

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

27. Jahrgang Heft 3/2015 August 2015

3/2015

Impressum

»Die $\TeX$ nische Komödie« ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Autoren wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nicht gestattet. Alle Rechte zur weiteren Verwendung außerhalb von DANTE e.V. liegen bei den jeweiligen Autoren.

Beiträge sollten in Standard- $\LaTeX$ -Quellcode unter Verwendung der Dokumentenklasse dtk erstellt und per E-Mail oder Datenträger (CD/DVD) an untenstehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, $\LaTeX$ -Pakete oder Schriften notwendig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden. Weitere Informationen für Autoren findet man auf der Projektseite <http://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit LuaTeX, Version beta-0.80.0 (TeX Live 2015) (rev 5238) erstellt. Als Standard-Schriften kamen Linux Libertine, Linux Biolinum, DejaVu Mono und XITS Math zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2500

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung $\TeX$ e.V.
Postfach 10 18 40
69008 Heidelberg

E-Mail: dante@dante.de (DANTE e.V.)
dtkred@dante.de (Redaktion)

Druck: Konrad Triltsch Print und digitale Medien GmbH
Johannes-Gutenberg-Str. 1–3, 97199 Ochsenfurt-Hohestadt

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit: Gert Ingold Eberhard Lisse Rolf Niepraschk
Heiko Oberdiek Christine Römer

Redaktionsschluss für Heft 4/2015: 15. Oktober 2015

ISSN 1434-5897

Editorial

Liebe Leserinnen und liebe Leser,

Diese Komödie ist ausschließlich dem Thema »KOMA-Script« gewidmet. Dankenswerterweise haben Elke Schubert und Markus Kohm hauptsächlich dazu beigetragen, dass es überhaupt möglich war, ein Themenheft zu erstellen. Sicher gibt es sowohl das Buch zu KOMA-Script, als auch die ausführlichen Dokumentationen, die mit jeder T_EX-Installation auf dem Rechner vorhanden sind. Trotzdem existieren noch reichlich Problemstellungen, die bislang nicht erfasst wurden. Weitere Beiträge zum Thema verfassten Rainer-Maria Fritsch und Rolf Niepraschk.

Diese Komödie erstellt die Teilbibliografien erstmalig mit dem Programm Biber, anstelle des sonst üblichen B_BT_EX. Dadurch sind zum einen komplexere Bibliografiedatensätze und zum anderen beliebige Unicodezeichen in den Bibliografieeinträgen möglich. Die Anwendung von Biber und bibl_atex für einzelne Dokumente ist mittlerweile völlig unproblematisch und allen Anwendern zu empfehlen. Das Perlprogramm Biber kommt mit allen benötigten Perlbibliotheken, sodass der Anwender sich nicht um eine Installation der Skriptsprache Perl kümmern muss. Eine etwas andere Situation liegt jedoch vor, wenn verschiedene Dokumente mit eigenen Bibliografien zu einem großen Ganzen zusammengefügt werden müssen, wie es bei der »Komödie« der Fall ist. Was auf den ersten Blick einfach aussah, stellte sich dann als problematisch heraus; mit B_BT_EX braucht man nur eine bb_l-Datei zu erstellen, die über einen einfachen Befehl `\input` eingelesen werden kann. Dies ist deswegen so einfach, weil die bb_l-Datei bereits die formatierte Bibliografie darstellt. Mit Biber und demzufolge auch dem Paket bibl_atex, ist dies nicht mehr so einfach, denn die nun von Biber erzeugte bb_l-Datei, enthält nur noch Anweisungen, die vom Paket bibl_atex erst ausgewertet werden müssen, um eine Bibliografie zu erstellen. Nur so ist es möglich, die vielfältigen Anforderungen an eine Bibliografie auf der T_EX-Ebene zu erfüllen.

Ein erster Versuch wurde mit der letzten Ausgabe vorgenommen, jedoch wieder verworfen; der Abgabetermin an die Druckerei bestimmte das Ende des Versuches. Für diese Ausgabe wurde eine andere Lösung gewählt, die aber aller Voraussicht nach auch wieder verworfen werden wird. Sicher ist, dass der Befehl `\refsection` von bibl_atex auf einfache Weise Teilbibliografien ermöglicht, die dann einzelnen Artikeln zugeordnet werden können. Für diese Ausgabe von »Die T_EXnische Komödie« wurde aber noch eine Gesamtdatei für die Bibliografiedaten erstellt. Dabei

kann es natürlich zu doppelten Schlüsseln (bibkeys) kommen, wenn in verschiedenen Beiträgen gleiche Schlüssel vorhanden sind. Um dies zu verhindern, müssen die Schlüssel so ergänzt werden, dass sie eindeutig sind. Da dies ein ziemlicher Arbeitsaufwand ist, wird in Zukunft das optionale Argument `label` verwendet werden, welches es dem Befehl `\addbibresource` erlaubt, Bibliografien einzelnen Abschnitten zuzuordnen. Ob dies letztlich die Lösung ist, muss sich erst noch zeigen; das Erstellen einer Ausgabe von »Die T_EXnische Komödie« ist aber ohnehin *immer* ein »Abenteuer«; Pakete, die in einzelnen Dokumenten selten oder nie gemeinsam genutzt werden, zeigen manchmal unerwartete Ergebnisse. Insofern wird sich nicht allzu viel ändern, wenn man einmal davon absieht, dass Rolf Niepraschk dankenswerterweise die Überarbeitung der Dokumentenklasse `dtk` übernommen hat und diese bei der nächsten Ausgabe zur Anwendung kommen soll und wir dann grundsätzlich `biblatex` verwenden werden. Aber bis dahin geht es erst einmal auf Reisen; die Herbsttagung in Graz steht bevor.

Ich wünsche Ihnen wie immer viel Spaß beim Lesen und verbleibe
mit T_EXnischen Grüßen,

Ihr Herbert Voß

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

der Chefredakteur mahnt berechtigterweise zur Eile, also los.

Eigentlich wollte ich bereits während der TUG-Tagung in Darmstadt mein Grußwort fertigstellen, doch hat dies nicht ganz geklappt. Nicht zuletzt war daran auch das interessante und dicht gedrängte Tagungsprogramm schuld, das keinen Vorwand für das »Vortragsschwänzen« bot.

Als hätten die Organisatoren bzw. die Referenten mein Grußwort aus dem letzten Heft aufmerksam gelesen, ging es am ersten Tag zunächst ausführlich um barrierefreie PDF-Dokumente sowie die verschiedenen PDF-Standards, die es mittlerweile gibt, und wie man sie mit $\text{\TeX}$ erzeugen kann. Wieder einmal zeigte sich, wie wichtig das Thema ist, doch hatte ich diesmal den Eindruck, dass sich die zugehörigen Entwicklungen innerhalb der $\text{\TeX}$ -Welt beschleunigen.

Anschließend gab es neben einigen eher theoretischen Vorträgen, z. B. zu Fragen des Unicode-Standards, auch einige spannende Einblicke in praktische Beispiele zum Einsatz von $\text{\TeX}$.

Einen Punkt möchte ich noch aufgreifen. Auch die TUG diskutierte bei ihrer Mitgliederversammlung das Thema »Brauchen wir in Zukunft noch Nutzergruppen wie die TUG«. Leider konnte aufgrund der äußerst knapp bemessenen Zeit keine tiefere Diskussion entstehen, doch war es interessant, zu sehen, welche ganz unterschiedliche Sichtweisen es gibt. Exemplarisch möchte ich die folgenden Fragen aufwerfen bzw. wiedergeben:

- Brauchen wir mehr $\text{\TeX}$ -Nutzer, damit ein paar davon auch Mitglied in den Anwendergruppen werden und so deren Überleben sichern oder brauchen die Nutzer in Wahrheit die Nutzergruppen, da ohne diese die gesamte Infrastruktur zusammenbrechen würde. Stimmt nicht eigentlich beides?
- Wen sehen wir überhaupt als typischen Nutzer, den wir für $\text{\TeX}$ gewinnen wollen? Die Studenten, speziell Erstsemester, an den Hochschulen oder doch schon die Schüler, vielleicht sogar die Grundschüler?

- Brauchen wir nur mehr Programmierer, um T_EX voranzubringen oder geht es um den effizienten Einsatz der verfügbaren Manpower?

Ich denke, die TUG 2015 hat nicht nur diesbezüglich einerseits gezeigt, wie heterogen die Protagonisten der T_EX-Welt sind, und andererseits auch die Dynamik und Vielseitigkeit demonstriert, aus der man als Anwender schöpfen kann. Der persönliche Austausch schafft neue Motivation und Sichtweisen, ermöglicht die Bildung von Interessengruppen, schmiedet neue Allianzen und sorgt letztlich dafür, dass viele der Teilnehmer auch nach Jahrzehnten immer wieder mit Begeisterung und Engagement dabei sind.

Ich freue mich, dass viele Mitglieder von DANTE e.V. die Möglichkeit wahrgenommen haben, die Tagung in Darmstadt zu besuchen und möchte mich insbesondere bei Klaus Höppner sowie Christine Detig und Joachim Schrod bedanken, die als »Locals« im Organisationsteam maßgeblich zu einer gelungenen Konferenz beigetragen haben.

Mit dieser Ausgabe der DTK erhalten Sie die aktuelle T_EX Collection 2015. Die Erstellung und Verteilung an die anderen europäischen Nutzergruppen hat mit der Erfahrung der Vorjahre routiniert funktioniert. Durch den Wechsel zu einem anderen Anbieter konnten wir nochmals einige Hundert Euro bei der Herstellung einsparen. Durch die Preisstaffelung haben wir zudem viele Exemplare zur freien Verfügung, die wir zur Werbung bei Veranstaltungen etc. einsetzen wollen. Mein Dank geht an alle Beteiligten, die zu diesem Erfolg beigetragen haben.

Bereits in wenigen Wochen findet die Herbsttagung in Graz statt, höchste Zeit also, sich anzumelden. Ein entsprechendes Formular finden Sie auf den Tagungsseiten unter <http://www.dante.de/events/Herbst2015.html>. Dort sind auch Informationen zur Anreise und zu den Übernachtungsmöglichkeiten aufgeführt. Ich hoffe sehr, viele von Ihnen in der schönen Stadt Graz zu treffen.

Zum Schluss wünsche ich Ihnen wie immer viel Spaß bei der weiteren Lektüre unserer DTK.

Herzlichst Ihr/Euer
Martin Sievers

KOMA-Script

Experimentelles KOMA-Script-Repositorym

Elke Schubert

Mit Hilfe des experimentellen Repositoriums können die neuen Möglichkeiten von KOMA-Script schon getestet, beziehungsweise genutzt werden, bevor sie mit der nächsten Hauptversion auf CTAN verfügbar sind.

KOMA-Script entwickelt sich sehr rasch – ständig kommen neue Möglichkeiten wie zusätzliche Optionen, Befehle oder mitunter sogar ein Paket hinzu. Außerdem muss natürlich immer mal wieder ein Fehler korrigiert werden. Damit wir Anwender schnell von den Neuerungen oder Korrekturen profitieren können und nicht warten müssen, bis diese gesammelt als neue Hauptversion auf CTAN veröffentlicht werden, sind seit Anfang 2014 auf <http://www.komascript.de> neben den Haupt- auch Zwischenversionen verfügbar. Damit findet man dort immer die neueste von Markus Kohm als stabil¹ eingestufte Version.

Wie diese neueste Version installiert wird, hängt von der verwendeten $\text{\TeX}$ -Distribution ab und ist ausführlich unter <http://www.komascript.de/current> erklärt. Eine Rückkehr zur Hauptversion ist jederzeit möglich.

$\text{\TeX}$ Live

Die Idee und die Bezeichnung als »Experimentelles Repositorym« haben ihren Ursprung in der seit $\text{\TeX}$ Live 2013 bestehenden Möglichkeit, mehrere Paketquellen gleichzeitig einbinden zu können. Das »experimentell« bezieht sich dabei nur auf die Art der Bereitstellung, die bei mir aber von Beginn an wunderbar funktioniert hat.

Sobald die zusätzliche Quelle entsprechend der Anleitung eingebunden ist, erhält man zukünftig beim Updaten automatisch die Zwischen- und Hauptversionen aus diesem Repositorym. Auf der Webseite findet sich jedoch auch eine Beschreibung,

¹ Die Veröffentlichungstauglichkeit gilt nicht für als »ALPHA« oder »BETA« gekennzeichnete Pakete oder Möglichkeiten sowie Pakete mit Versionsnummern kleiner 1.

wie man bei Bedarf die zusätzliche Paketquelle entfernen kann, um zur in $\text{\TeX}$ Live enthaltenen Hauptversion zurückzukehren.

MiK $\text{\TeX}$

Bei MiK $\text{\TeX}$ sind keine zusätzlichen Paketquellen vorgesehen, dafür ist es einfach, mehrere Verzeichnisse als $\text{\TeX}$ MF root einzubinden. Deshalb stellt Markus Kohm die aktuelle Version für diese Distribution als Installationspaket zur Verfügung, das sich selbst entpackt, das entsprechende Verzeichnis als $\text{\TeX}$ MF root registriert und die Filename Database aktualisiert. Auch das funktionierte bei mir sehr zuverlässig, ist aber aus meiner Sicht nur Nutzern zu empfehlen, die zum einen das root-Konzept von MiK $\text{\TeX}$ verstanden haben und vor allem bereit sind, selbst regelmäßig auf der Webseite nach neuen Versionen zu schauen. Andererseits lohnt sich der Aufwand schon deshalb, weil zwischen dem Erscheinen einer neuen Hauptversion auf CTAN und deren Verbreitung über MiK $\text{\TeX}$ immer etwas Zeit vergeht.

Dafür ist der Wechsel zwischen der in MiK $\text{\TeX}$ enthaltenen Version und der zusätzlich installierten einfach über den Eintrag des Verzeichnisses in der » $\text{\TeX}$ MF root«-Liste möglich. Entfernt man das Verzeichnis aus der Liste, wird die MiK $\text{\TeX}$ -Version genutzt. Trägt man es erneut in die Liste ein, wird die zusätzliche Version wieder genutzt.

Kontrolle über die Versionsnummer

Ob man gerade mit einer Zwischen- oder einer Hauptversion arbeitet, kann man an der Versionsnummer von KOMA-Script erkennen. Während die Nummer der CTAN-Version nur aus zwei durch einen Punkt getrennten Zahlen besteht, sind es bei einer Zwischenversion drei. Die Versionsnummer findet man zum einen in der log-Datei, man kann sie sich aber auch mit `\KOMAScriptVersion` ins Dokument ausgeben lassen.

Weitere Pakete

Auf dem gleichen Weg kann man auch die Alpha-Version des Paketes `titlepage` erhalten, das Vorlagen für Titelseiten enthält und bisher noch nicht auf CTAN veröffentlicht ist. Eventuell werden zukünftig weitere Pakete von Markus Kohm folgen.

Von scrpage2 zu scrlayer-scrpage

Elke Schubert

Über viele Jahre war scrpage2 das Kopf- und Fußzeilenpaket von KOMA-Script. Mit KOMA-Script 3.12 wurde es von scrlayer-scrpage abgelöst, welches wiederum auf dem ebenfalls neuen und sehr mächtigen Paket scrlayer basiert. Neue Dokumente sollten in jedem Fall mit scrlayer-scrpage erstellt werden. Ältere Dokumente kann man recht einfach auf die neue Variante umstellen.

Einfacher Wechsel des Paketes

Das neue Paket basiert zwar auf einem ganz anderen Ansatz als das alte scrpage – es verwendet Ebenen für die Kopf- und Fußzeilen – aber der Umstieg ist dank ähnlicher Syntax ganz leicht, zumindest wenn das alte Paket entsprechend seiner Dokumentation [3] verwendet wurde.

Als erster Schritt genügt in der Regel das Austauschen des zu ladenden Paketes.

```
\usepackage{scrlayer-scrpage}
```

Hat man das gemacht, sieht das Ergebnis nach dem nächsten Kompilieren dem bisherigen mindestens ziemlich ähnlich. Trotzdem und selbst wenn gar keine Unterschiede erkennbar sind, sollte man die aus Kompatibilitätsgründen noch immer funktionierenden alten Befehle gegen die neuen Einstellmöglichkeiten tauschen. In den meisten Fällen sind das Optionen mit Key-Value-Syntax, die teilweise auch eine erweiterte Funktionalität bieten.

Vordefinierte Seitenstile

Bei der Umstellung ist zu beachten, dass scrlayer-scrpage den Hauptseitenstil scrheadings bereits beim Laden des Paketes aktiviert. Dieser wird damit auch ohne \pagestyle{scrheadings} verwendet. Aber natürlich kann der Seitenstil danach mit den üblichen Befehlen gewechselt werden.

Anders als bisher ist der Seitenstil headings nur noch ein Alias für scrheadings. Damit ist es nicht mehr möglich, mit \pagestyle{headings} zum Seitenstil der Dokumentenklasse zurückzukehren. Gleichzeitig erleichtert das die Verwendung des neuen Paketes mit anderen Klassen.

Der Seitenstil ohne Kolummentitel heißt jetzt plain.scrheadings. Für diesen wurde, wegen der Kompatibilität zu scrpage2, neben plain auch scrplain als Alias definiert. Es gibt weiterhin zwei Voreinstellungen für scrheadings und den zugehörigen Seitenstil plain, deren Auswahl entweder per Option oder abhängig von der Klasse

automatisch erfolgt. Dabei löst `pagestyleset` mit zwei möglichen Werten die beiden alten Optionen `komastyle` und `standardstyle` ab:

- `pagestyleset=KOMA-Script` (Stil der KOMA-Script-Klassen)
- `pagestyleset=standard` (Stil der Standardklassen)

Eine detaillierte Beschreibung dieser beiden Voreinstellungen für die Seitenstile findet man in [1, Abschnitt 5.4] und [2].

Leeren von Kopf- und Fußzeile

Will man die Inhalte von Kopf- und Fußzeile selbst definieren, wird man die vordefinierten meistens vorher entfernen. Die dafür zuständigen alten Befehle von *scrpage2* sollten durch die folgenden drei neuen Befehle ausgetauscht werden:

- `\clearpaairofpagestyles` (Leeren beider Seitenstile)
- `\clearmainofpaairofpagestyles` (Leeren des Hauptseitenstils)
- `\clearplainofpaairofpagestyles` (Leeren des plain-Seitenstils)

Einfügen von Inhalten in Kopf- und Fußzeile

Die aus *scrpage2* bekannten Befehle zur Festlegung des Kopf- und Fußzeileninhaltes wie `\ihead` haben sich nicht geändert. Sie sind in [1, Abschnitt 5.4] und [2], aber auch [3, Abschnitt 1.1] aufgezählt und ausführlich beschrieben.

KOMA-Script 3.14 brachte noch eine angenehme Neuerung: Mit den neuen Sternversionen dieser Befehle kann man sich das Wiederholen gleichen Inhalts in den optionalen Argumenten sparen. Damit verkürzt sich zum Beispiel

```
\cfoot[\pagemark]{\pagemark}
```

zu

```
\cfoot*{\pagemark}
```

Damit *scrlayer-scrpage* die Kontrolle über den Markierungsmechanismus von der Klasse übernimmt, muss immer eine der nach wie vor aktuellen Optionen `automark` oder `manualmark` oder einer der Befehle `\automark` beziehungsweise `\manualmark` gesetzt werden. Sie bestimmen, ob und welche Gliederungsebenen automatisch Marken für Kopf oder Fuß setzen, die dann über `\rightmark`, `\leftmark` oder einfach `\headmark` im Kopf oder Fuß ausgegeben werden können. Im einseitigen Modus wirkt außerdem noch die Option `autooneside`, deren Voreinstellung sich allerdings zu `autooneside=false` geändert hat.

Mit der neu hinzugekommenen Sternversion `\automark*` kann man das automatische Setzen von Marken für die als Argumente angegebenen Gliederungsebenen veranlassen, ohne es für alle anderen Gliederungsebenen aufzuheben. So ist es mit

```
\automark[chapter]{chapter}
\automark*[section]{}

```

möglich, bis zum Beginn des ersten Abschnittes die Kapitelüberschrift nicht nur als linke sondern auch als rechte Marke zu setzen, während anschließend die Abschnittsüberschrift den Inhalt von `\rightmark` bestimmt.

Textauszeichnungen

Die Wahl von Schriftart, -größe, -ausrichtung etc. erfolgt wie bei KOMA-Script üblich mit den Befehlen `\setkomafont`, `\addtokomafont`, `\usekomafont`. Das klappt mit `scrlayer-scrpage` jetzt auch dann, wenn eine andere Klasse verwendet wird.

Zu den existierenden Elementen `pagefoot`, `pagehead`, `pageheadfoot` und `pagenumber` sind `footbotline`, `footsepline`, `headsepline` und `headtopline` hinzugekommen, die nur für Farbänderungen der Linien gedacht sind. Ob automatische Kolumnentitel beispielsweise immer komplett in Großbuchstaben ausgegeben werden, hängt von der Angabe für die neue Option `markcase` ab. Mögliche Werte sind `lower`, `upper`, `used` und `noupper`, deren Wirkungen in [1, Abschnitt 5.5] und [2] erklärt sind. Die drei noch funktionierenden Optionen `markuppercase`, `markusedcase` und `markuppercase` sind dagegen veraltet und erzeugen eine entsprechende Warnung.

