

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

24. Jahrgang Heft 4/2012 November 2012

4/2012

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
dtkred@dante.de (Redaktion)

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

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit: Rudolf Herrmann Gert Ingold Eberhard Lisse
Rolf Niepraschk Heiko Oberdiek Christine Römer
Gert Seidl Martin Sievers

Redaktionsschluss für Heft 1/2013: 15. Januar 2013

ISSN 1434-5897

Editorial

Liebe Leserinnen und liebe Leser,

der 24. Jahrgang unserer Vereinszeitschrift »Die \TeX nische Komödie« wird mit dieser Ausgabe beendet. In diesen vielen Erscheinungsjahren gab es immer wieder mal »Dürreperioden«, in denen der jeweilige Chefredakteur mit dem standardmäßigen Aufruf hausieren ging, doch bitte Beiträge einzureichen. Dieser Jahrgang kann als sehr gut bezeichnet werden, zumal diese Ausgabe einen ungewöhnlichen Umfang aufweist. Dies ist vor allen Dingen den Tagungsberichten geschuldet, die sich auf die TUG 2012 und die gerade zu Ende gegangene Euro \TeX 2012 beziehen.

Den Hauptteil dieser Ausgabe bildet der Beitrag von Walter Entenmann zum Thema »Deutsche Schriften«. Wer auf unserer Tagung in Bremen anwesend war, wird sich vielleicht an den Vortrag von ihm erinnern. Insbesondere die älteren Mitglieder von DANTE e.V. werden häufig mit »Sütterlin« ihre eigene Schulzeit oder Kindheit verbinden. Der Beitrag beschreibt eingehend die Vorgehensweise bei der Entwicklung einer Schreibrift mit METAFONT.

Markus Kohm gibt eine ausführliche Einführung in die Gestaltung eines Briefkopfes für sein eigenes Paket `scr1ttr2`, welches bekanntermaßen von vielen \TeX -Nutzern angewendet wird. Christine Römer beschreibt exemplarisch die Anwendung der verschiedenen Pakete zur Erstellung eines Glossars. Daneben gibt es noch zwei kurze Berichte zum Thema Schriften, wobei Sie zur Information auch eine Übersicht über die neue Mathematiksschrift TG Termes aus dem Projekt \TeX Gyre finden; die Fertigstellung ist wieder ein wichtiger Schritt zur Vervollständigung der PostScript-Schriften im Format OpenType.

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,

T_EX ist international – diese zunächst trockene und abstrakte Aussage wird regelmäßig auf Tagungen mit Leben gefüllt. So auch auf der diesjährigen EuroT_EX, bei der Teilnehmerinnen und Teilnehmer aus insgesamt zwölf Nationen vertreten waren. Über Grenzen hinweg haben sich in den vergangenen Jahren viele Kontakte und Freundschaften entwickelt. Ich denke, dass auch dies für sich genommen ein Erfolg der T_EX-Gemeinde ist.

Einen ausführlichen Bericht von Uwe Ziegenhagen zur EuroT_EX finden Sie in dieser Ausgabe. Für mich persönlich waren die Demonstration eines komplexen bidirektionalen Satzes am Beispiel von Sanskrit sowie der japanische Textsatz mit Laufrichtungen von rechts nach links und von unten nach oben ein weiterer Beleg für die große Stärke des T_EX-Systems insgesamt. In meinem beruflichen Umfeld sind unterschiedliche Laufrichtungen im Dokument oft geäußerte Wünsche bzw. Notwendigkeiten, die mit anderer Software – auch kommerzieller – nicht oder nur mäßig erfüllt werden können.

Neben interessanten Vortragsthemen bietet eine mehrtägige Tagung auch immer die Möglichkeit zum Ideenaustausch und zur Diskussion von Angesicht zu Angesicht – nicht nur innerhalb des Vorstands. Ein Thema dabei war auch diesmal die Zukunft von DANTE e.V.

»Wo sehen Sie sich in zehn Jahren?« ist eine oft gestellte Frage in Bewerbungsgesprächen. Doch auch für unseren Verein lohnt es sich, trotz oder gerade wegen seiner fast 25-jährigen Vergangenheit diese Frage zu stellen – und zu beantworten. Dafür bedarf es aus meiner Sicht nicht nur einer Menge Ideen – die hat der Verein über seine Mitglieder zweifelsohne –, sondern vor allem einer systematischen Bestandsaufnahme und Planung. Wir haben im Vorstand damit während der EuroT_EX begonnen und werden dies in den nächsten Monaten fortsetzen. Einbezogen werden sollen dabei interne Strukturen und Abläufe genauso wie Tagungen und der Außenauftritt insgesamt. Wir wollen damit den geänderten Gegebenheiten Rechnung tragen und den Verein fit für kommende Herausforderungen machen.

