

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

24. Jahrgang Heft 3/2012 August 2012

3/2012

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-0.70.2-2012052410 (format=lualatex 2012.7.8) (\TeX Live 2012) erstellt. Als Standard-Schriften kamen Linux Libertine, Linux Biolinum, Bera Mono und \TeX Gyre Pagella 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: Rudolf Herrmann Gert Ingold Eberhard Lisse

Rolf Niepraschk Heiko Oberdiek Christine Römer

Volker RW Schaa Dominik Waßenhoven

Redaktionsschluss für Heft 4/2012: 15. Oktober 2012

ISSN 1434-5897

Editorial

Liebe Leserinnen und liebe Leser,

man sollte bei einem etwas komplizierteren Objekt, wie es »Die \TeX nische Komödie« zweifelsohne darstellt, nicht zwischendurch Urlaub machen; nach vier Wochen wunderte ich mich, warum der Übersetzungsvorgang der Gesamtausgabe nicht mehr möglich war. Ich konnte mich aber auch nicht mehr erinnern, ob es vorher ohne Probleme ging. Die Erstellung der PDF-Datei erfolgt über ein Makefile gesteuert, wobei prinzipiell jedes Dokument erst einmal einzeln übersetzt wird, um u. a. eventuell vorhandene Bibliografien zu erstellen. Gleichzeitig werden alle gefundenen `\usepackage`-Anweisungen in eine externe Datei geschrieben. Das Gesamtdokument liest dann diese Datei als Präambel und fügt die einzelnen Dokumente über ein modifiziertes `\input` ein. Befehle wie `\documentclass`, `\usepackage` usw. werden dabei ignoriert, sodass ein großes Dokument mit allen Beiträgen entsteht, welches allerdings in schöner Regelmäßigkeit Probleme bereitet; die willkürlich zusammengesetzte Reihenfolge der Pakete in der Datei muss meistens korrigiert werden. Oder es musste, wie bei dieser Ausgabe, auf das Laden von `tikz` verzichtet werden, da es zusammen mit den speziellen \LaTeX 3-Befehlen nicht funktionierte. Das war insofern unproblematisch, als es nur für eine einzelne Grafik verwendet wurde. Diese kann man mithilfe der Dokumentenklasse `standalone` schnell erstellen, um sie dann über `includegraphics` einzubinden, sodass das problematische Paket nicht mehr geladen werden muss.

Diese Ausgabe zeigt wieder ein breites Spektrum der Anwendungen, die mit \LaTeX , $\text{Lua}\TeX$ oder \LaTeX 3 möglich sind. Den von mir geplanten Tagungsbericht zur TUG 2012, die im Juli in Boston stattfand, werden Sie voraussichtlich im nächsten Heft finden. Da diese Ausgabe die offizielle Einladung zur Herbsttagung mit der obligatorischen Mitgliederversammlung enthält, muss sie wegen der einzuhaltenden Einladungsfristen zeitnah erscheinen.

Ich wünsche Ihnen wie immer viel Spaß beim Lesen und verbleibe mit \TeX nischen Grüßen,

Ihr Herbert Voß

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

der Sommer 2012 hat anscheinend doch noch die Kurve bekommen. Ich hoffe, Sie können ihn genießen und haben erholsame Tage verbracht – ob im Urlaub auf Reisen oder beim Grillen auf »Balkonien« oder »Terrassia«.

Ein wenig hitzig geht es zuweilen auch auf unserer Mailingliste zu. Selbst wenn die Diskussionen meist recht schnell wieder abkühlen, kommt es leider immer wieder vor, dass sich Mitglieder durch den Umgangston angegriffen fühlen. Ich möchte daher alle Diskutanden bitten, auch bei Beiträgen im »unpersönlichen« World Wide Web stets fair miteinander umzugehen und gleichzeitig nicht jedes (zu schnell) geschriebene Wort sofort auf die Goldwaage zu legen. So trägt unsere Mailingliste dann hoffentlich wieder ausschließlich zum fruchtbaren Austausch unter Gleichgesinnten bei.

Anfang Juli fand die erste Sitzung des neu gewählten Vorstands im DANTE-Büro in Heidelberg statt. Dieser Termin wird auch in Zukunft fest in den Jahresplan integriert werden, so dass derzeit vier Termine für die Vorstandstreffen existieren: das Jahresanfangstreffen im Januar (Darmstadt), die DANTE Frühjahrstagung im März, die Vorstandssitzung Anfang Juli (Heidelberg) und die DANTE-Herbsttagung im Oktober. Für alle Mitglieder besteht jederzeit die Möglichkeit, Anregungen für die jeweilige Agenda zu machen – eine entsprechende E-Mail an vorstand@dante.de genügt.

Bei der ersten Sitzung wurden viele organisatorische Dinge besprochen. Darüber hinaus wurden aber auch einige Beschlüsse und Vorhaben gefasst, über die ich hier kurz berichten möchte: Die Situation an Hochschulen – aber nicht nur dort – im Bezug auf \TeX hat sich in den letzten Jahren stark verändert. Installationen und Support, die lange üblich waren, werden zunehmend zurückgefahren oder im Zuge von Umstellungen der Infrastruktur gänzlich aufgegeben.

Der Vorstand sieht derzeit keine Möglichkeit, mit vertretbarem Aufwand zentral Änderungen herbeizuführen. Gleichwohl gibt es zahlreiche Mitglieder, die sich auch bisher schon für die Verbreitung und den Erhalt » \TeX nischer« Strukturen

an ihrer Schule, Hochschule oder auch in ihrem Unternehmen eingesetzt haben. Dieses Engagement wollen wir fördern und ausbauen. Dazu können beispielsweise für die Auslage vor Ort alte DTK-Ausgaben oder auch unser Flyer dienen, der dazu derzeit von einer Designerin überarbeitet wird. DANTE bietet darüber hinaus schon seit langem die Möglichkeit, für Open-Source-Tage, Lehrerfortbildungen, Erstsemester-Tage etc. Büchertische zusammenzustellen, um vor Ort Präsenz zu zeigen. Ihnen als freiwilliger Helfer vor Ort und »verlängerter« Arm wird dabei so viel Arbeit abgenommen wie es möglich ist – nur anfragen unter office@dante.de müssen Sie noch selbst.

Die finanzielle Situation des Vereins ist durchaus erfreulich. Daher werden im Büro einige in die Jahre gekommene Geräte durch aktuelle ersetzt. Zudem werden für Tagungen und Messen voraussichtlich zwei große transportable Banner (»Roll-Up-Displays«) gekauft.

Zwei »Langzeithemen« waren ebenfalls Gegenstand der Diskussionen: Die »T_EXnischen Kontakte« als gedrucktes Mitgliederverzeichnis werden endgültig eingestampft, die zugrundeliegende Vorlage soll aber in anderem Zusammenhang wiederverwendet werden.

Unsere Internetseite ist sicherlich verbesserungswürdig, gerade auch, weil Inhalte über die Jahre hinweg »unkontrolliert« gewachsen sind. Hier soll durch einen »Style Guide«, in dem Regeln für die Darstellung bestimmter Begriffe und Elemente festgelegt werden, eine Konsolidierung erfolgen. Für die Überarbeitung der Inhalte, insbesondere für die FAQ-Seiten, werden weiterhin Mitstreiter gesucht. Auch bzgl. der Struktur sind wir als Vorstand für konstruktive Vorschläge dankbar. An der grundsätzlichen Verwendung des Content-Management-Systems Lenya soll aber festgehalten werden.

Zurück zur DTK: Auch diese Ausgabe bietet aus meiner Sicht einen breiten Blick auf die »T_EXnischen« Möglichkeiten aus ganz unterschiedlichen Perspektiven. Ein herzliches Dankeschön dafür an die Autoren der Beiträge. Ganz besonders bedanke ich mich bei Joachim Lammarsch, der den Nachruf auf unser verstorbene Ehrenmitglied Klaus Thull verfasst hat. Dieser war ein wichtiger Wegbereiter für T_EX im deutschsprachigen Raum, zu einer Zeit, wo bereits die Verbreitung von Software mit hohem persönlichen Einsatz und viel Herzblut verbunden war. Für dieses Engagement gilt ihm auch von Vereinsseite unser tief empfundener Dank.

Ich wünsche Ihnen allen interessante und unterhaltsame Stunden bei der Lektüre und hoffe, möglichst viele von Ihnen in Breskens bei der DANTE-Herbsttagung zu sehen.

Herzlichst Ihr/Euer
Martin Sievers

Einladung zur Herbsttagung und 47. Mitgliederversammlung von DANTE e.V.

Liebe DANTE-Mitglieder,

hiermit laden wir Sie herzlich zur Herbsttagung und 47. Mitgliederversammlung von DANTE e.V. ein. Zum ersten Mal werden wir im Rahmen der diesjährigen Euro \TeX vom 8.–12. Oktober Gast an der holländischen Küste, genauer im südlichsten Zipfel der Provinz Zeeland in unmittelbarer Nähe zu Belgien sein. Dadurch weicht der Tagungsablauf diesmal von unseren sonst üblichen Wochenendterminen ab.

Die Euro \TeX -Tagung wird von der Con \TeX t-Gruppe organisiert, deren eigenes Nutzertreffen zeitgleich stattfindet. Die Adresse des Tagungsorts lautet:

Hotel Restaurant Scaldis
Langeweg 3
4511 GA Breskens, Niederlande

Die Mitgliederversammlung von DANTE e.V. findet am Montag, den 8. Oktober 2012 um 17.00 Uhr im Anschluss an das reguläre Euro \TeX -Programm statt.

Die Tagesordnung besteht aus folgenden Punkten:

1. Begrüßung; Vorstellung des Vorstands; Tagesordnung
2. Bericht des Vorstands
3. Verschiedenes

Wie üblich sind auch Nichtmitglieder als Gäste herzlich willkommen.

Weitere Informationen zur Euro \TeX -Tagung finden Sie unter <http://www.ntg.nl/eurotex2012/>. Dort können Sie sich auch ganz einfach über ein Formular anmelden und sich über die Anreise informieren. Da diese für viele Teilnehmer etwas aufwendiger ist, gibt es auf der DANTE-Mailingliste bereits einen ersten Austausch zu privaten Mitfahrgelegenheiten und »Shuttelfahrten« bis zum Tagungsort.

Das übergreifende Thema der Konferenz lautet diesmal »Recreational uses of \TeX «, übersetzt etwa » \TeX -Einsatz in der Freizeit«. Allerdings dürfen wie immer auch andere Themen rund um » \TeX and friends« sehr gerne eingebracht werden. Falls Sie dies in Form eines Tutoriums oder Vortrags tun wollen, schreiben Sie bitte eine entsprechende E-Mail an eurotex@ntg.nl. Fügen Sie bitte eine Kurzzusammenfassung (Abstract) von maximal einer DIN A4-Seite als Text- oder \LaTeX -Datei bei.

Schriftliche Anmeldungen, Fragen, Wünsche und Anregungen rund um die Tagung richten Sie bitte an:

DANTE e.V.

Stichwort: Mitgliederversammlung von DANTE e.V.

Postfach 10 18 40

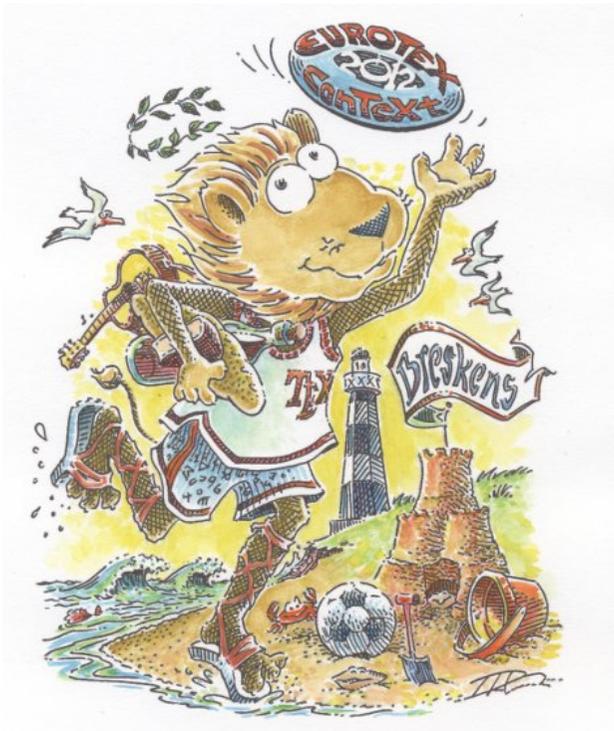
69008 Heidelberg

E-Mail (bevorzugt): mv47@dante.de

Wir freuen uns auf viele neue und alte Gesichter in Breskens, das nicht nur mit einem reizvollen touristischen Beiprogramm lockt.

Mit freundlichen Grüßen

Martin Sievers (Vorsitzender), Herbert Voß (Stellvertretender Vorsitzender)



Klaus Thull, eines unserer Ehrenmitglieder, ist verstorben.

Joachim Lammarsch

Er starb am 13. Mai 2012. Für viele kam es unerwartet, was wohl daran lag, dass niemand wusste, wie krank er war. Klaus Thull wurde am 22. März 1990 zum Ehrenmitglied von DANTE e.V. ernannt (<http://www.dante.de/DTK/Ausgaben/komoedie19901.pdf>). Das war unsere Anerkennung für die Unterstützung, die er unserem Verein zukommen ließ. Er hat unglaublich viel für DANTE e.V. getan, auch wenn er immer alle »Vereinsmeierei« weit von sich gewiesen hat. Das war nicht »sein Ding«. Trotzdem war er beispielsweise beim Eintüten der T_EXnischen Komödie, das früher von Hand an einem Abend im Rechenzentrum der Uni Heidelberg von fleißigen Händen erledigt wurde, immer dabei.

Die T_EXnische Geschichte von Klaus geht aber noch viel weiter in die Vergangenheit. Als ich Anfang der achtziger Jahre zum ersten Mal von T_EX gehört hatte, fand ich die Software sehr interessant. Klaus, ein Studienkollege und Werksstudent bei IBM in Heidelberg, meinte: »Ich schick's Dir mal zu.« Und es war Klaus, der dann in einer Nacht- und Nebelaktion über das erste Stück EARN, dem Vorläufer des Internet, die kompletten T_EX-Dateien ins Rechenzentrum sandte. Der Plattenplatz im URZ lief über ..., doch so etwas hat Klaus nicht gestört.

Er hat sich intensiv mit Sprachen und Schriften beschäftigt. Die Devanagari-Schriften lagen ihm besonders am Herzen. Darin begründet sich auch sein Nickname Mahilata. Aber die prägendste Aktion für mich passierte später. Es gab nicht mehr nur Großrechner sondern auch PCs. Für die PCs gab es zwei kommerzielle T_EX-Versionen von zwei amerikanischen Firmen. Ich erinnere mich noch heute sehr gut daran: Die Tür in meinem Zimmer im Rechenzentrum ging auf, Klaus kam herein und legte mir einen Stapel von 25 Disketten der Größe 5,25 Zoll auf den Tisch mit dem Kommentar, hier ist eine freie T_EX-Version für den IBM-PC, »Mach was gescheits draus«. Das war die Grundlage für die T_EX-Version, die wir in großen Stückzahlen in ganz Deutschland verteilen konnten, und die viel zum Erfolg von DANTE e.V. damals beigetragen hat. Aber nicht genug damit, hat Klaus auch ein METAFONT für IBM PCs geschaffen, das er ebenfalls dem Verein zur Verfügung stellte.

In den letzten Jahren war man gewohnt, dass Klaus verschwand und sporadisch immer wieder auftauchte. Jetzt wird er nicht mehr auftauchen. An diese Tatsache werde ich mich erst gewöhnen müssen.

Auszug aus einem »Interview« mit Yannis Haralambous (YH), geführt von Dave Walden (DH), nachzulesen in <http://www.tug.org/interviews/haralambous.html>

DW: Please tell me how you first became involved with \TeX .

YH: In 1987, the Math department of Lille University decided to buy two Mac Plus machines to run a piece of software called Textures. A few days later my Ph. D. advisor handed me a copy of The \TeX book. I loved it so much that I read it overnight (what I liked the most was the drawing of the computer drinking coffee). It was that event as well as my encounter with Klaus Thull, in 1988, that completely changed my life and made me switch from Mathematics to Computer Science.

[...]

DW: Who is Klaus Thull and what was your encounter with him about?

YH: Oh god! You don't know Klaus Thull? He's a great figure of early \TeX in Germany. He was the first to write a public domain METAFONT for MS-DOS, and for that reason he is an honorary DANTE member, for life. He worked on fractals and on Sanskrit and on many other interesting fields. Unfortunately I lost his trace many years ago. You should absolutely interview him.

Bretter, die die Welt bedeuten

KOMA-Script wird volljährig

Markus Kohm

1994 hat nicht nur $\text{\TeX} 2_{\epsilon}$, sondern auch KOMA-Script das Licht der Welt erblickt und wurde damit in diesem Jahr 18 Jahre alt. Im Unterschied zu $\text{\TeX} 2_{\epsilon}$, das bei seinem offiziellen Erscheinen bereits voll entwickelt war, musste KOMA-Script in den vergangenen 18 Jahren erst noch wachsen. Angefangen hat es als Baby, das nicht nur bei jeder Kleinigkeit geschrien hat, sondern auch sein Umfeld in Atem hielt. Seinen Vater, mich, kostete es einige Nerven, er will die Gelegenheit für einen kleinen Rückblick nutzen.

Die Empfängnis

Wann genau die Empfängnis von KOMA-Script war und wie lange ich damit schwanger ging, ist nachträglich schwer zu ermitteln. Eher zufällig stieß ich Anfang der 90er-Jahre des vorigen Jahrhunderts auf die SCRIPT-Sammlung von Frank Neukam. Darin war eigentlich alles enthalten, was ich für technische Dokumente während meines Studiums benötigte. Was fehlte, waren in erster Linie Anpassungen, die ich für literarische Texte benötigte, beispielsweise die Unterstützung für das A5-Format. Kurz entschlossen fügte ich die wichtigsten Änderungen in die Stile ein. Damals war die Lizenz von SCRIPT etwas unklar und der Autor außerdem für mich schwer zu erreichen. Deshalb beschloss ich irgendwann, meine modifizierte Version über die Maus KA¹ als Alternative anzubieten. Auch andere nahmen Änderungen daran vor und so gab es eine Zeit lang diverse inkompatible Versionen von SCRIPT. Als Frank Neukam im Dezember 1993 Script 2.0 herausbrachte, beschwerte er sich zu Recht über das Durcheinander. In Script 2.0 waren meine Änderungen vollständig enthalten.

¹Die Maus KA war eine Mailbox des MausNet. Dieses Mailbox-Netzwerk erfreute sich in der Zeit von Akustikkoppler und Modem in Deutschland, der Schweiz und Österreich großer Beliebtheit. Die Bezeichnungen der einzelnen Mailboxen richteten sich nach dem jeweiligen Autokennzeichen. Die Maus KA war demnach in Karlsruhe beheimatet.

Bereits kurz darauf hörte ich auch das erste Mal davon, dass ein Nachfolger von \LaTeX 2.09 in Arbeit sei, bei dem alles anders werden solle. Ich begann mir daraufhin Gedanken über die Zukunft meiner Dokumente zu machen, die inzwischen ganz auf die Verwendung von SCRIPT abgestellt waren. Ab Februar 1994 versuchte ich deshalb Frank Neukam zu erreichen, um ein neuerliches Versionschaos zu vermeiden. Ungefähr zu derselben Zeit – genau ist das nicht mehr festzustellen – gelang es mir endlich, eine Beta-Version von \LaTeX 2_ε auf meinen Computer zu bekommen. Um nicht wieder in fremdem Revier zu wildern – Frank Neukam antwortete weder auf Briefe noch E-Mails –, entwickelte ich aus Script 2.0, den Standardklassen und eigenen Ideen KOMA-Script 2.0. Die Versionsnummer und die Wahl der Grotesk für den Namenszug sollte dabei die Verbindung zu Script 2.0 herstellen. In erster Linie war das Ganze als Script 2.0 für \LaTeX 2_ε gedacht.

Die Entbindung

Die Entbindung erfolgte am 7. Juli 1994 in der Maus KA. Dieses Baby bestand lediglich aus den drei Basisklassen scrbook, scrreprt, scrartcl, dem Paket typearea und einer Anleitung, die lediglich eine leicht modifizierte Fassung der Anleitung zu Script 2.0 war.

Wie die meisten Babys konnte es schreien, wollte Aufmerksamkeit und hatte Hunger. Man konnte es lieb haben, davon genervt sein und es auch einfach nicht beachten. Fußball spielen konnte es noch nicht.

Das Säuglingsalter

Bereits in den ersten Tagen von KOMA-Script stürmten die Menschen auf mich ein. Nicht nur, dass jeder das Baby sehen wollte, sie wollten es berühren, fanden hier einen Finger zu klein, dort einen Zeh zu groß, die Augen zu blau, zu grün, zu grau, wollten hier ein Mützchen dazu haben und dort ein Jäckchen. Dem Papa gefiel das ganz gut und er bemühte sich, es jedem recht zu machen. Dazwischen musste er dem kleinen Schreihals hier ein Pflaster aufkleben, dort das Bäuchlein massieren, um all die Unzulänglichkeiten des Kleinen auszugleichen.

Bereits im Oktober 1994 war KOMA-Script auf die Verwendung von docstrip umgestellt. Dabei gab es dann eigene dtx-Dateien für den ursprünglichen Kern und die ersten hinzugekommenen Pakete.

Das Kleinkind

Da ich selbst kaum Geschäftsbriefe schrieb und für Privatbriefe den Füller vorzog, hatte ich auf diesen Bereich kein Augenmerk gelegt. Dies änderte sich rasch, nachdem Axel Kielhorn eine erste Anpassung von script_1.sty für \LaTeX 2_ε

zu KOMA-Script beisteuerte. So entstand die erste KOMA-Script-Briefklasse, `scrletter`.

Danach tat sich ein Kritiker ganz besonders dadurch hervor, dass er nach einer ordentlichen Pflegeanleitung verlangte. Als stolzer Papa, der neben Studium und KOMA-Script-Baby unter anderem noch mit DVI-Treiber- und Editor-Entwicklung für Atari ST und als Mitherausgeber einer Reihe von Amateur-Anthologien mehr als hinreichend belastet war, verbat ich mir solche Forderungen. So begann die \LaTeX -Karriere von Harald² mit der Erstellung einer überarbeiteten Anleitung zu KOMA-Script.