Höhe von Kopf- und Fußzeile

Das Paket `scrlayer-scrpage` passt die Höhe von Kopf- und Fußzeile automatisch an, wenn sie auf einer Seite für deren Inhalt nicht ausreicht. Diese Änderung gilt dann für die betroffene und alle folgenden Seiten. Die dabei ausgegebene Warnung mit Einstellhinweisen sollte man unbedingt beachten und entsprechende Anpassungen vornehmen. Während `typearea` dafür die Optionen `headlines` und `headheight` sowie `footlines` und `footheight` bereitstellt, kennt `geometry` nur Einstellmöglichkeiten für die Kopfzeile. Werden gar keine Pakete zum Einstellen des Seitenlayouts verwendet, kann man `\headheight` und die vom Paket `scrlayer` definierte Länge `\footheight` auch direkt anpassen.

Breite von Kopf- und Fußzeile

Die beiden Befehle zum Festlegen der Breiten wurden von den Optionen

- `headwidth=<breite>:<offset>:<abweichender offset linke seiten>`
- `footwidth=<breite>:<offset>:<abweichender offset linke seiten>`

abgelöst, bei denen alle drei Angaben optional sind. Stehen rechts von einem Doppelpunkt keine Werte, kann er weggelassen werden. Die Angabe des neuen Wertes `<abweichender offset linke seiten>`, der in der Voreinstellung gleich `<offset>` ist, macht nur bei zweiseitigen Dokumenten Sinn – in einseitigen gibt es keine linken Seiten.

An den Voreinstellungen für die Werte `<breite>` und `<offset>` hat sich mit einer Ausnahme nichts geändert: Wird `textwidthmarginpar` oder eine Länge, die diesem symbolischen Wert entspricht, als `<breite>` verwendet, dann ist `<offset>` in der Voreinstellung Null.

Zu den schon in `scrpage2` vorhandenen symbolischen Werten für die Breitenangaben sind weitere wie `marginpar` hinzugekommen, die in [1, Abschnitt 5.5] und [2] aufgelistet und erklärt sind.

Bei den Längenangaben ist zu berücksichtigen, dass sie voll expandiert werden, wenn die Optionen im optionalen Argument von `\documentclass` oder `\usepackage` angegeben werden. Möchte man zum Beispiel schon beim Laden des Paketes eine Kopfzeile einstellen, die 10 % breiter als der Satzspiegel ist, muss man

```
\usepackage[headwidth=\the\textwidth*11/10]{scrlayer-scrpage}
```

verwenden. In den Befehlen `\KOMAOption` oder `\KOMAOptions` sind dagegen auch die gewohnten Längenangaben möglich. Das Beispiel vereinfacht sich damit zu

```
\usepackage{scrlayer-scrpage}
\KOMAOptions{headwidth=1.1\textwidth}
```

Linien in Kopf- und Fußzeile

Die Optionen für das Ein- und Ausschalten der vier möglichen Trennlinien übernehmen jetzt auch das Festlegen von deren Dicke und Länge. Da die Farbe der Linien unabhängig von der Klasse über `\setkomafont` oder `\addtokomafont` festlegbar ist, sind die alten Befehle wie `\setheadsepline` damit obsolet. Die neue Syntax ist

- `headtopline=<dicke>:<länge>`
- `headsepline=<dicke>:<länge>`
- `footsepline=<dicke>:<länge>`
- `footbotline=<dicke>:<länge>`

Sowohl die Angabe von `<dicke>` als auch `<länge>` ist optional, die Voreinstellungen dafür haben sich nicht geändert. Lässt man `<länge>` weg, kann auch der Doppelpunkt entfallen. Analog zu Breite von Kopf und Fuß versteht `<länge>` einerseits die gleichen symbolischen Werte, und es muss andererseits daran gedacht werden, dass

Längenangaben im optionalen Argument von Paket oder Klasse voll expandiert werden.

Alternativ zu obiger Syntax kennen die vier Optionen auch einfache Schalterwerte, wie `true` oder `false`, siehe [1, Tabelle 2.5] und [2]. Als Dicke und Länge werden dann die Standardwerte verwendet. Da bei *scrlayer-scrpage* die Optionen für die Übernahme der Linien auf `plain`-Seiten die Werte für einfache Schalter verstehen, können alle eingefügten Trennlinien auch wieder entfernt werden.

Die Voreinstellung für die Ausrichtung der Trennlinien hat sich ebenfalls geändert. Standardmäßig ist jetzt `ilines` aktiviert, womit die Linien bündig mit dem inneren Rand von Kopf und Fuß ausgegeben werden. Möchte man sie zentriert zu Kopf und Fuß haben, muss man jetzt die Option `clines` explizit setzen. Für Linien mit der gleichen Breite wie Kopf oder Fuß macht das allerdings keinen Unterschied.

Besonderheiten der Optionen `headsepline` und `footsepline`

Es wird immer empfohlen, diese beiden Optionen nicht erst beim Laden des Paketes, sondern schon global bei der Klasse anzugeben. Hintergrund ist, dass die KOMA-Script-Klassen gleichnamige Optionen kennen, die insbesondere auch Schalterwerte für die `typearea`-Optionen `headinclude` und `footinclude` setzen und damit den Satzspiegel beeinflussen. Näheres dazu findet man in [1, Abschnitte 2.6 und 3.12] und [2].

Wird den Optionen nun aber ein Wert für `<dicke>` oder `<länge>` zugewiesen, dann sind sie keine Klassenoptionen mehr, das heißt `headinclude` und `footinclude` werden nicht mehr automatisch angepasst, sondern müssen explizit gesetzt werden.

Ausblick

Nach dem Wechsel zu *scrlayer-scrpage* und dem Austauschen der alten Befehle und Optionen, kann man weitere neue Möglichkeiten des zugrunde liegenden Paketes *scrlayer* nutzen. So können einzelne Ebenen modifiziert, entfernt oder hinzugefügt werden, oder ihre Ausgabe kann auf Seiten mit bestimmten Eigenschaften eingeschränkt werden. Zum Beispiel wird oft ein etwas größerer Abstand zwischen Fußzeile und Fußzeilenlinie gewünscht. Das kann man leicht erreichen, in dem man die Ebenen mit dieser Linie etwas nach oben schiebt.

```
\ModifyLayer[addvoffset=-1mm]{scrheadings.foot.above.line}
\ModifyLayer[addvoffset=-1mm]{plain.scrheadings.foot.above.line}
```

Komplexere Anwendungsbeispiele zeigt Markus Kohm in seiner Artikelserie in dieser Ausgabe. Auch zum Erstellen neuer Seitenstile gibt es ganz neue Wege. Man kann sie direkt aus neuen Ebenen aufbauen, als neue Seitenstil-Paare erzeugen oder

über Befehle wie `\defpagestyle` definieren. Ausführliche Informationen dazu finden sich in [1, Kapitel 17 und 18] und [2].

Zusammenfassung

Da nicht nur der Austausch des seit KOMA-Script 3.12 obsoleten Paketes `scrpage2` durch das neue `scrlayer-scrpage` empfohlen wird, sondern auch das Ersetzen etlicher, nur noch aus Kompatibilitätsgründen gültiger, alter Befehle und Optionen, wurden deren Nachfolger sowie einige weitere Neuerungen kurz vorgestellt.

Literatur und Software

- [1] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für $\text{\LaTeX}$ 2_ε*. 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [2] Markus Kohm. *KOMA-Script ein wandelbares $\text{\LaTeX}$ 2_ε-Paket*. Mai 2015. CTAN:macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 24. 06. 2015).
- [3] Markus Kohm. *The Obsolete Package scrpage2*. Juni 2014. CTAN:macros/latex/contrib/koma-script/doc/scrpage2.pdf (besucht am 24. 06. 2015).

Firmenlogo mit `scrlayer`

Markus Kohm

In den letzten Jahren gab es bei KOMA-Script einige kleinere und größere Neuerungen. Viele davon sind bei der Mehrzahl der KOMA-Script-Anwender unbemerkt geblieben. Obwohl ich im Support regen Gebrauch davon mache, bleibt so das Potential dieser Erweiterungen eher verborgen und ungenutzt. Bei der Tagung von DANTE e.V. im Herbst 2014 habe ich daher einige konkrete Aufgaben aus dem Support-Alltag beispielhaft aufgegriffen. Dieser Beitrag zeigt, wie man mit Hilfe von `scrlayer`, das seit KOMA-Script 3.12 fester Bestandteil der Sammlung ist, abhängig von verschiedenen Kriterien ein Firmenlogo auf den Seiten eines Dokuments platzieren kann.

Aufgabenstellung

Die Aufgabe wurde bereits mehrfach an mich herangetragen. Herausgreifen will ich die Anfrage eines Dienstleisters, der im Rahmen eines Projekts Dokumente wie

Spezifikationen und Fortschrittsberichte an die Kunden weiterreicht. Vom Kunden werden die Dokumente teilweise auch Dritten zugänglich gemacht. Zwecks Außendarstellung soll nun am oberen äußeren Rand das Firmenlogo des Dienstleisters angezeigt werden.

Lösung mit scrlayer

Zwar gibt es auf CTAN bereits einige Pakete, mit denen man ein Logo auf einer Seite platzieren könnte, seit KOMA-Script 3.12 geht das aber auch sehr einfach mit KOMA-Script-Mitteln, konkret mit dem Paket `scrlayer`. Im vorliegenden Fall wurde bereits das Paket `scrlayer-scrpage` verwendet, um den Seitenstil anzupassen. Damit lag die Verwendung von `scrlayer` nahe. Das Paket wird von

```
12 \usepackage{scrlayer-scrpage}
```

intern ohnehin geladen.

Das Logo selbst lag als JPEG vor. Zwar ist das aufgrund der verlustbehafteten Kompression in meinen Augen nicht das ideale Format für ein Logo, aber letztlich spielt dies für die Problemlösung eine untergeordnete Rolle. Für mich bedeutete das lediglich, dass für die Einbindung des Logos das Paket `graphicx` zu verwenden ist:

```
10 \usepackage{graphicx}
```

Als Ersatz für das echte Logo verwende ich in diesem Beispiel eine der Grafiken, die vom Paket `mwe` (siehe [4]) bereitgestellt werden:

```
11 \usepackage{mwe}
```

Das Paket `scrlayer` stellt ein Ebenenmodell für $\LaTeX$ bereit. Neue Ebenen werden mit `\DeclareNewLayer` deklariert und können verschiedene Eigenschaften besitzen:

```
12 \DeclareNewLayer[%
13   background,
14   topmargin,
15   contents={\hfill
16     \includegraphics[width=3cm,height=3cm]{example-image}%
17   }%
18 ]{Logo}
```

In der Voreinstellung werden Ebenen zweimal verwendet, einmal im Hintergrund der Seite, also hinter der normalen Textebene, und einmal im Vordergrund, also vor der eigentlichen Textebene. Im Beispiel wird die Ebene Logo mit der Eigenschaft `background` auf den Hintergrund beschränkt. Entsprechend könnte man die Ausgabe mit der Eigenschaft `foreground` auf den Vordergrund beschränken. Die Verwendung einer dieser beiden Eigenschaften ist meist sinnvoll, da es in den wenigstens Fällen einen Nutzen bringt, die Ebene zweimal auszugeben.

Die Eigenschaft `topmargin` ist eine so genannte zusammengesetzte Eigenschaft. Sie wird letztlich auf die Eigenschaften `align`, `voffset`, `height`, `hoffset` und `width` abgebildet, um Position und Ausdehnung der Ebene festzulegen. Hier handelt es sich um den Bereich des oberen Randes der Seite in kompletter Seitenbreite, siehe dazu auch [2, Tabelle 17.1] oder [1, Tabelle 17.1].

Über die Eigenschaft `contents` wird der Inhalt der Ebene festgelegt. Hier muss lediglich das Logo geladen und zuvor mit `\hfill` an den rechten Rand verschoben werden. Damit ist die Ebene fertig definiert. Um eine Ebene zu aktivieren, muss man sie bei `scrlayer` an einen Seitenstil binden:

```
19 \AddLayersToPageStyle{@everystyle@}{Logo}
```

Da das Logo laut Aufgabenstellung auf jeder Seite stehen soll, binden wir die Ebene an den Pseudo-Seitenstil `@everystyle@`.

Die Ebenen dieses Seitenstils werden grundsätzlich auf alle Seitenstile, die mit `scrlayer` definiert wurden, angewendet. Bereits beim Laden des Pakets wird der Seitenstil `empty` zu einem so genannten *Ebenenseitenstil* umdefiniert. Die Seitenstile `headings` und `plain` werden hingegen beim Laden von `scrlayer-scrpage` entsprechend ersetzt. Für das Beispiel selbst verwenden wir die Klasse `scrartcl`. Damit sind alle Seitenstile, die von der Klasse verwendet oder bereitgestellt werden, nach dem Laden von `scrlayer-scrpage` Ebenenseitenstile und verwenden die Ebenen von `@everystyle@`. Also tragen nun alle Seiten das Logo.

Mit dem vollständigen Beispielcode aus Listing 1 erhalten wir so unter anderem die drei Seiten aus Abbildung 1.

Listing 1: Beispieldokument in einseitigem Layout

```
9 \documentclass{scrartcl}
10 \usepackage{graphicx}
11 \usepackage{mwe}
12 \usepackage{scrlayer-scrpage}
13 \DeclareNewLayer[{}
14   background,
15   topmargin,
16   contents={\hfill
17     \includegraphics[width=3cm,height=3cm]{example-image}%
18   }%
19 ]{Logo}
20 \AddLayersToPageStyle{@everystyle@}{Logo}
21 \usepackage[ngerman]{babel}
22 \usepackage[utf8]{inputenc}
23 \usepackage{libertine}
24 \usepackage{blindtext}
25 \begin{document}
26 \title{Beispiel für ein Logo auf jeder Seite}
27 \author{Markus Kohm}
```

```

28 \date{Juli 2014}
29 \maketitle
30 \tableofcontents
31 \blinndocument
32 \end{document}

```

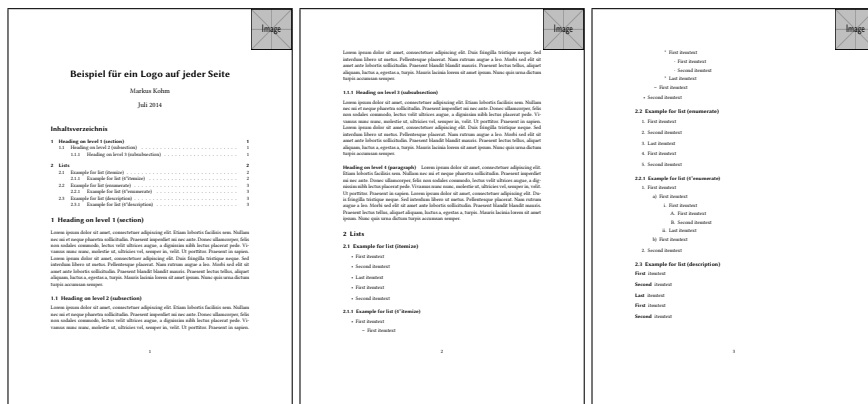


Abbildung 1: die ersten drei Seiten des Beispieldokuments in einseitigem Layout aus Listing 1

Abwandlung

Liest man die Aufgabenstellung noch einmal genauer, so fällt auf, dass dort nicht vom *rechten Rand*, sondern vom *äußeren Rand* die Rede ist. Die bisherige Lösung würde bei einem doppelseitigen Dokument aber sowohl auf linken als auch auf rechten Seiten das Logo an den rechten Rand schieben. Für ein doppelseitiges Dokument muss sie daher abgewandelt werden. Dafür bieten sich zwei unterschiedliche Ansätze an:

1. Für gerade und ungerade Seiten unterschiedliche Ebenen definieren.
2. Die Ebene so abwandeln, dass auf ungeraden Seiten weiterhin das Logo nach rechts verschoben wird, auf geraden Seiten die Verschiebung aber unterbleibt.

Der erste Ansatz ist denkbar einfach. Statt der einen Ebene Logo definieren wir nun zwei Ebenen Logo.odd und Logo.even:

```

13 \DeclareNewLayer[%
14 background,
15 oddpage,
16 topmargin,

```

```

17 contents={\hfill
18   \includegraphics[width=3cm,height=3cm]{example-image}%
19 }%
20 }]{Logo.odd}
21 \DeclareNewLayer[%
22   background,
23   evenpage,
24   topmargin,
25   contents={%
26     \includegraphics[width=3cm,height=3cm]{example-image}%
27   }%
28 }]{Logo.even}
29 \AddLayersToPageStyle{@everystyle@}{Logo.odd,Logo.even}

```

Dabei ist Logo.odd mit Logo nahezu identisch. Es wurde lediglich die Eigenschaft `oddpages` hinzugefügt. Dadurch wird die Verwendung der Ebene im doppelseitigen Layout auf rechte Seiten eingeschränkt. Im einseitigen Layout hat die Eigenschaft keine Auswirkung. Dagegen schränkt die Eigenschaft `evenside` von Ebene Logo.even die Verwendung der Ebene im doppelseitigen Layout auf linke Seiten ein und verhindert ihre Verwendung im einseitigen Layout komplett. Abbildung 2 zeigt eine Doppelseite des Ergebnisses.

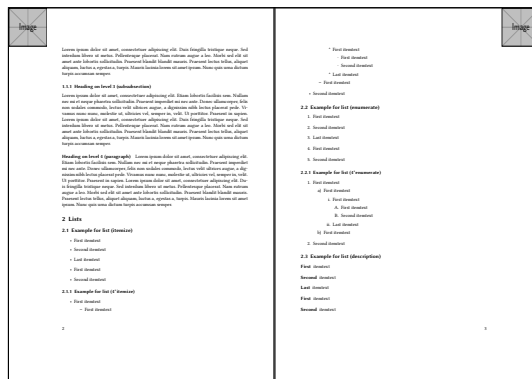


Abbildung 2: Doppelseite eines Beispieldokuments in zweiseitigem Layout

Just als ich dem Dienstleister diese Lösung vorstellte, kam er jedoch auf die Idee, dass auf linken Seiten eigentlich kein Logo erforderlich sei, da man diese ja nur bei beidseitigem Ausdruck habe und sie dann natürlich immer eine mit Logo bedruckte Vorderseite besitze. Statt die einmal definierte Ebene wieder zu entfernen, lautet die Lösung dafür einfach:

```

29 \AddLayersToPageStyle{@everystyle@}{Logo.odd}

```

Es wird also schlicht darauf verzichtet, die Ebene für linke Seiten dem Pseudo-Seitenstil `@everystyle@` hinzuzufügen. Sollte es sich der Auftraggeber doch noch einmal anders überlegen, ist der Weg zurück so denkbar einfach.

Das ist auch der Grund, warum ich mich letztlich gegen die Alternativimplementierung mit nur einer Ebene:

```
\DeclareNewLayer[{%
  background,
  topmargin,
  contents={\ifodd\value{page}\hfill\fi
  \includegraphics[width=3cm,height=3cm]{example-image}%
}%
]{Logo}
\AddLayersToPageStyle{@everystyle@}{Logo}
```

entschieden habe. Funktionieren würde der Vergleich mit `\ifodd` in diesem Fall, da zum Zeitpunkt der Anwendung eines Seitenstils und damit einer daran gebundenen Ebene der Zähler `page` immer den korrekten Wert der aktuell auszugebenden Seite besitzt. Es wäre hier also insbesondere nicht notwendig die KOMA-Script-Anweisung `\ifthispageodd` zu verwenden, um eine jederzeit gültige Entscheidung darüber zu fällen, ob die aktuelle Seite gerade oder ungerade ist.

Zusammenfassung

Es wurde dargestellt, wie man mit `scrlayer` neue Ebenen definieren und diese Seitenstilen hinzufügen kann. Darüber hinaus wurde gezeigt, wie man die Verwendung von Ebenen über Eigenschaften auf bestimmte Seiten einschränken kann. Es sei darauf hingewiesen, dass es neben den gezeigten Eigenschaften `oddpage` und `evenpage` weitere derartige Einschränkungsmöglichkeiten gibt, etwa für Gleitumgebungsseiten oder Seiten im einseitigen Satz. Näheres dazu ist [2] oder [1] zu entnehmen.

Literatur und Software

- [1] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für L^AT_EX 2_ε*. 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [2] Markus Kohm. *KOMA-Script ein wandelbares L^AT_EX 2_ε-Paket*. Apr. 2015. CTAN:macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 01.05.2015).
- [3] Markus Kohm. *Tipps und Tricks mit KOMA-Script*. Präsentation. Sep. 2014. URL: <http://www.komascript.de/dante1409> (besucht am 22.04.2015).
- [4] Martin Scharrer. *The mwe Package*. Mai 2012. CTAN:macros/latex/contrib/mwe/ (besucht am 01.09.2014).

Dokumentversion mit `scrlayer`

Markus Kohm

Dieser Beitrag zeigt, wie man mit Hilfe von `scrlayer`, das seit KOMA-Script 3.12 fester Bestandteil der Sammlung ist, Versionsinformationen zu einem Dokument platzieren kann.

Aufgabenstellung

Die Aufgabe wurde bereits mehrfach an mich herangetragen. Herausgreifen will ich die Anfrage eines Dienstleisters, der im Rahmen eines Projekts Dokumente wie Spezifikationen und Fortschrittsberichte an die Kunden weiterreicht. Naturgemäß kursieren von diesen Dokumenten häufig unterschiedliche Versionen sowohl beim Dienstleister selbst als auch beim Kunden. Um jeweils zu wissen, worüber man gerade redet, soll daher jede Seite des Dokuments eine komplette Versionsinformation tragen.