Freiwilliges Engagement ist dabei sicherlich ein Dauerthema. Auch der Erfolg von \TeX als Software wäre ohne den großen Einsatz einzelner Personen nicht denkbar. Aus diesem Grund vergibt DANTE e.V. seit 2010 jährlich einen Ehrenpreis für besonderen » \TeX nischen« Einsatz im Alltag. Die bisherigen Preisträger Heiko Oberdiek, Rolf Niepraschk und Ulrike Fischer sind in diesem Sinne auch herausragende Vertreter unseres Vereins, dessen Zweck es doch ist, »alle Interessenten in geeigneter Weise bei der Anwendung von \TeX -Software zu unterstützen, insbesondere durch Erfahrungsaustausch mit den Anwendern. Ferner soll die weitere Entwicklung von \TeX unterstützt werden.« (DANTE-Satzung, § 2 (1))

Für 2013 steht die Entscheidung bezüglich der Auszeichnung noch aus. Vielleicht fällt Ihnen ja spontan jemand ein, der immer wieder in Foren, auf Mailinglisten, Tagungen oder bei anderen Gelegenheiten als besonders fleißiger und kompetenter » \TeX -Botschafter« aufgefallen ist. Gerne können Sie mir bzw. dem Vorstand Ihren Vorschlag mit kurzer Begründung übersenden. Die Preisverleihung wird während der Frühjahrstagung in Gießen stattfinden, für die Sie in dieser Ausgabe auch die Einladung finden. Wir sind derzeit in der glücklichen Lage, dass es viele Mitglieder gibt, die als lokale Organisatoren eine DANTE-Tagung an ihrer Hochschule ausrichten wollen. Dafür an dieser Stelle ein herzliches Dankeschön.

Ein Thema auf der kommenden Tagung wird auch die FAQ bzw. das Wiki sein. Nachdem die Inhalte vor einigen Jahren auf Wunsch vieler Mitglieder mit entsprechendem Aufwand in eine »Mitmachform« überführt worden sind, fehlen leider diejenigen, die ein solches Angebot durch Aktualisierungen und Ergänzungen am Leben halten. Manch andere (kommerzielle) Plattform hat den klassischen »häufig gestellten Fragen« auf dante.de mittlerweile den Rang abgelaufen. Dennoch rechtfertigen aus meiner Sicht die Zugriffszahlen und vor allem unser eigener Anspruch als Anwendervereinigung \TeX e.V. eine erhöhte Anstrengung, um den Anwendern auch an dieser Stelle kompetent mit Rat und Tat zur Seite zu stehen. Dies ist unser »Kerngeschäft«, das wir etwas zu sehr vernachlässigt haben, während sich die \TeX -Welt um uns herum weiterentwickelt hat.

In Gießen möchten wir über das Konzept sprechen, andererseits aber auch Inhalte in einer Art »FAQ-Sprint« überarbeiten. Sie sind jetzt schon herzlich eingeladen, sich an diesen Arbeiten und der begleitenden Diskussion konstruktiv zu beteiligen.

Auch wenn der goldene Oktober in diesen Tagen gerade erst einem kühlen November Platz macht, so ist dies doch schon das letzte Grußwort im Jahr 2012. Daher möchte ich Ihnen, Ihren Familien und Freunden bereits jetzt eine besinnliche Adventszeit und ein gesegnetes Weihnachtsfest wünschen. Ich hoffe, wir sehen uns 2013 gesund und voller Tatendrang auf einer der zahlreichen \TeX -Tagungen wieder.

Herzlichst Ihr/Euer
Martin Sievers

Beschlüsse der 47. Mitgliederversammlung von DANTE e.V. am 8. Oktober 2012 in Breskens (NL)

Manfred Lotz

Zeit: 8. Oktober 2012, 18:00 Uhr – 18:21 Uhr
Ort: de Platte Knoop
Boulevard 38
4511 AC Breskens, Niederlande
Teilnehmer: 21 (anhand der ausgegebenen Stimmkarten)
Leitung: Martin Sievers (Vorsitzender von DANTE e.V.)
Protokollant: Manfred Lotz (Schriftführer von DANTE e.V.)

Die Mitgliederversammlung wurde satzungsgemäß eingeladen und ist beschlussfähig.