Bereits vor dem ersten Geburtstag war KOMA-Script so zu einem Sammelsurium an Klassen, Paketen und Beispieldateien angewachsen. Ganz besonders gab es diverse Pakete, welche für die unterschiedlichen Mützchen und Schühchen verantwortlich waren, die von all den Menschen nachgefragt wurden, die den Kleinen herumkrabbeln sahen. Da ich keine Lust mehr hatte, jede Woche neue Mützchen und Schühchen herzustellen, auf der anderen Seite das damals einzige für diesen Zweck verfügbare Paket, `fancyheadings`, ein Spielkamerad war, mit dem es immer wieder Streit gab, erstellte ich das Paket `scrpage`. Ab dann wurden alle Fragen nach Mützchen und Schühchen damit beantwortet.

Ein weiterer großer Schritt bestand in einigen Erweiterungen, die Werner Lemberg vorschlug. Werner Lemberg befasste sich sehr intensiv mit sogenannten CJK-Sprachen.³ Dafür benötigte er einige wenige Erweiterungen, die bei den Standardklassen nicht vorgesehen waren. In KOMA-Script waren sie rasch realisiert. So ganz genügte das allerdings nicht. An dieser Stelle zeigte sich, dass KOMA-Script internationaler werden musste. Werner Lemberg ist dann auch die erste, gekürzte, englische Übersetzung der deutschen KOMA-Script-Anleitung zu verdanken.

Die Kindheit

Zur Jahrtausendwende war KOMA-Script erneut zu einer vielfältigen Sammlung an Klassen und Paketen angewachsen. Die eher monolithische Struktur erschwerte mir auch zunehmend die Weiterentwicklung. Selbst wenn ich nur eine kleine Erweiterung vornehmen wollte, musste ich meist an drei bis vier weit auseinanderliegenden Stellen in einer viele tausend Zeilen langen Datei etwas ändern. Das lag schlicht daran, dass der Quellcode entsprechend der logischen Abarbeitung der Klassen aufgebaut war, also zuerst die Optionen, dann Grundeinstellungen, dann

²Der geneigte Leser kennt selbigen unter dem Namen Axel Sommerfeldt nicht nur vom unverzichtbaren Paket `caption`, sondern beispielsweise auch als gern gesehenen Helfer bei mannigfaltigen \LaTeX -Problemen. Ich selbst lernte ihn jedoch unter dem Namen Harald kennen.

³Chinesisch, Japanisch, Koreanisch.

darauf aufbauende Befehle und Umgebungen. Es war an der Zeit, hier eine mehr thematische Unterteilung vorzunehmen.

Bei der Anleitung zeichneten sich teils ähnliche, teils andere Probleme ab. Sie war eigentlich noch immer die Anleitung, die Axel Sommerfeldt vor Jahren erstellt hatte. Sie war nicht klar nach Klassen und Paketen strukturiert, sondern die Themen erschienen in einer eher wahllosen Abfolge, zwar weitgehend in sich geschlossen, aber schwer nachvollziehbar. Obwohl aus den ursprünglich rund 60 Seiten inzwischen nahezu 100 geworden waren, behandelte sie die meisten Themen nur sehr knapp. Während anfangs nur Nerds wie ich \LaTeX genutzt hatten und deshalb kurze und knappe, eher technische Beschreibungen ideal gewesen waren, hatte sich die Anwenderschaft inzwischen sehr gewandelt. \LaTeX mauserte sich zum System für jedermann.

Kurz gesagt, war es an der Zeit, dem Kind Struktur beizubringen. Seine Sätze mussten einen Zusammenhang bekommen und sollten nicht länger scheinbar unmotiviert in den Raum gestellt erscheinen. Das Problem dabei war, dass ich eigentlich nicht die Zeit dafür hatte. Ich hatte inzwischen wirklich einen kleinen Sohn und teilte mir mit meiner Frau Erziehung und Arbeitsstelle. Deshalb suchte ich Mitstreiter.

Die waren auch sehr schnell gefunden – vielleicht sollte ich sagen: verdächtig schnell. Als es an die konkrete Verteilung von Aufgaben bei der Erstellung einer neuen KOMA-Script-Anleitung ging, waren plötzlich alle verschwunden. Ich machte mich also zunächst im stillen Kämmerchen an die Restrukturierung des KOMA-Script-Quelltextes. Veranschlagt hatte ich dafür von Mitte 2001 bis Ende 2003. Bis dahin sollte auch trotz der Schwierigkeiten die neue Anleitung entstehen. Beim Zerpflücken der KOMA-Script-Quellen wurde mir außerdem klar, dass `scrlettr` an allen Ecken und Enden hakte. Ob das auf das ursprüngliche Design von Frank oder die rasche Umsetzung durch Axel oder meine Erweiterungen zurückzuführen ist, ist nachträglich nicht mehr zu sagen. Wahrscheinlich waren wir alle nicht ganz unschuldig.

Zur Kompensation des Frusts bei der Anleitung, entstand deshalb ein komplett neues, zumindest damals sehr anderes Design einer Briefklasse. Aus welchem Anlass damals der Kontakt zu Torsten Krüger entstand, weiß ich nicht mehr, aber er erwies sich während der Implementierung von `scrlettr2` als bester Tester, den ich je hatte. Das gab mir auch den Schwung, die neue Anleitung weiter voranzutreiben, was zusätzlich durch Jens-Uwe Morawski befügelt wurde, der kurzfristig einige kleinere Kapitel der neuen Anleitung auf Basis der Erklärungen der alten Anleitung erstellte und auch all die neuen Beispiele zu dem ebenfalls neuen Paket `scrpage2` erarbeitete, das `scrpage` abgelöst hatte.

Aus dem Kind wird ein Teenager

Als KOMA-Script ein zweistelliges Alter erreichte, gab es also auch eine neue Anleitung, ein neues Paket für Kopf- und Fußzeilen und eine neue Briefklasse mit einem neuen Bedienkonzept. Was es nicht gab, war die neue Quellcodebasis. Die bis Ende 2003 geplante Fertigstellung hatte sich tatsächlich erheblich verzögert. Inzwischen war ich in meinem Hauptberuf der Entwicklungsleiter eines Unternehmens geworden, hatte meine Tochter als zweites Kind und war dann aus dem Beruf ausgeschieden, um mich ganz um meine Kinder kümmern zu können. Das alles hatte zu einigen Turbulenzen geführt, die sich auch auf die Entwicklung von KOMA-Script auswirkten. Außerdem war es nicht leicht, gleichzeitig zwei vollkommen unabhängige Code-Stränge zu pflegen. Aber 2006 war es endlich so weit. Mit KOMA-Script 2.95 erfolgte am 5. Juli 2006, also zwei Tage vor dem zwölften Geburtstag von KOMA-Script, die endgültige Umstellung auf die neue Codebasis.

Seither ging die Entwicklung geradezu rasant weiter. Am 3. November 2008 waren mit der 17. Veröffentlichung auf Basis des neuen Codes dann endlich alle Änderungen, die ich für KOMA-Script bereits bei der Codeumstellung geplant hatte, fertig, was mit Versionsnummer 3 belohnt wurde. All die Konzepte, die ich mit `scr1tr2` getestet und inzwischen für gelungen befunden hatte, waren in dieser Version nicht nur für alle Klassen umgesetzt, sondern auch dokumentiert. Das machte sich gleichzeitig auch in einer neuerlich restrukturierten Anleitung bemerkbar.

Inzwischen gab es 34 Veröffentlichungen von KOMA-Script auf Basis des neuen Codes. Unter anderem mit `scr1file`, `toctbasic`, `scrjura` sind diverse neue Pakete für spezielle Anwendungsbereiche entstanden. Es gibt eine gedruckte, stark erweiterte Anleitung als Buch bereits in 4. offizieller Auflage⁴. Ein Ende ist trotzdem nicht abzusehen.

Wirklich erwachsen?

Am 7. Juli 2012 wird KOMA-Script 18 Jahre alt. Ich selbst habe mich gefragt, ob es damit erwachsen und ausgewachsen ist. Erwachsen ist es sicher. Aber erwachsen heißt weder ausgewachsen noch fertig. Wir alle können heutzutage nicht mit dem Lernen aufhören, nur weil wir angeblich unsere Lehrjahre hinter uns gebracht haben. Wir lernen und entwickeln uns immer weiter. Ebenso sehe ich bei KOMA-Script derzeit noch kein Ende. Tatsächlich habe ich mehr Ideen in der Schublade, als ich je werde ausführen können. Während es mich früher geärgert hat, wenn ich bereits

⁴ Vor der ersten offiziellen Auflage gab es bereits eine Vorabauflage von DANTE e.V. nur für Mitglieder. In Zusammenarbeit mit Christoph Kaeder und vor allem durch dessen persönlichen Einsatz ist dann nicht nur die Verlagsausgabe bei Lehmanns entstanden. Daraus hat sich auch die gesamte Buchreihe DANTE bei Lehmanns entwickelt, in der bisher vor allem diverse Titel von Herbert Voß erschienen sind.

Zeit investiert hatte und dann jemand anderer mit einem Paket zu demselben Thema schneller war, freut es mich daher heute, wenn ich meine Zeit anderweitig nutzen kann.

Obwohl das manchen erstaunen mag, war es nie mein Anliegen, möglichst viele Features in KOMA-Script einzubauen. Stattdessen wollte ich dem Anwender die Arbeit immer erleichtern. Das gilt zum einen für den typographischen Aspekt, zum anderen auch für das Schreiben mit \LaTeX . Weder sollte der Anwender erst mehrere Regalmeter an manchmal auch widersprüchlicher Fachliteratur wälzen müssen, um gut gesetzte Dokumente zu erstellen, noch sollte die Umsetzung der typographisch sinnvollen Änderungen das Erlernen der recht komplizierten \TeX -Sprache erfordern. Auch das Hinzufügen von möglicherweise inkompatiblen Paketen, nur um Kleinigkeiten der Form zu ändern, wie das bei den Standardklassen unumgänglich ist, sollte dem Anwender möglichst erspart werden. Typographisch sinnvolle Anpassungen sollten stattdessen mit möglichst einfachen Mitteln und einem klaren Konzept zu erreichen sein.

In der Frühphase von KOMA-Script setzten die Ressourcen der \TeX -Distributionen, der Computer selbst oder die etablierten Konzepte dem häufig Grenzen. Inzwischen sind neue Konzepte gerade in der Benutzerführung entstanden und wurden auf dem langen Weg zu KOMA-Script 3 auch von mir entwickelt und eingeführt. Hier liegt nach wie vor ein hohes Entwicklungspotential.

Einst war ich selbst ein rosa Baby mit Piepsstimme. In der Pubertät wurde die Stimme tief und Haare zierten mein Gesicht. Ebenso hat KOMA-Script von Version 2 hin zu Version 3 eine erste tiefgreifende Änderung erlebt. Inzwischen werden meine Augen schlechter, und die Haare wieder dünner – nur nicht dort, wo sie eher stören. Vielleicht wird das bei KOMA-Script früher oder später ganz ähnlich. Bei KOMA-Script bietet sich dann aber nicht nur die Möglichkeit einer Brille. Vielleicht kann man die Augen ersetzen und die Haare wirklich entfernen, statt sie immer wieder abzurazieren. Vielleicht kann man auch weite Teile des Körpers erneuern oder austauschen.

Was noch zu sagen wäre

Begonnen hatte ich die Arbeit an KOMA-Script einerseits aus Eigennutz, andererseits weil ich der Ideengemeinschaft, die mir \TeX beschert hatte, etwas zurückgeben wollte. Quasi über Nacht hatte das Projekt jedoch ein enormes öffentliches Interesse auf sich gezogen. Vor die Wahl gestellt, mein Baby eifersüchtig zu behüten und mir von niemandem dreinreden zu lassen oder den Bedürfnissen anderer ebenfalls Rechnung zu tragen, entschloss ich mich zu Letzterem.

Trotzdem hat nicht alles, was als Verbesserung gedacht war, auch den Weg in KOMA-Script und die Öffentlichkeit gefunden. Hier sei beispielhaft ein Layout-

Paket genannt, das von mir mit großem Aufwand entwickelt und einige Wochen genutzt wurde. Am Ende blieb davon nur ein sehr kleiner Teil als `scrpge2`. Der Rest wurde trotz aller Vorteile aufgrund der Komplexität der Benutzerschnittstelle verworfen.

Natürlich gab es genügend Momente in den letzten 18 Jahren, in denen ich nahe daran war, alles hinzuwerfen. Ich habe unzählige Stunden mit Support zugebracht, häufig eine Last bis hin zur Belästigung. Menschen haben mich, den notorischen Telefonhasser, nachts um halb elf angerufen, weil sie ein \LaTeX -Problem fern ab von KOMa-Script hatten. Dass eines Abends ein Professor vor meiner Tür stand und mir einen Nebenjob angeboten hat, war mir damals eher peinlich und ist heute amüsant. Dass noch nie jemand nachts um vier Uhr an meine Haustür hämmerte und Hilfe verlangte, kann schon ein wenig verwundern, aber wer weiß, was die Zukunft bringt.

Andererseits hat es mich sehr gefreut, wenn ich Menschen helfen konnte. In einer ganzen Reihe von Fällen hat der Support dazu beigetragen, meinen Horizont zu erweitern und mich sehr inspiriert. Manchmal musste ich mich gar daran erinnern, es in der Regel mit Fremden zu tun zu haben, um meine eigenen Prioritäten wieder neu auszurichten.

Ich hätte mir sehr gerne die unsäglichen Diskussionen über Lizenzverstöße erspart. Hier hätte ich bereits im Säuglingsalter von KOMa-Script wirkliche Hilfe gut gebrauchen können. Heute noch sehe ich als großen Fehler, Entwickler überhaupt mit der Realisierung und Durchsetzung ihrer diesbezüglichen Wünsche und Ansprüche zu belästigen.

Aber wenn ich mich heute umdrehe und zurückschaue, dann sehe ich nicht nur auf ein Gebirge von Code – viele zehntausend Zeilen – und ein immer dicker werdendes Buch zurück, sondern auch auf sehr viele überaus erfreuliche Begegnungen:

Da war beispielsweise Luzia, die bei der Veröffentlichung früher KOMa-Script-Versionen eine wichtige Rolle spielte; Bernd und David, die mir einige Erklärungen zur Arbeitsweise von \TeX gegeben haben; Ulrike⁵, die nicht müde wird, den Leuten meine absonderliche Logik zu erklären und mir meine Fehler zu melden; Robin, der seit Jahren die Domain zur Verfügung stellt und damals die Idee mit dem KOMa-Script Documentation Project hatte; Rainer, Alexander, Heiner und viele andere, die mir so manche Kiste Wein, Hardwarespenden und tröstende Worte haben zukommen lassen; Rolf, der als Ruhepol die in dieser Kombination nicht nur seltene, sondern auch seltsam anmutende Begabung hat, Ansporn zu sein; all die vielen Menschen, die mich getrieben, aufgemuntert, beruhigt, ermahnt, inspiriert und was nicht alles noch haben; die Wise Guys, die mir über manchen Frust und aus

⁵ Ich nutze sehr gerne die Gelegenheit ihr auf diesem Weg zum Ehrenpreis 2012 zu gratulieren!

manchem Tief heraus geholfen haben, obwohl sie von mir gar nichts wissen und die Begegnung daher sehr einseitiger Natur ist. Auch Menschen bei DANTE e.V. haben mich immer wieder unterstützt. Teilweise hat diese Unterstützung verhindert, dass ich alles hingeworfen habe.

Am Wichtigsten in all der Zeit war, dass ich Menschen kennen lernen und zu meinen Freunden rechnen durfte, die ich als wertvoll betrachte, auch wenn ich selten die Mühe aufbringe, diesen Freundschaften die Aufmerksamkeit zu widmen, die sie verdient haben.

Ach ja: Fußball spielen kann KOMA-Script noch immer nicht. Also bleibt auf jeden Fall noch etwas zu tun.

Kopfzeiltricks mit scrpage2

Markus Kohm

Für die 4. Auflage des KOMA-Script-Buchs habe ich eine spezielle Form der Kopfzeile verwendet, bei der die Seitenzahl durch einen kleinen schwarzen Balken vom Kolummentitel getrennt wird. Dabei ragt die Seitenzahl weit in den Randbereich. Realisiert wurde dies mit Hilfe des KOMA-Script-Pakets scrpage2.

Problemstellung

Im Buch zu KOMA-Script [1] wurde ein Layout mit einer langen, eher schmalen Textspalte und vergleichsweise breiten, stark besetzten Marginalien­spalte gewählt. Beim Standard-Seitenstil von KOMA-Script hätte sich hier das Problem ergeben, dass einerseits die Textspalte durch den Kolummentitel noch stärker betont worden wäre, andererseits die Seitenzahl im Fuß und bündig mit der Außenkante der Textspalte recht verloren gewirkt hätte und schwer aufzufinden gewesen wäre. Die Randproportionen wären durch die Seitenzahl ebenfalls gestört worden.

Grundsätzlich hätte hier die Möglichkeit bestanden, eine Kopfzeile zu wählen, die Textspalte und Marginalien­spalte überspannt, wobei der Kolummentitel außen bündig zu setzen gewesen wäre. Entsprechend hätte auch der Fuß breiter gesetzt werden müssen. Allerdings hätte dabei die Seitenzahl auf Seiten mit gering besetzter Marginalien­spalte sehr verloren gewirkt. Der untere Rand hätte sehr hoch gewählt werden müssen, um die Seitenzahl nicht in die Ecke zu quetschen.

Als Ausweg in solchen Fällen sieht man häufig, dass Kopf und Fuß mit einer horizontalen Linien abgetrennt werden. Diese Linie erzeugt eine Rahmenwirkung.

Persönlich kann ich daran allerdings wenig Gefallen finden, da sie mir den Seiteninhalt zu sehr einzwängt. Außerdem enthält das Buch bereits verschiedene Elemente mit Rahmenwirkung wie Tabellen und Syntax-Kästen oder Abbildungen von Seiten, die teilweise auch mit dem Kopf oder Fuß der Seite zusammentreffen können. Diese unmittelbare Häufung horizontaler Linien galt es zu vermeiden.

Daher entschied ich mich gegen horizontale Linien im Kopf oder Fuß. Der Fuß sollte insgesamt leer bleiben, um Probleme mit der Seitenzahl im Fuß zu vermeiden. Die Paginierung musste also im Kopf erfolgen. Es stellte sich lediglich noch die Frage, wo genau im Kopf. Das Problem der Seitenzahlen war mit der ersten Entscheidung also lediglich vom Fuß in den Kopf verschoben.

Seitenzahlen innen lehne ich grundsätzlich ab. Für mich ist die Erstellung eines Layouts nicht nur eine Frage der Optik sondern auch der Handhabung. Erschließt man sich den Inhalt über das Inhaltsverzeichnis oder ein Stichwortverzeichnis oder folgt Querverweisen in einem Buch, benötigt man Seitenzahlen häufig bei der Suche. Dabei ist es sehr unpraktisch, wenn die Seitenzahlen in der Nähe des Bundstegs stehen. Die Nähe des Außenstegs ist weit besser geeignet, da man dann das Buch nur wenig aufschlagen muss und so beim Blättern häufig schneller ist.

Lösung

Die Wahl fiel schließlich auf ein klassisches Design mit Kolummentitel innen und Seitenzahl außen im Kopf. Infolge der starken Besetzung der Marginalienspalte, die häufig auch für Abbildungen, Tabellen oder deren Titel mit verwendet wurde, war es erforderlich, die Seitenzahl nicht bündig mit der Außenkante der Textspalte zu setzen. Stattdessen wurde ein bündiger Abschluss mit der Marginalienspalte gewählt.

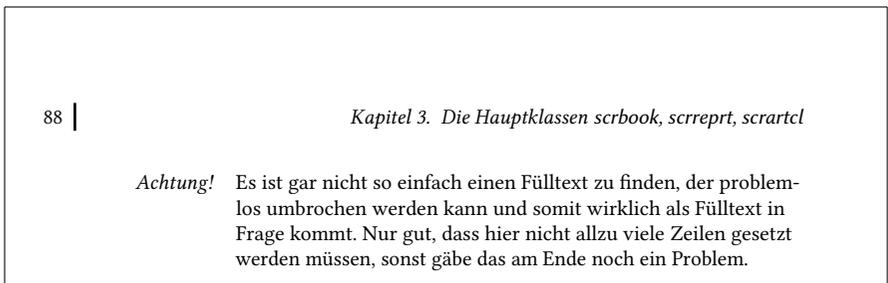


Abbildung 1: Die Kopfzeile, wie sie in [1] zu sehen ist.

Allerdings ist die Marginalienspalte nicht durchgehend stark besetzt. Es gibt auch Seiten, auf denen die Marginalienspalte frei bleibt und sogar Seiten bei denen die

Textspalte links über die gesamte Seitenhöhe eingezogen ist. Auf solchen Seiten wirkt die Seitenzahl, insbesondere solange sie nur ein- oder zweistellig ist, ein wenig verloren. Eine Betonung mit einer horizontalen Linie wollte ich, wie bereits erwähnt, vermeiden. Also wählte ich stattdessen eine etwas dickere vertikale Linie zum Inneren der Seite hin, wie Abbildung 1 zeigt.

Für die Realisierung wurde zunächst das Paket scrpage2 geladen, dessen Seitenstil scrheadings gesetzt und dieser so neu konfiguriert, dass im Kopf der Kolumnentitel innen und die Paginierung auch auf plain-Seiten wie Kapitelanfangsseiten außen gesetzt wird. Der Fuß bleibt dabei leer:

```

1 \usepackage{scrpage2}
2 \pagestyle{scrheadings}
3 \clearscrheadfoot
4 \ihead{\headmark}
5 \ohead[\pagemark]{\pagemark}

```

Erst im Anschluss daran wurde die Paginierung so umdefiniert, dass neben der Seitenzahl ein vertikaler Strich eingefügt wird. Dabei ist zwischen geraden und ungeraden Seiten zu unterscheiden:

```

6 \renewcommand*{\pagemark}{%
7   \usekomafont{pagenumber}%
8   \ifodd\value{page}\pnumbar\enskip\fi
9   \thepage
10  \ifodd\value{page}\else\enskip\pnumbar\fi
11 }%
12 \newcommand*{\pnumbar}{%
13   \raisebox{0pt}[\ht\strutbox][\dp\strutbox]{%
14     \rule[-\dp\strutbox]{1.5pt}{1.1\baselineskip}%
15   }%
16 }

```

Die Darstellung der Seitennummer mit vertikaler Linie ist also nicht Sache der Kopfdefinition selbst, sondern Bestandteil der Definition der Paginierung `\pagemark`. Die Höhe, Ausrichtung und Breite der Linie wurde durch Ausprobieren ermittelt. Die für die Ausrichtung relevante Höhe und Tiefe wurde mit Hilfe von `\raisebox` allerdings wieder auf die Normalhöhe und Normaltiefe einer Zeile festgesetzt. Dadurch wird eine korrekte vertikale Ausrichtung bezüglich des Kolumnentitels sichergestellt.