Lösung mit `scrlayer`

Wie bereits im vorherigen Artikel wird für die Realisierung das Paket `scrlayer` verwendet, das von `scrlayer-scrpage` geladen wird, und damit eine neue Ebene definiert:

```

9 \DeclareNewLayer[{%
10   background,
11   bottommargin,
12   contents={%
13     \parbox[t][\layerheight][c]{\layerwidth}{%
14       \hfill \jobname.pdf \textbullet\
15       \pdffilemoddate{\jobname.tex}\hspace*{2em}%
16     }%
17   }%
18 }]{Version}

```

Die Ebene `Version` wird mit der Eigenschaft `background` auf den Hintergrund beschränkt. Die Eigenschaft `bottommargin` legt die Position und Ausdehnung der Ebene fest. Hier handelt es sich um den Bereich des unteren Randes der Seite unterhalb der Fußzeile in kompletter Seitenbreite, siehe dazu auch [3, Tabelle 17.1] oder [2, Tabelle 17.1].

Über die Eigenschaft `contents` wird der Inhalt der Ebene festgelegt. In diesem Fall wird eine `\parbox` der Höhe und Breite der Ebene platziert, deren Inhalt über das dritte optionale Argument von `\parbox` vertikal zentriert wird. Die beiden Makros

`\layerwidth` und `\layerheight` sind nur während der Verarbeitung einer Ebene gültig und geben die definierte Ausdehnung der Ebene wieder. Dies kann wie im gezeigten Fall für den Inhalt einer Ebene recht nützlich sein. Innerhalb der Box wird an den rechten Rand verschoben der Dateiname der Hauptdatei des Dokuments, `\jobname`, versehen mit der Endung `.pdf`, gefolgt von einem dicken Punkt und dem Änderungsdatum der Hauptdatei des Dokuments, `\pdffilemmoddate{\jobname.tex}`, ausgegeben. Damit das ganze nicht komplett am rechten Rand klebt, wird abschließend mit `\hspace*{2em}` noch ein kleiner Abstand zu diesem eingefügt. Damit ist die Ebene fertig definiert. Um eine Ebene zu aktivieren, muss man sie bei `scrlayer` an einen Seitenstil binden:

```
19 \AddLayersToPageStyle{@everystyle@}{Version}
```

Da die Versionsinformation laut Aufgabenstellung auf jeder Seite stehen sollte, binden wir sie an den Pseudo-Seitenstil `@everystyle@`. Die Ebenen dieses Seitenstils werden grundsätzlich auf alle Seitenstile, die mit `scrlayer` definiert wurden, angewendet. Mit dem vollständigen Beispielcode aus Listing 1 erhalten wir so unter anderem die drei Seiten aus Abbildung 1.

Listing 1: Beispieldokument in einseitigem Layout

```
9 \documentclass{scrartcl}
10 \usepackage{scrlayer-scrpage}
11 \DeclareNewLayer[%
12   background,
13   bottommargin,
14   contents={%
15     \parbox[t][\layerheight][c]{\layerwidth}{%
16       \hfill \jobname.pdf \textbullet\
17       \pdffilemmoddate{\jobname.tex}\hspace*{2em}%
18     }%
19   }%
20 ]{Version}
21 \AddLayersToPageStyle{@everystyle@}{Version}
22 \usepackage[ngerman]{babel}
23 \usepackage{utf8}{inputenc}
24 \usepackage{libertine}
25 \usepackage{blindtext}
26 \begin{document}
27 \title{Beispiel für ein Dokumentversion auf jeder Seite}
28 \author{Markus Kohm}
29 \date{Juli 2014}
30 \maketitle
31 \tableofcontents
32 \blinddocument
33 \end{document}
```

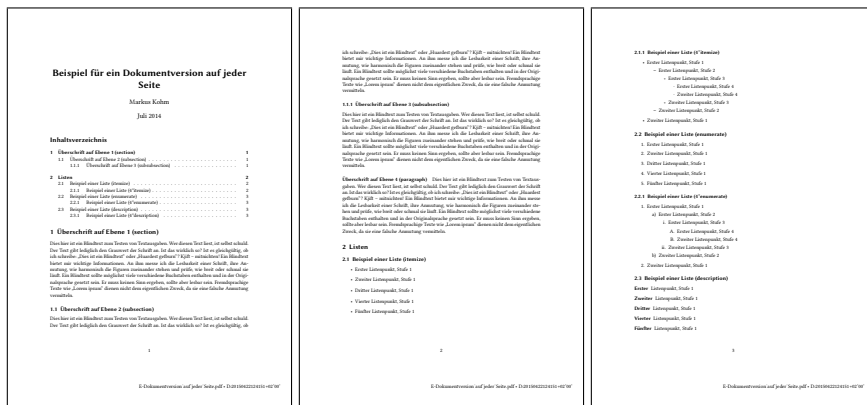


Abbildung 1: die ersten drei Seiten des Beispieldokuments in einseitigem Layout aus Listing 1

Abwandlung

Wie so oft wollte auch in diesem Fall der Auftraggeber umgehend einige Änderungen. So sollte die Versionsinformation bei doppelseitigen Dokumenten auf linken Seiten nicht rechts, sondern links platziert werden. Auch in diesem Fall bestand die Lösung dafür darin, zwei Ebenen zu definieren:

```
\DeclareNewLayer[%
  oddpage,
  background,
  bottommargin,
  contents={%
    \parbox[t][\layerheight][c]{\layerwidth}{%
      \hfill
      \jobname.pdf \textbullet\
      \pdffilemoddate{\jobname.tex}%
      \hspace*{2em}%
    }%
  }%
]{Version.odd}
\DeclareNewLayer[%
  evenpage,
  background,
  bottommargin,
  contents={%
    \parbox[t][\layerheight][c]{\layerwidth}{%
      \hspace*{2em}%
      \jobname.pdf \textbullet\
      \pdffilemoddate{\jobname.tex}%
    }%
  }%
```

```
}%
}}{Version.even}
```

Gleichzeitig sollte die Versionsinformation auf Vakantseiten, also den automatisch eingefügten leeren, linken Seiten vor Kapitelanfängen, nicht angezeigt werden. Diese Seiten werden bei KOMA-Script normalerweise im Seitenstil `empty` gesetzt. Daher wurden die beiden Ebenen nun nicht mehr dem Pseudo-Seitenstil `@everystyle`, sondern explizit nur noch den beiden Seitenstilen `scrheadings` und `plain.scrheadings` hinzugefügt:

```
1 \AddLayersToPageStyle{scrheadings}{Version.odd,Version.even}
2 \AddLayersToPageStyle{plain.scrheadings}{Version.odd,Version.even}
```

Allerdings war die Versionsinformation dadurch dann auch nicht mehr auf der Titelseite. Das wurde einfach gelöst, indem die Ausgabe der Titelseite wie folgt geändert wurde:

```
3 \AddLayersToPageStyle{empty}{Version.odd,Version.even}
4 \maketitle
5 \RemoveLayersFromPageStyle{empty}{Version.odd,Version.even}
```

Die letzte der drei Anweisungen entfernt die Ebenen nach Ausgabe der Titelseiten wieder aus der Verarbeitung des Seitenstils `empty`.

Oft werden Dokumente, die von mehreren Personen gepflegt werden, über ein Versionsmanagementsystem verwaltet. Auf CTAN existieren diverse Pakete, um die Metainformationen von Dateien in ein Dokument zu integrieren, beispielsweise `gitinfo` (siehe [4]) oder `svn-multi` (siehe [5]). Stephan Hennig hat mit `vc` auch ein System geschaffen, das mit unterschiedlichen Versionsmanagementsystemen arbeiten kann (siehe [1]). Die von diesen Paketen bereitgestellte Versionsinformation kann natürlich einfach an Stelle von `\pdffilemoddate` verwendet werden.

Zusammenfassung

Es wurde gezeigt, wie man mit `scrlayer` neue Ebenen definieren und diese Seitenstilen hinzufügen kann. Darüber hinaus wurde erläutert, wie man die Verwendung von Ebenen über Eigenschaften auf bestimmte Seiten einschränken kann. Es sei darauf hingewiesen, dass es neben den beschriebenen Eigenschaften `oddpaper` und `evenpage` weitere derartige Einschränkungsmöglichkeiten gibt, etwa für Gleitumgebungsseiten oder Seiten im einseitigen Satz. Näheres dazu ist [3] oder [2] zu entnehmen.

Nebenbei wurde außerdem gezeigt, dass `pdfTeX` bereits eine Anweisung bietet, um das Änderungsdatum einer Datei als Versionsinformation nutzen zu können. Möglichkeiten zur Verwendung der Metainformationen eines Versionsmanagementsystems wurden als Alternative genannt.

Literatur und Software

- [1] Stephan Hennig. *The vc bundle*. Apr. 2008. CTAN: support/vc (besucht am 01. 09. 2014).
- [2] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für $\LaTeX$ 2_ε*. 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [3] Markus Kohm. *KOMA-Script ein wandelbares $\LaTeX$ 2_ε-Paket*. Apr. 2015. CTAN: macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 01. 05. 2015).
- [4] Brent Longborough. *gitinfo.sty. A package for accessing metadata from the git DVCS*. Aug. 2011. CTAN: macros/latex/contrib/gitinfo/ (besucht am 01. 09. 2014).
- [5] Martin Scharrer. *The svn-multi Package*. Aug. 2011. CTAN: macros/latex/contrib/svn-multi/ (besucht am 01. 09. 2014).

Farbige, kleine Kapitelmarken am Rand mit `scrlayer`

Markus Kohm

Dieser Beitrag zeigt, wie man mit Hilfe von `scrlayer`, das seit KOMA-Script 3.12 fester Bestandteil der Sammlung ist, abhängig von verschiedenen Kriterien farbige, kleine Kapitelmarken am Rand der Seiten anbringen kann.

Aufgabenstellung

Die Aufgabe wurde erstmals vor ca. zehn Jahren an mich herangetragen. Damals habe ich dafür noch eine Lösung mit `scrpage2` und eine alternative Lösung mit `everyshi` angeboten. Aufgreifen will ich hier jedoch eine Anfrage von Anfang 2014. Dabei wollte ein Doktorand seinem Professor das Auffinden der Kapitel ein wenig erleichtern, indem er eine Art Register an den äußeren Rand der Seiten drucken wollte. Zunächst habe ich den Hilfesuchenden auf die Lösung aus [3, Anhang B] verwiesen. Allerdings stellte sich schnell heraus, dass er Marken ohne Text, dafür aber in unterschiedlichen Farben – je Kapitel eine andere – haben wollte.

Lösung mit scr_layer

Wie bereits erwähnt, gibt es bereits verschiedene Lösungsansätze für dieses Problem. Da ich aber ohnehin gerade in scr_layer vertieft war, bastelte ich in wenigen Minuten auch noch eine Lösung damit. In diesem Fall musste der Auftraggeber erst einmal überredet werden, von scr_{page}2 zu scr_layer-scr_{page} zu wechseln und

```
14 \usepackage{scrlayer-scrpage}
```

zu verwenden. Für die Farben verwendete ich das Paket xcolor:

```
12 \usepackage[svgnames]{xcolor}
```

weil es mit colorlet eine sehr schöne Möglichkeit bietet, vorhandene Farben oder Kombinationen daraus für neue Farbnamen zu verwenden (siehe [2]):

```
62 \colorlet{Kapitelmarke1}{DarkBlue}
63 \colorlet{Kapitelmarke2}{DarkRed}
64 \colorlet{Kapitelmarke3}{DarkGreen}
```

So konnten für die unterschiedlichen Farben der Marken einfach nummerierte Namen definiert werden.

Nun hätte man für die Nummern der Kapitelmarken einfach den Kapitelzähler verwenden können. Dies hätte sogar im Anhang funktioniert, wenn man in den Farbnamen \thechapter verwendet hätte. Allerdings sollten auch nicht nummerierte Kapitel wie die *Versicherung an Eides Statt* oder die Verzeichnisse eine Marke erhalten. Also erschien es sinnvoll, einen eigenen Zähler dafür zu definieren:

```
15 \newcounter{Kapitelmarke}
```

Daraus ergab sich aber das Problem, dass dieser Zähler natürlich auch zu erhöhen war. Statt dies manuell vorzunehmen, wurde die Erhöhung mit Hilfe von xpatch (siehe [1]) an die beiden Anweisungen \@makechapterhead und \@makeschapterhead gebunden und so automatisiert:

```
13 \usepackage{xpatch}
```

```
65 \makeatletter
66 \xapptocmd{\@makechapterhead}{\stepcounter{Kapitelmarke}}
67 {}{\undefined}
68 \xapptocmd{\@makeschapterhead}{\stepcounter{Kapitelmarke}}
69 {}{\undefined}
70 \makeatother
```

Außer der Nummer haben die Kapitelmarken weitere Eigenschaften, wie eine Höhe und Breite. Für die Berechnung der Position müssen wir dann außer der Nummer der Marke auch noch wissen, wo die oberste Marke platziert werden soll und wie viele Marken maximal untereinander angeordnet werden dürfen, bevor die nächste Marke wieder neu oben angesetzt wird:

```

16 \newlength\Kapitelmarkenhoehe
17 \setlength{\Kapitelmarkenhoehe}{1cm}
18 \newlength{\Kapitelmarkenbreite}
19 \setlength{\Kapitelmarkenbreite}{1cm}
20 \newlength\Kapitelmarkenoffset
21 \setlength{\Kapitelmarkenoffset}{\dimexpr 1cm+\topmargin\relax}
22 \newcommand*\MaxKapitelmarken{20}

```

Da eine optische Überlappung der Marken ausdrücklich nicht gewünscht war, entspricht der vertikale Versatz von Kapitel zu Kapitel ebenfalls der Höhe der Marken. Mit einer solchen Überlappung hätte man für den Versatz eine weitere Länge definiert. Die oberste Position, `\Kapitelmarkenoffset`, war recht willkürlich gewählt. Tatsächlich hatte ich in einem ersten Ansatz eine etwas aufwändigere Berechnung vorgeschlagen, was ich aber letztlich nicht zu bestimmen hatte:

```

\setlength{\Kapitelmarkenoffset}{\dimexpr \topmargin+1in
+ \headheight+\headsep\relax}

```

Wenn man Material mit gefärbtem Hintergrund randbündig drucken will – die Druckerei spricht dann von *randabfallend* –, so muss man in den Druckvorlagen grundsätzlich über den physischen Rand des fertig gebundenen und geschnittenen Werkes hinaus gehen. Damit sollen Toleranzen bei der Herstellung – etwa durch den verbreiteten, leichten Kugel- oder Hohlchnitt des Papierstapels – weniger stark ins Gewicht fallen. Üblich sind 3 mm. Für den Fall einer geforderten Abweichung wurde auch hierfür eine Länge definiert:

```

3 \newlength{\Anschnitt}
4 \setlength{\Anschnitt}{3mm}

```

Damit sind alle Einstellmöglichkeiten definiert und initialisiert.

Kommen wir zu den Marken selbst. Das Paket `scrlayer` stellt ein Ebenenmodell für $\LaTeX$ bereit. Neue Ebenen werden mit `\DeclareNewLayer` deklariert und können verschiedene Eigenschaften besitzen:

```

5 \newif\ifKapitelmarken
6 \DeclareNewLayer[{\%
7   background, oddpage, rightmargin,
8   contents={\%
9     \ifKapitelmarken
10      \parbox[t][\layerheight][t]{\layerwidth}{\%
11        \raggedleft
12        \vspace*{\dimexpr \Kapitelmarkenoffset
13          +\Kapitelmarkenhoehe*\XmodY{\value{\Kapitelmarke}}
14          {\MaxKapitelmarken}}}%
15      \makebox[\Kapitelmarkenbreite][l]{\%
16        \color{\Kapitelmarke}\theKapitelmarke}%
17      \rule{\dimexpr \Kapitelmarkenbreite+\Anschnitt}{\%
18        {\Kapitelmarkenhoehe}}%

```

```

19      }%
20    }%
21  \fi
22    }%
23  }}{Kapitelmarken}

```

In der Voreinstellung werden Ebenen zweimal verwendet, einmal im Hintergrund der Seite, also hinter der normalen Textebene, und einmal im Vordergrund, also vor der eigentlichen Textebene. Im Beispiel wird die Ebene Kapitelmarken mit der Eigenschaft `background` auf den Hintergrund beschränkt. Darüber hinaus wird mit `oddpaper` dafür gesorgt, dass sie nur auf ungeraden Seiten gedruckt werden. Die Erfahrung zeigt, dass das ausreichend ist und Probleme mit minimalem vertikalem Versatz beim Druck etwas reduziert.

Die Eigenschaft `rightmargin` ist eine so genannte zusammengesetzte Eigenschaft. Sie wird letztlich auf die Eigenschaften `align`, `offset`, `height`, `offset` und `width` abgebildet, um Position und Ausdehnung der Ebene festzulegen. Hier handelt es sich um den Bereich des rechten Randes der Seite in kompletter Seitenhöhe, siehe dazu auch [4, Tabelle 17.1] oder [3, Tabelle 17.1].

Über die Eigenschaft `contents` wird der Inhalt der Ebene festgelegt. Hier muss also die Marke ausgegeben werden. Um dies abschalten zu können, ohne die Ebene aus dem Seitenstil der betreffenden Seiten entfernen zu müssen, wurde zuvor ein neuer Schalter `\ifKapitelmarken` definiert. In der Voreinstellung entspricht dieser `\iffalse`, so dass keine Marken ausgegeben werden.

Die Ausgabe der Marken erfolgt in einer `\parbox` der Höhe und Breite der Ebene, `\layerheight` und `\layerwidth`. Die Marke soll im rechten Rand rechtsbündig platziert werden. Die vertikale Position wird mit einem vertikalen Abstand erreicht. Um die Marke für den Anschnitt über den rechten Rand hinaus schieben zu können, wird eine `\makebox` in Markenbreite mit linksbündiger Ausrichtung verwendet. Das um `\Anschnitt` breitere Rechteck in der Farbe der aktuellen Markennummer ragt damit genau um diese Überbreite über den Rand der Seite hinaus. In einem PDF-Betrachter wird dies in der Regel nicht zu erkennen sein, da dieser normalerweise nur das Material anzeigt, das innerhalb der Seite liegt.

Damit ist die Ebene fertig definiert. Um eine Ebene zu aktivieren, muss man sie bei `scrlayer` an einen Seitenstil binden:

```

60 \AddLayersToPageStyle{@everystyle@}{Kapitelmarken}

```

Im einfachsten Fall binden wir sie an den Pseudo-Seitenstil `@everystyle@`. Die Ebenen dieses Seitenstils werden grundsätzlich auf alle Seitenstile, die mit `scrlayer` definiert wurden, angewendet. Für das Beispiel selbst verwenden wir die Klasse `scrbook`. Damit sind alle Seitenstile, die von der Klasse verwendet oder bereitgestellt werden, nach dem Laden von `scrlayer-scrpage` Ebenenseitenstile und verwenden die Ebenen

von @everystyle@. Also tragen nun alle ungeraden Seiten Kapitelmarken, sobald diese mit `\Kapi telmarkent rue` aktiviert wurden, was man natürlich erst nach den Titelseiten macht. Ein minimaler Dokumentkörper könnte dann so aussehen und unter anderem die ungeraden Seiten aus Abbildung 1 liefern:

```

72 \usepackage{blindtext}
73
74 \begin{document}
75 \title{Einfache farbige Kapitelmarken}
76 \author{Markus Kohm}
77 \date{Juli 2014}
78 \maketitle
79 \cleardoubleoddpag
80 \Kapi telmarkent rue
81 \tableofcontents
82 \blinddocument
83 \blinddocument
84 \end{document}

```

Die erste Seite ist dabei die Titelseite, die vor `\Kapi telmarkent rue` ausgegeben wurde und daher keine Marke trägt.