TOP 1: Begrüßung, Tagesordnung und Vorstellung des Vorstands

TOP 1.1: Begrüßung und Vorstellung

Martin Sievers begrüßt die Teilnehmer der 47. Mitgliederversammlung von DANTE e.V. in Breskens und stellt die Tagesordnung vor:

1. Begrüßung, Tagesordnung und Vorstellung des Vorstands
 - Begrüßung und Tagesordnung
 - Vorstellung des Vorstands
2. Bericht des Vorstands
 - Vergangene Tagungen
 - Kommende Tagungen
 - Finanzen und Gemeinnützigkeit
 - DTK
 - Projektförderung
 - T_EX Collection
 - T_EXnische Kontakte
3. Verschiedenes

Die Tagesordnung wird ohne Einspruch akzeptiert.

TOP 1.2: Vorstellung des Vorstands

Bis auf Klaus Höppner sind die derzeitigen Vorstandsmitglieder anwesend und werden von Martin Sievers vorgestellt: Martin Sievers (Vorsitzender), Herbert Voß (stellvertretender Vorsitzender), Manfred Lotz (Schriftführer), Patrick Gundlach (Beisitzer), Volker RW Schaa (Beisitzer), Uwe Ziegenhagen (Beisitzer).

Der Verein unterhält in Heidelberg ein Büro, das von Frau Karin Dornacher geleitet wird.

TOP 2: Bericht des Vorstands

TOP 2.1: Vergangene Tagungen

- 20. BachoTeX in Bachotek, Polen vom 29. 4.–3. 5. 2012
- TUG 2012 in Boston, Massachusetts, USA vom 16. 7.–18. 7. 2012
- 10. BayTeX in München am 21. 7. 2012

TOP 2.2: Kommende Tagungen

- Frühjahrstagung 2013 vom 6. 3.–8. 3. in Gießen
Lokaler Organisator ist Günter Partosch.
- Herbsttagung 2013 in Köln
Lokaler Organisator ist Uwe Ziegenhagen.
- TUG Tagung 2013 in Tokio, voraussichtlich vom 22.–26. Oktober.
- Frühjahrstagung 2014 in Heidelberg.
Die lokalen Organisatoren sind Marion und Joachim Lammarsch. DANTE e.V. feiert 2014 den 25. Geburtstag in seiner »Geburtsstadt« Heidelberg.
Die TUG hat in Boston gegenüber dem Vorstand von DANTE e.V. erklärt, dass sie eine TUG-Tagung in Deutschland begrüßen würde, was jedoch frühestens 2014 möglich wäre. Der Vorstand wird dies eruieren.
- Herbsttagung 2014 in Stralsund

TOP 2.3: Finanzen und Gemeinnützigkeit

Die Finanzlage ist gut. Es sind Investitionen im Büro geplant, wie die Anschaffung eines neuen Monitors und eines neuen Kopierers. Ein Rollup-Display wurde bereits angeschafft.

Der Folgeantrag bezüglich der Gemeinnützigkeit des Vereins wurde beim Finanzamt eingereicht.

TOP 2.4: DTK

Alle Ausgaben der DTK, die älter als ein Jahr sind, werden gemäß Beschluss der 46. MV seit April 2012 in digitaler Form zur Verfügung gestellt.

TOP 2.5: Projektförderung

TOP 2.5.1: Genehmigte Projekte

- Extended Euler Font
Konvertierung AMS Euler Type 1 Fonts in OpenType und Erweiterung
 - ▷ Rahmen: 3.100 €
 - ▷ Basisversion vorhanden, zur Zeit ruht das Projekt wegen anderer Verpflichtungen.
- \TeX Gyre Math Fonts
Die \TeX Gyre Math Fonts sind fertiggestellt. Einen Bericht hierzu gab es bei der Euro \TeX 2012 in Breskens.

TOP 2.5.2: Beantragte Projekte

- Unicode Arabic OpenType Font
 - ▷ Rahmen: 7.500 \$
 - ▷ Erste Rate von 3.000 \$ wurde überwiesen.
- Oriental \TeX BIDI
A Bidirectional Layout Model for Critical Editions.
 - ▷ Rahmen: 7.500 \$

Dieses Projekt wurde nicht genehmigt.