Zum Schluss wurde noch dafür gesorgt, dass der Kopf die gesamte Breite von Textspalte und Marginalienspalte einnimmt. Damit die Änderung der Kopfbreite auch eine eventuelle Neuberechnung des Seitenspiegels überlebt, wurde sie über den `hook \AfterCalculatingTypearea` definiert und sogleich per `\recalctypearea` aktiviert:

```

17 \AfterCalculatingTypearea{%
18   \setheadwidth[0pt]{%
19     \dimexpr\textwidth+\marginparsep+\marginparwidth\relax
20   }%
21 }%
22 \recalc\typearea

```

Hier ist besonders das optionale Argument `0pt` zu beachten, das dafür Sorge trägt, dass die Kopfzeile innen bündig ausgerichtet wird.

Anwendung auf einen Standard-Seitenspiegel

Wie bereits erwähnt, verwendet [1] einen speziellen Seitenspiegel mit breiter Marginalienspalte, die optisch teilweise zum Satzspiegel zu rechnen ist. Bei Verwendung eines Standard-Layouts mit unbesetzter oder dünn besetzter Marginalienspalte, die nicht als Teil des Satzspiegels betrachtet werden kann, stellt sich deshalb die Frage, ob der Kopf, wie gehabt, verwendet werden kann. Abbildung 2 zeigt hierfür ein Beispiel.

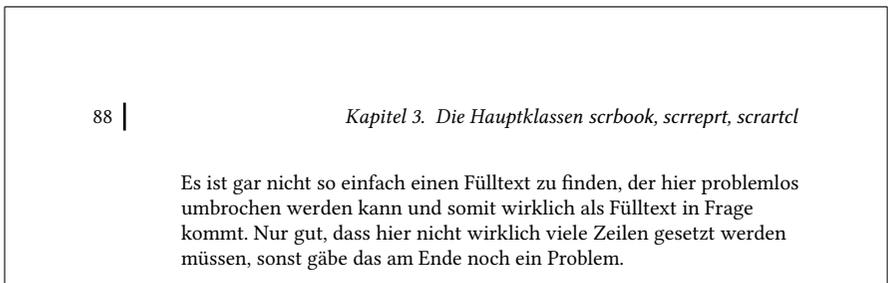


Abbildung 2: Die Kopfzeile aus Abbildung 1 in Kombination mit einem Standard-Layout

Nach meinem Verständnis steht die Seitenzahl hier etwas weit weg vom Textbereich. Als Alternative bietet sich an, die Paginierung in der Marginalienspalte innen statt außen zu setzen, wie in Abbildung 3 gezeigt.

Die Zeilen 8–10 des Listings sind hierfür wie folgt zu ändern:

```

8   \ifodd\value{page}\makebox[\marginparwidth][l]{\pnumbar\enskip\thepage
↪ }%
9   \else\makebox[\marginparwidth][r]{\thepage\enskip\pnumbar}%
10  \fi

```

Es ist gar nicht so einfach einen Fülltext zu finden, der hier problemlos umbrochen werden kann und somit wirklich als Fülltext in Frage kommt. Nur gut, dass hier nicht wirklich viele Zeilen gesetzt werden müssen, sonst gäbe das am Ende noch ein Problem.

Abbildung 3: Abwandlung der Kopfzeile aus Abbildung 2 für ein Standard-Layout

Weitere Modifikationen

Die gezeigte Kopfzeile ist in dieser Form für doppelseitigen Satz verwendbar. Im einseitigen Satz entfällt schlicht die Unterscheidung zwischen geraden und ungeraden Seiten. Es wird empfohlen, stattdessen immer die Variante für ungerade Seiten zu verwenden.

Abgesehen von der Ausrichtung des Kolumnentitels am inneren Rand ist als Alternative auch eine Ausrichtung des Kolumnentitels am äußeren Rand der Textspalte denkbar. In diesem Fall müsste die `\ihead`-Anweisung aus Zeile 4 entfallen und die `\ohead`-Anweisung aus Zeile 5 entsprechend ergänzt werden. Alternativ bietet sich hier ein Wechsel zu den Anweisungen `\rohead` und `\lehead`, wie folgt, an:

```

4 \lehead[\pagemark]{\makebox[\marginparwidth][l]{\pagemark}\hspace{
   ↪\marginparsep}\headmark}
5 \rohead[\pagemark]{\headmark\hspace{\marginparsep}
   ↪\makebox[\marginparwidth][r]{\pagemark}}
```

Soll hingegen die Linie breiter, höher, tiefer oder in einer anderen Farbe gesetzt werden, so ist lediglich die Anweisung `\pnumber` anzupassen. Die eigentliche Definition der Paginierung oder des Seitenstils bleibt davon unberührt.

Natürlich kann die Paginierung auch wieder im Fuß erfolgen, wie das bei den KOMA-Script-Klassen voreingestellt ist. Hierfür muss wiederum nicht die Definition der Paginierung `\pagemark` selbst umdefiniert werden, sondern lediglich deren Verwendung in der Konfigurierung des Seitenstils.

Schlussbemerkung

In diesem Artikel wurde gezeigt, wie man mit Hilfe von `scrpage2` sehr einfach Kopf und Fuß der Seite an ein gewünschtes Layout anpassen kann. Dabei wurde zunächst die Entscheidungsfindung für eine geeignete Form erläutert. Für die gewählte Form

wurde ein Implementierungsvorschlag erstellt, der in der Realität auch tatsächlich Anwendung findet.

Varianten für abweichende Randeinstellungen wurden erläutert und teilweise ebenfalls ausgeführt. Abschließend wurde ein Ausblick auf weitere Änderungsmöglichkeiten gegeben. Wie am Beispiel der Modifikationen gezeigt wurde, birgt der Implementierungsvorschlag den Vorteil, dass unterschiedliche Aspekte der Definition einzeln und hierarchisch implementiert wurden. Dadurch können Teilbereiche unabhängig voneinander geändert werden. Auf diese Weise wäre es beispielsweise auch möglich, unterschiedliche Paginierungsanweisungen für unterschiedliche Seiten oder Seitenstile zu definieren.

Literatur

- [1] Markus Kohm und Jens-Uwe Morawski: *KOMA-Script – Eine Sammlung von Klassen und Paketen für $\LaTeX 2_{\epsilon}$* ; Lehmanns Media; Berlin; 4. Aufl.; 2012.

Das xtemplate Paket

Clemens Niederberger

Einer der wesentlichen Punkte bei der Entwicklung von $\LaTeX 3$ stellt grob gesagt die Trennung von Implementierung, Layout-Design und Anwenderschnittstelle dar. Das Paket xtemplate steht zwischen den ersten beiden, im *Designer Interface Foundation Layer*. Anhand eines nicht zwingend praxisnahen Beispiels soll nun der Gedanke hinter dem Paket und seine Verwendung vorgestellt werden.

Einführung

Kürzlich habe ich mir zum ersten Mal das Paket xtemplate, das Teil des l3packages-Bundles¹ ist, genauer angesehen. Hatte ich erst einmal das Prinzip verstanden, war ich sofort von der Idee, die dahinter steckt, begeistert. Aber welches Prinzip steckt dahinter?

Frank Mittelbach hat zum Beispiel auf der TUG Konferenz 2011 [3] die grundlegende Struktur, die für $\LaTeX 3$ angedacht ist, diskutiert. Das will ich hier natürlich nicht wirklich wiederholen. Ein wesentlicher Teil – die Hauptidee sozusagen – ist die

¹ CTAN:macros/latex/contrib/l3packages

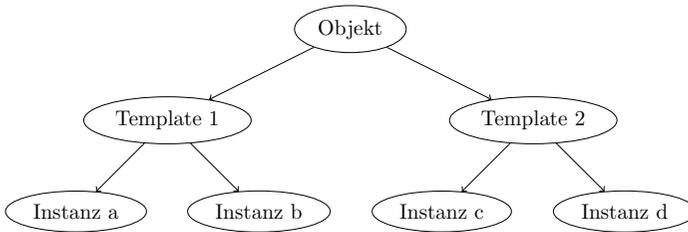


Abbildung 1: Schematische Darstellung der Verhältnisse Objekt zu Template zu Instanz.

strikte(-re) Trennung in verschiedene Schichten oder Layer. Den *Core Programming Layer* `expl3` haben Sie bestimmt schon einmal kennenlernen können, etwa im Beitrag von Dominik Waßenhoven in der Komödie 2/2012 über aufrechte Klammern in kursivem Text [2].

Das `xtemplate` Paket ist im *Designer Interface Foundation Layer* angesiedelt. Es richtet sich also vornehmlich an Paket- und Klassenautoren, dürfte aber auch in einem \TeX 3-Kernel eine große Rolle spielen. Neu ist die Idee hinter `xtemplate` nicht, wie zum Beispiel in »New Interfaces for \TeX Class Design« (1999) [4] nachzulesen ist.

Es geht in etwa um folgendes: Eine Klasse beispielsweise muss für verschiedene Objekte, zum Beispiel Überschriften oder Listen, ein passendes Design bereitstellen, bevorzugt mit einer einfachen Schnittstelle, die es Autoren erlaubt, verschiedene Parameter den eigenen Wünschen oder an bestimmte Vorgaben anzupassen. Oder, um es mit den Worten der Dokumentation von `xtemplate` [6] zu sagen:

1. semantic elements such as the ideas of sections and lists;
2. a set of design solutions for representing these elements visually;
3. specific variations for these designs that represent the elements in the document.

Sowohl der Definition als auch der Schnittstelle sollte man die Anzahl und gegebenenfalls die Art der Argumente ansehen können.

Mit `xtemplate` können nun Objekte festgelegt und sogenannte Templates für diese Objekte definiert werden. Für jedes der Templates lassen sich zu guter Letzt konkrete Instanzen definieren (Abbildung 1). Die Benutzerschnittstelle wird dann mit `xparse`² [5] definiert.

Ich für meinen Teil lerne mit Beispielen immer am besten und ich nehme an, dass ich damit nicht alleine bin.

² Ebenfalls Teil des `l3packages`-Bundles.

Ich werde also anhand eines Beispiels, das nicht notwendigerweise eine sinnvolle Anwendung von xtemplate darstellt, das Prinzip des Konzeptes demonstrieren. Inspiriert von einer Frage auf tex.stackexchange.com [7] und meiner Antwort darauf, werde ich im Folgenden das Objekt `names` und die beiden Templates `full-name` und `initial` definieren. Das Festlegen konkreter Instanzen wird dann zeigen, wie flexibel und leicht erweiterbar das Konzept ist.

Ein kleiner Vorgeschmack: die Eingabe

```
\name{Jack Skellington} \\
\name[init-it-rev]{Jack Skellington}
```

wird zu folgender Ausgabe führen:

Jack Skellington
Skellington, J.

Eine kleine Warnung: Wenn Sie mit `expl3`, der »Programmiersprache« von $\text{\LaTeX}3$ nicht vertraut sind, können die Details, vor allem der eigentliche Code, wohl etwas kryptisch wirken. Das Beispiel ist aber so kurz gehalten, dass man dem Code meiner Meinung nach auch ohne großes Vorwissen leicht folgen kann.

Die wichtigsten Befehle

Es gibt im Wesentlichen vier Befehle, die für das Definieren der Strukturen wichtig sind:

```
\DeclareObjectType{<object>}{<number of args>}
\DeclareTemplateInterface{<object>}{<template>}{<number of args>}
  {<interface>}
\DeclareTemplateCode{<object>}{<template>}{<number of args>}
  {<parameter>}{<code>}
\DeclareInstance{<object>}{<instance>}{<template>}{<parameter>}
```

Mit dem ersten wird das Objekt mit einer bestimmten Anzahl von Argumenten festgelegt. Dann kann mit dem zweiten ein Template für dieses Objekt erklärt werden. Genauer: ein Interface für das Template. Das heißt, es wird bestimmt, mit welchen Parametern welcher Art das Template später beeinflusst werden kann.

Im dritten Befehl findet die eigentliche Implementierung statt. Den Parametern, die im Interface bestimmt wurden, werden nun konkrete Variablen zugeordnet und der Code definiert, der dann das Verhalten des Templates bestimmt.

Mit dem vierten Befehl schließlich wird eine Instanz definiert, die ein Template mit konkreten Werten für die Parameter aufruft. Jede einer solchen Instanz kann dann an anderer Stelle mit `\UseInstance` aufgerufen werden.

Ein Beispiel

Das Objekt

Zunächst einmal muss man sich die grundlegende Schnittstelle überlegen. Der Benutzerbefehl unseres Objektes `names` soll `\name` mit einem Argument sein, in das Vor- und Nachname durch ein Leerzeichen getrennt eingegeben wird. Auf einer tieferen Ebene sollten wir aber Vor- und Nachnamen als einzelne Argumente behandeln. Das Objekt wird also zwei Argumente bekommen:

```
\usepackage{xtemplate,xparse}
% ich werde expl3 verwenden, also Umschalten auf den expl3-Raum:
\ExplSyntaxOn
% #1: Vorname, #2: Nachname
\DeclareObjectType { names } { 2 }
```

Das war es dann auch schon. Als nächstes müssen wir die Templates bestimmen.

Die Templates

Die Templates werden für ein bestehendes Objekt definiert. Zunächst muss das Interface festgelegt werden. Dabei werden die Anzahl der Argumente des Templates sowie eine mögliche Anzahl von Parametern festgelegt. Jeder Parameter wird einem bestimmten Typ zugeordnet und erhält unter Umständen einen voreingestellten Wert.

```
% das Interface für das Template 'fullname':
\DeclareTemplateInterface { names } { fullname } { 2 }
{
  % Parameter      : Typ      = Default
  reversed        : boolean  = false ,
  use-last-name   : boolean  = true  ,
  use-first-name  : boolean  = true  ,
  last-name-format : tokenlist
  first-name-format : tokenlist      ,
}
% das Interface für das Template 'initial':
\DeclareTemplateInterface { names } { initial } { 2 }
{
  reversed          : boolean  = false ,
  use-last-name     : boolean  = true  ,
  use-first-name    : boolean  = true  ,
  last-name-format  : tokenlist
  first-name-format : tokenlist      ,
  last-name-initial : boolean  = false ,
  first-name-initial : boolean  = true  ,
}
```

Die Parameter, die hier festgelegt werden, können ohne Aufwand erweitert werden und ergeben sich natürlich aus der Art des Objektes und der gewünschten Kom-

plexität und Flexibilität, die das jeweilige Template haben soll. Hier habe ich mich für eine begrenzte Anzahl entschieden, mit denen das Konzept leicht demonstriert werden kann.

Nachdem das Festlegen der Interfaces zu den Templates nun geschehen ist, kann der eigentlich Code für die Templates definiert werden. Fangen wir mit `fullname` an. Für die Parameter benötigen wir entsprechend ihres Typs die passenden Variablen oder Funktionen. Mit `\AssignTemplateKeys` können diese dann im Code-Teil gesetzt und mit den aktuellen Werten verwendet werden. Der Code überprüft nun lediglich die Wahrheitswerte der booleschen Variablen und schreibt je nach Wert die beiden Argumente in der gewünschten Reihenfolge. Die Lösung ist nicht die eleganteste, wird aber für diese Demonstration ausreichen:

```
% erst einmal die Variablen:
\bool_new:N \l_names_reversed_bool
\bool_new:N \l_names_use_last_bool
\bool_new:N \l_names_use_first_bool
\tl_new:N   \l_names_last_format_tl
\tl_new:N   \l_names_first_format_tl

% und dann der Template-Code:
% #1: Vorname, #2: Nachname
\DeclareTemplateCode { names } { fullname } { 2 }
{
  reversed           = \l_names_reversed_bool   ,
  use-last-name      = \l_names_use_last_bool   ,
  use-first-name     = \l_names_use_first_bool  ,
  last-name-format   = \l_names_last_format_tl ,
  first-name-format  = \l_names_first_format_tl ,
}
{
  \AssignTemplateKeys
  \bool_if:NTF \l_names_reversed_bool
  {
    \bool_if:NT \l_names_use_last_bool
    { { \tl_use:N \l_names_last_format_tl #2 } }
    \bool_if:nT { \l_names_use_first_bool && \l_names_use_last_bool }
    { , ~ }
    \bool_if:NT \l_names_use_first_bool
    { { \tl_use:N \l_names_first_format_tl #1 } }
  }
}
{
  \bool_if:NT \l_names_use_first_bool
  { { \tl_use:N \l_names_first_format_tl #1 } }
  \bool_if:nT { \l_names_use_first_bool && \l_names_use_last_bool }
  { \tl_use:N \c_space_tl }
  \bool_if:NT \l_names_use_last_bool
  { { \tl_use:N \l_names_last_format_tl #2 } }
}
}
```

Für das Template `initial` können wir die meisten Variablen wiederverwenden, benötigen aber eine Hilfsfunktion, die von einem Namen alles außer der Initialie verschluckt. Der Code wird ein klein wenig umfangreicher. Er könnte natürlich ebenfalls eleganter formuliert werden, für unsere Demonstrationszwecke aber wird er genügen.

```
% zwei weitere Variablen:
\bool_new:N \l_names_last_initial_bool
\bool_new:N \l_names_first_initial_bool

% Hilfsfunktion:
\cs_new:Npn \names_get_initial:w #1#2 \q_stop { #1 . }

% der Template-Code:
% #1: Vorname, #2: Nachname
\DeclareTemplateCode { names } { initial } { 2 }
{
  reversed           = \l_names_reversed_bool      ,
  use-last-name      = \l_names_use_last_bool      ,
  use-first-name     = \l_names_use_first_bool     ,
  last-name-format   = \l_names_last_format_tl     ,
  first-name-format  = \l_names_first_format_tl    ,
  last-name-initial  = \l_names_last_initial_bool  ,
  first-name-initial = \l_names_first_initial_bool ,
}
{
  \AssignTemplateKeys
  \bool_if:NTF \l_names_reversed_bool
  {
    \bool_if:NT \l_names_use_last_bool
    {
      \group_begin:
      \tl_use:N \l_names_last_format_tl
      \bool_if:NTF \l_names_last_initial_bool
      { \names_get_initial:w #2 \q_stop }
      { #2 }
      \group_end:
    }
  }
  \bool_if:nT { \l_names_use_first_bool && \l_names_use_last_bool }
  { , ~ }
  \bool_if:NT \l_names_use_first_bool
  {
    \group_begin:
    \tl_use:N \l_names_first_format_tl
    \bool_if:NTF \l_names_first_initial_bool
    { \names_get_initial:w #1 \q_stop }
    { #1 }
    \group_end:
  }
}
{
```

```

\bool_if:NT \l_names_use_first_bool
{
  \group_begin:
  \tl_use:N \l_names_first_format_tl
  \bool_if:NTF \l_names_first_initial_bool
  { \names_get_initial:w #1 \q_stop }
  { #1 }
  \group_end:
}
\bool_if:nT { \l_names_use_first_bool && \l_names_use_last_bool }
{ \tl_use:N \c_space_tl }
\bool_if:NT \l_names_use_last_bool
{
  \group_begin:
  \tl_use:N \l_names_last_format_tl
  \bool_if:NTF \l_names_last_initial_bool
  { \names_get_initial:w #2 \q_stop }
  { #2 }
  \group_end:
}
}
}

```

Nachdem das erledigt ist, fehlt eigentlich nicht mehr viel. Es müssen für jedes Template noch wenigstens eine Instanz und natürlich noch der Benutzerbefehl definiert werden.

Die Instanzen

Das Definieren der Instanzen ist nicht mehr weiter schwer. Man wählt das zu verwendende Template aus und übergibt den Parametern die gewünschten Werte. Ich habe mich für je drei Instanzen für jedes Template entschieden:

```

% ein paar Instanzen, erst mal mit dem Template 'fullname':
\DeclareInstance { names } { standard } { fullname } {}
\DeclareInstance { names } { it-rev } { fullname }
{
  first-name-format = \itshape ,
  reversed          = true
}
\DeclareInstance { names } { first-only } { fullname }
{ use-last-name = false }
% und dann mit 'initial':
\DeclareInstance { names } { init-first } { initial } {}
\DeclareInstance { names } { init-it-rev } { initial }
{
  first-name-format = \itshape ,
  reversed          = true
}
\DeclareInstance { names } { init-all } { initial }
{ last-name-initial = true }

```

Weitere Instanzen zu definieren wäre nun überhaupt kein Problem mehr. Mit jeder weiteren Instanz erhielte der Benutzerbefehl, den wir gleich noch festlegen werden, eine weitere Option.

Der Benutzerbefehl

Für die Definition dieses Befehl verwenden wir das Paket xparse, das Marco Daniel in der Komödie 2/2012 [1] vorgestellt hat. Damit können wir ohne Aufwand ermöglichen, dass die Eingabe mit nur noch einem Argument, nämlich Vor- und Nachname getrennt durch ein Leerzeichen, erfolgen kann.