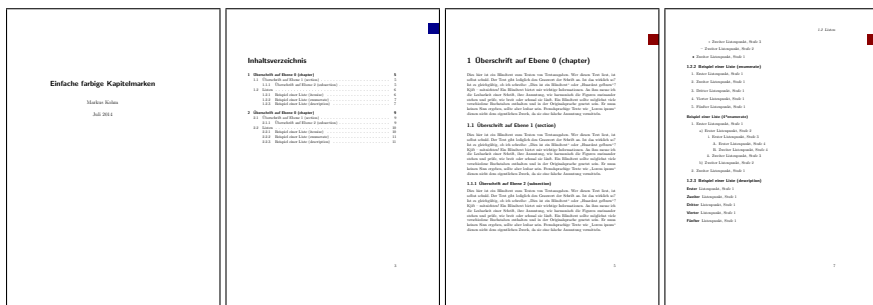


Abbildung 1: vier Seiten ohne und mit Kapitelmarken

Abwandlung

Statt die Kapitelmarken auf alle ungeraden Seiten zu drucken, könnte man sie auch auf die Kapitelanfangsseiten beschränken. Dazu definieren wir zunächst einen neuen Seitenstil:

```

45 \DeclareNewPageStyleByLayers{chapter}{%
46   plain.scrheadings.head.above.line,%
47   plain.scrheadings.head.odd,%
48   plain.scrheadings.head.even,%
49   plain.scrheadings.head.oneside,%

```

```

50 plain.scrheadings.head.below.line,%
51 plain.scrheadings.foot.above.line,%
52 plain.scrheadings.foot.odd,%
53 plain.scrheadings.foot.even,%
54 plain.scrheadings.foot.onside,%
55 plain.scrheadings.foot.below.line,%
56 Kapitelmarken%
57 }

```

Statt alle Ebenen für diesen Stil neu festzulegen, wurden dabei einfach die Ebenen des Seitenstils `plain.scrheadings` verwendet. Hinzu kam lediglich die neu definierte Ebene `Kapitelmarken`. Alternativ hätte man auch direkt die Ebenen kopieren können:

```

\DeclareNewPageStyleByLayers{chapter}{Kapitelmarken}
\ForEachLayerOfPageStyle*{plain.scrheadings}{%
  \AddLayersToPageStyle{chapter}{#1}%
}

```

Egal wie man den neuen Seitenstil definiert, man muss noch dafür sorgen, dass er auf Kapitelanfangsseiten auch verwendet wird. Das geht bei den KOMA-Script-Klassen `scrbook` und `scrreprt` deshalb besonders einfach, weil man den Seitenstil der Kapitelanfangsseiten explizit festlegen kann:

```

58 \renewcommand*{\chapterpagestyle}{chapter}

```

Bei Verwendung einer anderen Klasse müsste man gegebenenfalls mit einem expliziten `\thispagestyle{chapter}` nach der Kapitelüberschrift dafür sorgen. Auch dies könnte man aber über `xpatch` bei Bedarf automatisieren.

Zusammenfassung

Es wurde gezeigt, wie man mit `scrlayer` neue Ebenen definieren und diese Seitenstilen hinzufügen kann. Darüber hinaus wurde erläutert, wie man die Verwendung von Ebenen über Eigenschaften auf bestimmte Seiten einschränken kann. Es sei darauf hingewiesen, dass es neben der gezeigten Eigenschaft `oddpages` weitere derartige Einschränkungsmöglichkeiten gibt, etwa für Gleitumsgebungsseiten oder Seiten im einseitigen Satz. Näheres dazu ist [4] oder [3] zu entnehmen. Nebenbei wurde eine Anweisung aus `xcolor` verwendet, die nicht nur für dieses Beispiel sehr nützlich sein kann, und es wurde in die Theorie randabfallender Seiteninhalte eingeführt.

Literatur und Software

- [1] Enrico Gregorio. *The xpatch package. Extending etoolbox patching commands*. Okt. 2012. CTAN:macros/latex/contrib/xpatch/ (besucht am 21. 04. 2015).
- [2] Uwe Kern. *Extending L^AT_EX's color facilities: the xcolor package*. Jan. 2007. CTAN:macros/latex/contrib/xcolor/ (besucht am 21. 04. 2014).

- [3] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für $\text{\LaTeX} 2_{\epsilon}$* . 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [4] Markus Kohm. *KOMA-Script ein wandelbares $\text{\LaTeX} 2_{\epsilon}$ -Paket*. Apr. 2015. CTAN: /macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 01.05.2015).

Ändern des Kapitelformats¹

Elke Schubert

Das aktuelle KOMA-Script verträgt sich nicht mit dem Paket `titlesec`, welches sehr oft benutzt wird, wenn das Layout von Überschriften verändert werden soll. Dieser Beitrag zeigt, wie dies ausschließlich mit KOMA-Script-Mitteln möglich ist.

Die folgenden Vorschläge basieren auf der Annahme, dass es sich um einzeilige Überschriften handelt. Eine der Möglichkeiten ist die Modifikation des Makros `\chapterheadstartvskip`:

```

1 \documentclass[ngerman]{scrreprt}
2 \usepackage{ifluatex}
3 \ifluatex\else\usepackage[utf8]{inputenc}\fi
4 \usepackage{libertine}
5 \usepackage{babel}
6
7 \usepackage{blindtext}
8 \usepackage{xcolor}
9 \usepackage{etoolbox}
10
11 \appto\chapterheadstartvskip{%
12   \smash{\makebox[0pt][l]{%
13     \textcolor{gray}{%
14       \usesizeofkomafont{chapter}%
15       \rule[\dimexpr-2\ht\strutbox-\dp\strutbox]{\linewidth}{2\baselineskip}}}}
16   \addtokomafont{chapter}{\color{white}}
17 \begin{document}
18 \chapter{Ein Kapitel}
19 \blindtext
20 \chapter*{Ein Kapitel ohne Nummer}
21 \blindtext
22 \end{document}
```

¹ Zuerst veröffentlicht auf <http://tex.stackexchange.com/questions/257247/same-chapterformat-for-chapter-and-chapter-without-titlesec>

Das Paket `etoolbox` ermöglicht dabei mit `\appto` (»append to«) ein einfaches Anhängen von Code an ein bestehendes Makro, hier `\chapterheadstartvskip`. Mit `\makebox[Opt][l]{...}` kann ein Inhalt von beliebiger Breite linksbündig gesetzt werden, ohne dass sich der aktuelle Punkt verändert; der Boxinhalt hat keine Breite und eine eventuelle Tiefe bleibt durch `\smash` unberücksichtigt. Dadurch wird der Titel »1 Ein Kapitel« vertikal zentriert ausgegeben.

1 Ein Kapitel

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest geburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Soll der graue Hintergrundbalken über die ganze Seitenbreite gehen, so ist als optionales Argument von `\makebox` »c« zu benutzen. Da dies der Standardvorgabe entspricht, kann man diese Option auch ignorieren. Durch das Zentrieren in einer Box mit `Opt` können Inhalte, die breiter als die aktuelle Textbreite sind, symmetrisch auf die Seite platziert werden. Zusätzlich wird in folgendem Beispiel noch das Makro `\chapterformat` verändert:

```

8 \usepackage{xcolor}
9 \usepackage{etoolbox}
10 \appto\chapterheadstartvskip{%
11 \smash{\makebox[0pt]{%
12 \textcolor{gray}{%
13 \usesizeofkomafont{chapter}%
14 \rule[\dimexpr-2\ht\strutbox-\dp\strutbox]{2\paperwidth}{2\baselineskip}}}%
15 \renewcommand\chapterformat{%
16 \makebox[0pt][r]{\thechapter\
17 \textcolor{orange}{\smash{\rule[-\dp\strutbox]{2.5pt}{\baselineskip}}}}%
18 \enskip}
19 \addtokomafont{chapter}{\color{white}}
20
21 \begin{document}
22 \chapter{Ein Kapitel}
23 \blindtext
24 \chapter*{Ein Kapitel ohne Nummer}
25 \blindtext
26 \end{document}

```

1 | Ein Kapitel

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

In mehreren Beiträgen in dieser Ausgabe von »Die T_EXnische Komödie« wurde bereits die Anwendung des Paketes `scrlayer-scrpage` zum Erstellen von Seitenstilen gezeigt. Es eignet sich auch für diesen Fall, um auf der ersten Seite eines Kapitels die Überschrift mit einem grauen Hintergrund zu versehen. Hierbei wurde der vertikale Abstand der Überschrift manuell gesetzt.

```

8 \usepackage{xcolor}
9 \usepackage[automark]{scrlayer-scrpage}
10 \DeclareNewLayer[
11   background, topmargin, addheight=\headheight,
12   contents={\color{gray}\rule{\layerwidth}{\layerheight}}
13 ]{chapterbackground}
14
15 \DeclareNewPageStyleByLayers{chapterpage}{chapterbackground}
16
17 \ForEachLayerOfPageStyle*{plain}{%
18   \ifstrstart{#1}{plain.scrheadings.foot}{\AddLayersToPageStyle{chapterpage
19   ↳}{#1}}{}}
20
21 \renewcommand\chapterpagestyle{chapterpage}
22 \renewcommand\chapterheadstartvskip{\vspace*{-4.7\baselineskip}}
23 \RedeclareSectionCommand[afterskip=2.8\baselineskip]{chapter}
24 \renewcommand\chapterformat{%
25   \makebox[0pt][r]{\thechapter\
26     \textcolor{orange}{\smash{\rule[-\dp\strutbox]{2.5pt}{\baselineskip}}}}%
27   \enskip}
28 \addtokomafont{chapter}{\color{white}}
29 \begin{document}
30 \tableofcontents
31 \Blinddocument
32 \chapter{Ein Kapitel}
33 \blindtext
34 \chapter*{Ein Kapitel ohne Nummer}
35 \blindtext
36 \end{document}

```


natürlich fehleranfällig ist, oder aber die Anweisung `\chapter` so geändert werden musste, dass sie das Argument für die `.toc`-Datei zusätzlich in eine weitere Verzeichnisdatei schrieb. Bei der Umdefinierung von `\chapter` ergibt sich das Problem, dass dabei sowohl die Sternform als auch Versionen mit und ohne optionales Argument weiterhin funktionsfähig bleiben müssen. Außerdem verbergen sich hinter `\chapter` eine ganze Reihe von geschachtelten, internen Anweisungen, die sich jederzeit ändern können.

Mit Einführung der Anweisung:

```
\addchaptertocentry{<Nummer>}{<Text>}
```

gibt es einen weiteren Ansatzpunkt. Diese seit KOMA-Script 3.08 neue Anweisung, die in [2, Abschnitt 21.3] und [1, Kapitel 21.3] zu finden ist, wird bei KOMA-Script dazu verwendet, Kapiteleinträge in die `.toc`-Datei für das Inhaltsverzeichnis zu schreiben. Dabei ist `<Nummer>` die Gliederungsnummer des Kapitels und `<Text>` der Text der Überschrift, wie er im Inhaltsverzeichnis erscheinen soll. Bei Einträgen ohne Gliederungsnummer bleibt das erste Argument einfach leer.

Um nun die Kapiteleinträge nicht nur in die `.toc`-Datei, sondern auch in einer weitere Datei mit Endung `.ctoc` zu schreiben, muss lediglich `\addchaptertocentry` umdefiniert werden:

```
10 \newcommand*{\savedaddchaptertocentry}{}
11 \let\savedaddchaptertocentry\addchaptertocentry
12 \renewcommand*{\addchaptertocentry}[2]{%
13   \savedaddchaptertocentry{#1}{#2}%
14   \ifstr{#1}{}{%
15     \addxcontentsline{ctoc}{chapter}{#2}%
16   }{%
17     \addxcontentsline{ctoc}{chapter}[{#1}]{#2}%
18   }%
19 }
```

In Zeile 10 und 11 wird zunächst die ursprüngliche Bedeutung der Anweisung `\addchaptertocentry` in das neue `\savedaddchaptertocentry` kopiert. Danach wird die Anweisung selbst ab Zeile 12 umdefiniert. In Zeile 13 wird dabei die kopierte Originalanweisung ausgeführt, um nach wie vor den Eintrag ins Inhaltsverzeichnis vorzunehmen. In den Zeilen 14 bis 18 wird abhängig davon, ob eine Gliederungsnummer angegeben ist oder nicht, mit `\addxcontentsline` ein Eintrag in die neue `.ctoc`-Datei geschrieben. Die Anweisung `\ifstr` stellt einen einfachen Vergleich zwischen der Expansion der ersten beiden Argumente her. Im Falle der Gleichheit wird das dritte Argument verwendet, im Falle der Ungleichheit das vierte. Siehe dazu [2, Abschnitt 12.3] oder [1, Abschnitt 12.3].

Die Anweisung:

```
\addxcontentsline{<Datei>}{<Typ>}[<Nummer>]{<Text>}
```

wird von Paket `tocbasic` zur Verfügung gestellt, das von allen Klassen von KOMA-Script für die Verwaltung der Verzeichnisse verwendet wird. Sie arbeitet ähnlich zu `\addcontentsline`, wobei allerdings die Gliederungsnummer ein eigener optionaler Parameter ist. Näheres ist in [2, Abschnitt 15.2] oder [1, Abschnitt 15.2] zu finden.

Damit sind die Kapiteleinträge ins Inhaltsverzeichnis bereits in die neue Kapitelübersicht kopiert. Was fehlt, sind die Zusammenfassungen. Dafür wird eine neue Anweisung definiert:

```
20 \newcommand*{\chaptershortabstract}[1]{%
21   \addxcontentsline{ctoc}{chapterabstract}{#1}%
22 }
```

Diese schreibt ihr Argument einfach nur als Eintrag des Typs `chapterabstract` in die `.ctoc`-Datei.

Für den neuen Eintragstyp `chapterabstract` wird dann natürlich ebenfalls eine Anweisung benötigt, die den Eintrag im Verzeichnis ausgibt:

```
24 \newcommand*{\l@chapterabstract}[2]{% 1: Text, 2: Seite
25   {\leftskip=1.5em\small #1\par}%
26 }
```

Wie im Kommentar angegeben ist das erste Argument dieser Anweisung der auszugebende Text und das zweite die Seitenzahl der Seite, auf welcher der Eintrag erstellt wurde. Die Seitenzahl verwenden wir nicht. Stattdessen wird nur der Text des Eintrags mit einem festen Abstand vom linken Rand und in kleinerer Schrift ausgegeben.

Was jetzt noch fehlt, ist die Definition des Verzeichnisses selbst:

```
28 \DeclareNewTOC[listname={The Chapters}]{ctoc}
```

Dies geschieht wieder mit Hilfe des Pakets `tocbasic`. Dabei wird nicht nur die Endung `.ctoc` quasi reserviert, sondern auch gleich der Titel des Verzeichnisses als »The Chapters« festgelegt und die Anweisung `\listofctocs` für die Ausgabe des Verzeichnisses definiert (siehe [2, Abschnitt 15.5] oder [1, Abschnitt 15.5])

Wie sich bei einem ersten Test zeigte, kann sich allerdings bei einem derartigen Verzeichnis der vertikale Ausgleich mit `\flushbottom`, wie er bei doppelseitigen Dokumenten üblicherweise voreingestellt ist, sehr unschön auswirken. Deshalb wird mit

```
29 \AfterTOCHead[ctoc]{\raggedbottom}
```

zusätzlich dafür gesorgt, dass die Kapitelübersicht ohne vertikalen Ausgleich gesetzt wird. Auch dabei ist wieder ein Anweisung aus `tocbasic` behilflich (siehe [2, Abschnitt 15.2] oder [1, Abschnitt 15.2]).

Mit dem vollständigen Beispielcode aus Listing 1 ergeben sich so die beiden Seiten aus Abbildung 1. Die linke Seite zeigt dabei die Kapitelübersicht, während die rechte das ganze normale Inhaltsverzeichnis ist.

Listing 1: Beispieldokument mit Kapitelverzeichnis und Inhaltsverzeichnis

```

9 \documentclass{scrbook}
10 \usepackage{blindtext}
11 \newcommand*{\savedaddchaptertocentry}{}
12 \let\savedaddchaptertocentry\addchaptertocentry
13 \renewcommand*{\addchaptertocentry}[2]{%
14   \savedaddchaptertocentry{#1}{#2}%
15   \ifstr{#1}{}{%
16     \addxcontentsline{ctoc}{chapter}{#2}%
17   }%
18   \addxcontentsline{ctoc}{chapter}[[#1]]{#2}%
19 }%
20 }
21 \newcommand*{\chaptershortabstract}[1]{%
22   \addxcontentsline{ctoc}{chapterabstract}{#1}%
23 }
24 \makeatletter
25 \newcommand*{\l@chapterabstract}[2]{% 1: Text, 2: Seite
26   {\leftskip=1.5em\small #1\par}%
27 }
28 \makeatother
29 \DeclareNewTOC[lisname={The Chapters}]{ctoc}
30 \AfterTOCHead[ctoc]{\raggedbottom}
31 \begin{document}
32 \listoftocs
33 \tableofcontents
34 \chapter{First example chapter}
35 \chaptershortabstract{This is the first example chapter
36   of the document. \blindtext}
37 \Blindtext
38 \chapter{Second example chapter}
39 \chaptershortabstract{This is the second example chapter of the
40   document. \blindtext}
41 \Blindtext
42 \blinddocument
43 \chaptershortabstract{This is the last example chapter of the
44   document. \blindtext}
45 \end{document}

```

Zusammenfassung

Am Beispiel eines Verzeichnisses mit einer Kapitelübersicht wurde gezeigt, wie man `\addchaptertocentry` dazu verwenden kann, an die Erstellung der Kapiteleinträge ins Inhaltsverzeichnis weitere Aktionen zu knüpfen. Hierzu wurden die Einträge zusätzlich in eine weitere Hilfsdatei geschrieben.

Abbildung 1: Kapitelübersicht und Inhaltsverzeichnis aus Listing 1

Literatur und Software

- [1] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für L^AT_EX₂ ϵ* . 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [2] Markus Kohm. *KOMA-Script ein wandelbares L^AT_EX₂ ϵ -Paket*. Apr. 2015. CTAN:macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 01. 05. 2015).

Kombination von Kapitelmarken mit einer Kapitelübersicht

Markus Kohm

Dieser Beitrag ist in gewisser Weise eine Kombination der beiden vorhergehenden von Seite 24 und Seite 33, bringt also Kapitelmarken und Kapitelübersicht zusammen, wobei in der Kapitelübersicht die Markierungen der Kapitelmarken übernommen werden sollen.

Aufgabenstellung

Diese Anforderung wurde so konkret nie an mich gestellt, liegt aber eigentlich nach dem Lesen der Beiträge von Seite 24 und 33 nahe. Will man in einem Werk sowohl Kapitelmarken als auch eine Kapitelübersicht, wäre es schön, wenn die Kapitelmarken auch in der Kapitelübersicht direkt bei den entsprechenden Kapitelüberschriften und deren Kurzbeschreibungen zu sehen wären. Damit hätte man einen direkten Bezug zwischen der Marke und dem Inhalt. Allerdings ergeben sich daraus zusätzliche Design-Probleme:

- Hat man in der Kapitelübersicht zusätzliche Beschreibungstexte, dann stehen die Kapitelmarken in größerer vertikaler Entfernung.
- Sind einige Kapitelüberschriften mehrzeilig, so wären entweder die Kapitelmarken unterschiedlich hoch oder sie stünden in unterschiedlichem vertikalem Abstand zueinander.

Im weiteren Text des Werks sollen die Marken aber immer möglichst dicht aufeinander folgen, um ein möglichst übersichtliches Register zu schaffen. Dadurch ginge dann aber die räumliche Verbindung zwischen den Marken in der Kapitelübersicht und im restlichen Text verloren. Um dies zu vermeiden, gibt es unterschiedliche Lösungsmöglichkeiten:

- Auf Kapitelbeschreibungen wird verzichtet und Kapitelüberschriften dürfen im Verzeichnis nur einzeilig sein.
- Die Kapitelmarken sind mindestens so hoch wie der längste Kapiteleintrag einschließlich Kurzbeschreibung.

Kombinationen aus beiden Ansätzen sind denkbar.

Lösung

Für die im Vortrag realisierte Lösung wurde der Ansatz verwendet, dass zum einen auf die Kurzbeschreibungen in der Übersicht verzichtet wurde und zum anderen die Kapitelüberschriften eine konfigurierbare Markenhöhe einhalten müssen.

Da es Kapitel mit und ohne Kapitelmarkierungen geben kann, mussten die Einträge in das Kapitelverzeichnis leicht modifiziert werden:

```

53 \renewcommand*{\addchaptertocentry}[2]{%
54   \savedaddchaptertocentry{#1}{#2}%
55   \ifKapitelmarken
56     \ifstr{#1}{ }{%
57       \addxcontentsline{ctoc}{markedchapter}{#2}%
58     }{%
59       \addxcontentsline{ctoc}{markedchapter}[{#1}]{#2}%
60     }%
61   \else
62     \ifstr{#1}{ }{%
63       \addxcontentsline{ctoc}{unmarkedchapter}{#2}%
64     }{%
65       \addxcontentsline{ctoc}{unmarkedchapter}[{#1}]{#2}%
66     }%
67   \fi
68 }

```

Hier wird also für Kapitel mit Kapitelmarke ein Eintrag vom Typ `markedchapter` erzeugt, während im anderen Fall der Eintragstyp `unmarkedchapter` verwendet wird. Der Befehl für die Verzeichniseinträge der unmarkierten Kapitel ist recht einfach gehalten:

```

99 \newcommand*{\l@unmarkedchapter}[2]{%
100   \par
101   \ifdim \dimexpr\pagegoal-\pagetotal\relax
102     <\Kapitelmarkenhoehe
103     \cleardoubleoddpages
104   \fi
105   \begingroup
106     \setlength{\lineskip}{0pt}% kein Mindestabstand
107     \noindent%
108     \parbox[b][\Kapitelmarkenhoehe][c]{\textwidth}{%
109       \l@chapter{#1}{#2}%
110     }%
111   \endgroup
112 }

```

Dabei wird zunächst sichergestellt, dass auf der aktuellen Seite noch genügend Platz für eine weitere Kapitelmarke und damit für einen weiteren, kompletten Eintrag ist. Falls das nicht der Fall ist, wird nicht einfach nur auf die nächste Seite, sondern auf die nächste rechte Seite gewechselt. Das ist notwendig, weil wir Kapitelmarken nur

auf rechten Seiten haben und damit auch die Übersicht – insbesondere natürlich bei den im Anschluss gezeigten markierten Einträgen – nur auf rechten Seiten sinnvoll ist.

Anschließend wird der normale Kapiteleintrag vertikal zentriert in einer Box der Höhe einer Kapitelmarke ausgegeben. Die Änderung von `\lineskip` sorgt dabei dafür, dass diese Boxen unmittelbar untereinander stehen können und nicht durch einen Mindestabstand voneinander getrennt werden.

