in [Abschnitt 20](#) 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 23](#) 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^AT_EX-Nutzern nicht bekannt sind (Makros könnten verwendet werden, doch das würde umständlich sein, außer vielleicht mit Hilfe von ε -T_EX `\nicht`

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

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 -TeX zu verwenden, \ nicht expandierte Primitive. Sehen Sie dazu zum Beispiel [Abschnitt 22](#).

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

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

\begingroup
\etocsetstyle{part}{}{}{\PreparePart \appendtotok\toctabletok\toctablepiece}{}

\etocsetstyle{section}{}{}{\PrepareSection \appendtotok\toctabletok\toctablepiece}{}

\etocsetstyle{subsection}{}{}{\PrepareSubsection\appendtotok\toctabletok\toctablepiece}{}

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

Teil VIII.

etoc und die Außenwelt

42. Kompatibilität mit anderen Paketen

Die Inhalte der .toc-Datei (falls bereits vorhanden) werden einmal in den Speicher von **etoc** getan, bei `\begin{document}`.⁴⁵

Man sollte **etoc** *nach* babel laden.

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 L^AT_EX am Ende der Kompilierung geschrieben wurde.

etoc lädt das Paket `multicol`.

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). So ist es inkompatibel mit der beamer-Klasse. Wie auch immer, wenn beamerim article-Modus verwendet wird, d. h. die article-Klasse in Verbindung mit dem beamerarticle-Paket, dann sollte **etoc** arbeiten.

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 toc-Eintrag.). Beispielsweise sind die Inhaltsverzeichnisse des vorliegenden Dokuments komplett verlinkt. Es ist kein Thema, ob **etoc** vor oder nach hyperref geladen wird.

Die Release 1.07k fügt Kompatibilität mit dem Paket tocloft hinzu:⁴⁶ Schritte wurden getan, um eine Neudefinition von `\tableofcontents` zu vermeiden, getan von tocloft an `\begin{document}`.

Solange **etoc** ungeschickt ist im Kompatibilitätsmodus, wird von tocloft die Anpassung sowohl für den Linienstil 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.

⁴⁵Neu mit 1.07m. Früherer Versionen benutzten die .toc-Datei bei `\usepackage{etoc}`. Danke an Denis Bitouzé für ein Babel-etoc Problem entdecken zu haben.

⁴⁶<http://ctan.org/pkg/tocloft>

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

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

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 eher⁴⁷ aus, gerade vor dem Schließen des Teils vom toc-Anzeigestil (und damit innerhalb einer Gruppe).

etoc zu verwenden, kann unvereinbar sein mit dem Paket tocvsec⁴⁸; jetzt lebt es, sortierend, wie es deaktiviert die Modifikationen des \tableofcontents durch tocvsec2 (getan im Stile von oben) und es kanzelt auch andere toc-bezogene Macros, aber es reimplantiert partiell ihre Funktionalität mit **\etocsettocdepth.toc**. In der Art und Weise, die letzten beiden L^AT_EX-Läufe sind für das erneute Nehmen des Befehls in einem Dokument notwendig, um einen Effekt im Inhaltsverzeichnis zu haben.

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^AT_EX-Lauf angezeigt werden.

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.

Es ist möglich, **etoc** und **tableof**⁴⁹ 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**-

⁴⁷Im Gegensatz zu \etocaftertochook, \etocaftercontentshook wird nicht ausgeführt, wenn tocdepth das Drucken des TOC nicht ermöglicht.

⁴⁸<http://ctan.org/pkg/tocvsec2>

⁴⁹<http://ctan.org/pkg/tableof>

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.

43. T_EX-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 vorzeitige Expansion geschützt. Auch `\etociffirst` und `\etoccontentsline`.

Andererseits `\etocthename`, `\etocthenumber`, `\etocthepage`, 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^AT_EX. Alle TOCs werden in Gruppierungen gesetzt.

44. 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: Insbesondere, 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 dies 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^AT_EX eine bestimmte Anzahl von Durchführungen haben, um für das Dokument das endgültige Aussehen zu erhalten (mindestens zweimal durchlaufen).

Teil IX.

Der Code

45. Zeitstempel

Dies ist die deutsche Dokumentation von 2015/08/29, erzeugt von der Quelldatei mit dem Stempel Time-stamp: <09-09-2016 at 21:49:18 CEST>. Die dokumentierte Paketversion ist v1.08g von 2015/08/29.

46. Änderungen

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. `\etocsetlevel` more economical.

v1.08e [2015/04/17]

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

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

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

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

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

v1.08d [2015/04/09]

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

Thanks to Christine Römer!

v1.08c [2015/03/30]

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

v1.08b [2015/03/18]

Bug fixes:

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

- `\etocpartname` (a macro used by the package own default line styles) was defined to be `\partname`, but this is not compatible at least with babel+fr 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`, `\etocxifnumbered`, `\etocglobaldefs` and `\etoclocaldefs`.

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

It is mandatory for such uses to issue `\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!

v1.07i [2014/04/22]

Fixes a bug with the 1.07k compatibility layer with `tocloft` which had broken the 1.07k (sic) compatibility with memoir (yes, memoir class 1.07k testing had been done before adding the `tocloft` thing to the source code ...). Also, etoc when detecting `tocvsec2` now checks if this is

under the memoir class, as then nothing special needs to be done to rescue `\tableofcontents`, contrarily to the situation with the native `tocvsec2`.

v1.07k [2014/03/06]

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

v1.07j [2013/12/03]

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

v1.07i [2013/10/21]

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

v1.07h [2013/10/16]

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

v1.07g [2013/10/13]

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

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

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

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

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

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

v1.07f [2013/03/07]

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

47. Ausführung

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`, `\etocthe-
page`, `\etoclink`,

`\etoccontentsline`, `\etoccontentsline*`

`\etocnopar`, `\etocaftercontentshook`

Modified command: `\etocmulticolstyle`

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

v1.06 [2012/12/07]

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

v1.05 [2012/12/01]

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

Compatibility with KOMA-script and memoir document classes.

v1.04 [2012/11/24]

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

`\tableofcontents \label{toc:1}`

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

`\tableofcontents \ref{toc:1}`

v1.02 [2012/11/18]

Initial version.

47. Ausführung

Diese Dokumentation wurde ohne Einbeziehung des Quellcodes erstellt.

Um die englische Dokumentation mit dem Quellcode zu produzieren, ist einzubeziehen:

kompilet etex mit `etoc.dtx` oder `etoc.ins`, um `etoc.tex` zu produzieren,
dann viermal latex mit `etoc.tex`, dann `dvipdfmx` mit `etoc.dvi`.

Um die deutsche Dokumentation mit Einbeziehung des Quellcodes zu erstellen, ist zu kompilet etex mit `etoc.dtx` oder `etoc.ins`, um `etoc-DE.tex` zu erzeugen, dann ist `\NoSourceCode` in `etoc-DE.tex` auf 0 zu setzen, dann ist latex dreimal mit `etoc-DE.tex` laufen zu lassen, dann `dvipdfmx` mit `etoc-DE.dvi`.

(Ignorieren Sie `dvipdfmx`-Warnungen oder, im Falle von Problemen mit einer alten Version von `dvipdfmx`, setzen Sie `\Withdvipdfmx` auf 0 in `etoc.tex` oder `etoc-DE.tex` um dann `pdflatex` laufen zu lassen.)