Der Befehl wird als optionales Argument den Namen der zu verwendenden Instanz bekommen und testen, ob die gewählte Instanz existiert. Falls nicht, wird die Instanz standard verwendet, mit der der eingegebene Name dargestellt wird.

```
% eine weitere Hilfsvariable:
\tl_new:N \l_names_instance_tl
\tl_set:Nn \l_names_instance_tl { standard }

% der interne Befehl:
\cs_new:Npn \names_typeset_name:nnn #1#2#3
{
  \IfInstanceExistTF { names } { #1 }
  { \UseInstance { names } { #1 } }
  { \UseInstance { names } { standard } }
  { #2 } { #3 }
}
\cs_generate_variant:Nn \names_typeset_name:nnn { V }

% der Benutzerbefehl:
\DeclareDocumentCommand \name { o > { \SplitArgument { ~ } { 1 } } m }
{
  \group_begin:
  \IfNoValueF { #1 } { \tl_set:Nn \l_names_instance_tl { #1 } }
  % #2 expandiert zu zwei Argumenten in {}
  \names_typeset_name:Vnn \l_names_instance_tl #2
  \group_end:
}
% noch den expl3-Raum beenden:
\ExplSyntaxOff
```

Nun können wir endlich die Beispiele vom Anfang (und noch ein paar weitere) nachvollziehen:

```
\name{Jack Skellington} \\
\name[it-rev]{Jack Skellington} \\
\name[first-only]{Jack Skellington} \\
\name[init-first]{Jack Skellington} \\
\name[init-it-rev]{Jack Skellington} \\
\name[init-all]{Jack Skellington}
```

Und das Ergebnis:

Jack Skellington
Skellington, *Jack*
Jack
J. Skellington
Skellington, *J.*
J. S.

Zum Schluss

Ich hoffe, dieser kleine Ausflug konnte Ihnen einen ersten Einblick in die Funktionalität von xtemplate vermitteln. Sehr viel größer ist mein eigener Einblick zwar auch noch nicht. Meiner Meinung nach hat das Konzept dahinter aber eine große Zukunft und ich freue mich schon jetzt darauf zu sehen, wie es in L^AT_EX₃ eines Tages umgesetzt sein wird.

Quellen

Literatur

- [1] Marco Daniel: *Das Paket xparse – Dokumentenmakros auf Basis exp13; Die T_EXnische Komödie*; 24(2), S. 39–47; 2012.
- [2] Dominik Waßenhoven: *Aufrechte Klammern in kursivem Text; Die T_EXnische Komödie*; 24(2), S. 51–53; 2012.
- [3] Frank Mittelbach: *L^AT_EX₃ architecture and current work in progress*; 2011.
- [4] Frank Mittelbach, David Carlisle und Chris Rowley: *New Interfaces for L^AT_EX Class Design*; 1999.
- [5] The L^AT_EX₃ Project: *The xparse Package*; 2012-04-23.
- [6] The L^AT_EX₃ Project: *The xtemplate Package*; 2012-04-23.
- [7] Emit Taste: *Macro for formatting names (initials or full name)*; <http://tex.stackexchange.com/q/57636/5049>; zuletzt besucht: 2012-06-13.

Lyrik mit \LaTeX

Christine Römer

Mit verse und seinen Erweiterungen in memoir können sowohl Gedichte ansprechend gesetzt, als auch Interpretationshilfen für die Textanalyse eingebracht werden. Reserven gibt es noch bei der Veranschaulichung des Versmaßes.

Einführung

Lyrische Texte wurden früher nach strengen Regeln gedichtet. Deshalb ist für die Literaturwissenschaft »für die Lyrikinterpretation [...] die Kenntnis der Metrik, der Reimformen, der wichtigsten Strophen- und Gedichtformen« grundlegend. [1, S. 8] Heutige Lyrik lehnt jedoch oftmals Formregeln ab, dass es sich um Lyrik handeln soll, wird nur noch durch die Einteilung in Verse angezeigt. Diese Entwicklung ist jedoch noch nicht in das allgemeine Wissen eingegangen. In einer aktuellen Diskussion zu einem Gedicht wurde dies kürzlich sichtbar. Der nachfolgende Limerick [2] aus einer Satirezeitschrift nimmt dies auch auf die Schippe.

Was gesagt werden muss

Es gab mal zwei Staaten in Nahost
die war'n über'nander erbost.
Da kam Günter Grass
und dichtete was,
und das hat sich nicht mal gereimt.

Mit $\text{T}_{\text{E}}\text{X}$ und seinen Verwandten kann man Gedichte nicht nur typografisch ansprechend setzen und ihre Form der Sprache individuell anpassen, sondern auch die formalen Merkmale von Gedichten anzeigen.

Gedichte setzen

In [3, S. 94] wird das Setzen von Gedichten als etwas Schwieriges charakterisiert, dass »vom Typografen äußerstes Feingefühl« erfordert. »Schriftcharakter, Schriftgrößen, Zeilenabstände, Laufweite und Wortzwischenraum [...] müssen aufs Sorgfältigste der Sprache angepasst werden.«

Zum Setzen von Gedichten wird in den L^AT_EX-Nachschnlagewerken in der Regel auf die Umgebung »verse« aus dem gleichnamigen Paket (»Typesetting simple verse with L^AT_EX«) verwiesen.

```

1 \usepackage{verse}
2
3 \begin{verse} Gedicht \end{verse}

```

In dieser Umgebung werden die Verse als eine Art von Liste eingerückt und zentriert ausgegeben und Strophen mit einer Leerzeile beendet. Für die Strukturierung eines Gedichtes gibt es grundlegende Befehle, einige werden nachfolgend aufgeführt, weitere können in [4, S. 5 f.] nachgelesen werden:

<code>\</code>	neuer Vers (Zeile)
<code>\\!</code>	leere Zwischenzeile vor neuer Strophe
<code>\poemtitle{TITEL}</code>	Gedichtstitel
<code>\vin</code>	Einrücken des Verses um 1,5 em
<code>\poemlines{ZAHL}</code>	entsprechend der Zahl werden die Verse nummeriert
<code>\poemtitlefont</code>	spezifiziert den Font und die Position des Titels
<code>\verselinebreak</code>	Versumbrechen
<code>\\></code>	Abkürzung für <code>\verselinebreak</code>

L^AT_EX benötigt zur korrekten Berechnung der Ausrichtung des Gedichtes und eventuellen Zeilenumbrüchen eine Referenzzeile. Nach dieser richtet sich dann die genaue Position des Gedichtes. Diese Referenzzeile muss nicht der längste Vers des Gedichtes sein, sondern kann und sollte in der Regel dem Mittelwert entsprechen. Weiterhin werden Verse, die länger als die Referenzzeile sind, automatisch umbrochen und in der nächsten Zeile passend weitergeführt. [5]

```

1 \settowidth{\versewidth}{Referenzvers}
2 \begin{verse}[\versewidth] Gedicht \end{verse}

```

FORTSCHRITT

UND wieder rauscht mein tiefes Leben lauter, als ob es jetzt in breitem Ufern ginge. Immer verwandter werden mir die Dinge und alle Bilder immer angeschauter.	4
Dem Namenlosen fühl ich mich vertrauter: Mit meinen Sinnen, wie mit Vögeln, reiche ich in die windigen Himmel aus der Eiche und in den abgebrochenen Tag der Teiche	8

Alles ist tot, alles bang in der Stadt Nein.
 Sie ist ein Zimmer tapeziert mit Pein.
 [...]

Jewgeni Jewtuschenko

```

1 \renewcommand{\poemtitlefont}{%
2 \normalfont\large\itshape\centering}
3 \poemtitle{Zwei Städte}
4 \settowidth{\versewidth}{zwischen der Stadt Nein und der Stadt Ja}
5 % lange Zeile für mittige Gedichtpositionierung
6 \poemlines{0}
7
8 \begin{verse}[\versewidth]
9   Ich bin ein Zug,\
10  \hspace{2.3cm} der fährt jahraus -- jahrein\
11  zwischen der Stadt Ja\
12  \hspace{3.2cm} und der Stadt Nein.\
13  Meine Nerven sind gespannt\
14  \hspace{4.3cm} aus Eisendraht\
15  zwischen der Stadt Nein\
16  \hspace{3.7cm} und der Stadt Ja.\
17  Alles ist tot, alles bang in der Stadt Nein.\
18  Sie ist ein Zimmer tapeziert mit Pein.\
19  \relax [\ldots]\
20 \end{verse}
21 \attrib{Jewgeni Jewtuschenko}

```

In »memoir« hat der Paketautor von »verse« weitere Befehle zur Verfügung gestellt [5, Kap. 14], beispielsweise nachfolgende:

```

\PoemTitle*      für \poemtitle
\PoemTitle       dem Titel wird eine Nummer vorangestellt
\PlainPoemTitle  ohne Titel
\vinphantom      simuliert Text, Vers rückt um Größe TEXT ein

```

Es ist mit `\hspace{ABSTAND}` ziemlich schwierig, Versteile bündig untereinander zu setzen, wie oben an dem Jewtuschenko-Gedicht zu sehen ist. Außerdem müssen bei einem Schriftwechsel die Abstände neu eingerichtet werden. Das Bündigsetzen kann nun mit `\vinphantom` exakt erfolgen:

```

1 Ich bin ein Zug,\
2 \vinphantom{Ich bin ein Zug,}
3 der fährt jahraus -- jahrein

```

Mit den neuen Makros `\brokenline` und `\brokenlinethree` von H. Oberdiek, die `\vinphantom` mit `\providecommand` in `verse` umdefinieren, kann die doppelte Tipparbeit eingespart werden, die `\vinphantom` verlangt. Diese Makros funktionieren auch in `memoir`. Mit ihnen kann alternativ auch `\vinphantom` ignoriert wer-

den und gleich `\leavevmode\phantom` in den Definitionen von `\brokenline` und `\brokenlinethree` verwendet werden.

In einem kleinen Ausschnitt aus dem Gedicht »Einsamkeit« von Jewgeni Jewtuschenko kommen beide neuen Makros zum Einsatz.

```

1 \documentclass[12pt]{article}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage[ngerman]{babel}
5
6 \usepackage{tgpagella}
7 \usepackage{verse}
8 \pagestyle{empty}
9 \begin{document}
10
11 \newcommand{\attrib}[1]{\nopagebreak{\raggedleft\footnotesize #1\par}}
12 % Erstellen des Kommandos, um den Autor anzeigen lassen zu können
13
14 \renewcommand{\poemtitlefont}{\normalfont\large\itshape\centering}
15 \poemtitle{Zwei Städte}
16 \settowidth{\versewidth}{uns den erstbesten Kumpels in die Arme wirft}
17 % lange Zeile für mittige Gedichtpositionierung
18 \poemlines{0}
19
20 \providecommand*{\vinphantom}[1]{\leavevmode\phantom{#1}}
21 \newcommand*{\brokenline}[2]{#1\\ \vinphantom{#1}#2}
22 \newcommand*{\brokenlinethree}[3]{%
23   #1\\%
24   \vinphantom{#1}#2\\%
25   \vinphantom{#1}\vinphantom{#2}#3%
26 }
27
28 \begin{verse}[\versewidth]
29   [\ldots]\\
30   \brokenlinethree{Wir,}{schämen uns, und Angst vor solcher}{Einsamkeit
31   ↪}\\
32   uns den erstbesten Kumpels in die Arme wirft,\\
33   \relax [\ldots]\\
34   Wie sinnlos trifft man, schüttelt man die Hände sich --\\
35   \brokenline{hier säuft und säuft man bloß,}{und find't ein Ende nicht
36   ↪.}\\
37   \relax [\ldots]
38 \end{verse}
39 \attrib{Jewgeni Jewtuschenko}
40 \end{document}

```



```

12  Noch ein paar Schritte\\
13  Und dann war ich da mit dem Schlüssel zu dieser Tür\\!
14  \end{altverse}
15
16  \begin{altverse}
17  \flagverse{\textsc{Refrain}\quad} Dieser Weg wird kein leichter sein\\
18  Dieser Weg wird steinig und schwer\\
19  Nicht mit vielen wirst du dir einig sein\\
20  Doch dieser Leben bietet so viel mehr\\
21  \end{altverse}
22  \linenumberfrequency{0}
23  \end{verse}

```

Das neue poetrytex-Paket von Sam Whited (Juli 2012) ist primär für das Setzen von poetischen Anthologien geschaffen worden. Die verse-Umgebung wurde darin etwas modifiziert und in eine poem-Umgebung umgebaut.

Darstellen von Reim, Rhythmus und Metrum

Klassische Gedichte reimen nach bestimmten Schemata. So nennt man beispielsweise ein Gedicht aus Vierzeilern *Quartett*. Wenn sich die ersten und die beiden letzten Zeilen jeweils beim Aussprechen reimen, wird dies als *Paarreim* bezeichnet. Wenn sich das Ende von Verszeilen reimt, ist dies ein *Endreim*, den es wieder in verschiedenen Ausprägungen gibt. Veranschaulicht wird dies in der Regel mit Kleinbuchstaben:

aabb Paarreim
abab Kreuzreim
abba umarmender Reim
... ...

Rhythmus und Metrum werden auch *Versmaß* genannt. Die regelmäßige Abfolge von betonten und unbetonten Silben ist der *Rhythmus*, der in der lyrischen Sprache nach einem sich wiederholenden Takt (*Metrum*) aufgebaut sein kann. Eine metrisch gebundene Verszeile umfasst mehrere Takte und wird damit als zwei-, drei- oder vierhebzig bezeichnet. Man unterscheidet traditionell nach den Metren verschiedene Rhythmen in den Wortsilben: den *Jambus* (unbetont, betont) et cetera. Um mehr Übersichtlichkeit bei der Vermittlung und Bestimmung des Versmaßes zu bekommen wird die Abfolge von Hebungen und Senkungen auch graphisch dargestellt. Das übliche Modell nimmt dafür den Backslash (\) für eine Hebung und »ein kleines breites U« (U) für die Senkung [1, S. 17].

Beispielsweise wurde der *Daktylus*, der schon in der griechischen *Ilias* Verwendung fand, während der Klassikepoche in der deutschen Verslehre als Abfolge einer betonten und zweier unbetonten Silben definiert.

betont - unbetont - unbetont: \ U U

```
1 \verb|\| $\cup$ $\cup$
```

Um sich die Arbeit etwas zu erleichtern, ist es sinnvoll, Befehle zu definieren. Es dürfen dafür aber keine Befehle genommen werden, die es schon gibt:

```
1 \usepackage{xspace}
2
3 \newcommand\He{\textbackslashxspace}% Hebung
4 \newcommand\Se{$\cup$xspace}% Senkung
```

Der Daktylus tritt in dem folgenden Beispiel auf:

Würde der Frauen

Ehret die Frauen!	1
sie flechten und weben	2
Himmlische Rosen ins irdische Leben,	3
Flechten der Liebe beglückendes Band,	4
Und in der Grazie züchtigem Schleier	5
Nähren sie wachsam das ewige Feuer	6
Schöner Gefühle mit heiliger Hand.	

Friedrich Schiller

Das Schema dazu:

Vers	Metrum	Reim
	\ U U \ U	
1	\ U U \ U U \ U U \ U	a
2	\ U U \ U U \ U U \ U	a
3	\ U U \ U U \ U U \	b
4	\ U U \ U \ U U \ U	c
5	\ U U \ U U \ U U \ U	c
6	\ U U \ U U \ U U \	b

```
1 \begin{tabular}{@{}c|c|c@{}}
2 Vers & Metrum & Reim \\
3 \hline
4 & \He\Se \Se \He\Se \\
5 1 & \He\Se \Se \He\Se \Se \He\Se \Se \He\Se & a \\
6 2 & \He\Se\Se \He\Se \Se \He\Se\Se \He\Se & a \\
7 3 & \He\Se \Se \He\Se \Se\He\Se\Se \He & b \\
8 4 & \He \Se \Se \He\Se \He\Se\Se \He\Se & c \\
9 5 & \He\Se \Se \He\Se \Se \He\Se\Se \He\Se & d \\
10 6 & \He\Se \Se\He\Se \Se \He\Se\Se \He & d \\
11 \end{tabular}
```

Wie zu sehen ist, unterscheiden sich die Verse bei der Betonung der letzten Silbe; ist die letzte Silbe im Vers betont, liegt eine männliche, ist sie unbetont eine weibliche Kadenz vor. Der Titel, Vers 1, 2, 4, 5 haben weibliche und Vers 3 und 6 haben männliche Kadenz.

Wenn man sehen will, wie sich die Hebungen und Senkungen auf die einzelnen Wörter verteilen, kann der Wortzwischenraum beispielsweise durch einen Unterstrich markiert werden. Damit keine uneinheitlichen Abstände um den Unterstrich entstehen, kann man ein Makro für den Wortzwischenraum (`Zr`) verwenden:

```
\newcommand*{\Zr}{\unskip\textunderscore}
```

Würde der Frauen: `\U_U_\U = \He\Se\Zr\Se\Zr\He\Se`

Es ist auch möglich komplexe »Bausteine« zu definieren. Beispielsweise `\daktylus`: `\U U` (Daktylus) oder `\anapaest U U \` (Anapäst).

```
1 \newcommand\daktylus{${\textbackslash\cup\cup$\xspace}
2 \newcommand\anapaest{${\cup\cup\textbackslash$\xspace}
```

Literatur

- [1] Stefan Neuhaus: *Grundriss der Literaturwissenschaft*; A. Francke Verlag; Tübingen und Basel; 3. Aufl.; 2009.
- [2] Atze Svoboda: *Was gesagt werden muss*; *Eulenspiegel*; 5, S. 14; 2012.
- [3] Hans Peter Willberg und Friedrich Forsman: *Lesetypografie*; Verlag Hermann Schmidt; Mainz; 2010.
- [4] Peter R. Wilson: *Typesetting simple verse with \LaTeX* ; 2009; CTAN:macros/latex/contrib/verse/verse.pdf; zuletzt besucht: 2012-08-14.
- [5] Peter R. Wilson und Lars Madsen: *The Memoir Class for Configurable Typesetting – User Guide*; The Herries Press; Normandy Park, WA (USA); 2011; CTAN:macros/latex/contrib/memoir/memman.pdf.

europass Lebensläufe setzen mit \LaTeX

Uwe Ziegenhagen

In diesem Artikel sollen mit `ecv` und `europcv` zwei Pakete vorgestellt werden, die den Satz von Lebensläufen nach dem europäischen `europass` Muster ermöglichen.

Der Europäische Lebenslauf

Die `europass` Initiative der Europäischen Union hat das Ziel, ein Rahmenwerk für grenzüberschreitende Bewerbungsunterlagen zu schaffen. Innerhalb eines Landes lassen sich verschiedene Bewerbungen noch relativ einfach vergleichen, bei Bewerbungen aus verschiedenen europäischen Ländern wird dies jedoch schwer, da es in den einzelnen Ländern unterschiedliche Bewertungsmaßstäbe gibt. Mit `europass` ist ein Standard geschaffen worden, der Qualifikationen, Fähigkeiten und Kompetenzen europaweit verständlich darstellen will. Ein komplettes Qualifikationsprofil besteht dabei aus den folgenden Dokumenten:

- `europass` Lebenslauf
- `europass` Sprachenpass
- `europass` Mobilität
- `europass` Erläuterung zum Hochschulabschluss
- `europass` Zeugniserläuterungen

Mehr Informationen zu `europass` finden sich unter [1] und [4]. Im Folgenden beschäftigen wir uns nur mit dem Lebenslauf.

Das `ecv` Paket

Das `ecv` Paket von Chr. Neumann und B. Haberstumpf bietet eine eigene Dokumentenklasse, die in ihrer aktuellen Form deutsch und englisch als globale Optionen (`[german]` bzw. `[english]`) unterstützt. Beide Sprachen lassen sich in einem Dokument nutzen, die entsprechenden `ecv`-Befehle unterstützen alle optionalen Argumente für die Angabe der Sprache. In seiner aktuellen Version 0.3 vom April 2011 setzt das Paket `latin9`-Encoding voraus, dies lässt sich jedoch durch einfaches Austauschen gegen beispielsweise `utf8` in der `ecv.cls` beheben.

Für zentrale Angaben, wie den in der Fußzeile genutzten Namen oder die Angabe des Bewerbungsfotos, stellt das Paket die folgenden Befehle bereit:

- `\ecvPortrait{Datei}` setzt das Bewerbungsfoto
- `\ecvName{Vorname Nachname}` setzt den Namen in der Fußzeile
- `\ecvSig{Name}{Ort}` setzt die Unterschrift

Definition der Überschriften

Die einzelnen Lebensabschnitte werden dann innerhalb einer `ecv`-Umgebung gesetzt. Um einen Lebenslauf zu unterteilen, gibt es eine Reihe von Kommandos:

- `\ecvSec[lang]{Text}` Section-Befehl
- `\ecvBSec[lang]{Text}` Section-Befehl mit vertikalem Abstand
- `\ecvSub[lang]{Text}` Subsection-Befehl
- `\ecvBSub[lang]{Text}` Subsection-Befehl mit vertikalem Abstand
- `\ecvERSub[lang]{Text}{Text}` Subsection-Befehl mit rechtsbündiger Beschreibung
- `\ecvBERSub[lang]{Text}{Text}` Subsection-Befehl mit rechtsbündiger Beschreibung und Abstand
- `\ecvEBSub[lang]{Text}{Text}` Subsection-Befehl mit Beschreibung im Blocksatz
- `\ecvBESub[lang]{Text}{Text}` Subsection-Befehl mit Beschreibung im Blocksatz und Abstand

Verschiedene Befehle stehen für lokalisierte Strings bereit, die je nach Sprach-einstellung (`[german]` bzw. `[english]`) den deutschen bzw. englischen Terminus setzen:

- `\ecvPerson` für »Person« bzw. »Personal Information«
- `\ecvProfession` für »Beruf« bzw. »Profession«
- `\ecvResearch` für »Forschung« bzw. »Research«
- `\ecvEducation` für »Ausbildung« bzw. »Education«
- `\ecvPublications` für »Publikationen« bzw. »Publications«
- `\ecvAwards` für »Auszeichnungen« bzw. »Awards«
- `\ecvScholarships` für »Stipendien« bzw. »Scholarships«
- `\ecvJobs` für »Berufserfahrung« bzw. »Jobs«
- `\ecvLanguages` für »Sprachen« bzw. »Languages«
- `\ecvLanguageTravels` für »Sprachreisen« bzw. »Language Travels«
- `\ecvAbilities` für »Fähigkeiten« bzw. »Abilities«
- `\ecvConferences` für »Konferenzen« bzw. »Conferences«

- `\ecvSpeeches` für »Vorträge« bzw. »Speeches«
- `\ecvTraining` für »Fortbildungen« bzw. »Training«
- `\ecvAttachements` für »Anhänge« bzw. »Attachments«

Für die einzelnen Einträge innerhalb eines Abschnitts gibt es ebenfalls verschiedene Befehle, die gleichzeitig »Tag« und »Text« setzen, dabei entspricht »Tag« der auf der linken Seite stehenden Bezeichnung des Eintrags, »Text« der rechtsseitig stehenden Beschreibung selbst:

- `\ecvEPR[lang]{Tag}{Text}` setzt einen einfachen Marker, rechtsbündig
- `\ecvEPB[lang]{Tag}{Text}` setzt einen einfachen Marker im Blocksatz
- `\ecvEFR[lang]{Tag}{Text}` setzt einen rechtsbündigen Marker (Bullet)
- `\ecvEFB[lang]{Tag}{Text}` setzt einen Marker im Blocksatz (Bullet)
- `\ecvENR[lang]{Tag}{Text}` setzt einen rechtsbündigen Eintrag
- `\ecvENB[lang]{Tag}{Text}` setzt einen Eintrag im Blocksatz

Es gibt auch die Möglichkeit, »Tag« und »Text« getrennt zu setzen, hier sei auf die Anleitung des Pakets [2] verwiesen. Ein komplettes Beispiel zeigt Listing 1.