Der Befehl für die Verzeichniseinträge mit Kapitelmarkierung ist dagegen etwas aufwändiger:

```

71 \newcommand*{\l@markedchapter}[2]{%
72   \par
73   \ifdim \dimexpr\pagegoal-\pagetotal\relax
74     <\Kapitelmarkenhoehe
75     \cleardoubleoddpag
76   \fi
77   \begingroup
78     \setlength{\lineskip}{0pt}% kein Mindestabstand
79     \noindent%
80     \parbox[b][\Kapitelmarkenhoehe][c]{\textwidth}{%
81       \l@chapter{#1}{#2}%
82     }%
83     \stepcounter{MyLocalKapitelmarke}%
84     \expandafter\xdef
85     \csname Kapitelmarkengoal\theMyLocalKapitelmarke
86     \endcsname{%
87       \the\pagetotal
88     }%
89     \makebox[0pt][l]{%
90       \hspace*{\dimexpr\paperwidth
91         -\textwidth-\oddsidemargin-1in
92         -\Kapitelmarkenbreite}%
93       \colorifexists{Kapitelmarke\theMyLocalKapitelmarke}%
94       \rule{\dimexpr\Kapitelmarkenbreite+\Anschnitt}%
95         {\Kapitelmarkenhoehe}%
96     }%
97   \endgroup
98 }
```

Auch hier muss natürlich sichergestellt werden, dass der Eintrag noch auf die Seite passt. Ebenfalls wird anschließend der Kapiteleintrag gesetzt.

Allerdings muss nun die vertikale Position der Marke zusätzlich gespeichert werden, damit diese Marke später im Text an derselben Stelle im Rand ausgegeben werden kann. Dazu wird ein Makro nach dem Schema `\Kapitelmarkengoal<Nummer>` definiert, dessen Inhalt der aktuelle Wert von `\pagetotal`, also der aktuellen Füllhöhe der Seite ist.

Zum Schluss muss noch neben der zuvor ausgegebenen Box eine Farbmarkierung in den Rand geschrieben werden. Dies ist über eine horizontale Box der angeblichen Breite 0 einfach möglich.

Die Farbe wird dabei nicht direkt über die von den Farbpaketen `color` und `xcolor` bekannte Anweisung `\color`, sondern über eine selbst definierte Anweisung `\colorifexists` gesetzt:

```

133 \newcommand*{\colorifexists}[1]{%
134   \scr@ifundefinedorrelax{\string\color@#1}{%
135     \GenericWarning{%
136       }{%
137         \string\colorifexists\space Warning: Color `#1' undefined%
138       }%
139     }{\color{#1}}}%
140 }

```

Diese Anweisung verwendet internes Wissen aus den genannten Farbpaketen, um zunächst zu prüfen, ob die Farbe auch definiert ist. Ist eine Farbe mit dem angegebenen Namen nicht definiert, so wird eine Warnung ausgegeben und es bleibt bei der aktuellen Farbe, also normalerweise Schwarz. Natürlich ist die Verwendung solch internen Wissens immer mit dem Risiko behaftet, dass es bei einer Änderung des Pakets nicht mehr funktioniert. Leider bieten die beiden Pakete selbst keine Benutzeranweisung um festzustellen, ob eine Farbe definiert wurde. In unserem Fall würde eine Änderung der Interna immerhin insoweit auffallen, als dann plötzlich alle Farben als nicht definiert gemeldet würden.

Die Namen der für die Marken vorgesehenen Farben setzen sich einfach aus »Kapitelmarke« und einer fortlaufenden Nummer zusammen. Im Beispiel werden sie als

```

119 \colorlet{Kapitelmarke1}{DarkBlue}
120 \colorlet{Kapitelmarke2}{DarkRed}
121 \colorlet{Kapitelmarke3}{DarkGreen}

```

definiert.

Normalerweise werden zwischen den Kapiteleinträgen von `\l@chapter` auch vertikale Abstände eingefügt. Dies erfolgt mit `\addvspace` und ist bei unserem Verzeichnis nicht erwünscht. Statt `\addvspace` bei jedem einzelnen Eintrag zu deaktivieren, machen wir das genau ein Mal:

```

113 \AfterTOCHead[ctoc]{%
114   \raggedbottom
115   \let\addvspace\@gobble
116 }

```

Bei der Gelegenheit wird auch der vertikale Ausgleich im Verzeichnis deaktiviert, was allerdings durch die Verwendung von `\cleardoubleoddpages` oben normalerweise gar nicht zwingend notwendig wäre.

Was uns jetzt noch fehlt, sind die Marken im Rand:

```

25 \DeclareNewLayer[%
26   background,oddpages,textarea,
27   align=tr,offset=\paperwidth,width=\Kapitelmarkenbreite,
28   contents=%
29   \ifKapitelmarken
30     \expandafter\providecommand\expandafter*\expandafter{%
31       \csname Kapitelmarkengoal\theKapitelmarke\endcsname
32     }%
33     \parbox[t][\layerheight][t]{\layerwidth}%
34     \raggedleft
35     \vspace*{\dimexpr
36       \csname Kapitelmarkengoal\theKapitelmarke\endcsname
37       -\ht\strutbox
38     }%
39     \makebox[\Kapitelmarkenbreite][l]{%
40       \colorifexists{Kapitelmarke\theKapitelmarke}%
41       \rule{\dimexpr\Kapitelmarkenbreite+\Anschnitt}%
42       {\Kapitelmarkenhoehe}%
43     }%
44   }%
45 \fi
46 }%
47 }%{Kapitelmarken}
48 \AddLayersToPageStyle{@everystyle@}{Kapitelmarken}

```

Diese sind wie bereits im Artikel von Seite 24 mit Hilfe einer Ebene realisiert, die allen Seitenstile hinzugefügt wird. Der Hauptunterschied ist dabei, dass die in `\Kapitelmarkengoal<Nummer>` gespeicherte Position verwendet wird, damit die Marke dieselbe vertikale Position erhält wie im Kapitelverzeichnis. Damit ist auch automatisch die Frage geklärt, was geschieht, wenn es mehr Kapitel gibt, als Kapitelmarken untereinander passen würden: Es wird automatisch wieder oben angefangen.

Mit dem vollständigen Beispielcode aus Listing 1 ergeben sich so das Verzeichnis und die Kapitelanfangsseiten aus Abbildung 1. Die erste abgebildete Seite ist dabei das Kapitelverzeichnis ohne eigene Marke, die zweite das Inhaltsverzeichnis mit eigener Marke. Es folgen die Kapitel 1 bis 4.

Listing 1: Beispieldokument mit Kapitelverzeichnis und Kapitelmarken

```

9 \documentclass{scrbook}
10 \usepackage[ngerman]{babel}
11 \usepackage{blindtext}
12
13 % Marken -----
14 \usepackage{svgnames}{xcolor}

```

```

15 \usepackage{etoolbox}
16 \usepackage{scrlayer-scrpage}
17 \newlength\Kapitelmarkenhoehe
18 \setlength\Kapitelmarkenhoehe{2cm}
19 \newcounter{Kapitelmarke}
20 \newlength\Kapitelmarkenbreite
21 \setlength\Kapitelmarkenbreite{1cm}
22 \newlength\Anschnitt
23 \setlength\Anschnitt{3mm}
24 \newif\ifKapitelmarken
25 \DeclareNewLayer[{\%
26   background, oddpage, textarea,
27   align=tr, hoffset=\paperwidth, width=\Kapitelmarkenbreite,
28   contents={\%
29     \ifKapitelmarken
30       \expandafter\providecommand\expandafter*\expandafter{\%
31         \csname Kapitelmarkengoal\theKapitelmarke\endcsname
32       }{0pt}\%
33       \parbox[t][\layerheight][t]{\layerwidth}{\%
34         \raggedleft
35         \vspace*{\dimexpr
36           \csname Kapitelmarkengoal\theKapitelmarke\endcsname
37           -\ht\strutbox
38         }{\%
39           \makebox[\Kapitelmarkenbreite][l]{\%
40             \colorifexists{Kapitelmarke\theKapitelmarke}{\%
41               \rule{\dimexpr\Kapitelmarkenbreite+\Anschnitt}{\%
42                 {\Kapitelmarkenhoehe}{\%
43               }{\%
44             }{\%
45           }{\fi
46         }{\%
47       }]{Kapitelmarken}
48 \AddLayersToPageStyle{@everystyle@}{Kapitelmarken}
49
50 % Verzeichnis -----
51 \newcommand*\savedaddchaptertocentry{}{}
52 \let\savedaddchaptertocentry\addchaptertocentry
53 \renewcommand*\addchaptertocentry[2]{\%
54   \savedaddchaptertocentry{#1}{#2}\%
55   \ifKapitelmarken
56     \ifstr{#1}{\%
57       \addxcontentsline{ctoc}{markedchapter}{#2}\%
58     }{\%
59       \addxcontentsline{ctoc}{markedchapter}[{#1}]{#2}\%
60     }{\%
61   \else
62     \ifstr{#1}{\%
63       \addxcontentsline{ctoc}{unmarkedchapter}{#2}\%
64     }{\%
65       \addxcontentsline{ctoc}{unmarkedchapter}[{#1}]{#2}\%
66     }{\%

```

```

67 \fi
68 }
69 \makeatletter
70 \newcounter{MyLocalKapitelmarke}
71 \newcommand*{\l@markedchapter}[2]{%
72   \par
73   \ifdim \dimexpr\pagegoal-\pagetotal\relax
74     <\Kapitelmarkenhoehe
75     \cleardoubleoddpages
76   \fi
77   \begingroup
78     \setlength{\lineskip}{0pt}% kein Mindestabstand
79     \noindent%
80     \parbox[b][\Kapitelmarkenhoehe][c]{\textwidth}{%
81       \l@chapter{#1}{#2}%
82     }%
83     \stepcounter{MyLocalKapitelmarke}%
84     \expandafter\xdef
85     \csname Kapitelmarkengoal\theMyLocalKapitelmarke
86     \endcsname{%
87       \the\pagetotal
88     }%
89     \makebox[0pt][l]{%
90       \hspace*{\dimexpr\paperwidth
91         -\textwidth-\oddsidemargin-1in
92         -\Kapitelmarkenbreite}%
93       \colorifexists{Kapitelmarke\theMyLocalKapitelmarke}%
94       \rule{\dimexpr\Kapitelmarkenbreite+\Anschnitt}%
95         {\Kapitelmarkenhoehe}%
96     }\par
97   \endgroup
98 }
99 \newcommand*{\l@unmarkedchapter}[2]{%
100   \par
101   \ifdim \dimexpr\pagegoal-\pagetotal\relax
102     <\Kapitelmarkenhoehe
103     \cleardoubleoddpages
104   \fi
105   \begingroup
106     \setlength{\lineskip}{0pt}% kein Mindestabstand
107     \noindent%
108     \parbox[b][\Kapitelmarkenhoehe][c]{\textwidth}{%
109       \l@chapter{#1}{#2}%
110     }\par
111   \endgroup
112 }
113 \AfterTOCHead[ctoc]{%
114   \raggedbottom
115   \let\addvspace\@gobble
116 }
117 \makeatother
118 \DeclareNewTOC[lisname={The Chapters}]{ctoc}

```

```

119 \colorlet{Kapitelmarke1}{DarkBlue}
120 \colorlet{Kapitelmarke2}{DarkRed}
121 \colorlet{Kapitelmarke3}{DarkGreen}
122
123 % Erhöhen der Marken
124 \makeatletter
125 \apptocmd{\@makechapterhead}{\ifKapitelmarken\stepcounter{Kapitelmarke}\fi}
126   {}{\undefined}
127 \apptocmd{\@makeschapterhead}{\ifKapitelmarken\stepcounter{Kapitelmarke}\fi}
128   {}{\undefined}
129 \makeatother
130
131 % Kleine Erweiterung von xcolor:
132 \makeatletter
133 \newcommand*{\colorifexists}[1]{%
134   \scr@ifundefinedorrelax{string\color@#1}{%
135     \GenericWarning{%
136       }{%
137         \string\colorifexists\space Warning: Color `#1' undefined%
138       }%
139     }\color{#1}}%
140   }
141 \makeatother
142 \usepackage{hyperref}
143
144 \begin{document}
145 \title{Einfache farbige Kapitelmarken}
146 \author{Markus Kohm}
147 \date{Juli 2014}
148 \maketitle
149 \listoftocs
150
151 \cleardoubleoddpag
152 \Kapitelmarkenttrue
153 \addxcontentsline{ctoc}{markedchapter}{\contentsname}
154 \tableofcontents
155 \blinddocument
156 \blinddocument
157 \Kapitelmarkenfalse\blinddocument\Kapitelmarkenttrue
158 \blinddocument
159 \end{document}

```

Wie zu sehen ist, sind Kapitelverzeichnis und Kapitel 3 ohne Randmarkierung. Die Marke zu Kapitel 4 bleibt schwarz, da nur drei Farben definiert wurden.

Zusammenfassung

Die Aufgabenstellungen aus den Artikeln von Seite 24 und 33 dieser Artikelserie wurden kombiniert. Dabei wurde das Zusammenspiel verschiedener Teile von KOMA-Script, beispielsweise der Klassenanweisung `\addchaptertocentry` und der Pakete `scrlayer`, `scrlayer-scrpage` und `tocbasic` gezeigt. Nebenbei wurde erwähnt,

The Chapters Inhaltsverzeichnis 5 1 Überschrift auf Ebene 0 (chapter) 7 2 Überschrift auf Ebene 0 (chapter) 11 3 Überschrift auf Ebene 0 (chapter) 15 4 Überschrift auf Ebene 0 (chapter) 19	Inhaltsverzeichnis 1 Überschrift auf Ebene 0 (chapter) 7 1.1 Überschrift auf Ebene 1 (section) 7 1.2 Überschrift auf Ebene 2 (subsection) 7 1.3 Überschrift auf Ebene 3 (subsection) 7 1.4 Überschrift auf Ebene 4 (subsection) 7 1.5 Überschrift auf Ebene 5 (subsection) 7 2 Überschrift auf Ebene 0 (chapter) 11 2.1 Überschrift auf Ebene 1 (section) 11 2.2 Überschrift auf Ebene 2 (subsection) 11 2.3 Überschrift auf Ebene 3 (subsection) 11 2.4 Überschrift auf Ebene 4 (subsection) 11 3 Überschrift auf Ebene 0 (chapter) 15 3.1 Überschrift auf Ebene 1 (section) 15 3.2 Überschrift auf Ebene 2 (subsection) 15 3.3 Überschrift auf Ebene 3 (subsection) 15 3.4 Überschrift auf Ebene 4 (subsection) 15 4 Überschrift auf Ebene 0 (chapter) 19 4.1 Überschrift auf Ebene 1 (section) 19 4.2 Überschrift auf Ebene 2 (subsection) 19 4.3 Überschrift auf Ebene 3 (subsection) 19 4.4 Überschrift auf Ebene 4 (subsection) 19	1 Überschrift auf Ebene 0 (chapter) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält. 1.1 Überschrift auf Ebene 1 (section) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält. 1.1.1 Überschrift auf Ebene 2 (subsection) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält.
2 Überschrift auf Ebene 0 (chapter) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält. 2.1 Überschrift auf Ebene 1 (section) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält. 2.1.1 Überschrift auf Ebene 2 (subsection) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält.	3 Überschrift auf Ebene 0 (chapter) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält. 3.1 Überschrift auf Ebene 1 (section) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält. 3.1.1 Überschrift auf Ebene 2 (subsection) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält.	4 Überschrift auf Ebene 0 (chapter) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält. 4.1 Überschrift auf Ebene 1 (section) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält. 4.1.1 Überschrift auf Ebene 2 (subsection) Dies hier ist ein Blöckchen aus Texten von Testtexten. Wie dieses Text hier, ist selbst selbst. Der Text gibt lediglich den Gesamt der Sicht an. In der Wirklichkeit ist es gleichgültig, ob ich schreibe „Dies ist ein Blöckchen“ oder „Blöckchen gefahren“? Richtig: selbstverständlich! Ein Blöckchen kann aus mehreren Informationen. An den Namen ist die Bedeutung des Blöckchens. Die Aussage, wie konstruiert die Figuren zusammengefasst werden und jenseits, wie leicht oder schwer es ist. Ein Blöckchen sollte möglichst viele verschiedene Informationen enthalten und in der Ökonomie der Dinge sein. Es kann keine Blöcke haben, selbst aber keine Blöcke. Blöckchen der Text von „Lauter“ kann nicht ohne Ökonomie sein, da es eine falsche Aussage enthält.

Abbildung 1: Kapitelübersicht und Kapitelanfangsseiten aus Listing 1

wie man den aktuellen Füllstand der Seite zwischen zwei Absätzen ermitteln kann oder wie man feststellen kann, ob eine Farbe definiert ist.

Dennoch wird für das vollständige Verständnis dieses Teils nicht nur das Vorwissen aus den vorherigen Teilen, sondern auch grundlegendes $\text{\TeX}$ -Wissen benötigt, das hier nur oberflächlich berührt wurde. Für weiterführende Informationen sei ausdrücklich auf [1] oder [2] sowie die Implementierungsdokumentation der Pakete `color` und `xcolor` und natürlich auf [4] und [3] hingewiesen.

Literatur und Software

- [1] Victor Eijkhout. *$\text{\TeX}$ by Topic. A $\text{\TeX}$ nician's Reference*. DANTE e.V. und Lehmanns Media, 2014. ISBN: 978-3-86541-590-5.

- [2] Donald E. Knuth. *Computers & Typesetting*. Bd. A: *The T_EX book*. Reading, Mass.: Addison-Wesley, 1984.
- [3] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für L^AT_EX 2_ε*. 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [4] Markus Kohm. *KOMA-Script ein wandelbares L^AT_EX 2_ε-Paket*. Apr. 2015. CTAN:macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 01.05.2015).

Anhangsverzeichnis

Markus Kohm

Das Thema Anhangsverzeichnis verfolgt mich im Support seit Jahren oder sogar schon seit Jahrzehnten. Dabei geht es darum, dass vor allem Studenten und Doktoranden in ihrer Abschlussarbeit vor dem Anhang ein eigenes Verzeichnis wünschen. In der Regel sollen dann die Überschriften des Anhangs nicht mehr im Hauptinhaltsverzeichnis aufgeführt werden, sondern nur noch im Anhangsverzeichnis. Dieses wiederum soll selbst noch einen Eintrag im Inhaltsverzeichnis des Dokuments erhalten.

Vorüberlegungen

Diverse Pakete auf CTAN sind bereits heute in der Lage, ein solches Verzeichnis zu produzieren. Dabei werden unterschiedliche Anforderungen an das Dokument gestellt. Eines der mächtigsten Werkzeuge in diesem Bereich ist vermutlich das Paket etoc [1]. Ich selbst habe dieses jedoch bisher nie verwendet, und auch die Anwender, die mit entsprechenden Fragen an mich herantreten, wollen häufig eine möglichst *kleine* Lösung.

Schon 2006 habe ich deshalb eine erste Lösung, die auf Manipulationen innerhalb der toc-Datei basiert, im Rahmen des KOMA-Script Documentation Projects [2] gezeigt. Diese wurde 2010 als [5] durch eine verallgemeinerte, auf dem Paket tocstyle basierende Lösung ergänzt und seither immer wieder verbessert. Inzwischen kursiert der Code aus den Vorschlägen von [2] in diversen Foren und Dokumentvorlagen, leider in unterschiedlichen Versionen und Modifikationen und meist in einer rechtlichen Dunkelzone ohne Nennung der Quelle. Die KOMA-Script-Klassen selbst liefern allerdings seit einigen Versionen bereits einen Großteil der Mittel mit, die für ein entsprechendes Verzeichnis benötigt werden. So ist es mit `\DeclareNewTOC`

jederzeit möglich, ein neues Verzeichnis anzulegen. Die Einträge ins Inhaltsverzeichnis selbst werden von KOMA-Script-Klassen auch nicht mehr direkt per `\addcontentsline` vorgenommen, sondern mit Befehlen wie `\addchaptertocentry`, `\addsectiontocentry` etc., die ihrerseits auf `\addtocentrydefault` basieren. Hier könnte man also mit einem zusätzlichen Verzeichnis ansetzen.

Lösung

Der erste Schritt der Lösung besteht darin, mit Hilfe von `\DeclareNewTOC` ein neues Verzeichnis zu deklarieren:

```

14 \DeclareNewTOC[%
15   owner=toc,
16   types=appendices,
17   listname=\appendixname
18 ]{atoc}

```

Für die Deklaration wird die Anweisung `\DeclareNewTOC` verwendet, die in [6, Abschnitt 15.5] oder [4, Abschnitt 15.5] dokumentiert ist. Als Endung für die neue Hilfsdatei wird `atoc` gewählt. Dabei wird mit der Option `listname=\appendixname` auch gleich der Name des neuen Verzeichnisses sprachabhängig deklariert. Die Angabe `types=appendices` erlaubt lediglich, später auf das Verzeichnis mit dem Befehl `\listofappendices` statt mit dem eher unhandlichen `\listofatocs` zuzugreifen.

Eine besondere Bedeutung kommt noch der Option `owner=toc` zu. Voreinstellung wäre `owner=float`. Damit wäre das Anhangsverzeichnis ein Gleitumgebungsverzeichnis wie das Abbildungs- oder das Tabellenverzeichnis. Allerdings werden in diese Verzeichnisse in der Voreinstellung bei jedem Kapitel automatisch Abstände eingefügt. Da wir im Anhangsverzeichnis die Kapiteleinträge selbst haben, wollen wir diese zusätzlichen Abstände nicht. Deshalb wählen wir mit besagter Option einen anderen Besitzer.

Wie bereits eingangs erwähnt, soll das Anhangsverzeichnis mit seiner Überschrift ins normale Inhaltsverzeichnis eingetragen werden. Das erreichen wir ganz einfach mit:

```

19 \setuptoc{atoc}{totoc}

```

Die Anweisung `\setuptoc` einschließlich der möglichen Einstellungen findet sich in [6, Abschnitt 15.2] oder [4, Abschnitt 15.2].

Als Nächstes muss dafür gesorgt werden, dass das Verzeichnis am Anfang des Anhangs auch wirklich ausgegeben wird. Dazu wird die Anweisung `\appendix` mit Hilfe des Pakets `xpatch` (siehe [3]) modifiziert:

```

22 \xapptocmd{\appendix}{%
23   \listofappendices

```

Es wird also schlicht die oben erwähnte Anweisung `\listofappendices` am Ende der Umschaltung auf den Anhang ausgeführt.

Doch das ist nur die halbe Miete. Es muss nun auch noch dafür gesorgt werden, dass ab dem Anhang nicht mehr in die `toc`-Datei, sondern in die neue `atoc`-Datei geschrieben wird. Dafür wird im Patch von `\appendix` noch eine weitere Modifikation eingefügt:

```
24 \xpatchcmd{\addtocentrydefault}{\toc}{\atoc}{\PaTchFaileD}%
```

Es wird also mit Hilfe von `\xpatchcmd` in der Definition der Anweisung `\addtocentrydefault` das erste Auftreten von `\toc` durch `\atoc` ersetzt. Die Anweisung `\addtocentrydefault` wird in der Voreinstellung intern von allen Gliederungsbefehlen verwendet, um den Inhaltsverzeichniseintrag zu erzeugen. Näheres dazu ist [6, Abschnitt 21.3] oder [4, Abschnitt 21.3] zu entnehmen. Die Originaldefinition dieser Anweisung in KOMA-Script lautet:

```
\newcommand{\addtocentrydefault}[3]{%
  \tocbasic@addxcontentsline{\toc}{#1}{#2}{#3}%
}
```

Durch die Modifikation wird daraus:

```
\renewcommand{\addtocentrydefault}[3]{%
  \tocbasic@addxcontentsline{\atoc}{#1}{#2}{#3}%
}
```

Als letztes Argument von `\xpatchcmd` wird die undefinierte Anweisung `\PaTchFaileD` verwendet. Sollte die Modifikation fehlschlagen, wird daher eine Fehlermeldung wegen des nicht definierten `\PaTchFaileD` ausgegeben, die wir als entsprechenden Hinweis interpretieren können. Natürlich könnte man stattdessen auch eine passende Fehlermeldung beispielsweise mit `\GenericError` bauen.

Damit kann abschließend auch die Modifikation von `\appendix` auf diese Weise beendet werden:

```
33 }{\PaTchFaileD}
```

Führen wir nun einen ersten Test mit dem angegebenen Code aus, so erhalten wir auf den ersten Blick genau das gewünschte Ergebnis.

Bei Verwendung des Pakets `hyperref` (siehe [8]) fällt allerdings in den *Outlines* oder *Bookmarks* (siehe Abbildung 1), die im Deutschen etwas verwirrend ebenfalls *Lesezeichen* genannt werden, auf, dass diese nur bis zum Anhang reichen. Die Einträge für die Überschriften im Anhang fehlen.

Meist wird aber gewünscht, dass die Einträge in den Lesezeichen durchaus vorhanden sind, dort jedoch unterhalb von »Anhang« einsortiert werden. Um das zu

Überschrift auf Ebene 0 (chapter)	3
Überschrift auf Ebene 1 (section)	3
Überschrift auf Ebene 2 (subsection)	3
Listen	4
Beispiel einer Liste (terme)	4
Beispiel einer Liste (enumerate)	5
Beispiel einer Liste (descripton)	5
Anhang	7

Abbildung 1: Die Lesezeichen ohne hyperref-Hack

Überschrift auf Ebene 0 (chapter)	3
Überschrift auf Ebene 1 (section)	3
Überschrift auf Ebene 2 (subsection)	3
Listen	4
Beispiel einer Liste (terme)	4
Beispiel einer Liste (enumerate)	5
Beispiel einer Liste (descripton)	5
Anhang	7
Überschrift auf Ebene 0 (chapter)	7
Überschrift auf Ebene 1 (section)	9
Listen	10

Abbildung 2: Die Lesezeichen mit hyperref-Hack

erreichen, muss die Modifikation von `\appendix` um das Setzen solcher Lesezeichen erweitert werden. Dazu greifen wir tief in die Trickkiste:

```

21 \makeatletter
22 \xapptocmd{\appendix}{%
23   \listofappendices
24   \xpatchcmd{\addtocentrydefault}{\{toc\}}{\{atoc\}}{\PaTchFaileD}%
25   \providetocentry{\Hy@writebookmark}[5]}{%
26   \xapptocmd{\addtocentrydefault}{%
27     \Hy@writebookmark{\curname the#1\endcsname}%
28       {#3}%
29       {\@currentHref}%
30       {\the\numexpr\curname toplevel@#1\endcsname+1\relax}%
31       {toc}%
32   }}{\PaTchFaileD}%
33 }}{\PaTchFaileD}
34 \makeatother

```

Es wird dabei die gleiche Anweisung `\Hy@writebookmark` genutzt, die auch `hyperref` für seinen Patch von `\addcontentsline` verwendet. Da `hyperref` diese Anweisung nur aufruft, wenn in die `toc`-Datei geschrieben wird, muss sie hier für die `atoc`-Datei explizit ausgeführt werden. Das vierte Argument gibt dabei die Gliederungsebene an. Damit eine Einordnung unterhalb von »Anhang« erfolgt, muss der normale Wert um eins erhöht werden. Abbildung 2 zeigt die so erweiterten Lesezeichen.

Wer stattdessen will, dass die Einträge auf denselben Ebenen erfolgen wie im Hauptteil des Dokuments, ersetzt

```

30 {\the\numexpr\curname toplevel@#1\endcsname+1\relax}%

```

einfach durch

```

30 {\curname toplevel@#1\endcsname}%

```

Den vollständigen Code des Beispiels finden Sie in Listing 1. Abbildung 3 zeigt dessen Inhaltsverzeichnis und das Anhangsverzeichnis nebeneinander.

Listing 1: Ein Beispiel mit Anhangsverzeichnis

```

7 \documentclass{scrbook}
8 \usepackage[ngerman]{babel}
9 \usepackage{fontspec}
10 \usepackage{mwe}
11 \usepackage{xpatch}
12 \usepackage[colorlinks]{hyperref}
13
14 \DeclareNewTOC[%
15   owner=toc,
16   types=appendices,
17   listname=\appendixname
18 ]{atoc}
19 \setuptoc{atoc}{totoc}
20
21 \makeatletter
22 \xapptocmd{\appendix}{%
23   \listofappendices
24   \xpatchcmd{\addtocentrydefault}{\{toc\}}{\{atoc\}}{\PaTchFaILED}%
25   \providecommand*{\Hy@writebookmark}[5]{}%
26   \xapptocmd{\addtocentrydefault}{%
27     \Hy@writebookmark{\csname the#1\endcsname}%
28       {\#3}%
29       {\@currentHref}%
30       {\the\numexpr\csname toplevel@#1\endcsname+1\relax}%
31       {toc}%
32   }}{\PaTchFaILED}%
33 }{\PaTchFaILED}
34 \makeatother
35
36 \begin{document}
37 \tableofcontents
38 \blinddocument
39 \appendix
40 \blinddocument
41 \end{document}

```

Ergänzende Hinweise

Die Verwendung der internen `hyperref`-Anweisung `\Hy@writebookmark` ist natürlich ein etwas unsauberer Hack. Man hätte stattdessen auch `\pdfbookmark` oder dessen Erweiterung mit Hilfe des Pakets `bookmark` (siehe [7]) verwenden können. Um jedoch den aktuellen Ankernamen zu erhalten, wäre auch in diesen Fällen die Verwendung der internen Anweisung `\@currentHref` notwendig gewesen. Solange selbige nicht offiziell für Paketautoren dokumentiert ist, kommt man also ohnehin nicht ohne Hack aus.

Werden im Anhang Inhaltsverzeichniseinträge an KOMA-Script vorbei vorgenommen, so schlägt die hier gezeigte Lösung fehl und die Einträge landen wieder im

Inhaltsverzeichnis		Anhang	
1. Überschrift auf Ebene 0 (chapter)	3	A. Überschrift auf Ebene 0 (chapter)	9
1.1. Überschrift auf Ebene 1 (section)	3	A.1. Überschrift auf Ebene 1 (section)	9
1.1.1. Überschrift auf Ebene 2 (subsection)	3	A.1.1. Überschrift auf Ebene 2 (subsection)	9
1.2. Liste	4	A.2. Liste	10
1.2.1. Beispiel einer Liste (Roman)	4	A.2.1. Beispiel einer Liste (Roman)	10
1.2.2. Beispiel einer Liste (numerate)	5	A.2.2. Beispiel einer Liste (numerate)	11
1.2.3. Beispiel einer Liste (description)	5	A.2.3. Beispiel einer Liste (description)	11
Anhang	7		

Abbildung 3: Inhaltsverzeichnis und Anhangsverzeichnis aus dem Beispieldokument mit Listing 1

Inhaltsverzeichnis statt im Anhangsverzeichnis. Dasselbe gilt, wenn beispielsweise `\addchaptertocentry` ohne die Verwendung von `\addtocentrydefault` undefiniert wird. In einem solchen Fall muss `\appendix` um entsprechende Patches erweitert werden. Es kann deshalb diskutiert werden, ob `\addtocentrydefault` tatsächlich der richtige Ansatzpunkt ist oder nicht besser direkt `\addtocontents` modifiziert werden sollte. Das ist tatsächlich ebenfalls möglich. Man müsste dann jedoch zunächst prüfen, ob überhaupt in die toc-Datei geschrieben werden soll und nur in diesem Fall stattdessen die atoc-Datei verwenden. Ein solcher Code könnte beispielsweise so aussehen:

```
\let\mainaddtocontents\addtocontents
\renewcommand{\addtocontents}[1]{%
  \ifstr{#1}{toc}{\mainaddtocontents{atoc}}{\mainaddtocontents{#1}}%
}%
```

Der Code könnte übrigens nicht direkt per `xpatch` in `\appendix` eingefügt werden. Stattdessen wäre eine Anweisung zu definieren, die genau diesen Code ausführt. Die

zusätzliche Anweisung könnte dann in der Modifizierung von `\appendix` verwendet werden.

Während die Lösung von [5] auch den Fall vorsieht, dass ein Anhang wieder endet und danach die Einträge wieder im normalen Inhaltsverzeichnis vorgenommen werden sollen, ist das bei der Lösung aus diesem Artikel bisher nicht vorgesehen. Aber auch dafür gibt es natürlich Möglichkeiten, indem man beispielsweise den Patch von `\addtocentrydefault` wieder rückgängig macht.

Einige Anwender, die mit KOMA-Script näher vertraut sind, werden sich vielleicht fragen, warum ich `\appendix` modifiziere, statt `\appendixmore` zu definieren. Man könnte das tatsächlich tun. Da aber `\appendixmore` abhängig von der Option `appendixprefix` bereits definiert sein kann, müsste man letztlich auch für dessen Erweiterung mit `xpatch` arbeiten und dabei zusätzlich den Fall behandeln, dass die Anweisung noch nicht definiert ist. Deshalb habe ich den einfacheren Weg gewählt und die immer vorhandene Anweisung `\appendix` erweitert.

Zusammenfassung

Es wurde gezeigt, wie man mit KOMA-Script-Mitteln für die bekannten KOMA-Script-Klassen ein Anhangsverzeichnis ergänzen kann. Dabei wurde auch auf die Problematik der Lesezeichen mit `hyperref` eingegangen. Ergänzend wurden alternative Implementierungen und Möglichkeiten der Abwandlung erwähnt.

Literatur und Software

- [1] Jean-François Burnol. *The etoc package*. Apr. 2015. CTAN:macros/latex/contrib/etoc (besucht am 16. 06. 2015).
- [2] Community. *The KOMA-Script Documentation Project*. URL: <http://www.komascript.de> (besucht am 16. 06. 2015).
- [3] Enrico Gregorio. *The xpatch package. Extending etoolbox patching command*. Okt. 2012. CTAN:macros/latex/contrib/xpatch (besucht am 19. 06. 2015).
- [4] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für L^AT_EX 2_ε*. 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [5] Markus Kohm. *Anhangsverzeichnis verallgemeinert mit tocstyle*. Foren-Antwort. Aug. 2010. URL: <http://www.komascript.de/comment/3447#comment-3447> (besucht am 16. 06. 2015).
- [6] Markus Kohm. *KOMA-Script ein wandelbares L^AT_EX 2_ε-Paket*. Apr. 2015. CTAN:macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 01. 05. 2015).

- [7] Heiko Oberdiek. *The bookmark package*. Dez. 2011. CTAN:macros/latex/contrib/oberdiek (besucht am 19.06.2015).
- [8] Sebastian Rahtz und Heiko Oberdiek. *Hypertext marks in L^AT_EX: a manual for hyperref*. Nov. 2012. CTAN:macros/latex/contrib/hyperref (besucht am 19.06.2015).

Beschränkung von chapteratlists=entry auf Kapitel mit Verzeichniseinträgen¹

Markus Kohm

Es geht darum, dass ein Anwender die Kapiteleinträge auch in den Gleitumgebungsverzeichnissen – namentlich dem Abbildungsverzeichnis – haben will. Allerdings soll dies auf die Kapiteleinträge von Kapiteln beschränkt bleiben, die auch mindestens eine entsprechende Gleitumgebung – namentlich eine Abbildung – haben. Die Frage ist dabei, ob man diese Anforderung manuell oder sogar automatisch umsetzen kann.

Automatische Lösung

Da eine manuelle Lösung oft fehleranfällig ist, stellt sich die Frage, wie man das ganze automatisch machen könnte. Dazu müsste man zunächst einmal bei jedem Kapitel wissen, ob es eine entsprechende nummerierte Gleitumgebung hat. Ich klinke mich dazu in die Anweisung `\addcontentsline` ein. Wird darüber ein Eintrag in das entsprechende Gleitumgebungsverzeichnis geschrieben, ist klar, dass auch der Kapiteleintrag geschrieben werden muss. Diese Tatsache schreibe ich einfach in der Form `\chapterhas{Kapitelnummer}{Dateiendung}` in die aux-Datei:

```

1 \let\chapterhas@original@addcontentsline\addcontentsline
2 \renewcommand*{\addcontentsline}[1]{%
3   \immediate\write\@auxout{\string\chapterhas{\thechapter}{#1}}%
4   \chapterhas@original@addcontentsline{#1}%
5 }

```

Jetzt müssen die `\chapterhas`-Anweisungen auch noch beim Lesen der aux-Datei ausgewertet werden. Ich definiere in diesem Fall einfach eine Art Markeranweisung `\chapterhas@Kapitelnummer@Dateiendung`:

¹ Der Beitrag erschien zuerst auf <http://www.komascript.de/comment/5070#comment-5070>

```

1 \newcommand*{\chapterhas}[2]{%
2   \global\@namedef{chapterhas@#1@#2}{true}%
3 }

```

Später kann dann geprüft werden, ob die entsprechende Anweisung definiert ist. `\global` ist hier notwendig, weil alle Definitionen aus der `aux`-Datei nur lokal innerhalb einer Gruppe wirksam sind. Jetzt muss noch die Erzeugung des Kapiteleintrags in das jeweilige Gleitumgebungsverzeichnis geregelt werden. Das passiert bei KOMA-Script über die Anweisung `\addchaptertocentry`. In die Originaldefinition

```

1 \newcommand*{\addchaptertocentry}[2]{%
2   \addtocentrydefault{chapter}{#1}{#2}%
3   \if@chaptertolists
4     \doforeachtocfile{%
5       \iftocfeature{\@currentt}{chapteratlist}{%
6         \addxcontentsline{\@currentt}{chapteratlist}[{#1}]{#2}%
7       }{}%
8     }%
9     \@ifundefined{float@addtolists}{}{\scr@float@addtolists@warning}%
10  \fi
11 }

```

muss nun noch ein Test auf die Markeranweisung eingefügt werden:

```

1 \renewcommand*{\addchaptertocentry}[2]{%
2   \addtocentrydefault{chapter}{#1}{#2}%
3   \if@chaptertolists
4     \doforeachtocfile{%
5       \iftocfeature{\@currentt}{chapteratlist}{%
6         \ifundefinedorrelax{chapterhas@thechapter @\@currentt}{%
7           }{}%
8         \addxcontentsline{\@currentt}{chapteratlist}[{#1}]{#2}%
9       }{}%
10    }{}%
11  }%
12  \@ifundefined{float@addtolists}{}{\scr@float@addtolists@warning}%
13  \fi
14 }

```

Wie man sieht, ist die Änderung nicht sehr groß, die man auch mit Hilfe des Paketes `xpatch` »hineinpatchen« könnte. [1] Das folgende Listing zeigt ein Gesamtbeispiel:

```

1 \documentclass[chapteratlists=entry,DIV=13,ngerman]{scrbook}
2 \usepackage{libertine}
3 \usepackage{babel}
4
5 \makeatletter
6 \let\chapterhas@original@addcontentsline\addcontentsline
7 \renewcommand*{\addcontentsline}[1]{%
8   \immediate\write\@auxout{\string\chapterhas{\thechapter}{#1}}%
9   \chapterhas@original@addcontentsline{#1}%

```

```

10 }
11 \newcommand*{\chapterhas}[2]{%
12   \global\@namedef{chapterhas@#1@#2}{true}%
13 }
14 \renewcommand*{\addchaptertocentry}[2]{%
15   \addtocentrydefault{chapter}{#1}{#2}%
16   \if@chaptertolists
17     \doforeachtocfile{%
18       \iftocfeature{\@current}{chapteratlist}{%
19         \ifundefinedorrelax{chapterhas@thechapter \@current}{%
20           }{%
21             \addxcontentsline{\@current}{chapteratlist}[\@#1]{\@#2}%
22           }%
23         }{}%
24       }%
25       \@ifundefined{float@addtolists}{}{\scr@float@addtolists@warning}%
26     \fi
27   }
28 \makeatother
29
30 \usepackage{mwe}
31 \begin{document}
32 \tableofcontents
33 \listoffigures
34 \listoftables
35 \Blinddocument
36 \begin{figure}
37   \centering
38   \includegraphics{example-image}
39   \caption{Testfigure}
40   \label{fig:test}
41 \end{figure}
42 \Blinddocument
43 \begin{table}
44   \centering
45   \begin{tabular}{ll}
46     test & table
47   \end{tabular}
48   \caption{Testtable}
49   \label{tab:test}
50 \end{table}
51 \Blinddocument
52 \Blinddocument
53 \begin{figure}
54   \centering
55   \includegraphics{example-image-a}
56   \caption{Testfigure A}
57   \label{fig:testA}
58 \end{figure}
59 \Blinddocument
60 \begin{table}
61   \centering

```

```

62 \begin{tabular}{ll}
63   test & table
64 \end{tabular}
65 \caption{Testtable A}
66 \label{tab:testA}
67 \end{table}
68
69 \end{document}

```

Nach dem zweiten L^AT_EX-Lauf enthalten das Abbildungs- und Tabellenverzeichnis keine Kapiteleinträge; nach dem dritten Lauf werden auch diese angezeigt (siehe Abbildung 1).

Inhaltsverzeichnis	Abbildungsverzeichnis	Tabellenverzeichnis
1 Überschrift auf Ebene 0 (Kapitel) 7	1 Überschrift auf Ebene 0 (Kapitel) 7	2 Überschrift auf Ebene 0 (Kapitel) 23
1.1 Überschrift auf Ebene 1 (Unterkapitel) 8	1.1 Teilfiguren 26	2.1 Tabelle 27
1.2 Liste 15	1.2 Beispiel einer Liste (numerisch) 15	2.2 Überschrift auf Ebene 0 (Kapitel) 71
1.2.1 Beispiel einer Liste (numerisch) 15	1.2.1 Teilfiguren 26	2.2.1 Tabelle A 28
1.2.2 Beispiel einer Liste (alphabetisch) 16		
2 Überschrift auf Ebene 0 (Kapitel) 23		
2.1 Überschrift auf Ebene 1 (Unterkapitel) 26		
2.1.1 Überschrift auf Ebene 2 (Unterunterkapitel) 26		
2.1.2 Liste 27		
2.1.3 Beispiel einer Liste (numerisch) 27		
2.1.4 Beispiel einer Liste (alphabetisch) 28		
3 Überschrift auf Ebene 0 (Kapitel) 30		
3.1 Überschrift auf Ebene 1 (Unterkapitel) 30		
3.1.1 Überschrift auf Ebene 2 (Unterunterkapitel) 30		
3.1.2 Liste 30		
3.1.3 Beispiel einer Liste (numerisch) 30		
3.1.4 Beispiel einer Liste (alphabetisch) 30		
4 Überschrift auf Ebene 0 (Kapitel) 36		
4.1 Überschrift auf Ebene 1 (Unterkapitel) 36		
4.1.1 Überschrift auf Ebene 2 (Unterunterkapitel) 36		
4.1.2 Liste 36		
4.1.3 Beispiel einer Liste (numerisch) 36		
4.1.4 Beispiel einer Liste (alphabetisch) 36		
5 Überschrift auf Ebene 0 (Kapitel) 71		
5.1 Überschrift auf Ebene 1 (Unterkapitel) 71		
5.1.1 Überschrift auf Ebene 2 (Unterunterkapitel) 71		
5.1.2 Liste 71		
5.1.3 Beispiel einer Liste (numerisch) 71		
5.1.4 Beispiel einer Liste (alphabetisch) 71		

Abbildung 1: Kapiteleinträge in der Liste der Abbildungen und Tabellen.