Listing 1: Beispiel für das `ecv` Paket, siehe Abbildung 1 auf Seite 43

```

\documentclass[german]{ecv}
\usepackage[T1]{fontenc} \usepackage{babel}
\ecvPortrait{../Bilder/Mustermann.png}
\begin{document}
\begin{ecv}
\ecvSec{\ecvPerson}
\ecvEPR{Name}{Erika Mustermann}
\ecvEPR[german]{Adresse}{Beethovenallee~37a~\newline 146612~Berlin}
\ecvEPR[english]{Address}{Beethovenallee~37a~\newline 146612~Berlin}
\ecvEPR[german]{Telefon}{(030)6912415}
\ecvEPR[english]{Phone}{+49306912415}
\ecvEPR[E-Mail]{erika.mustermann@gmx.net}
\ecvEPR[german]{Staatsangehörigkeit}{Deutsch}
\ecvEPR[english]{Nationality}{German}
\ecvEPR[german]{Geburtsdatum}{13.~September~1964}
\ecvEPR[english]{Birthdate}{September 13th 1964}
\ecvBSec{\ecvEducation}
\ecvBSub[german]{Schule}
\ecvBSub[english]{School}
\ecvEPR[german]{08/1971--05/1983}{Grundschule und Gymnasium
in Berlin (Leistungskurse Mathematik und Physik)}
\ecvEPR[english]{08/1971--05/1983}{Elementary school and grammar school
in Berlin (major fields mathematics and physics)}
\end{ecv}
\ecvSig{Erika Mustermann}{Berlin}
\end{document}

```

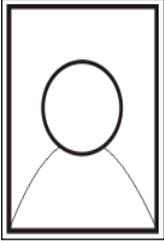
LEBENS LAUF		
ZUR PERSON		
Name	Erika Mustermann	
Adresse	Beethovenallee 37a 146612 Berlin	
Telefon	(030)6912415	
E-Mail	erika.mustermann@gmx.net	
Staatsangehörigkeit	Deutsch	
Geburtsdatum	13. September 1964	
BILDUNG		
SCHULE		
08/1971–05/1983	Grundschule und Gymnasium in Berlin (Leistungskurse Mathematik und Physik)	
Berlin, 21. Januar 2012		
Erika Mustermann		
Seite 1		

Abbildung 1: Ausgabe von Listing 1, deutsches ecv-Beispiel

Das europecv Paket

Im Gegensatz zu ecv unterstützt das europecv Paket von N. Vitacolonna aus dem Jahr 2006 quasi alle europäischen Sprachen, isländisch und maltesisch dürften hierbei sicherlich die exotischsten Vertreter sein. Wie ecv stellt europecv eine eigene Klasse bereit, der andere Pakete wie totpages oder booktabs per Klassenoption übergeben werden können. Der Lebenslauf selbst wird dann in einer europecv Umgebung gesetzt. Für die Angabe von persönlichen Daten wie Name, Adresse und Telefonnummer bietet europecv eine Reihe von Befehlen, die automatisch in die in der Klassenoption angegebenen Sprache übersetzt werden.

- `\ecvname{Text}` setzt den Namen
- `\ecvfootername{Text}` setzt den Namen in der Fußzeile
- `\ecvaddress{Text}` setzt die Adresse
- `\ecvtelephone[mobil]{TelNr}` setzt die Telefonnummer
- `\ecvfax{Faxnummer}` setzt die Faxnummer
- `\ecvemail{Text}` setzt die E-Mail Adresse
- `\ecvnationality{Text}` setzt die Angabe der Nationalität
- `\ecvdateofbirth{Datum}` setzt das Geburtsdatum
- `\ecvgender{Text}` setzt die Angabe des Geschlechts
- `\ecvpicture{Bild}` setzt das Bewerbungsfoto

Sind die entsprechenden Angaben definiert, lassen sie sich durch `\ecvpersonalinfo` setzen. Für die Gruppierung und den Satz der einzelnen Elemente des Lebenslaufs stellt europecv zwei Befehle bereit:

- `\ecvsection[vspace]{Text}` Abschnittsname mit optionalem vertikalem Abstand
- `\ecvitem[vspace]{Tag}{Beschreibung}` Eintrag mit Tag und Beschreibung und optionalem vertikalem Abstand

Sprachkenntnisse

Die Beschreibung der Sprachkenntnisse der Bewerber ist in europecv besonders umfassend gelöst; hier implementiert das Paket die Vorgabe von *europass*. *europass* unterteilt die Beherrschung einer Fremdsprache in fünf Kategorien: verstehendes Hören (I1), verstehendes Lesen (I2), kommunikatives Sprechen (I3), erzählendes Sprechen (I4) und Schreiben (I5). Die Kenntnisse jeder dieser Kategorien soll jeder Bewerber von A1 bis C2 bewerten, für Details zu den einzelnen Niveaus siehe die Anleitung des Pakets [3] oder den entsprechenden Wikipedia-Artikel [5].

- `\ecvAOne` basic user (A1)
- `\ecvATwo` basic user (A2)
- `\ecvBOne` independent user (B1)
- `\ecvBTwo` independent user (B2)
- `\ecvCOne` proficient user (C1)
- `\ecvCTwo` proficient user (C2)

Die Sprachbefähigung wird dann in Form einer Tabelle ausgegeben. Die folgenden Befehle setzen die einzelnen Teile der Tabelle.

- `\ecvmothertongue[vspace]{Sprache}` setzt die Muttersprache
- `\ecvlanguageheader[vspace]{Symbol}` setzt den Kopf der Sprachtabelle mit Referenz auf die CEF-Fußnote, `Symbol` definiert das Fußnotensymbol
- `\ecvlanguagefooter[vspace]{Symbol}` druckt die CEF-Zeile, `Symbol` definiert das Fußnotensymbol
- `\ecvlanguage[vspace]{Sprache}{11}{12}{13}{14}{15}` setzt eine Zeile der Tabelle
- `\ecvlastlanguage[vspace]{Sprache}{11}{12}{13}{14}` setzt dann die letzte Zeile, wenn `booktabs` per Klassenoption geladen wurde

Listing 2: Beispiel für das `europcv` Paket, siehe Abbildung 2 auf Seite 47

```
\documentclass[a4paper, helvetica, narrow, german, latin1, totpages, booktabs]{
  ↪europcv}
\usepackage[a4paper, top=1.5cm, left=1cm, right=1.3cm, bottom=2.0cm]{geometry}
\usepackage[T1]{fontenc}
\usepackage{babel}
\usepackage[]{graphicx}

\ecvname{Erika Mustermann}
\ecvfootername{Erika Mustermann}
\ecvaddress{Beethovenallee~37a\\ 146612~Berlin}
\ecvtelephone{0172-123\, 456\, 789}{(030)~6\, 91\, 24\, 15}
\ecvfax{(030)~6\, 91\, 24\, 15}
\ecvemail{erika.mustermann@gmx.net}
\ecvnationality{Deutsch}
\ecvdateofbirth{13. September 1964}
\ecvgender{weiblich}
\ecvpicture{../Bilder/Mustermann.png}

\begin{document}
\begin{europcv}
\ecvpersonalinfo

\ecvsection{Ausbildung}
```

```

\ecvitem{08/1971--05/1983}{Grundschule und Gymnasium in Berlin
(Leistungskurse Mathematik und Physik)}
\ecvitem{10/1983--09/1988}{Medienwirtschaft in Passau}

\ecvsection{Sprachen}
\ecvmothertongue[12pt]{Deutsch}
\ecvlanguageheader{*}
\ecvlanguage{Englisch}{\ecvAOne}{\ecvATwo}{\ecvBTwo}{\ecvBOne}{\ecvCOne}
\ecvlastlanguage{Spanisch}{\ecvAOne}{\ecvATwo}{\ecvBTwo}{\ecvBOne}{\
↪ecvCTwo}
\ecvlanguagefooter{*}
\end{europecv}
\end{document}

```

Die Abbildungen auf den Seiten 48 und 49 zeigen im Original die ersten beiden Seiten einer Anwendung des Paketes von Eberhard Lisse.

Fazit

Beide Pakete bieten sehr interessante Möglichkeiten, einen professionell aussehenden Lebenslauf nach EU-Standard zu setzen. Welches Paket man am Ende vorzieht, sollte man anhand eines kleinen Musterdokuments selbst entscheiden. Anhand des Funktionsumfangs der beiden Pakete lässt sich keine klare Empfehlung für das eine und gegen das andere aussprechen. Das *ecv*-Paket punktet beim Satz der deutschen und englischen Version des Lebenslaufs; *europecv* ist im Umgang etwas einfacher und bietet eine sehr ansprechende Darstellung der Sprachkompetenz.

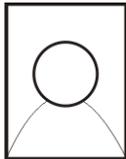
Die Beispieldateien zu diesem Artikel liegen unter <http://www.uweziegenhagen.de/?p=1333>.

Literatur

- [1] Bundesinstitut für Berufsbildung: *europass Deutschland*; <http://www.europass-info.de>; zuletzt besucht: 2012-01-21.
- [2] C.P. Neumann und B. Haberstumpf: *ecv – A fancy Curriculum Vitae Class*; 2011; CTAN:macros/latex/contrib/ecv/; zuletzt besucht: 2012-08-14.
- [3] N. Vitacolonna: *europecv: an unofficial class for European curricula*; 2006; CTAN:macros/latex/contrib/europecv/; zuletzt besucht: 2012-08-14.
- [4] Wikipedia.org: *Europäischer Bildungspass*; http://de.wikipedia.org/wiki/Europ%C3%A4ischer_Bildungspass; zuletzt besucht: 2012-01-2012.
- [5] Wikipedia.org: *Gemeinsamer Europäischer Referenzrahmen*; http://de.wikipedia.org/wiki/Gemeinsamer_Europ%C3%A4ischer_Referenzrahmen; zuletzt besucht: 2012-04-03.



**Europass
Lebenslauf**



Angaben zur Person

Nachname(n) / Vorname(n) **Erika Mustermann**

Adresse(n) **Beethovenallee 37a**
146612 Berlin

Telefon **(030) 6 91 24 15** Mobil: **0172-123 456 789**

Fax **(030) 6 91 24 15**

E-mail **erika.mustermann@gmx.net**

Staatsangehörigkeit(en) **Deutsch**

Geburtsdatum **13. September 1964**

Geschlecht **weiblich**

Ausbildung

08/1971–05/1983 **Grundschule und Gymnasium in Berlin (Leistungskurse Mathematik und Physik)**
10/1983–09/1988 **Medienwirtschaft in Passau**

Sprachen

Muttersprache(n) **Deutsch**

Selbstbeurteilung
Europäische Kompetenzstufe

		Verstehen		Sprechen		Schreiben				
		Hören	Lesen	An Gesprächen teilnehmen	Zusammenhängendes Sprechen					
Englisch	A1	Elementare Sprachverwendung	A2	Elementare Sprachverwendung	B2	Selbständige Sprachverwendung	B1	Selbständige Sprachverwendung	C1	Kompetente Sprachverwendung
	Spanisch	A1	Elementare Sprachverwendung	A2	Elementare Sprachverwendung	B2	Selbständige Sprachverwendung	B1	Selbständige Sprachverwendung	C2

*Kompetenzstufe des gemeinsamen europäischen Referenzrahmens

Seite 1 / 1 - Lebenslauf von
Erika Mustermann

Abbildung 2: Ausgabe von Listing 2, europecv



Europass Lebenslauf



Angaben zur Person

Nachname(n) / Vorname(n)

Adresse(n)

Telefon

Fax

E-mail

Staatsangehörigkeit(en)

Geburtsdatum

Geschlecht

Lisse, Eberhard Wolfgang

PO Box 8421, Bachbrecht, Namibia

+264 61 236403 Mobil: +264 81 124 6733

+264 88 624273

el@lisse.NA

Deutsch

1957-02-19

Männlich

Berufserfahrung

Berufsfeld

2004 – heute
Wichtigste Tätigkeiten und
Zuständigkeiten

2002 – 2004
Wichtigste Tätigkeiten und
Zuständigkeiten

2000 – 2002
Wichtigste Tätigkeiten und
Zuständigkeiten

1999 – 2000
Wichtigste Tätigkeiten und
Zuständigkeiten

1995 – 1999
Wichtigste Tätigkeiten und
Zuständigkeiten

1990 – 1995

Frauenheilkunde & Geburtshilfe

Niedergelassener Frauenarzt

Komplette (konservative und operative) Frauenheilkunde und Geburtshilfe als Belegarzt, Windhuk, Namibia

1. Oberarzt

Staatskrankenhaus Oshakati, Namibia, einem ländlichen Krankenhaus der Regelversorgung (5000 Geburten per annum)

Facharzt

Kreiskrankenhaus Anhalt-Zerbst, (250 Geburten per annum).

Assistenzarzt

Facharztausbildung, Städtisches Krankenhaus, Neunkirchen.

Senior Medical Officer (Assistenzarzt)

Staatskrankenhaus Swakopmund, Namibia.

Medical/Senior Medical Officer (Assistenzarzt)

Seite 1 / 4 - Lebenslauf von
Eberhard Wolfgang Lisse

Weitere Informationen finden Sie unter <http://europass.cedefop.eu.int>
© Europäische Gemeinschaften, 2003.

Wichtigste Tätigkeiten und
Zuständigkeiten

Windhoek Central Hospital, Namibianisches Nationalkrankenhaus
Maximalversorgung mit 7000 Geburten per annum

1990 – 1990

Wichtigste Tätigkeiten und
Zuständigkeiten

Praxisassistent
Belegarztpraxis in Swakopmund, Namibia

1988 – 1990

Wichtigste Tätigkeiten und
Zuständigkeiten

Medical Officer (Assistenzarzt)
Staatskrankenhaus Oshakati, Namibia,

**Schul- und
Berufsbildung**

2000

Bezeichnung der erworbenen
Qualifikation

Facharztprüfung
Facharzt für Frauenheilkunde und Geburtshilfe

Name und Art der Bildungs-
oder Ausbildungseinrichtung

Ärztelkammer des Saarlandes

1981 – 1983

Bezeichnung der erworbenen
Qualifikation

Hochschule
Ärztliche Prüfung (Staatsexamen)

Name und Art der Bildungs-
oder Ausbildungseinrichtung

Medizinische Fakultät der Rheinisch Westfälischen Technischen
Hochschule, Aachen

1981 – 1983

Bezeichnung der erworbenen
Qualifikation

Hochschule
Ärztliche Vorprüfung (Physikum)

Name und Art der Bildungs-
oder Ausbildungseinrichtung

Medizinische Fakultät der Rheinischen Friedrich-Wilhelms-Univer-
sität, Bonn

1966 – 1976

Bezeichnung der erworbenen
Qualifikation

Gymnasium
Zeugnis der Reife (Abitur)

Name und Art der Bildungs-
oder Ausbildungseinrichtung

Stätisches Neusprachliches Gymnasium, Brüsseler Ring, Aachen

Wehrdienst

Derzeitiger Dienstgrad

Oberstabsarzt d.R.

1976 – 1982

Laufbahn

Grundwehrdienst, Soldat auf Zeit

Letzter Dienstgrad

Unteroffizier in Sanitätsdienst

Stabsunteroffizier

**Persönliche
Fähigkeiten und
Kompetenzen**

Muttersprache(n)

Deutsch

Sonstige Sprache(n)

Selbstbeurteilung

Europäische

Kompetenzstufe⁽⁴⁾

Verstehen		Sprechen	Schreiben
Hören	Lesen	An Gesprä- chen teilnehmen	Zusammen- hängendes Sprechen

Briefumschläge beschriften und frankieren mit \LaTeX

Uwe Ziegenhagen

Briefe lassen sich nicht mehr nur auf die traditionelle Art mittels Briefmarken frankieren, die man am Automaten oder direkt beim Post-Angestellten gekauft hat. Auch mit selbst gedruckten Briefmarken oder gar Handyperto kann ein Brief auf die Reise geschickt werden. In diesem Artikel soll gezeigt werden, wie Briefumschläge über \LaTeX mit Briefmarken aus dem Internet bedruckt werden können.

Motivation

Ausgangspunkt meiner Überlegungen war der Wunsch, Briefumschläge nicht mehr von Hand beschriften zu müssen, sondern diese – natürlich mit \LaTeX – zu bedrucken. Fertige Pakete, die die gewünschte Funktionalität bereitstellten, waren nicht zu finden, die Lösung war aber dank `geometry.sty` auch schnell allein umgesetzt. Listing 3 zeigt den entsprechenden Quellcode. Über das Paket `geometry` wird das entsprechende Maß des Umschlags gesetzt, die Ränder wurden so gewählt, dass der für die Ausgabe genutzte Laserdrucker nicht über den Umschlag hinaus druckt. Der geneigte Leser, der dies selbst nutzen möchte, muss hier gegebenenfalls Anpassungen vornehmen.

Listing 3: Quellcode für einen Briefumschlag, Ausgabe siehe Abbildung 1.

```

1 \documentclass[12pt]{scrartcl}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage[paperwidth=229mm, paperheight=114mm,
5   left=15mm, top=10mm, right=5mm, bottom=5mm]{geometry}
6 \usepackage{helvet}
7 \renewcommand\familydefault{\sfdefault}
8 \setlength\parindent{0pt}
9
10 \begin{document}
11 Max Mustermann \\
12 Mustergasse 19\\
13 12345 Musterstadt
14
15 \vspace*{3cm}
16 \hspace*{12cm}
17 \parbox{6cm}{
18 Erika Mustermann \\
19 Musterweg 39\\
20 12346 Musterhausen}
21 \end{document}

```

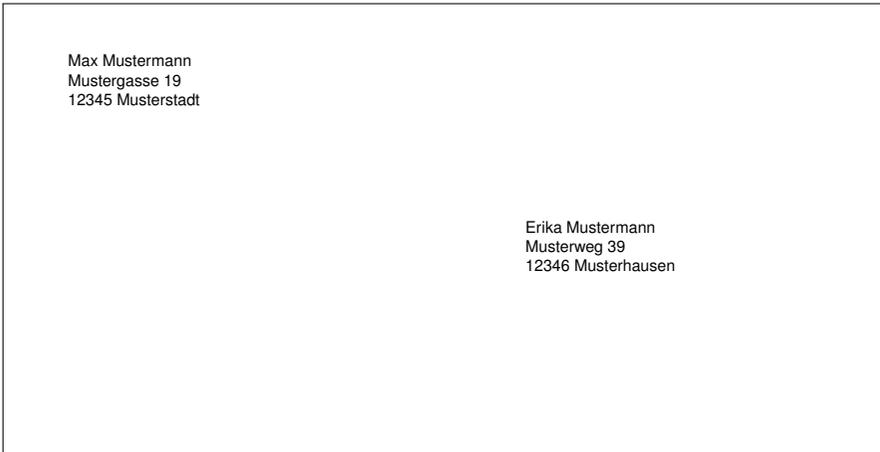


Abbildung 1: Fertiger Briefumschlag von Listing 3.

Frankierung

Nachdem die Drucklösung für die Adressen recht zügig umgesetzt war, bestand die nächste Aufgabe darin, die Frankierung des Briefs gleich mit auf den Umschlag zu drucken.

Für Microsoft Word stellt die Deutsche Post AG das »E-Porto Add-In« bereit, mit dem man direkt aus Word heraus eine Briefmarke kaufen und diese als Bild in den Briefumschlag einfügen kann. Für \LaTeX gibt es leider keine entsprechende vorgefertigte Schnittstelle. Bestellt man die Briefmarken jedoch über die Webseite der Deutschen Post, so erhält man standardmäßig eine PDF-Datei im Format DIN A4 mit den Marken zum Ausdrucken (siehe Abbildung 2). Ich habe für die folgenden Experimente einen Bogen von 20 Stück genutzt, bei anderen Stückzahlen müssen die Maße entsprechend angepasst werden.

Um die Marken zu trennen, nutzt man am besten `pdfcrop`, das unter anderem Bestandteil von `TEXlive` ist. Ruft man `pdfcrop` ohne spezielle Parameter auf (außer dem Namen der Eingabedatei), so entfernt es den gesamten nicht bedruckten Bereich der Eingabedatei.

`pdfcrop` hat jedoch mit `--bbox <top bottom right left>` einen optionalen Parameter, mit dem wir die Größe der Bounding Box – also des Bereichs, der noch in der Ausgabe sichtbar sein soll – explizit angeben können. Ein Hinweis an dieser Stelle: `pdfcrop` ändert nur den sichtbaren Bereich, nicht die Datei selbst. In jeder

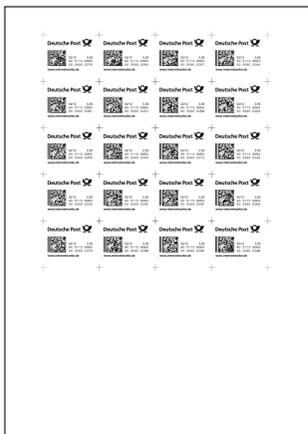


Abbildung 2: PDF-Bogen mit Briefmarken

generierten Briefmarke steckt also die komplette Datei, es wird jedoch nur die einzelne Marke angezeigt.

Die Ermittlung der Bounding Boxen für die 20 Briefmarken erwies sich dann als relativ mühselig. Der erste, etwas naive Ansatz bestand darin, die Datei im Sumatra PDF-Viewer zu öffnen, auf der Kommandozeile `pdfcrop` mit geeigneten Werten aufzurufen und anhand des Ergebnisses die Werte für den Parameter `--bbox` anzupassen.

Nach diversen Versuchen war jedoch klar, dass mittels dieser » $\pi \times$ Daumen«-Methode kein zufriedenstellendes Ergebnis zu erreichen war. Die linke untere und die rechte obere Briefmarke waren zwar sauber beschnitten, die Marken in der Mitte hingegen wiesen Verschiebungen beim Beschnitt auf.

Die Lösung dieses Problems bestand letztlich darin, den gesamten Bogen in CorelDraw X4 (in Inkscape geht es genauso) zu importieren und Hilfslinien direkt auf den Schnittmarken zu zeichnen. Da CorelDraw in neueren Versionen diese Hilfslinien automatisch einrasten lassen kann, konnte eine Genauigkeit von drei Nachkommastellen erreicht werden. Abbildung 3 zeigt eine Schnittmarke mit den entsprechenden Hilfslinien.

Die gemessenen Werte wurden dann mit allen Nachkommastellen in ein Excel-Arbeitsblatt eingetragen, dort fanden dann auch die Umrechnung in Postscript-Punkte sowie die Berechnungen für die Maße aller Briefmarken statt. Tabelle 1 zeigt die ermittelten Werte für einen DIN A4 Bogen von 20 Briefmarken.

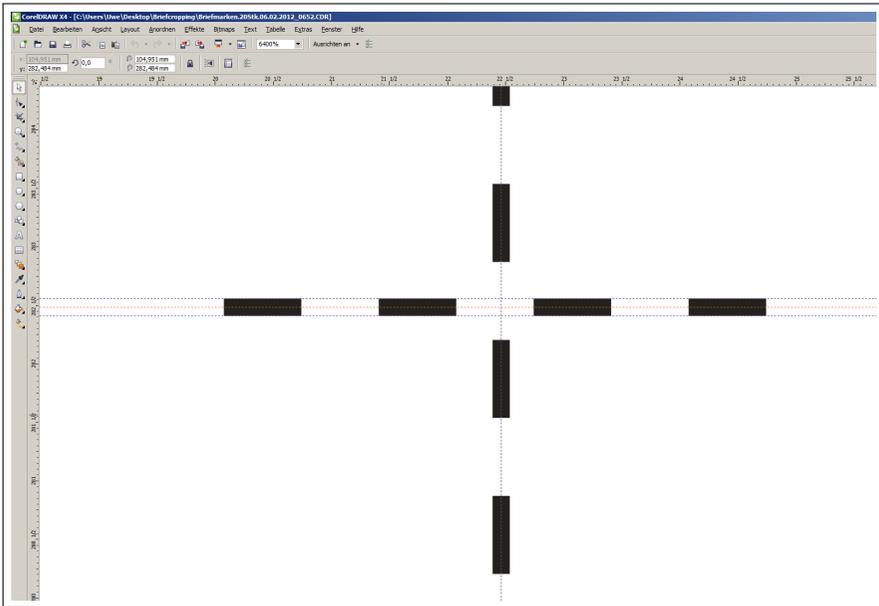


Abbildung 3: CorelDraw Hilfslinien.

Tabelle 1: Maße für einen 20-Marken-Bogen, gemessen von der linken unteren Ecke.

<i>Abstand</i>	<i>Maß in mm</i>	<i>in PS-Punkten</i>
links	22.461	63.66897638
unten	117.485	333.02834650
rechts	182.460	517.20944880
oben	282.484	800.68818900

Mittels Dreisatz und der Information, dass 25,4 mm (entsprechend einem Zoll) 72 Postscript-Punkten entsprechen (<http://de.wikipedia.org/wiki/Schriftgrad>), lassen sich die passenden Werte für alle Briefmarken bestimmen. Über die Excel-Funktionen zur Verkettung von Strings wurden dann die entsprechenden Aufrufe von pdfcrop erzeugt, sodass sich der Ablauf zur Erzeugung der einzelnen Marken auf einfaches Kopieren und Einfügen in die Kommandozeile beschränkt.

Nachdem die einzelnen Marken entsprechend beschnitten waren, wurden sie mit der `\AddtoShipoutPicture*` Routine aus dem `eso-pic` Paket auf den Umschlag gesetzt, Probieren ergab auch hier passende Werte.



Abbildung 4: Fertiger Briefumschlag mit Briefmarke.

Erweiterung & Fazit

Nachdem die grundlegenden Mechanismen jetzt definiert sind, sind noch weitere Schritte der Automatisierung denkbar. So wäre es wünschenswert, verschiedene Umschlaggrößen bedrucken zu können, neue und bereits verbrauchte Marken erfassen zu können und die Adressen in einer kleinen Datenbank vorzuhalten.

Dies soll in einer kleinen Python-Anwendung erfolgen, über den Projektfortschritt werde ich unter <http://uweziiegenhagen.de/?p=1714> berichten. Dort finden sich auch die Beispieldateien zu diesem Artikel.

Eintragsindividuelle und schlüsselwortgesteuerte Randnotizen bei Einträgen im Literaturverzeichnis mit dem Paket `biblatex`

Manuel Kühner

Einleitung

In der \LaTeX -Welt war lange Zeit `BibTeX` das (Hilfs-)Mittel der Wahl, um das Zitieren und das Erstellen von Literaturverzeichnissen zu ermöglichen. `BibTeX` litt leider unter einigen wesentlichen Einschränkungen – vor allem im Bereich der Konfigurierbarkeit und dem Umgang mit verschiedenen Zeichenkodierungen.¹

Mit dem `biblatex`-Paket stellte Philipp Lehman [1] den \LaTeX -Nutzern eine von diesen Einschränkungen befreite und zudem stark erweiterte Neuimplementierung zur Verfügung, siehe auch [4, 5].

`biblatex` – »abgehakt!«

In einem früheren Artikel dieser Zeitschrift hat Rolf Niepraschk [3] gezeigt, wie man mit Hilfe des `biblatex`-Pakets, beziehungsweise den durch dieses Paket bereit gestellten `Hook`² `\AtEveryBibitem`, einzelne Einträge im Literaturverzeichnis mit einer Kennzeichnung versieht. Die Kennzeichnung in Form eines Symbols im Randbereich soll anzeigen, ob die entsprechende Quelle im Original \checkmark oder als Kopie $[\checkmark]$ vorhanden ist:

Literatur

- [1] Albert Einstein. »Zur Elektrodynamik bewegter Körper«. In: *Annalen der Physik* $[\checkmark]$ 17 (1905), S. 891–921.
- [2] Donald E. Knuth. *Computers & Typesetting*. Bd. A: *The $T_{\text{E}}X$ book*. Reading, Mass.: Addison-Wesley, 1984. \checkmark
- [3] Donald E. Knuth. *Computers & Typesetting*. Bd. C: *The METAFONTbook*. Reading, Mass.: Addison-Wesley, 1986.

¹Umlaute waren zum Beispiel problematisch, siehe auch [2]

²Ein `Hook` (engl. für *Haken*) bezeichnet in der \LaTeX -Welt einen vom Anwender editierbaren Befehl, der zu einem bestimmten Zeitpunkt beim Kompilieren ausgeführt wird. Ein prominenter `Hook` ist z. B. `\AtBeginDocument`.

Um die einzelnen Einträge in der Literaturdatenbank (der sogenannten bib-Datei) mit dem entsprechenden Attribut (copy oder exist) zu versehen, wurde das Feld keywords verwendet.

Alternative Implementierung

In einer meiner eigenen Arbeiten wollte ich in Form von Randnotizen angeben, ob sich eine Quelle in Papierform (Papier), in PDF-Form (PDF) oder gar nicht (-) in meinem Besitz befindet:

Literatur		
[1]	Max Mustermann. <i>Fiktive Personen rund um den Globus</i> . Kuhweg, 1999.	-
[2]	Vorname Nachname. <i>Buchtitel</i> . Hüpf, 2010.	PDF
[3]	Homer Simpson. <i>Mit Kerntechnik in die Zukunft. Das Wort heißt nukular!</i> Piercings, 2002.	Papier

Die relativ simple Implementierung dieser Funktionalität ist Gegenstand des weiteren Beitrags. Ich denke, dass diese vor allem für Nicht-Experten gut geeignet ist.

Grundlagen

Kernstück der Implementierung ist der bereits weiter oben angesprochene Hook `\AtEveryBibitem` des `biblatex`-Pakets. Zudem ist wichtig zu wissen, dass das `biblatex`-Paket die Möglichkeit bietet, Einträgen der Literaturdatenbank – einzelne oder mehrere durch Kommas getrennte – frei wählbare Schlüsselwörter (keywords) zu vergeben:

```
@Book{Homer2002,
  author   = {Simpson, Homer},
  title    = {Mit Kerntechnik in die Zukunft},
  subtitle = {Das Wort heißt \textsf{nukular}!},
  publisher = {Piercings},
  year     = {2002},
  keywords = {key:Papier,key:Kerntechnik,key:Book},
}
```

Der mit `\AtEveryBibitem` definierte Code wird zu Beginn jedes einzelnen Eintrags im Literaturverzeichnis ausgeführt. Zu diesem Zeitpunkt sind sämtliche bibliografische Daten – auch die Schlüsselwörter – des entsprechenden Eintrags bekannt. Weiterführende Informationen finden sich in [1, 4.10.6 General Purpose Hooks].

Mit der `biblatex`-Version 1.2 (12.02.2011) – also vermutlich nach Redaktionsschluss für [3] – wurde die Abfrage `\ifkeyword{keyword}{true}{false}` hinzugefügt.

Damit kann man auf einfache Art und Weise in Abhängigkeit der (Nicht-)Existenz eines Schlüsselworts reagieren.

Die Randnotiz selbst wird mit Hilfe des `marginnote`-Pakets von Markus Kohm gesetzt, da dieses als verlässlicher als der `TeX`-Standard-Befehl `\marginpar` gilt.

Herzstück

Die Formatierung der Randnotiz wird über ein neu definiertes Makro festgelegt:

```
% Formatierungsmakro für Randnotiz
\newcommand{\myMarginFormat}{\footnotesize\sffamily}
```

Die Definition des `\AtEveryBibitem`-Hooks besteht aus zwei ineinander verschachtelten `\ifkeyword`-Abfragen:

```
% Wird bei jedem Eintrag ausgeführt
\AtEveryBibitem{%
  \ifkeyword{key:PDF}% Schlüsselwort "key:PDF" vorhanden?
  % True -> PDF
  {\marginnote{\myMarginFormat PDF}}%
  % False -> Papier oder Nix
  \ifkeyword{key:Papier}% Schlüsselwort "key:Papier" vorhanden?
  % True -> Papier
  {\marginnote{\myMarginFormat Papier}}%
  % False -> Nix
  {\marginnote{\myMarginFormat --}}%
}%
}%
```

Kompletter Beispielcode

Die hier verwendete `bib`-Datei mit dem Dateinamen `Literatur.bib` ist nachfolgend abgedruckt; die seltsamen Kommentarzeilen dienen lediglich zur Bereichsdefinition für das `listings`-Paket. Damit wurden die Quellcode-Beispiele in diesem Artikel gesetzt:

```
@Book{Nachname2010,
  author   = {Vorname Nachname},
  title    = {Buchtitel},
  publisher = {Höpper},
  year     = {2010},
  keywords = {key:PDF},
}
%myStart
@Book{Homer2002,
  author   = {Simpson, Homer},
  title    = {Mit Kerntechnik in die Zukunft},
  subtitle = {Das Wort heißt \textsf{nukular}!},
  publisher = {Piercings},
```

```

    year      = {2002},
    keywords  = {key: Papier, key: Kerntechnik, key: Book},
  }
%myStop
@Book{Mustermann1999,
  author     = {Max Mustermann},
  title      = {Fiktive Personen rund um den Globus},
  publisher  = {Kuhweg},
  year       = {1999},
  keywords   = {},
}

```

Nachfolgend die komplette und kommentierte \LaTeX -Datei `my_LaTeX_File.tex` zur Erzeugung des Beispiels:

```

\documentclass[
  fontsize=11pt,
  parskip=true,
  paper=a4,
  papersize=auto,
]{scrartcl}

% Standardpakete
%\usepackage[latin1]{inputenc}
%\usepackage[T1]{fontenc}
\usepackage{libertineotf}
\usepackage[ngerman]{babel}

% Bessere Randnotizen (von Markus Kohm)
\usepackage{marginnote}

% biblatex laden und konfigurieren
\usepackage[
  style=numeric,
  backend=biber,
  language=german,
]{biblatex}

% Paket zur "cleveren" Anpassung von Zitaten
% (ist vom gleichen Paketautor wie biblatex)
\usepackage[
  autostyle=true, % Voraussetzung: babel-Paket
]{csquotes}

% bib-Datei einbinden
\addbibresource{Literatur.bib}

%% eigene Anpassungen des Literaturverzeichnisses
% -----
%myStartA
% Formatierungsmakro für Randnotiz
\newcommand{\myMarginFormat}{\footnotesize\sffamily}

```

```

%myStopA

%myStartB
% Wird bei jedem Eintrag ausgeführt
\AtEveryBibitem{%
  \ifkeyword{key:PDF}% Schlüsselwort "key:PDF" vorhanden?
  % True -> PDF
  {\marginnote{\myMarginFormat PDF}}%
  % False -> Papier oder Nix
  {\ifkeyword{key:Papier}% Schlüsselwort "key:Papier" vorhanden?
  % True -> Papier
  {\marginnote{\myMarginFormat Papier}}%
  % False -> Nix
  {\marginnote{\myMarginFormat --}}%
  }%
}%
%myStopB
% -----
% Abstände ändern (Literaturverzeichnis)
\setlength\bibparsep{0em}
\setlength\bibitemsep{1em}
\setlength\bibhang{1em}

\begin{document}
% Keine Seitenzahlen und Kopf- und Fußzeile ausgeben
% alle Einträge aus der bib-Datei
% im Verzeichnis angeben
\pagestyle{empty}
\nocite{*}
\printbibliography
\end{document}

```

Unter Windows kompiliert man die Datei beispielsweise mit der folgenden Batch-Datei:

```

lualatex my_LaTeX_File.tex
biber my_LaTeX_File
lualatex my_LaTeX_File.tex
lualatex my_LaTeX_File.tex

```

Fazit

Mit dem biblatex-Paket lassen sich nahezu alle Wünsche erfüllen – allerdings bereitet diese Fülle an Möglichkeiten und die damit verbundene sehr ausführliche Dokumentation vor allem Anfängern, aber auch Studenten in der Endphase ihres Projekts manchmal Kopfzerbrechen.

Ich hoffe, dass das hier vorgestellte Grundgerüst für den einen oder anderen eine Hilfestellung für eigene Ideen bietet.

Literatur

- [1] Philipp Lehman: *The biblatex Package. Programmable Bibliographies and Citations*; Nov. 2011.
- [2] Philipp Lehman: *Zu den Nachteilen von Bib \TeX ; Die \TeX nische Komödie*; 4/2011, S. 66–67; Nov. 2011.
- [3] Rolf Niepraschk: *biblatex – abgehakt!*; *Die \TeX nische Komödie*; 1/2011, S. 55–59; Febr. 2011.
- [4] Dominik Waßenhoven: *Bibliographien erstellen mit biblatex (Teil 1)*; *Die \TeX nische Komödie*; 2/2008, S. 53–75; Mai 2008.
- [5] Dominik Waßenhoven: *Bibliographien erstellen mit biblatex (Teil 2)*; *Die \TeX nische Komödie*; 4/2008, S. 31–51; Nov. 2008.

Meine ersten Schritte mit Lua \TeX

Ralf Meyer

Ich erkläre in einigen Auszügen, wie ich Lua \TeX benutzt habe, um Ergebnisse aus externen Tabellen zu analysieren und in gewohnter \TeX -Schönheit auszugeben. Dann beschreibe ich einige Probleme von Lua \TeX mit Sonderzeichen und mit der Lokalisierung, die mir dabei aufgefallen sind. Insbesondere erwähne ich hier ein Problem mit einem kürzlich erschienenen Artikel von Herbert Voß.

Datenverarbeitung und -ausgabe mit Lua \TeX

Lua \TeX ergänzt das bekannte \TeX um eine Programmiersprache namens Lua. Ich habe Lua \TeX zuerst in meiner Funktion als Kassenwart meines Chores eingesetzt. Neben der Pflege der Mitgliederliste ist es eine meiner Pflichten, den Mitgliedern zum Jahreswechsel eine Spendenbescheinigung über ihre Mitgliedsbeiträge auszustellen und über die Finanzen des Chores zu berichten. Diese Spendenbescheinigungen, den Finanzbericht und die Mitgliederliste habe ich basierend auf Luafunktionen mit \TeX formschön ausgegeben. Dabei sind mir aber auch noch einige Kinderkrankheiten von Lua \TeX aufgefallen – bei Lua \TeX handelt es sich noch um eine β -Version.

Als Mathematiker benutze ich zwar \LaTeX jeden Tag, aber nur als Anwender, und auch bezüglich Skriptsprachen bin ich eher Anfänger. Insofern ist mein Code wahrscheinlich nicht immer vorbildlich, andererseits ist es aber auch ermutigend, was man ohne Lua-Vorkenntnisse mit vertretbarem Aufwand erreichen kann.

Allgemein betrachtet ist mein Problem die Analyse und Ausgabe von Daten, wobei Lua die Analyse und \TeX die Ausgabe übernimmt. In meinem Fall war eine Rückkopplung von \TeX zu Lua nicht notwendig. Ich hätte auch eine beliebige Skriptsprache benutzen können, um aus meinen Daten eine \TeX -Eingabedatei zu erzeugen, und diese dann mit dem althergebrachten \TeX weiterverarbeiten können.

Da ich allerdings auch regelmäßig Einführungskurse in \LaTeX für Studierende anbiete, möchte ich in Sachen \TeX auf dem neuesten Stand bleiben; deshalb habe ich mir vorgenommen, Lua \TeX als einen heißen Kandidaten für die \TeX -Nachfolge einmal auszuprobieren.

Eine Analyse und Verarbeitung von externen Daten mit Lua \TeX kann auch für wissenschaftliche Veröffentlichungen benutzt werden. Man kann mit Lua die rohen Messwerte aus einer Tabellendatei auslesen und daraus auf transparente Weise die Endergebnisse berechnen und als schön gestaltete \LaTeX -Tabelle oder gar als tikzpicture-Umgebung ausgeben. Viele meiner Studierenden nutzen \LaTeX zum Erstellen von Versuchsprotokollen. Die grafische Darstellung der Versuchsergebnisse, etwa mit gnuplot, ist dort in der Regel nicht sehr ansehnlich. Ich hoffe, dass sie mit Lua \TeX bessere Ergebnisse erzielen können. Dies ist freilich ein Projekt für die Zukunft.

Daten einlesen mit Lua

Die Daten, die es zu analysieren gilt, sind in meinem Fall zwei csv-Dateien (csv steht für *comma separated values*). Eine enthält die Mitgliederliste des Chores mit Adressdaten, die andere ist die Liste der Kontobewegungen auf den Vereinskonten, die ich direkt von der Sparkasse bekomme. Die Mitgliederliste des Chores war ursprünglich als eine Excel-Tabelle geführt worden, aber jede Tabellenkalkulation kann ihre Einträge auch als csv-Datei speichern.

In der Mitgliederdatei enthalten die Zeilen jeweils folgende Angaben zu jedem Mitglied: 1) Stimmgruppe, 2) Vorname, 3) Name, 4) Straße, 5) PLZ, 6) Ort, 7) Telefon privat, 8) Telefon dienstlich, 9) Mobil, 10) E-Mail, 11) Beitrag. Damit ich meine csv-Dateien später noch verstehen kann, enthält die erste Zeile eine Erklärung, was die Einträge in den folgenden Zeilen jeweils bedeuten. Der eigentliche Inhalt beginnt in der zweiten Zeile. Lua soll diese Dateien öffnen und in eine Tabelle von Tabellen einlesen, wobei der i, j -te Eintrag der j -te Eintrag in der $i + 1$ -ten Zeile der Datei sein soll. Die erste Zeile kommt in die Einträge Legende, j . In Lua können Tabelleneinträge praktischerweise sowohl mit Zahlen als auch mit Zeichenketten indiziert werden. Es gibt Funktionen, die nur die mit Zahlen indizierten Einträge benutzen und andere, die alle benutzen. Meistens werden die Einträge in Legende, j von mir ignoriert, aber es ist beruhigend, dass sie noch da sind, wenn ich sie mal brauchen sollte.

Nach dem Einlesen der Datei soll also beispielsweise `Mitglieder[8][3]` der Name des achten Mitglieds sein und `Mitglieder[8][11]` der Beitrag, den er zahlen *soll* (und in der Regel, aber nicht immer, auch zahlt). Genauer sind alle diese Einträge vom Datentyp `string`. Lua unterscheidet zwischen der Zahl `37073` und der Zeichenkette `"37073"`.

Da ich vorher noch nicht mit Lua gearbeitet habe, habe ich mich im Internet nach Tutorien umgesehen, um die notwendigen Funktionen kennenzulernen und [2] gefunden. Dort habe ich genug über Lua gelernt, um Dateien zu öffnen und zu lesen und mit Zeichenketten und Tabellen zu operieren. Außerdem habe ich unter [1] eine schon fertige Lua-Funktion `fromCSV()` gefunden, mit der ich die Zeilen meiner `csv`-Datei analysieren kann (der Code wird hier nicht kopiert). Zum Beispiel liefert der Funktionsaufruf

```
fromCSV({"Müller, Peter",Hilbertstraße 4,37075,"Göttingen"})
```

als Wert eine Tabelle mit den vier Einträgen `{Müller, Peter}`, `{Hilbertstraße 4}`, `{37075}` und `{Göttingen}`. Dabei werden die Anführungszeichen entfernt, und Kommas innerhalb von Anführungszeichen werden korrekt behandelt. Die Zeichenkombination `[...]` dient in Lua wie `"..."` zur Begrenzung einer Zeichenkette und kann mit letzteren verschachtelt werden.