Manuelle Lösung

In diesem Fall setzt man ohne `chapteratlists=entry` nach der `\chapter`-Anweisung betroffener Kapitel einfach ein

```

1 \addxcontentsline{lof}{chapteratlist}{\thechapter}{Text, der im Verzeichnis
  ↳ angezeigt werden soll}

```

Unschön ist dabei, dass man den Text für den Eintrag selbst wiederholen muss. Daher könnte man auf die Idee kommen, die entsprechende Überschrift als

```

1 \KOMAOption{chapteratlists}{entry}
2 \chapter{Überschrift}
3 \KOMAOptions{chapteratlists}{Opt}

```

zu setzen. Das funktioniert, solange man nur ein Gleitumgebungsverzeichnis hat. Sobald man aber beispielsweise sowohl ein Abbildungs- als auch ein Tabellen-

verzeichnis hat, geht es nicht mehr, weil die Option sozusagen eine »Alles-oder-Nichts«-Option ist, sich also immer auf alle Verzeichnisse bezieht. Da das Ganze jedoch über TOC-Eigenschaften (siehe `tocbasic`-Kapitel in der Anleitung oder im Buch) gelöst ist, kann man tatsächlich die entsprechende Eigenschaft lokal setzen:

```

1 \documentclass[listof=chapterentry]{scrbook}% Feature generell einschalten ...
2 \makeatletter
3 % ... aber für alle Dateiendungen erst einmal deaktivieren
4 \doforeachtocfile{%
5   \unsettoc{@currentx}{chapteratlist}%
6 }
7 \makeatother
8 \usepackage{mwe}
9 \begin{document}
10 \tableofcontents
11 \listoffigures
12 \listoftables
13 \setuptoc{lof}{chapteratlist}% Kapitel mit Abbildungen
14 \chapter{First Chapter}
15 \unsettoc{lof}{chapteratlist}
16 \blindtext
17 \begin{figure}
18   \centering
19   \includegraphics{example-image}
20   \caption{Testfigure}
21   \label{fig:test}
22 \end{figure}
23
24 \setuptoc{lot}{chapteratlist}% Kapitel mit Tabellen
25 \chapter{Second Chapter}
26 \unsettoc{lot}{chapteratlist}
27 \blindtext
28 \begin{table}
29   \centering
30   \begin{tabular}{ll}
31     test & table
32   \end{tabular}
33   \caption{Testtable}
34   \label{tab:test}
35 \end{table}
36 \Blinddocument
37
38 \setuptoc{lof}{chapteratlist}
39 \chapter{Fourth Chapter}
40 \unsettoc{lof}{chapteratlist}
41 \begin{figure}
42   \centering
43   \includegraphics{example-image-a}
44   \caption{Testfigure A}
45   \label{fig:testA}
46 \end{figure}
47

```

```

48 \setuptoc{lot}{chapteratlist}
49 \chapter{Fifth Chapter}
50 \unsettoc{lot}{chapteratlist}
51 \begin{table}
52   \centering
53   \begin{tabular}{ll}
54     test & table
55   \end{tabular}
56   \caption{Testtable A}
57   \label{tab:testA}
58 \end{table}
59
60 \end{document}

```

Wie man sieht, muss man dabei auch die Option `listof=chapterentry` verwenden, danach jedoch erst einmal für alle Endungen die TOC-Eigenschaft `chapteratlist` wieder wegnehmen. Man setzt sie dann nur individuell bei den Kapiteln, bei denen man sie haben will. Natürlich ist es anfällig für Fehler, wenn man in einem Kapitel Gleitumgebungen hinzufügt oder entfernt.

Literatur und Software

- [1] Enrico Gregorio. *The xpatch package. Extending etoolbox patching command*. Okt. 2012. CTAN:macros/latex/contrib/xpatch (besucht am 19. 06. 2015).
- [2] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für L^AT_EX 2_ε*. 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [3] Markus Kohm. *KOMA-Script ein wandelbares L^AT_EX 2_ε-Paket*. Apr. 2015. CTAN:macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 01. 05. 2015).

Rechnungen erstellen

Rainer-M. Fritsch

Mehr Flexibilität mit den Variablen der Briefklasse `scr1ttr2`

Im World Wide Web finden sich verschiedene Ansätze zum Erstellen von Rechnungen mit $\text{\LaTeX}$, beispielsweise `rechnung.sty` [1] unter Verwendung von KOMA-Script, `invoice.sty` [4] oder `scr1ttr2.cls` [3].

Mit der Dokumentenklasse `scr1ttr2` von KOMA-Script hat man jedoch ein wunderbares Werkzeug für professionell gestaltete Briefe in der Hand, das einem das Drucken auf speziellen Briefbögen erspart. Gerade mit der *letter-class-option-Datei* `asymTypB.lco` [2, ab S. 613] lässt sich ein sehr schöner Briefbogen gestalten, mit linker Infospalte und einem Logo über die gesamte Briefbreite. Mit diesem Briefbogen soll nun auch eine Rechnung erstellt werden. Im heutigen Geschäftsverkehr läuft vieles auch per E-Mail, so dass auch ein professionell gestaltetes PDF mit einer Rechnung generiert werden soll.¹

Als Freiberufler rechne ich eher Honorare ab, muss jedoch auch gelegentlich Fremdleistungen Dritter oder technische Produkte mit in die Rechnung aufnehmen können. Ich habe nach einer flexiblen Lösung für Rechnungen gesucht, die standardisierte Prozesse über Variablen abbilden kann, jedoch auch die Abrechnung zusätzlicher Leistungen ermöglicht und das Ganze automatisch berechnet – ohne zwischendurch ein Tabellenkalkulationsprogramm benutzen zu müssen. All diese Voraussetzungen erfüllt KOMA-Script zusammen mit den Paketen `fp` (fixed point arithmetic for $\text{\TeX}$), `spreadtab` (für die Berechnung in den Tabellen), `numprint` (zur typografisch richtigen Ausgabe der Zahlwerte mit Tausender-Trennzeichen).

Der Quelltext

```

1 \documentclass[%
2   % asymTypB,
3   DIV=8,
4   fontsize=11pt,
5   parskip=half,
6   paper=a4,
7   enlargefirstpage=true,
8   firstfoot=false

```

¹ Auf die Ausgabe des asymmetrischen Firmenbriefbogens wird hier allerdings verzichtet.

```

9    ]{\scr\ltrim2}
10   \usepackage[utf8]{inputenc}
11   \usepackage[ngerman]{babel}
12
13   %% Signatur linksbündig setzen
14   \renewcommand*{\raggedsignature}{\raggedright}
15
16   %% Euro-Symbol rechts vom Betrag setzen
17   \usepackage[right]{eurosym}
18
19
20   %% zum Rechnen in Tabellen
21   \usepackage{fp, spreadtab, numprint}
22   %% spreadtab -- Werte müssen weiterhin mit Punkt geschrieben sein, Ausgabe
23   %→erfolgt mit Komma
24   \STsetdecimalsep{,}
25   %% spreadtab -- auf 2 Ziffern runden
26   \STautoround*{2}
27   %% numprint Typografie Tausendertrennung
28   \selectlanguage{ngerman}
29
30   %% Paket varsfromjobname von Uwe Ziegenhagen: extract information from the file-
31   %→name, einen Wert aus dem Dateinamen als fortlaufende Rechnungsnummer nutzen
32   \usepackage{varsfromjobname}
33
34   %% Variablen setzen - Diese sollten zentral in einer *.lco-Datei abgelegt werden
35   \setkomavar{fromlogo}{}
36   \setkomavar{fromname}{Schnell und Gründlich}
37   \setkomavar{fromaddress}{Musterstr. 38\\D-12345 Musterstadt}
38   \setkomavar{fromemail}{info@schnell-und-gruendlich.de}
39   \setkomavar{fromurl}{schnell-und-gruendlich.de}
40
41   \setkomavar{fromphone}[Tel.: ]{+49 30 12 34 56 78}
42   \newkomavar{mobil}
43   \setkomavar{mobil}{Mobil: 0123 45 67 89 10}
44   \setkomavar{fromfax}{+49 30 12 34 56 78}
45   \setkomavar{backaddressseparator}{ - }
46
47   \setkomavar{frombank}{Unsere Bank\\[0.15\baselineskip]
48   IBAN: DE00\,1234\,5678\,9101\,1213\,00\\[0.15\baselineskip] BIC: DEXXXXXXXX}
49
50   \newkomavar{zahlungsziel}
51   \setkomavar{zahlungsziel}{Der Rechnungsbetrag ist bei Erhalt der Rechnung ohne
52   %→Abzug fällig.\\Vielen Dank für Ihren Auftrag.\\[0.5\baselineskip]Mit
53   %→freundlichen Grüßen}
54
55   \setkomavar{signature}{Max Gründlich}
56   \newkomavar{stundensatz}
57   \setkomavar{stundensatz}{120.00}
58   %% KOMA-Variable "stundensatz" in einen numerischen Wert speichern, der von
59   %→spreadtab verarbeitet werden kann
60   \usekomavar[\def\nstundensatz]{stundensatz}

```

```

56 \newkomavar{mwst}
57 \setkomavar{mwst}{19}
58 %% KOMA-Variable "mwst" in einen numerischen Wert speichern, der von spreadtab
59   ↳ verarbeitet werden kann
60 \usekomavar[\def\nmwst]{mwst}
61
62
63 %% zweites Glied (getrennt durch Bindestriche) aus dem Dateinamen als
64   ↳ fortlaufende Rechnungsnummer nutzen
65 \setkomavar{invoice}{\gettwofromjobname-2015}
66
67 \setkomavar{date}{30.06.2015}
68
69 %% für Honorarabrechnungen
70 \newkomavar{stunden}
71 \setkomavar{stunden}{10}
72 %% KOMA-Variable "stunden" in einen numerischen Wert speichern, der von spreadtab
73   ↳ verarbeitet werden kann
74 \usekomavar[\def\nstunden]{stunden}
75
76 %% steuerliche Pflichtangabe Leistungszeitraum
77 \newkomavar{leistung}
78 \setkomavar{leistung}{im Monat Juni 2015}
79
80 %% Rechnungsempfänger
81 \newkomavar{adresse}
82 \setkomavar{adresse}{Firma\\Rechnungsempfänger\\Rechnungstr. 48\\[1.1]
83   ↳ baselineskip}12345 Rechnungsort}
84
85 \setkomavar{signature}{}
86
87 \begin{document}
88 \begin{letter}{\usekomavar{adresse}}
89 \opening{}
90
91
92 für die \usekomavar{leistung} erbrachten Leistungen berechnen wir:
93
94 \begin{spreadtab}[\STsavecell\gesamtsumme{c5}]{\tabular}{p{55mm}p{41mm}r}}
95   %% Zeile 1
96   %% Zelle a1
97   :={\nstunden} Stunden & %
98   %% Zelle b1
99   @ Dies und Das & %
100  %% Zelle c1
101  \EUR{\numprint{:=a1*\nstundensatz}}\\
102  %% Zeile 2
103  %% Zelle a2
104  :={2} Stück Jenes & %

```

```

104 %% Zeile b2
105 je \EUR{\numprint{:=30.00}}& %
106 %% Zeile c2
107 \EUR{\numprint{:=a2*b2}}\\
108 %% Zeile 3
109 %% Zeile a3
110 @Druckkosten &
111 %% Zeile b3
112 @ 1\,000 Flyer &
113 %% Zeile c3
114 \EUR{\numprint{:=1200.00}}\\
115 %% Zeile 3
116 %% Zeile a3
117 @ & %
118 %% Zeile b3
119 @zuzüglich \nmwst\,% USt. & %
120 %% Zeile c3
121 \EUR{\numprint{:=\nmwst*(c1+c2+c3)/100}}\\
122 %% Zeile 5
123 %% Zeile a5
124 @ & %
125 %% Zeile b5
126 @Summe & %
127 %% Zeile c5
128 \EUR{\numprint{:=c1+c2+c3+c4}}\\
129 \end{spreadtab}
130
131 \vspace{17pt}
132 Bitte überweisen Sie den Rechnungsbetrag von \EUR{\numprint{\gesamtsumme}} unter
133 ↳Angabe der Rechnungsnummer \usekomavar{invoice} auf das Konto:
134
135 \usekomavar{fromname}\\[0\baselineskip]
136 \usekomavar{frombank}
137 \closing{\usekomavar{zahlungsziel}}
138 \end{letter}%
139 \end{document}

```

Ein typischer Flüchtigkeitsfehler innerhalb der Umgebung `spreadtab` ist, bei Dezimalzahlen das Komma statt des Punktes zu benutzen (siehe auch Zeile 20 des Quelltextes).

`spreadtab` verlangt zwingend einen leeren numerischen Wert mit `:={}` oder alternativ mit dem Zeichen »@« (Zeile 94 usw.). In Zeile 89 wird die Zelle, die die Gesamtsumme enthält, in eine Variable gespeichert, die auch außerhalb der Umgebung `spreadtab` genutzt werden kann, um beispielsweise den Rechnungsbetrag noch einmal mit der Rechnungsnummer ausgeben zu können.

KOMA-Script bietet mit seiner Briefklasse und den vielfältigen Möglichkeiten Variablen anzulegen, sehr flexible Möglichkeiten, $\LaTeX$ im Büroalltag einzusetzen.

Wer auch Angebote schreiben muss und dabei nicht auf `\section` und `\subsection` verzichten möchte, könnte diesen Quelltext in seine *.lco-Datei übernehmen².

```
\newcounter{section}
\newcounter{subsection}

\providecommand{\section}[1]{%
  \par                                % neuer Absatz
  \refstepcounter{section}%          % Zähler raufsetzen
  \setcounter{subsection}{0}%        % ggf. Zähler für Unterabschnitte
  ↪ zurücksetzen
  \vspace{1em}                        % eine m-Breite Abstand nach oben
  {\noindent                          % nicht einziehen
   \large                             % das folgende fett
   \arabic{section}%                 % die Section-Zahl in arab. Zahlen
   \hspace{1em}%                     % eine m-Breite seitlichen Abstand
   #1                                % Den Text der Überschrift
  \par}                              % neuer Absatz
}

\providecommand{\subsection}[1]{%
  \par                                % neuer Absatz
  \refstepcounter{subsection}%        % Zähler raufsetzen
  \vspace{1em}%                       % eine m-Breite Abstand nach oben
  {\noindent                          % nicht einziehen
   \large                             % das folgende fett
   \arabic{section}.\arabic{subsection}%
   \hspace{1em}%                     % eine m-Breite seitlichen Abstand
   #1                                % Den Text der Überschrift
  \par}                              % neuer Absatz
}
```

Literatur und Software

- [1] M. G. Berberich. *rechnung.sty*. Nov. 1998. URL: <https://www.forwiss.uni-passau.de/~berberic/TeX/Rechnung/> (besucht am 04. 07. 2015).
- [2] Markus Kohm. *KOMA-Script. Eine Sammlung von Klassen und Paketen für L^AT_EX 2_ε*. 5. Aufl. Lehmanns Media, Apr. 2014. ISBN: 978-3-86541-613-1.
- [3] meinnoteblog. *L^AT_EX-Vorlagen für Briefe und Rechnung*. Nov. 2010. URL: <https://meinnoteblog.wordpress.com/2010/11/12/latex-vorlagen-fur-briefe-und-rechnung/> (besucht am 04. 07. 2015).
- [4] Uwe Ziegenhagen. *Invoice*. Juni 2010. URL: <http://uweziegenhagen.de/latex/documents/invoice/invoiceSample.tex> (besucht am 04. 07. 2015).

² Siehe auch <http://www.mrunix.de/forums/archive/index.php/t-39984.html>

Briefkopien mit scrlltr2 leicht gemacht

Rolf Niepraschk

Nicht selten möchte man von einem mit $\LaTeX$ gesetzten Brief nach dem Versenden eine Kopie behalten. Es existieren dafür etliche Wege. Der einfachste wäre, den Brief zweimal auszudrucken. Etwas mehr Aufwand ist nötig, wenn man die Kopie auch noch als solche kennzeichnen möchte. Im Folgenden wird ein Weg gezeigt, wie mit $\TeX$ selbst zusätzlich gekennzeichnete Kopien aller Seiten eines Briefes erzeugt werden können.

Man kann die Art, wie Briefe mit der KOMA-Script-Briefklasse »scrlltr2« gesetzt werden, mittels so genannter »letter class option«-Dateien (Endung .lco) beeinflussen (siehe dazu [1], S. 212). Für den Endanwender bietet es sich beispielsweise an, Angaben für einen bestimmten Absender oder für ein bestimmtes Briefformat in einer solchen Datei zu hinterlegen. Gibt man den Namen dieser Datei als Klassenoption an, werden diese Angaben wirksam, bevor der Brief beginnt. In ähnlicher Weise kann man innerhalb einer lco-Datei auch $\TeX$ -Code unterbringen, der anderweitig den Ablauf beim Setzen des Briefdokumentes beeinflusst. Der für die Briefseitenkopien benutzte Code befindet sich in einer solchen lco-Datei.

Gibt man beim Laden der Klasse scrlltr2 die Option copy an, wird die Datei copy.lco geladen. Sie kann von der in [2] angegebenen Adresse beschafft werden und muss in ein von $\TeX$ durchsuchbares Verzeichnis installiert werden. Der folgende $\LaTeX$ -Code erzeugt einen Brief und automatisch Kopien aller Seiten:

```
\documentclass[copy,ngerman,fontsize=11pt,enlargefirstpage]{scrlltr2}
\usepackage[utf8]{inputenc}
\usepackage{babel}
\usepackage{blindtext}
\setkomavar{fromname}{Peter Silie}
\setkomavar{fromaddress}{Hinter dem Tal 2\\ 54321 Musterheim}
\setkomavar{date}{26.\,5.\,2015}

\begin{document}
\setkomavar{subject}{Ihr Schreiben vom 11.\,5.\,2015}

\begin{letter}{%
Ellen Bogen \\
Dorfstraße 1\\
12345 Musterhausen
}
\opening{Sehr geehrte Frau Bogen,}
\blindtext[2]
\closing{Mit freundlichen Grüßen}
\end{letter}
\end{document}
```

Die so entstehende Briefseite und ihre Kopie ist in Abbildung 1 zu sehen.

Der grau hinterlegte Hintergrundtext kann durch Überschreiben der Variablen `copy` in der folgenden Weise geändert werden:

```
\setkomavar*{copy}{Zweitschrift}
```

Möglich ist auch, sie leer zu definieren, was die Kennzeichnung ganz unterdrückt.

Das Makro `\LetterCopyMarker` bestimmt die grundsätzliche Art, wie der Hintergrundtext ausgegeben wird. Standardmäßig ist es folgendermaßen definiert:

```
\newcommand*\LetterCopyMarker{%
  \put(0.5\paperwidth,-0.5\paperheight){%
    \makebox(0,0)[c]{\resizebox{\textwidth}{!}{\rotatebox{45}{%
      \textsf{\textbf{\color{lightgray}\strut\usekomavar*{copy}}}}}%
    }%
  }
}
```

Zum prinzipiellen Ablauf beim Erzeugen und Ausgeben der Seitenkopien: Mit Hilfe des Makros `\AtBeginShipout` aus dem Paket `atbegshi` (siehe [3]) wird zu dem Zeitpunkt, da die aktuelle Briefseite komplett von $\TeX$ gesetzt ist – innerhalb der so genannten Output-Routine – jeweils ein neues Boxregister reserviert, sofern es nicht bereits existiert. Sein Name enthält die aktuelle Seitenzahl, sodass später gezielt auf seinen Inhalt zugegriffen werden kann. Im nächsten Schritt wird das Register mit dem Inhalt der gerade fertiggestellten Seite gefüllt (enthalten in der Box `\AtBeginShipoutBox`). Dies geschieht mit allen folgenden Seiten so lange, bis eine neue Umgebung `letter` beginnt oder das Ende des Dokuments erreicht ist. In beiden Fällen werden alle bis dahin aufgesammelten Seiten fortlaufend ausgegeben, d. h. in die pdf-Datei geschrieben. Die dazu benutzte Box wird aus dem Hintergrundtext sowie dem Inhalt einer der früher gefüllten Kopie-Boxen zusammengesetzt. Bei einem zweihundertseitigen Brief mit nicht übermäßig komplexem Seiteninhalt hat $\TeX$ keine Probleme, die Kopien im Speicher zu halten und später auszugeben.