Meine Funktion zum Lesen einer `csv`-Datei sieht folgendermaßen aus:

```
1 function ReadCSV (Datei)
2   local Lesen, Zeile, i,v, Daten
3   Daten = {}
4   Datei = Datei .. ".csv"
5   Lesen = io.lines(Datei)
6   Daten.Legende = Lesen()
7   for Zeile in Lesen do
8     table.insert(Daten, Zeile)
9   end
10  for i,v in pairs(Daten) do
11    Daten[i]=fromCSV(v)
12  end
13  return Daten
14 end
```

Die erste Zeile ist eine Funktionsdeklaration in Lua und sagt, dass die Funktion `ReadCSV` heißt und einen Parameter annimmt, der im Code `Datei` heißt. Die zweite Zeile erklärt alle im Folgenden benutzten Variablen als lokal. Dadurch wird verhindert, dass eventuell schon existierende Variablen mit diesen Namen einfach überschrieben werden. Bei Variablen, die nicht weiter benutzt werden sollen, ist das immer zu empfehlen. Zeile 3 initialisiert die Variable `Daten` als leere Tabelle. In Zeile 4 werden Zeichenketten mittels des Operators `..` zusammengehängt. Denn der Aufruf `ReadCSV(Mitglieder)` soll die Datei `Mitglieder.csv` lesen. Zeile 5 benutzt

eine Standardfunktion der Ein- und Ausgabebibliothek von Lua und definiert eine sogenannte Iteratorfunktion mit Namen `Lesen`. Danach liefert jeder Aufruf von `Lesen()` eine Zeile aus der Datei; sind alle Zeilen ausgegeben, so liefert `Lesen()` den booleschen Wert `nil`, der die folgende `for`-Schleife in den Zeilen 7–9 verlässt. Zeile 6 liest die erste Zeile der Datei in den Tabelleneintrag `Daten.Legende`. Zeile 7 liest jeweils die nächste Zeile der Datei und schreibt das Ergebnis in die Variable `Zeile`. In Zeile 8 wird diese Variable in die Tabelle `Daten` eingefügt, und zwar durch Zahlen indiziert. Die Einträge in `Daten` sind zunächst jeweils Zeichenketten. Die folgende `for`-Schleife in den Zeilen 10–12 ersetzt jede dieser Zeichenketten durch eine Tabelle. Danach ist also jeder Eintrag von `Daten` wieder eine Tabelle. Schließlich wird diese Matrix in Zeile 13 als Endergebnis der Funktion zurückgegeben.

Also erzeugt der Funktionsaufruf

```
Mitglieder=ReadCSV("Mitglieder")
```

eine Matrix namens `Mitglieder` aus der Datei `Mitglieder.csv`.

Auf die gleiche Weise erzeuge ich aus der Umsatzliste der Bank eine Matrix der Umsätze.

Die Funktion `ReadCSV` oben reicht für meine Zwecke aus, hat aber einige Einschränkungen. Insbesondere bricht Lua mit einer Fehlermeldung ab, wenn die `csv`-Datei nicht korrekt formatiert ist oder wenn einer der Einträge ein Zeilenende enthält. Denn `io.lines` achtet nicht auf die `"`-Zeichen und liest bis zum nächsten Zeilenende, egal ob die `"`-Zeichen aufgehen.

Auf das Einlesen der Daten folgt eine Menge Lua-Code, um die Kontobewegungen zu klassifizieren: Handelt es sich um einen Mitgliedsbeitrag? Wenn ja, von wem? Ist es ein Honorar für unseren Übungsleiter, eine Mietzahlung, sind es Konzerteinnahmen? Das Ergebnis dieser Analyse wird als weiterer Tabelleneintrag in der Umsatztabelle vermerkt. Dann werden entsprechende Rückverweise in der Mitgliedertabelle angelegt, um später alle Beitragszahlungen eines Mitglieds auflisten und addieren zu können. Diese Daten werden für die Spendenbescheinigungen benötigt. Außerdem werden für jede Kategorie von Zahlungen jeweils die Ein- und Ausgaben dieser Kategorie addiert, beispielsweise alle Einnahmen und Ausgaben unserer Konzerte. Diese Daten werden für den Bericht an die Mitgliederversammlung gebraucht.

Die Analyse der Kontobewegungen ist recht kompliziert, so dass ich hier nicht auf die Details eingehe. Ich spreche nur einige Techniken und Probleme an, die mir verallgemeinerbar erscheinen.

Um einen Umsatz einem Mitglied zuzuordnen, wird überprüft, ob der Name des Mitglieds im Verwendungszweck oder im Namen des Umsatzes auftaucht. Allerdings müssen diese Zeichenketten für den Vergleich erst vorbereitet werden. Denn

in den Kontodaten der Sparkasse werden *fast* alle Wörter groß geschrieben und *fast* alle Umlaute nach dem Muster ä→ae, ß→ss, usw. umgewandelt. Schließlich werden manchmal im Verwendungszweck unerwünschte Leerzeichen eingefügt, die beim Vergleich stören. Deshalb werden der Verwendungszweck und der Name des Zahlungspartners sowie die Namen der Mitglieder in den Tabellen jeweils mit folgender Funktion vereinfacht:

```
local Vereinfache
function Vereinfache(text)
  local Umlaute = {"ä"]="ae", ["ö"]="oe", ["ü"]="ue", ["Ä"]="AE",
    ["Ö"]="OE", ["Ü"]="UE", ["ß"]="ss", [" "]=""}
  for s,t in pairs(Umlaute) do
    text=string.gsub(text,s,t)
  end
  return string.upper(text)
end
```

Diese Funktion ist lokal, das heißt, sie kann von anderen Lua-Modulen später nicht mehr aufgerufen werden. Die Funktion legt eine lokale Tabelle Umlaute mit Substitutionsregeln an. Dann wird in einer for-Schleife jede dieser Substitutionen mit der Systemfunktion `string.gsub` auf die gegebene Zeichenkette angewendet, beginnend mit der Transformation ä→ae. Schließlich wird mit der Funktion `string.upper` die Zeichenkette in Großbuchstaben verwandelt und als Wert der Funktion zurückgegeben.

Eine weitere Transformation betrifft die Eurobeträge aus der Umsatzliste. Dies sind Zeichenketten der Form -12,40 mit Dezimalkomma. Da Lua standardmäßig Dezimalpunkte statt -kommata erwartet, brauche ich eine weitere Funktion `KommaZahl`, die zunächst Kommas durch Punkte ersetzt und dann die resultierende Zeichenkette mit der Funktion `tonumber` in eine Fließkommazahl umwandelt. Letzteres ist übrigens nicht zwingend notwendig, weil Lua solche Typkonvertierungen automatisch durchführt, wenn sie syntaktisch nötig sind, etwa um zwei Zeichenketten zu addieren. Mit folgender Schleife werden die Einträge der Umsatzliste für die Analyse vorbereitet:

```
for i=1,#Umsatz do
  Umsatz[i][9]=KommaZahl(Umsatz[i][9])      -- Betrag
  Umsatz[i].ZWECK=Vereinfache(Umsatz[i][5]) -- Verwendungszweck
  Umsatz[i].NAME=Vereinfache(Umsatz[i][6])  -- Zahlungspartner
end
```

Etwas Ähnliches geschieht mit der Mitgliederliste.

Die Funktion

```
string.find(Text,Suchmuster)
```

überprüft, ob ein Text ein Suchmuster enthält. Die Funktion gibt Anfangs- und Endposition der ersten Textstelle zurück, die dem Suchmuster entspricht, oder nil, wenn das Suchmuster nicht auftritt. Mit

```
if string.find(w.ZWECK,v.NAME) then ...
```

teste ich in einer for-Schleife, ob der (vereinfachte) Nachname eines Mitglieds im (vereinfachten) Verwendungszweck eines Umsatzes enthalten ist.

Hierbei ist noch zu beachten, dass Bindestriche in Suchmustern eine Sonderfunktion haben. Damit Doppelnamen richtig erkannt werden, müssen deshalb beim Vereinfachen der Mitgliedernamen durch

```
string.gsub(name, "-", "%-%")
```

Bindestriche jeweils in %- umgewandelt werden.

Zu beachten ist weiter, dass die string-Bibliothek in Lua nichts von UNICODE versteht. Für Lua ist eine Zeichenkette eine Folge von 8-Bit-Zeichen. Da die deutschen Sonderzeichen in der UTF-8-Kodierung zwei Byte belegen, hat also die Zeichenkette »Bär« für Lua die Länge 4. In der Zeichenkette »Bär« findet Lua das Suchmuster »[ö]«, nicht aber »ö«, weil in der UTF-8-Kodierung das erste der beiden Bytes von ä und ö dasselbe ist.¹ Die oben erwähnte Funktion string.upper verwandelt »Bär« in »BÄR«. In der Funktion Vereinfache oben ist es also wichtig, dass string.upper erst nach den Substitutionen benutzt wird. Man sieht hieran, dass die eingebaute string-Bibliothek in Lua nicht für komplexere Operationen mit deutschen Zeichenketten ausreicht.

Letztlich muss der obige Lua-Code von \TeX ausgeführt werden. Ich habe den Großteil des Lua-Codes in eine separate Lua-Datei Analyse_Umsaetze.lua geschrieben. Diese wird von Lua \TeX mit

```
\directlua{dofile("Analyse_Umsaetze.lua")}
```

eingelesen. Die nichtlokalen Variablen und Funktionen, die dort deklariert werden, können dann später durch weitere Aufrufe von directlua benutzt werden. Für den Finanzbericht ist es zum Beispiel wichtig, die Ein- und Ausgaben der verschiedenen Kategorien (Mitgliedsbeitrag, Honorare, usw.) zu addieren. Diese Daten werden von Analyse_Umsaetze.lua in einer globalen Tabelle Salden gespeichert, wobei

¹ Bei »[ö]« sieht Lua nicht das »ö« als Zeichen, sondern zwei Bytes, aus denen das »[ö]« in UTF-8 zusammengesetzt ist. In Bytes ausgedrückt:

```
<B> < ä > <r> < ö >
```

```
0x42 0xc3 0xa4 0x72 und 0xc3 0xb6
```

Bei »[ö]« wird somit nach den Bytes 0xc3 oder 0xb6 Ausschau gehalten und 0xc3 in »Bär« gefunden, obwohl dort kein »ö« vorkommt.

etwa `Salden.Konzert.m` und `Salden.Konzert.p` die Ausgaben und Einnahmen der Konzerte enthalten. Durch

```
\directlua{tex.sprint(Salden.Konzert.p-Salden.Konzert.m)}
```

kann ich zum Beispiel die Differenz von Einnahmen und Ausgaben aus Konzerten in meinen Text einfügen.

Was kann Lua noch nicht gut?

Ein weiterer Anlass, diesen Artikel zu schreiben, war der Artikel [3], in dem Herbert Voß die Nutzung der Datumsfunktionen in Lua \TeX erklärt. Da ich diese Funktionalität bei meinen Experimenten mit Lua \TeX vermisst hatte, habe ich sie gleich ausprobiert. Leider hat der Code erst nach einer Modifikation funktioniert.

In [3] wird folgender Code empfohlen, um das Datum im deutschen Format anzuzeigen:

```
function Datum()
  os.setlocale("de_DE", "time")
  tex.print(os.date())
end
```

Mit `os.setlocale` werden deutsche Konventionen für die Zeitausgabe eingeschaltet. Interessant ist auch `os.setlocale("de_DE", "numeric")`. Damit sollten dann Dezimalkommata anstelle von Dezimalpunkten beim Konvertieren zwischen Zeichenketten und Zahlen benutzt werden. Leider hatte der Befehl `os.setlocale` oben auf meinem Rechner keine Wirkung. Eine Anfrage an die Mailingliste `luatex@tug.org` brachte mich erfreulich schnell auf die richtige Fährte. Die Bibliothek `os` ist, wie der Name schon andeutet, vom Betriebssystem abhängig. Auf meinem Ubuntu-Rechner finde ich mit dem Shell-Befehl `locale -a` heraus, welche Lokalisierungen möglich sind und stelle fest, dass überhaupt nur deutsche und englische Lokalisierungen vorhanden sind und die deutsche bei mir `de_DE.utf8` und nicht `de_DE` heißt; auf manchen Windowsrechnern heißt sie angeblich `German`, aber das konnte ich nicht überprüfen. Die Funktion `os.setlocale` gibt im Erfolgsfall die eingestellte Lokalisation zurück und ansonsten `nil`. Besser als der Code oben wäre also

```
if not os.setlocale("de_DE.utf8", "time") then
  if not os.setlocale("de_DE", "time") then
    texio.write_nl("Keine deutsche Lokalisation!")
  end
end
```

Das probiert zwei mögliche Namen für die deutsche Lokalisation aus und gibt eine Warnung aus, wenn beide nicht funktionieren.

Mein Eindruck ist allerdings, dass man `os.setlocale` besser gar nicht benutzen sollte. Viele Leute haben mehrere Rechner, einen im Büro, einen zu Hause, vielleicht

noch einen tragbaren dazu. Damit ich meine Lua \TeX -Dateien auf allen meinen Rechnern kompilieren kann, muss ich also für jedes vorkommende Betriebssystem verschiedene Fälle unterscheiden, was unbequem und fehleranfällig ist – und beim nächsten Betriebssystem-Update muss ich vielleicht neuen Code schreiben. Paketautoren müssen noch mehr Fälle berücksichtigen. Das lohnt sich vor allem deshalb nicht, weil die Funktion `os.setlocale` das Lokalisationsproblem ohnehin nur teilweise löst, weil sie keinen Einfluss auf die string-Bibliothek hat. Wie oben erläutert, heißt das, dass Mustererkennung und -substitution in Texten mit Sonderzeichen nicht funktioniert. Früher oder später ist man also darauf angewiesen, leistungsfähigere Bibliotheken zur Lokalisation zu benutzen, die auch das können. Da Lua C-Bibliotheken verwenden kann, stehen also sehr mächtige Funktionen zur Lokalisation und zur Behandlung von Unicode-Zeichenketten in Lua \TeX zur Verfügung. Dies deckt auch die lokalisierte Zeitausgabe ab, so dass man auf `os.setlocale` und `os.time` nicht mehr angewiesen ist. Allerdings habe ich noch nicht genug über Lua und C gelernt, um das selber zu bewerkstelligen. Stattdessen hoffe ich darauf, dass dieses Problem bald von Entwicklern durch ein geeignetes Paket gelöst werden wird.

Ein anderes Problem in Lua \TeX betrifft den Umgang mit Unicode-Zeichen, die im aktuellen Zeichensatz nicht enthalten sind. In der Mitgliederliste verwende ich zum Beispiel im Tabellenkopf das Piktogramm \mathfrak{A} . In Unicode hat dieses Zeichen die Nummer 9742 (U+260E), so dass ich es eigentlich in Lua \TeX direkt eingeben könnte. Aber im Standardzeichensatz, der von `fontspec` geladen wird, ist dieses Zeichen (noch) nicht enthalten. Erstaunlicherweise gibt Lua \TeX in so einem Fall keine Fehlermeldung aus – es ignoriert das fehlende Zeichen einfach. Erst beim Betrachten der Ausgabe fällt der Fehler auf. Ich vermisse in Lua \TeX den Befehl `DeclareUnicodeCharacter`, um Texte, die solche Zeichen enthalten, korrekt auszugeben, ohne für Piktogramme von Hand den Zeichensatz wechseln zu müssen.

Literatur

- [1] Lua-users: *Beispielcode CsvUtils*; <http://lua-users.org/wiki/CsvUtils>; zuletzt besucht: 2012-08-14.
- [2] Lua-users: *Tutorial*; <http://lua-users.org/wiki/TutorialDirectory>; zuletzt besucht: 2012-08-14.
- [3] Herbert Voß: *Datumfunktionen mit Lua \TeX ; Die \TeX nische Komödie*; 24(1), S. 13–19; 2012.

Worttrennungen überprüfen mit Lua \LaTeX

Patrick Gundlach

Für Lua \LaTeX gibt es zwei Pakete, die dem Autor bei der Überprüfung der Silbentrennung helfen. Damit lassen sich fehlerhafte oder unschöne Trennungen vermeiden.

Das Problem

Das Verfahren von \TeX zur automatischen Silbentrennung kann durchaus als sehr gut bezeichnet werden. Dennoch treten immer mal wieder unschöne Worttrennungen auf oder es wird fehlerhaft getrennt, beispielsweise bei Ab-fal-lauf-be-rei-tung, wobei dieser Fall inzwischen korrigiert ist. Problematisch können auch Fälle wie β -Galaktosidasen sein, bei denen Wörter im Quelltext mit einem normalen Bindestrich zusammengefasst werden. Dadurch kann \TeX nicht mehr in den einzelnen Teilwörtern trennen. Korrekt wäre bei Verwendung von babel und der Sprachoption ngerman die Schreibweise β =Galaktosidasen. Oft kann es passieren, dass der Text dann über den rechten Rand hinausragt, ohne dass die Ursache offensichtlich ist:

Außer der bakteriellen Lactase,
die ebenfalls eine β -Galaktosidase
ist, gibt es im Menschen noch
eine Lactase, die nicht ...

Ein Tausch von - durch "=" verhindert ein Überstehen in der zweiten Zeile:

Außer der bakteriellen Lactase,
die ebenfalls eine β -Galaktosi-
dase ist, gibt es im Menschen
noch eine Lactase, die nicht ...

Anzeigen aller Trennstellen

Da diese Probleme nicht offensichtlich sind, hilft ein Einbinden des (Lua \LaTeX) Pakets showhyphens, das alle Trennstellen mit einer kleinen Markierung versieht. Ein einfaches

```
\usepackage{showhyphens}
```

erzeugt die Markierungen:

Außer der bakteriellen Lactase, die ebenfalls eine β -Galaktosidase ist, gibt es im Menschen noch eine Lactase, die nicht...

Hier sieht man deutlich, dass das Wort *Galaktosidase* nicht getrennt wird. Man kann die schon erwähnte Gegenmaßnahme ergreifen. Das, zusammen mit der Dokumentklassenoption `draft`, um übervolle Boxen zu markieren, hilft bei der Suche nach schlechten Trennmöglichkeiten schon sehr viel weiter.

Überprüfen der gewählten Trennstellen

In der Regel möchte man die gewählten Trennstellen erst zum Schluss überprüfen, kurz vor der Fertigstellung eines Textes. Was passiert aber, wenn nach der Überprüfung noch hier und da Textstellen verändert werden? Im schlimmsten Fall muss das gesamte Dokument erneut durchgesehen werden, was aber oftmals aus Zeitmangel unterbleibt. Somit schleichen sich immer wieder schlechte Worttrennungen in einen Text ein.

Für dieses Problem gibt es nun das Paket `lua-check-hyphen`, das auch mit Lua \LaTeX funktioniert. Es erstellt während des Übersetzungsvorgangs mit Lua \LaTeX eine Liste der Worttrennungen in der Log-Datei und in einer externen Datei mit der Endung `.uhy`.

In alten Zeiten, als das Wü-
schen noch geholfen hat, lebte ein-
mal ein König, der hatte wunder-
schöne Töchter. Die Jüngste von
ihnen war so schön, dass die Son-
ne selber, die doch so vieles schon
gesehen hat, sich wunderte, so
oft sie ihr ins Gesicht schien. Nahe
bei dem Schlosse war ein großer,
dunkler Wald, und mitten darin,
unter einer alten Linde, war ein
Brunnen.

Abbildung 1: Ausschnitt aus dem Froschkönig (Gebrüder Grimm)

Der Text aus Abbildung 1 enthält vier Worttrennungen, die in der Datei `frosch.uhy` aufgelistet sind:

```
Son-ne
Wün-schen
wunder-schöne
ein-mal
```

Die Worttrennungen werden nur einmal aufgeführt, auch wenn sie mehrfach im Dokument vorkommen. Mit dieser Datei lassen sich die verwendeten Trennungen überprüfen.

Um jedoch nicht jedes Mal dieselben Wörter zu überprüfen, kann man eine sogenannte *whitelist* erstellen. Das ist eine Datei in genau demselben Format wie die erzeugte Datei. In die *whitelist* werden alle gültigen Trennungen eingetragen, beispielsweise:

```
Wün-schen
wunder-schöne
```

Diese Datei wird dann im Dokument mit

```
\LuaCheckHyphen{whitelist=gutetrennungen.txt}
```

in der Dokumentpräambel eingebunden. Worttrennungen, die in dieser Datei vorhanden sind, werden ignoriert. Das hat zur Folge, dass die Datei mit den unbekannteren Worttrennungen immer weniger Einträge enthält und somit weniger Aufwand betrieben werden muss, um gültige Trennstellen zu überprüfen.

Um die unbekannteren Trennstellen zusätzlich im Dokument zu markieren, kann die Option *mark* beim Laden des Pakets oder bei `\LuaCheckHyphen` angegeben werden:

```
\documentclass[ngerman]{article}
\usepackage{fontspec,babel,lua-check-hyphen}
\LuaCheckHyphen{mark,whitelist=gutetrennungen.txt}
\begin{document}
In alten Zeiten, als das Wünschen noch geholfen hat, lebte einmal ein
König, der hatte wunderschöne Töchter. Die Jüngste von ihnen war so
schön, dass die Sonne selber, die doch so vieles schon gesehen hat,
sich verwunderte, so oft sie ihr ins Gesicht schien. Nahe bei dem
Schlosse war ein großer, dunkler Wald, und mitten darin, unter einer
alten Linde, war ein Brunnen.
\end{document}
```

Die Datei `gutetrennungen.txt` hat den Inhalt wie oben:

```
Wün-schen
wunder-schöne
```

Daraus wird ein PDF mit Markierungen erzeugt:

In alten Zeiten, als das Wünschen noch geholfen hat, lebte einmal ein König, der hatte wunderschöne Töchter. Die Jüngste von ihnen war so schön, dass die Sonne selber, die doch so vieles schon gesehen hat, sich verwunderte, so oft sie ihr ins Gesicht schien. Nahe bei dem Schlosse war ein großer, dunkler Wald, und mitten darin, unter einer alten Linde, war ein Brunnen.

Alle unbekanntem Worttrennungen werden mit einem Kästchen markiert. Wird die Option `final` dem Paket übergeben (durch eine globale Option bei der Dokumentenklasse oder durch eine der beiden genannten Möglichkeiten), ist das Paket nicht aktiv und es werden weder Markierungen gezeichnet, noch werden unbekannte Trennungen ausgegeben.