Literatur und Software

- [1] Markus Kohm. *KOMA-Script*. Mai 2015. CTAN:macros/latex/contrib/koma-script/doc/scrguide.pdf (besucht am 09. 07. 2015).
- [2] Rolf Niepraschk. *scrlltr2copy*. 2015. URL: <https://github.com/rolfn/scrlltr2Copy> (besucht am 09. 07. 2015).
- [3] Heiko Oberdiek. *The atbegshi package*. Okt. 2011. CTAN:macros/latex/contrib/oberdiek/atbegshi.pdf (besucht am 09. 07. 2015).

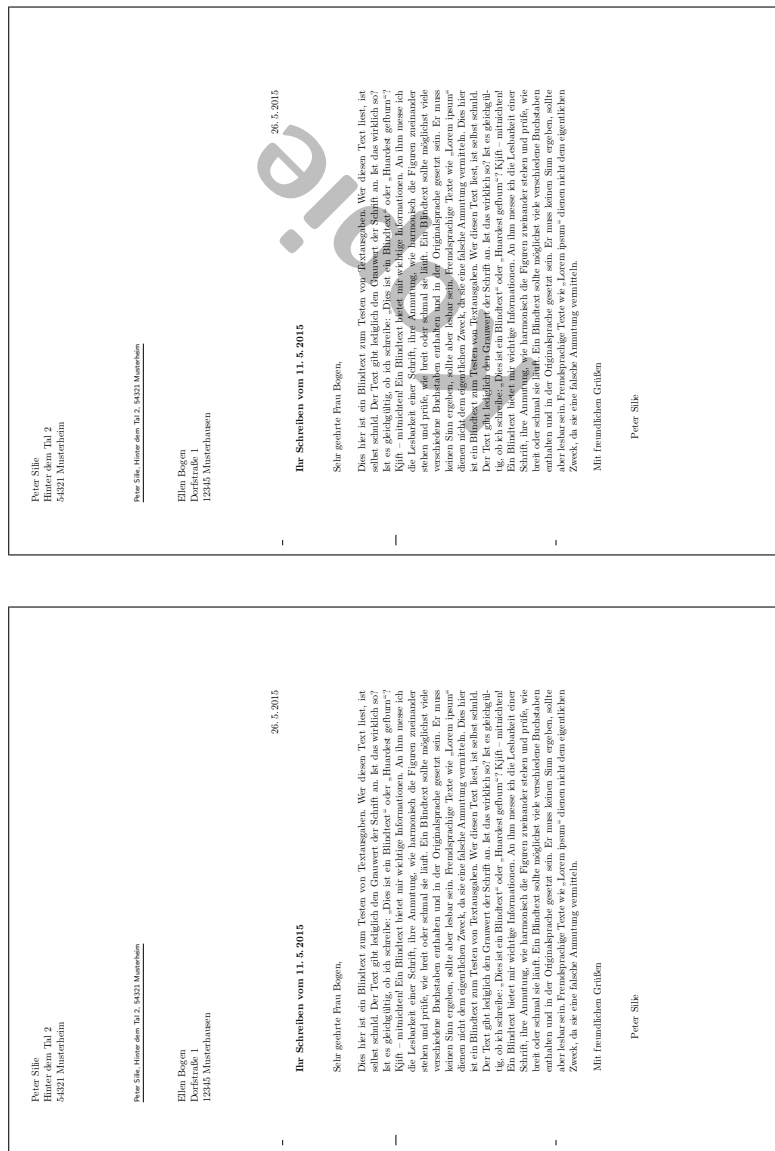


Abbildung 1: Briefseite und deren Kopie

Von fremden Bühnen

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss vor. Bloße Updates können auf der moderierten *CTAN-ann*-Mailingliste verfolgt werden, die auch auf Twitter als @ctanannounce verfügbar ist.

bewerbung von *André Hilbig* ist das erste $\text{\LaTeX}$ -Paket, das Makros zum Anfertigen kompletter Bewerbungsmappen mitsamt der ergänzenden Dokumentation bereitstellt, und alle Bewerbungsversuche können am Ende in einer *argetabelle* für den regelmäßigen Bericht an das Jobcenter dokumentiert werden. Die Daten werden aus einer einfachen CSV-Datei übernommen.

CTAN:macros/latex/contrib/bewerbung

cloze von *Josef Friedrich* ist ein $\text{\LuaTeX}$ -Paket zum Setzen von Lückentexten, wie sie in Fragebögen oder in Übungsaufgaben für die Schule verwendet werden.

CTAN:macros/luatex/latex/cloze

quran von *Seiied-Mohammad-Javad Razavian* stellt Makros zum Setzen des Korans bereit.

CTAN:macros/xetex/latex/quran

comicneue von *Silke Hofstra* enthält die Schriftart *Comic Neue* in den Formaten OpenType und Type 1 mitsamt der dazu passenden $\text{\LaTeX}$ -Unterstützung. Der Release erfolgte zum Comic-Sans-Tag am 3. Juli 2015.

CTAN:fonts/comicneue

bidihl von *Vafa Khalighi* ist ein experimentelles Paket, das Highlighting für bidirektional gesetzten Text bereitstellt.

CTAN:macros/xetex/latex/bidihl

tex4ebook von *Michal Hoftich* konvertiert $\text{\LaTeX}$ in die E-Book-Formate EPUB, MOBI und EPUB3 mit Hilfe von $\text{\TeX}$ 4ht.

CTAN:support/tex4ebook

make4ht von *Michal Hoftich* ist ein einfaches Build-System für $\text{\TeX}$ 4ht. Es enthält ein ausführbares Skript für den Aufruf des Konverters sowie eine Library, um im Einzelfall passende Workflows zu entwerfen.

CTAN:support/make4ht

br-lex von *Youssef Cherem* ist eine Klasse zum Setzen juristischer Texte nach den in Brasilien üblichen Vorgaben.

CTAN:macros/latex/contrib/br-lex

elements von *Clemens Niederberger* gibt die Eigenschaften der chemischen Elemente bis einschließlich der Ordnungszahl 112 aus.

CTAN:macros/latex/contrib/elements

semproc von *Tobias Columbus* ist eine Klasse zum Setzen von Seminar-Proceedings auf der Grundlage von KOMA-Script und etoc. Die Anpassungen umfassen unter anderem

den Befehl `\chapter` mit dem Autorennamen und Inhaltsverzeichnis für die einzelnen Kapitel.

CTAN:macros/latex/contrib/semproc

ucharcat von *David Carlisle* implementiert den $\text{\XeTeX}$ -Befehl `\ucharcat`, der gerade neu eingeführt worden ist, in Lua für $\text{\LuaTeX}$.

CTAN:macros/latex/contrib/ucharcat

rmathbr von *Denis Ryabov* wiederholt den mathematischen Operator zu Beginn der neuen Zeile, wie es in der russischen Typographie mit kyrillischen Buchstaben üblich ist.

CTAN:macros/latex/contrib/rmathbr

biblatex-opcit-booktitle von *Maïeul Rouquette* stellt die Abkürzung *op. cit.* für Bibliografien mit $\text{\BibTeX}$ für die Fälle bereit, in denen zwei unterschiedliche *Titel* aus demselben Werk zitiert werden.

CTAN:macros/latex/contrib/biblatex-contrib/biblatex-opcit-booktitle

academicons von *Diogo A. B. Fernandes* und *James Walsh* enthält den Font *Academicons* mit den Logos von 20 akademischen sozialen Netzwerken samt $\text{\TeX}$ -Unterstützung.

CTAN:fonts/academicons

autosp von *Bob Tennent* ist ein Programm zum Aufbereiten von $\text{\TeX}$ -Quellcode für den Musiknotensatz mit $\text{\MusixTeX}$.

CTAN:support/autosp

pdfbook2 von *Johannes Neumann* ist eine neue Fassung des Skripts *pdfbook* in Python für den Broschürendruck.

CTAN:support/pdfbook2

whuthesis von *Eric Zhang* ist eine Vorlage für Abschlussarbeiten an der chinesischen Universität Wuhan.

CTAN:macros/latex/contrib/whuthesis

bookdb von *Peter R. Wilson* ist ein $\text{\BibTeX}$ -Stil zum Katalogisieren der Heimbibliothek.

CTAN:biblio/bibtex/contrib/bookdb

termmenu von *Sean Allred* stellt terminalbasierte Menüs mithilfe bereit. Benötigt `expl3`.

CTAN:macros/generic/termmenu

xpiano von *Enrico Gregorio* ist eine Erweiterung zu dem Paket *piano* von *Émile Daneault*, die aufgrund einer Diskussion auf Stackexchange erarbeitet worden war.

CTAN:macros/latex/contrib/xpiano

typicons von *Arthur Vigil* ermöglicht mit dem Paket *fontspec* für $\text{\XeTeX}$ und $\text{\LuaTeX}$ den Zugriff auf die 336 Icons aus dem Font *Typicons*.

CTAN:fonts/typicons

copyedit von *CV Radhakrishnan*, *CV Rajagopal* und *SK Venkatesan* stellt Makros bereit, mit denen es möglich ist, einen englischen Text gemischt in den Varianten britisch, amerikanisch, kanadisch oder australisch korrekt zu setzen.

CTAN:macros/latex/contrib/copyedit

comment_io von *Carl Martin Henrik Larsson* ist ein Python-Skript, das Kommentierungen in $\text{\TeX}$ -Quellen einfügen und entfernen kann.

CTAN:support/comment_io

tagpair von *Andreas Nolda* dient zum Erstellen von Interlinearglossierungen.

CTAN:macros/latex/contrib/tagpair

hang von *Andreas Nolda* stellt Umgebungen für hängende Absätze und Listenelemente bereit.

CTAN:macros/latex/contrib/hang

biblatex-subseries von *Maïeul Rouquette* stellt neue Felder bereit, um Unterserien in einer Buchreihe mit $\text{\BibTeX}$ verarbeiten zu können.

CTAN:macros/latex/contrib/biblatex-contrib/biblatex-subseries

sourceserifpro von *Silke Hofstra* enthält die Schriftart *Source Serif Pro* in den Formaten OpenType und Type1 samt $\text{\LaTeX}$ -Unterstützung. CTAN:fonts/sourceserifpro

colorspace von *Javier Bezos* ist ein Paket, um Farben nach Farbräumen in PDFs zu bestimmen.

CTAN:macros/latex/contrib/colorspace

diadia von *Josef Kleber* ist ein Paket für insulinpflichtige Diabetiker, die $\text{\TeX}$ en, mit dem man ein Therapietagebuch (*diabetes diary*) führen kann – mit Tabellen und grafischen Auswertungen für Kohlehydratmenge, Insulindosis und HbA1c, Dosierungsanleitungen und Infoboxen. Bisher werden die Sprachen Deutsch und Englisch unterstützt.

CTAN:macros/latex/contrib/diadia

elocalloc von *David Carlisle* enthält die neuen *allocation macros* für $\text{\LaTeX}$ 2015-01-01.

CTAN:macros/latex/contrib/elocalloc

numending von *Dariya Kirillova* erzeugt die morphologischen Endungen von Zahlen in den drei Sprachen Russisch, Ukrainisch und Weißrussisch. Die Anleitung liegt in russischer Sprache vor.

CTAN:macros/latex/contrib/numending

gzt von *Denis Bitouzé* enthält ein Bundle von Klassen zum Setzen der Zeitschrift *La Gazette des Mathématiciens*.

CTAN:macros/latex/contrib/gzt

cjk-gs-integrate von *Norbert Preining* ist ein Skript, das die Verzeichnisse mit CJK-Schriften durchsucht und alles Nötige einrichtet, damit sie mit Ghostscript verwendet werden können.

CTAN:fonts/utilities/cjk-gs-integrate

cntperchap von *Christian Hupfer* ist ein Paket, das die Werte von bestimmten, vorher bezeichneten Zählern kapitelweise speichert, so dass man beim darauffolgenden $\text{\LaTeX}$ -Lauf wieder darauf zugreifen kann. Es funktioniert derzeit mit den Klassen book, memoir und scrbook.

CTAN:macros/latex/contrib/cntperchap

esrelation von *David Reinfurt* enthält eine Sammlung von Symbolfonts, mit denen man geordnete Paare beschreiben kann.

CTAN:fonts/esrelation

gfnotation von *Udo Wermuth* enthält Makros für plain $\text{\TeX}$ zum Setzen von Ausdrücken in Gottlob Freges Begriffsschriftnotation.

CTAN:macros/plain/contrib/gfnotation

datetime2-it-fulltext von *Claudio Duchi* stellt die Stile *it-fulltext* und *it-fulltext-twenty-four* für das Paket *datetime2* von *Nicola Talbot* bereit.

CTAN:macros/latex/contrib/datetime2-contrib/datetime2-it-fulltext

denisbdoc von *Denis Bitouzé* ist ein Paket zum Schreiben von Paket-Dokumentationen.

CTAN:macros/latex/contrib/denisbdoc

Spielplan

2015

5. 9. **Herbsttagung**

und 53. Mitgliederversammlung von DANTE e.V.
Technische Universität Graz, Österreich
<http://www.dante.de/events/Herbst2015.html>

14. 9. – 18. 9. **9th International ConT_EXt Meeting**

Ferme Equestre de Monts d'Aubrac, Nasbinals (Frankreich)
<http://meeting.contextgarden.net/2015/>

17. 10. **GuIT Meeting 2015**

Polo scientifico e tecnologico »Fabio Ferrari«, Trento (Italien)
<http://www.guitex.org/home/en/guit-meeting-2015>

7. 11. – 8. 11. **OpenRheinRuhr**

Freie Software und Netzpolitik
Rheinisches Industriemuseum (RIM) Oberhausen
<http://www.openrheinruhr.de/>



(Foto: Reinhard Kotucha – TUG 2015)

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von T_EX-Anwendern statt, die für jeden offen sind. Im WWW gibt es aktuelle Informationen unter <http://projekte.dante.de/Stammtische/WebHome>.

Aachen

Torsten Bronger,
bronger@physik.rwth-aachen.de
Gaststätte Knossos, Templergraben 28, 52062 Aachen
Zweiter Donnerstag im Monat, 19.00 Uhr

Berlin

Michael-E. Voges, Tel.: (03362) 50 18 35,
mevoges@t-online.de
Mantee – Café Restaurant, Chausseestraße 131, 10115 Berlin
Zweiter Donnerstag im Monat, 19.00 Uhr

Bielefeld

Jürgen Schwarze, Tel.: (0521) 5 57 39 06,
juergen.schwarze@bitel.net
Ferdis Pizza Pinte, Schmiedestraße 9, 33613 Bielefeld,
Zweiter Montag im Monat, 19.30 Uhr

Bremen

Winfried Neugebauer, Tel.: 0176 60 85 43 05,
tex@wphn.de
Wechselnder Ort
Erster Donnerstag im Monat, 18.30 Uhr

Dresden

Daniel Borchmann, daniel@algebra20.de, <http://tug-dd.kxqp.de>
Weltcafé Dresden, Schillingstraße 7, 01159 Dresden
ca. alle 6 Wochen, 19.30 Uhr

Erlangen

Walter Schmidt, Peter Seitz,
w.a.schmidt@gmx.net
Gaststätte »Deutsches Haus«, Luitpoldstraße 25, 91052 Erlangen
Dritter Dienstag im Monat, 19.00 Uhr

Frankfurt

Harald Vajkonny,
<http://wiki.lug-frankfurt.de/TeXStammtisch>
Restaurant »Zum Jordan«, Westerbachstr. 7, 60489 Frankfurt
Zweimonatlich, Vierter Donnerstag im Monat, 19.30 Uhr

Göttingen

Holger Nobach,
holger.nobach@nambis.de, <http://goetex.nambis.de/>
Restaurant Mazzoni Cucina Italiana,
Hermann-Rein-Straße 2, 37075 Göttingen
Dritter Donnerstag im Monat, 18.00 Uhr

Hamburg

Lothar Fröhling,
lothar@thefroehlings.de
Letzter Dienstag im Monat an wechselnden Orten, 19.00 Uhr

Hannover

Mark Heisterkamp,
 heisterkamp@rrzn.uni-hannover.de
Seminarraum RRZN, Schloßwender Straße 5, 30159 Hannover
Zweiter Donnerstag im Monat, 18.30 Uhr

Heidelberg

Martin Wilhelm Leidig, Tel.: (06203) 40 22 03,
 moss@moss.in-berlin.de
 Anmeldeseite zur Mailingliste: <http://mailman.moss.in-berlin.de/mailman/listinfo/stammtisch-hd-moss.in-berlin.de>
Wechselnder Ort
Letzter Freitag im Monat, ab 19.30 Uhr

Karlsruhe

Klaus Braune, Tel.: (0721) 608-4 40 31,
 klaus.braune@kit.edu,
SCC (Steinbuch Centre for Computing) des KIT (vormals Universität Karlsruhe, Rechenzentrum),
Zirkel 2, 2. OG, Raum 203, 76131 Karlsruhe
Erster Donnerstag im Monat, 19.30 Uhr

Köln

Uwe Ziegenhagen
Dingfabrik, Erzbergerplatz 9, 50733 Köln
Zweiter Dienstag im Monat, 19.00 Uhr

München

Uwe Siart,
 uwe.siard@tum.de, <http://www.siard.de/typografie/stammtisch.xhtml>
Erste Woche in geradzahligen Monaten an wechselnden Tagen, 19.00 Uhr

Stuttgart

Bernd Raichle,
 bernd.raichle@gmx.de
Bar e Ristorante »Valle«, Geschwister-Scholl-Straße 3, 70197 Stuttgart
Zweiter Dienstag im Monat, 19.30 Uhr

Trier

Martin Sievers,
 ttt@schoenerpublizieren.de
 Anmeldeseite zur Mailingliste: <http://lists.schoenerpublizieren.de/cgi-bin/mailman/listinfo/ttt>
Universität Trier
nach Vereinbarung

Wuppertal

Andreas Schrell, Tel.: (02193) 53 10 93,
 as@schrell.de
Restaurant Croatia »Haus Johannisberg«, Südstraße 10, 42103 Wuppertal
Zweiter Donnerstag im Monat, 19.30 Uhr

Würzburg

Bastian Hepp,
 LaTeX@sning.de
nach Vereinbarung

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

Tel.: (0 62 21) 2 97 66 (Mo., Mi.–Fr., 10.00–12.00 Uhr)
Fax: (0 62 21) 16 79 06
E-Mail: dante@dante.de

Konto: VR Bank Rhein-Neckar eG
BLZ 670 900 00 IBAN DE67 6709 0000 0002 3100 07
Kontonummer 2 310 007 SWIFT-BIC GENODE61MA2

Präsidium

Vorsitzender:	Martin Sievers	president@dante.de
stv. Vorsitzender:	Herbert Voß	vice-president@dante.de
Schatzmeister:	Klaus Höppner	treasurer@dante.de
Schriftführer:	Manfred Lotz	secretary@dante.de
Beisitzer:	Volker RW Schaa	
	Dominik Wagenführ	
	Uwe Ziegenhagen	

Ehrenmitglieder

Peter Sandner	22.03.1990	Klaus Thull †	22.03.1990
Yannis Haralambous	05.09.1991	Barbara Beeton	27.02.1997
Luzia Dietsche	27.02.1997	Donald E. Knuth	27.02.1997
Eberhard Mattes	27.02.1997	Hermann Zapf †	19.02.1999
Joachim Lammarsch	12.04.2014	Rainer Schöpf	12.04.2014

Server

DANTE: <http://www.dante.de/> (Rainer Schöpf, Joachim Schrod)
CTAN: <http://mirror.ctan.org/>

FAQ

DTK: <http://projekte.dante.de/DTK/WebHome>
T_EX: <http://projekte.dante.de/DanteFAQ/WebHome>

Autoren/Organisatoren

Jürgen Fenn Friedensallee 174/20 63263 Neu-Isenburg juergen.fenn@gmx.de	[68]	Rolf Niepraschk Persiusstr. 12 10245 Berlin Rolf.Niepraschk@gmx.de	[64]
Rainer-M. Fritsch Reuterstraße 38 12047 Berlin mail@rmf.berlin	[60]	Elke Schubert Hagenstraße 5 76297 Stutensee elke.schubert@kabelbw.de	[7, 8, 30]
Markus Kohm Freiherr-von-Drais-Straße 66 68535 Edingen-Neckarhausen komascript@gmx.info	[14, 20, 24, 33, 37, 47, 54]	Martin Sievers siehe Seite 74	[5]
		Herbert Voß Wasgenstraße 21 14129 Berlin herbert@dante.de	[3]

Die T_EXnische Komödie

27. Jahrgang Heft 3/2015 August 2015

Impressum

Editorial

Hinter der Bühne

5 Grußwort

KOMA-Script

- 7 Experimentelles KOMA-Script-Repository
- 9 Von scrpage2 zu scrlayer-scrpage
- 14 Firmenlogo mit scrlayer
- 20 Dokumentversion mit scrlayer
- 24 Farbige, kleine Kapitelmarken am Rand mit scrlayer
- 30 Ändern des Kapitelformats
- 33 Kapitelübersicht mit Kurzbeschreibung
- 38 Kombination von Kapitelmarken mit einer Kapitelübersicht
- 47 Anhangsverzeichnis
- 54 Beschränkung von chapteratlists=entry
- 60 Rechnungen erstellen
- 65 Briefkopien mit scrlltr2 leicht gemacht

Von fremden Bühnen

68 Neue Pakete auf CTAN

Spielplan

- 71 Termine
- 72 Stammtische

Adressen

75 Autoren/Organisatoren