Anmerkung

Die beiden Pakete zeigen die Flexibilität, die Lua_T_EX bietet. Mit einfachen Mitteln kann Programmcode während des Laufs ausgeführt werden. Das Paket `lua-check-hyphen` beispielsweise führt Programmcode aus, kurz bevor die Absätze umbrochen werden, um alle möglichen Trennstellen zu ermitteln. Zusätzlich wird Programmcode nach dem Absatzumbruch ausgeführt, um zu ermitteln, welche dieser Trennstellen genutzt wurden und um die Markierungen zu zeichnen. Die beiden genannten Pakete sind noch in der Entstehungsphase. Es ist möglich, dass sich dort noch der eine oder andere Fehler versteckt hat. Fehlermeldungen und Verbesserungsvorschläge nimmt der Autor gerne entgegen.

T_EX Gyre Pagella Math

Herbert Voß

Mit der aktuellen Version von T_EXLive steht eine weitere freie Mathematikschrift als OpenType zur Verfügung, die Pagella Math (Palatino) aus dem Projekt T_EX Gyre (<http://www.gust.org.pl/projects/e-foundry/tex-gyre/>). Dies ist ein

wichtiger Schritt zur Vervollständigung dieser Schrift. Das folgende Beispiel zeigt eine willkürliche Zusammensetzung von Zeichen der Pagella:

Theorem 1 (Residuum). Für eine in einer punktierten Kreisscheibe $D \setminus \{a\}$ analytische Funktion f definiert man das *Residuum* im Punkt a als

$$\operatorname{Res}_{z=a} f(z) = \operatorname{Res}_a f = \frac{1}{2\pi i} \int_C f(z) dz,$$

wobei $C \subset D \setminus \{a\}$ ein geschlossener Weg mit $n(C, a) = 1$ ist (z. B. ein entgegen dem Uhrzeigersinn durchlaufener Kreis).

ΑΑΔΝΒCDEΣΕFFGHIIJKLΜNOΘΩΡΦΠΕQRSTUVWXYZZ

ABCDabcd1234

ααββcδdδεεεζζξγγηηιικκλλμμννθθoσςφφϖρρρqrσττπυμννσσωωω

xyz[∞] ∝ ∅y = f(x)

$$\Sigma \int \Pi \prod \int \sum \sum_a^b \int_a^b \prod_a^b \sum_a^b \int_a^b \prod_a^b$$

Die Einbindung in ein Dokument, welches mit Xe_LT_EX oder Lua_LT_EX zu übersetzen ist, könnte wie folgt geschehen:

```
\usepackage{fontspec}
\usepackage{unicode-math}
[...]
\defaultfontfeatures{Ligatures=TeX}
\setmainfont[
  BoldFont=texgyrepagella-bold.otf,
  ItalicFont=texgyrepagella-italic.otf,
  BoldItalicFont=texgyrepagella-bolditalic.otf]{texgyrepagella-regular.otf}
\setmathfont{texgyrepagella-math.otf}
[...]
```

Die für das Beispiel verwendeten Schriften sind:

IXOXA+TeXGyrePagella-Bold	CID Type 0C	yes yes yes	7	0
OKQJUY+TeXGyrePagella-Regular	CID Type 0C	yes yes yes	8	0
TROBYC+TGPagellaMath-Regular	CID Type 0C	yes yes yes	9	0
QPUAJN+TeXGyrePagella-Italic	CID Type 0C	yes yes yes	10	0

Von fremden Bühnen

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss vor. Die Liste folgt der umgekehrten chronologischen Reihenfolge. Reine Updates werden nicht aufgeführt. Die Angaben können auf der *ctan-ann*-Mailingliste verfolgt werden, die auch über Twitter und Identi.ca als @ctanannounce verfügbar sind.

odsfile von *Michal Hoftich* ist ein Paket, mit dem OpenDocument-Tabellen mit Lua \TeX eingelesen und als \TeX -Tabellen gesetzt werden können.

CTAN:macros/latex/contrib/odsfile

poetrytex von *Sam Whited* dient zum Setzen von Anthologien.

CTAN:macros/latex/contrib/poetrytex

bgug von *Jim J. Green* ist ein Font, der ein Mengenzeichen in mehreren Größen für die Freges Begriffsschrift bereithält.

CTAN:fonts/bgug

lastbib von *Berik I. Tuleuov* ermöglicht den Zugriff auf den Bibliographiezähler mit dem Befehl `\LastBib`.

CTAN:macros/latex/contrib/lastbib

frege von *Quirin Pamp* dient zum Setzen von Freges Begriffsschrift in \TeX .

CTAN:macros/latex/contrib/frege

biblatex-phys von *Joseph Wright* ist ein biblatex-Stil gemäß AIP/APS.

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

enotez von *Clemens Niederberger* ermöglicht geschachtelte Endnoten. `hyperref` wird unterstützt und die Endnotenliste ist konfigurierbar.

CTAN:macros/latex/contrib/enotez

fitr von *Don P. Story* ermöglicht das Öffnen eines PDFs für die FitR-Ansicht.

CTAN:macros/latex/contrib/fitr

minion2newtx von *Michael Sharpe* enthält die Fontmetriken für die Schriftart MinionPro v2 zur Verwendung mit `newtxmath`.

CTAN:fonts/minion2newtx

footnotebackref von *Holger Lange* fügt Hyperlinks von Fußnoten zu den Fußnotenankern ein.

CTAN:macros/latex/contrib/footnotebackref

embrac von *Clemens Niederberger* definiert `\emph` und `\textit` so, dass runde und eckige Klammern nicht kursiv gesetzt werden.

CTAN:macros/latex/contrib/embrac

incgraph von *Thomas F. Sturm* dient zum Einfügen von Grafiken im Seitenformat. Hauptanwendungsfall sind mehrere eingescannte Seiten, die in einem PDF gesammelt werden sollen.

CTAN:macros/latex/contrib/incgraph

lua-check-hyphen von *Patrick Gundlach* ist eine Rechtschreibkontrolle für Dokumente in Lua_T_EX.

CTAN:macros/luatex/latex/lua-check-hyphen

typeface von *Geoffrey Jones* vereinfacht das Setup für Type1-Fonts.

CTAN:macros/latex/contrib/typeface

foreign von *Philip G. Ratcliffe* dient dazu, ausländische Wörter hervorzuheben, z. B. kursiv.

CTAN:macros/latex/contrib/foreign

basque-book von *Edorta Ibarra* ist eine Anpassung der Book-Klasse für Dokumente in baskischer Sprache.

CTAN:macros/latex/contrib/basque-book

fixmetodonotes von *Gioele Barabucci* stellt Befehle für Anmerkungen zur Überarbeitung von Dokumenten bereit (»fixme«, »todo«, »check«, »cite«).

CTAN:macros/latex/contrib/fixmetodonotes

python von *Martin R. Ehmsen* dient zum Einbetten von Python-Quelltexten in \LaTeX .

CTAN:macros/latex/contrib/python

csvtolatex von *Hannes Pietsch* ist ein Konverter für CVS-Tabellen nach \LaTeX .

CTAN:support/csvtolatex

calculator von *Robert Fuster* implementiert algebraische Funktionen.

CTAN:macros/latex/contrib/calculator

bakoma-mac von *Vasili Malyshev* ist eine Portierung von bakoma für Mac OS X (ab Version 10.5/Leopard).

CTAN:systems/mac/bakoma-mac

adobecaslön von *Boris Veytsman* enthält die \LaTeX -Unterstützung für die Schriftart Adobe Caslon.

CTAN:fonts/psfonts/adobe/adobecaslön

xpinyin von *Qing Lee* dient zum Setzen der chinesischen Pinyin-Umschrift.

CTAN:macros/latex/contrib/xpinyin

lmacs von *D. P. Story* ist ein einfaches Paket zum Zusammenfassen der Präambel in einer externen Datei.

CTAN:macros/latex/contrib/lmacs

autonom von *Patrick Häcker* nummeriert nur die mathematischen Formeln, die referenziert wurden.

CTAN:macros/latex/contrib/autonom

xpunctuate von *Philip G. Ratcliffe* ist eine Erweiterung des Pakets *xspace*.

CTAN:macros/latex/contrib/xpunctuate

fnpct von *Clemens Niederberger* verändert Formatierung und Kerning von multiplen und geschachtelten Fußnoten.

CTAN:macros/latex/contrib/fnpct

zhnumber von *Qing Lee* gibt mithilfe von \LaTeX 3 Zahlen auf Chinesisch aus.

CTAN:macros/latex/contrib/zhnumber

chickenize von *Arno Trautmann* ist ein Paket, mit dem man \LaTeX -Dokumente bearbeiten kann (überwiegend spielerisch, aber auch ernsthaft).

CTAN:macros/luatex/generic/chickenize

multienv von *Martin Scharrer* stellt eine alternative Syntax zum Definieren von geschachtelten Umgebungen bereit.

CTAN:macros/latex/contrib/multienv

harnon-cv von *William Hart* ist eine Klasse zum Erstellen eines Lebenslaufs mit einer vertikalen Timeline, die die Berufserfahrung anzeigt.

CTAN:macros/latex/contrib/harnon-cv

basque-date von *Edorta Ibarra* gibt das aktuelle Datum in baskischer Sprache nach den Regeln der Königlichen Akademie der Baskischen Sprache aus.

CTAN:macros/latex/contrib/basque-date

dynblocks von *Claudio Fiandrino* erleichtert die Definition dynamischer Blocks in einer Präsentation mit *beamer*.

CTAN:macros/latex/contrib/beamer-contrib/dynblocks

statrep von *Tim Arnold* dient zum Verarbeiten von *Datastep and Sascode* (SAS) mit \LaTeX .

CTAN:/makros/latex/contrib/statrep

famt von *Bhagyashree Chavan*, *Sadaf Desai* und *Shilpa Mahadik* ist eine Klasse für Berichte der *Finolex Academy of Management and Technology*, Ratnagari.

CTAN:macros/latex/contrib/famt

eitl von *Rolf Niepraschk* und *Herbert Voß* ist ein Skript zur leichteren Installation der aktuellen Version von \TeX Live über das Internet unter Linux.

CTAN:support/texlive/eitl

dowith von *Uwe Lück* stellt Makros bereit, mit denen man denselben Befehl auf alle Elemente einer selbstdefinierten Liste anwenden kann.

CTAN:macros/generic/dowith

biblatex-bwl von *Herbert Voß* ist ein *biblatex*-Stil, der für den Fachbereich Wirtschaft der FU Berlin entwickelt wurde.

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

latexfileinfo-pkgs von *Uwe Lück* besteht aus einer HTML-Datei, die Metadaten zu \LaTeX -Paketen anzeigt.

CTAN:info/latexfileinfo-pkgs

lgrx von *Günter Milde* enthält die Unterstützung, um griechische Texte mit dem LGR-Encoding zu setzen, den De-facto-Standard von *babel*.

CTAN:macros/latex/contrib/lgrx

codicefiscaleitaliano von *Claudio Beccari* dient dazu, die italienische Steuernummer auf Richtigkeit zu prüfen.

CTAN:macros/latex/contrib/codicefiscaleitaliano

lstaddons von *Martin Scharrer* bringt Erweiterungen, mit denen die Ausgabe des Pakets *listings* verbessert wird.

CTAN:macros/latex/contrib/lstaddons

mwe von *Martin Scharrer* dient zum Erstellen von Minimalbeispielen.

CTAN:macros/latex/contrib/mwe

articleingud von *Omar Salazar Morales* ist eine Klasse für die Zeitschrift der *Universidad Distrital Francisco José de Caldas*.

CTAN:macros/latex/contrib/articleingud

imsproc von *Vafa Khalighi* dient zum Setzen von Tagungsberichten für die *Iranian Mathematical Society* (IMS) mit \XeTeX .

CTAN:macros/xetex/latex/imscls

substitutefont von *Günter Milde* ermöglicht es vorzugeben, welche Schriftfamilie eine andere in einem Dokument bei Bedarf ersetzen kann.

CTAN:macros/latex/contrib/substitutefont

shadowtext von *Yori Zwols* dient dazu, einen Text mit einem Schattenwurf zu versehen.

CTAN:macros/latex/contrib/shadowtext

biblatex-swiss-legal von *Adrien Vion* stellt die notwendigen Anpassungen zu *biblatex* für juristische Bibliographien bereit, wie sie in der Schweiz benötigt werden, vorerst nur auf Französisch, eine deutsche und eine italienische Übersetzung ist aber in Vorbereitung.

CTAN:macros/latex/expt1/biblatex-contrib/biblatex-swiss-legal

pgfkeyx von *Ahmed Musa* enthält einige Korrekturen und Erweiterungen für das Paket *pgfkeys*.

CTAN:macros/latex/contrib/pgfkeyx

pgf-blur von *Martin Giese* ist eine Bibliothek für *pgf/Tikz*, mit der man in Grafiken einen Schatten erzeugen kann.

CTAN:graphics/pgf/contrib/pgf-blur

luatex-ja von *Hironori Kitagawa* dient zum Setzen von japanischen Texten mit \LaTeX .

CTAN:macros/luatex/generic/luatexja

countriesofeuropa von *Herbert Voß* enthält die Unterstützung für den gleichnamigen Type-1-Font, der unter der Open Font Licence steht und alle Länder Europas als einzelne Zeichen enthält.

CTAN:fonts/countriesofeuropa

regexpatch von *Enrico Gregorio* verallgemeinert die Patches, die in Philip Lehmans Paket *etoolbox* enthalten sind.

CTAN:macros/latex/contrib/regexpatch

romansh von *Claudio Beccari* enthält die Sprachdefinitionsdateien für *babel* und *polyglossia* zu der Sprache Rätoromanisch.

CTAN:macros/latex/contrib/babel-contrib/romansh

mycv von *Andrea Ghersi* ist eine weitere Klasse zum Schreiben von Lebensläufen. Sie unterstützt mehrere Layouts sowie das Paket *tikz*.

CTAN:macros/latex/contrib/mycv

hobete von *Tobias Görlach* ist das *beamer*-Theme für die Universität Hohenheim.

CTAN:macros/latex/contrib/hobete

ifthenx von *Geoffrey Jones* erweitert das Paket *ifthen* von *David Carlisle* um einige weitere Tests.

CTAN:macros/latex/contrib/ifthenx

geschichtsftrl von *Jonathan Zachhuber* ist ein $\text{Bib}\TeX$ -Stil, der für den Fachbereich Geschichte der Universität Freiburg entwickelt wurde.

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

coolthms von *Michael Fütterer* und *Jonathan Zachhuber* dient zum Referenzieren von Listenelementen, die in einer theoremartigen Umgebung geschachtelt vorliegen.

CTAN:macros/latex/contrib/coolthms

tikzpfleile von *Michael Fütterer* und *Jonathan Zachhuber* sorgt dafür, dass Pfeile, die mit *PGF/TikZ* gezeichnet werden, genauso aussehen wie Pfeile, die \TeX im Mathematikmodus erzeugt, um das Dokument im ganzen konsistent erscheinen zu lassen.

CTAN:graphics/pgf/contrib/tikzpfleile

nicefilelist von *Uwe Lück* dient, wie *longnamefilelist*, dazu, die Liste, die von `\listfiles` ausgegeben wird, schöner auszugeben.

CTAN:macros/latex/contrib/nicefilelist

appendixnumberbeamer von *Jerome Lelong* sorgt dafür, dass der Anhang in einer Präsentation mit *beamer* als Folie mitgezählt wird.

CTAN:macros/latex/contrib/beamer-contrib/appendixnumberbeamer

monofill von *Uwe Lück* kümmert sich um die Ausrichtung von reinem Text bzw. Festbreitenschrift.

CTAN:macros/latex/contrib/monofill

usebib von *Enrico Gregorio* ist ein Hack, der unter $\text{Bib}\TeX$ den Zugriff auf einzelne Felder in einer `.bib`-Datei erlaubt.

CTAN:macros/latex/contrib/usebib

typeoutfileinfo von *Uwe Lück* will ein minimalistischer Ersatz für die Pakete *latexfileversion* und *ltxfileinfo* sein: Ein Shell-Skript, das \TeX für eine lokale Datei aufruft und mit Hilfe des Pakets *readprov* Informationen über diese Datei ausgibt.

CTAN:support/typeoutfileinfo

vocaltract von *Dimitrios Ververidis* dient zum Zeichnen des Vokaltrakts mit Hilfe von \TeX und *PSTricks*.

CTAN:graphics/pstricks/contrib/vocaltract

Bücher und Rezensionen

Edition *dante* – Neuerscheinung

Herbert Voß:

Presentations with \LaTeX ; 1. Auflage;

DANTE e.V., Lehmanns Media, 2012

212 Seiten; ISBN 978-3-86541-496-0;

19,95 € (Ladenpreis) bzw. 17,- € für Mitglieder
von DANTE e.V., jeweils versandkostenfrei.



Bestellung

Bitte schicken Sie eine E-Mail an office@dante.de mit Angabe von *Name, Anschrift, Mitgliedsnummer* und *Anzahl der Exemplare*, und überweisen Sie den Betrag auf das Konto von DANTE e.V. oder bezahlen Sie per PayPal. Die Kontonummer finden Sie am Ende dieses Heftes und Informationen zu PayPal auf <http://www.dante.de/index/Intern/Zahlung.html>.

Bitte beachten Sie für Bestellungen bei DANTE e.V. folgende Informationen zum Widerrufsrecht: Käufer können bei Bestellungen per E-Mail, Internet, Brief oder Telefon den Kaufvertrag innerhalb einer Frist von 14 Tagen ab Erhalt der Ware per Brief, Fax oder E-Mail oder durch Rücksendung der Ware widerrufen (siehe Kontaktadresse). Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs oder der Ware. Bei einem Bestellwert bis 40,- € hat der Besteller die Rücksendekosten zu tragen. Bei Verschlechterung der Ware, die über die übliche Prüfung der Ware hinausgeht, hat der Besteller gegebenenfalls Wertersatz zu leisten.

Spielplan

Termine

2012

8. 10. – 12. 10. **EuroT_EX 2012**, 6th International ConT_EXt Meeting und DANTE-Herbsttagung mit 47. Mitgliederversammlung
Breskens (Niederlande)
<http://meeting.contextgarden.net/2012/>

2013

16. 3. – 17. 3. **Chemnitzer Linuxtage 2013**
Technische Universität Chemnitz
09107 Chemnitz
<http://www.chemnitzer.linux-tage.de>
22. 5. – 25. 5. **Linuxtag Berlin**
Messegelände
14055 Berlin
<http://www.linuxtag.org/>

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßige Treffen von \TeX -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
Ort derzeit wechselnd
Zweiter Donnerstag im Monat, 19.00 Uhr

Bremen

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

Darmstadt

Karlheinz Geyer, geyerk.fv.tu@nds.tu-darmstadt.de, <http://www.da-tex.org>
Wechselnder Ort
Erster Freitag im Monat, ab 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

Hamburg

Lothar Fröhling, lothar@thefroehlings.de
Restaurant Sandstuv, Neue Straße 17, 21073 Hamburg-Harburg
Letzter Dienstag im Monat, 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

Helmut Siegert

Institut für Kristallographie, Zülpicher Straße 49b, 50674 Köln

Letzter Dienstag im Monat, 19.30 Uhr

München

Uwe Siart, uwe.siart@tum.de, <http://www.siart.de/typografie/stammtisch.xhtml>

Erste Woche des Monats 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, tex-stammtisch-trier@schoenerpublizieren.de

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 \TeX 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

Präsident: Martin Sievers president@dante.de

Vizepräsident: Herbert Voß vice-president@dante.de

Schatzmeister: Klaus Höppner treasurer@dante.de

Schriftführer: Manfred Lotz secretary@dante.de

Beisitzer:
Patrick Gundlach
Volker RW Schaa
Uwe Ziegenhagen

Ehrenmitglieder

Peter Sandner	22. März 1990	Klaus Thull †	22. März 1990
Yannis Haralambous	5. Sep. 1991	Barbara Beeton	27. Feb. 1997
Luzia Dietsche	27. Feb. 1997	Donald E. Knuth	27. Feb. 1997
Eberhard Mattes	27. Feb. 1997	Hermann Zapf	19. Sep. 1999

Server

DANTE: <http://www.dante.de/> (Rainer Schöpf, Joachim Schrodt)

CTAN: <http://mirror.ctan.org/>

FAQ

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

\TeX : <http://projekte.dante.de/DanteFAQ/WebHome>

Autoren/Organisatoren

Jürgen Fenn Friedensallee 174/20 63263 Neu-Isenburg juergen.fenn@gmx.de	[73]	Clemens Niederberger Am Burgrain 3 71083 Herrenberg contact@mychemistry.eu	[22]
Patrick Gundlach Eisenacher Straße 101 10781 Berlin Germany	[68]	Christine Römer Institut für germanistische Sprachwissenschaft FSU Jena Christine.Roemer@uni-jena.de	[31]
Markus Kohm Freiherr-von-Drais-Straße 66 68535 Edingen-Neckarhausen komascript@gmx.info	[10, 17]	Martin Sievers Im Treff 8 54296 Trier martin@dante.de	[4, 6]
Manuel Kühner Stömmmerstr. 9c 85055 Ingolstadt manuel.kuehner@gmail.com	[55]	Herbert Voß Wasgenstraße 21 14129 Berlin herbert@dante.de	[3, 6, 71, 78]
Joachim Lammarsch Joachim.Lammarsch@urz.uni-heidelberg.de	[8]	Uwe Ziegenhagen Köln uwe@ziegenhagen.info	[40, 46]
Ralf Meyer Mathematisches Institut Bunsenstraße 3-5 37073 Göttingen rameyer@uni-math.gwdg.de	[60]		

Die T_EXnische Komödie

24. Jahrgang Heft 3/2012 August 2012

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 6 Herbsttagung
- 8 Klaus Thull

Bretter, die die Welt bedeuten

- 10 KOMA-Script wird volljährig
- 17 Kopfzeilentricks mit `scrpage2`
- 22 Das `xtemplate` Paket
- 31 Lyrik mit \LaTeX
- 40 *europass* Lebensläufe setzen mit \LaTeX
- 50 Briefumschläge beschriften und frankieren mit \LaTeX
- 55 Randnotizen im Literaturverzeichnis
- 60 Meine ersten Schritte mit Lua \TeX
- 68 Worttrennungen überprüfen mit Lua \LaTeX
- 71 \TeX Gyre Pagella Math

Von fremden Bühnen

- 73 Neue Pakete auf CTAN

Bücher und Rezensionen

- 78 Edition *dante* – Neuerscheinung

Spielplan

- 79 Termine
- 80 Stammtische

Adressen

- 83 Autoren/Organisatoren