- DejaVu Font
Erweiterung der Monospace-Variante um mathematische Zeichen, damit sie im Editor eine brauchbare Unicode-Schrift wird.
 - ▷ Rahmen 4.000 €

Soll gleichzeitig ein Muster für die Font-Technologie sein beziehungsweise werden, um andere Fonts zu erweitern.
- Generalizing Tool Chain For Font Extensions (Math and other)
 - ▷ Rahmen 6.000 €

TOP 2.6: Server-Upgrade

Es wurde ein Upgrade des Servers `comedy.dante.de` bei Hosteurope auf das Paket »Rootserver Professional M« durchgeführt. Der bestehende Vertrag wird zum Ende Februar gekündigt, so dass genügend Zeit bleibt, den Server geordnet umzuziehen.

TOP 2.7: T_EX Collection

Die T_EX Collection 2012 wurde zusammen mit der DTK ausgeliefert. Da der Versand von den USA nach Deutschland teuer ist und die Koordination auch dieses Jahr nicht optimal gelaufen ist, wird die Möglichkeit einer erneuten Eigenproduktion in Erwägung gezogen. Die weitere Vorgehensweise wird 2013 mit der TUG diskutiert werden.

TOP 2.8: T_EXnische Kontakte

Die Bemühungen zur Erstellung der T_EXnischen Kontakte wurden eingestellt. Das Design soll aber für andere Zwecke weiterverwendet werden.

TOP 3: Verschiedenes

Unter diesem Tagungsordnungspunkt wurden keine Beschlüsse gefasst.

Martin Sievers schließt die Versammlung um 18:21 Uhr.

Martin Sievers
(Versammlungsleiter)

Manfred Lotz
(Protokollant)

DANTE 2013 – Einladung zur Mitgliederversammlung und »Call for Papers«

Martin Sievers, Günter Partosch

Liebe Mitglieder von DANTE e.V.,

die Frühjahrstagung DANTE 2013 findet vom 6.–8. März 2013 an der Justus-Liebig-Universität in Gießen statt.

Veranstaltungsort ist der Seminarraum B 301, der sich im Interdisziplinären Forschungszentrum (IFZ) befindet (<http://www.uni-giessen.de/cms/fbz/zentren/ifz>), etwa 50 Meter vom Hochschulrechenzentrum entfernt. Das Tagungsbüro wird im benachbarten Raum B 302 eingerichtet werden.

Am Mittwoch sind Tutorien geplant, Donnerstag und Freitag sind für Vorträge und die 48. Mitgliederversammlung von DANTE e.V. vorgesehen. Die Mitgliederversammlung beginnt am Donnerstag, den 7. März 2013 um 9.00 Uhr in der

Justus-Liebig-Universität
Interdisziplinäres Forschungszentrum (IFZ)
Raum B 301
Heinrich-Buff-Ring 26
35392 Gießen

Die Tagesordnung lautet:

1. Begrüßung und Tagesordnung
2. Bericht des Vorstands
3. Finanzbericht
4. Bericht der Kassenprüfer
5. Entlastung des Vorstands
6. Wahl von Kassenprüfern
7. Verschiedenes

Ihre Stimmunterlagen erhalten Sie direkt vor Ort, um vorherige Anmeldung wird gebeten. Eine Übertragung des Stimmrechts ist im Rahmen des § 13 (4) der Vereinsatzung möglich. Wie üblich sind auch Nichtmitglieder als Gäste herzlich willkommen.

Falls Sie ein Tutorium oder einen Vortrag anbieten wollen, reichen Sie Ihren Vorschlag bitte mit dem entsprechenden Formular auf der Tagungsseite im Internet oder per E-Mail an dante2013@dante.de ein. Fügen Sie hierfür bitte eine Kurzzusammenfassung (Abstract) als Text- oder \TeX -Datei in einem der üblichen Formate (\LaTeX , $X_{\text{L}}\TeX$, $\text{Lua}\TeX$ oder $\text{Con}\TeX$ t) bei.

Unter <http://www.dante.de/events/dante2013.html> finden Sie die Tagungsseite mit allen weiteren Informationen rund um die Veranstaltung. Da die Zahl der für den Tagungszeitraum verfügbaren Hotelplätze beschränkt ist, wird um eine frühzeitige Reservierung gebeten.

Für Fragen, Wünsche und Anregungen nutzen Sie bitte obige E-Mail-Adresse oder wenden Sie sich an

DANTE e.V.
Stichwort: DANTE 2013
Postfach 10 18 40
69008 Heidelberg

Mit freundlichen Grüßen

Martin Sievers (Vorsitzender DANTE e.V.)
Günter Partosch (Universität Gießen)

DANTE 2013
in Gießen

TUG 2012: A First-Time Attendee

David S. Latchman

On July 16–18, 2012, I attended the 33rd annual meeting of the T_EX Users Group in Boston, MA. It was not only my first time attending such an event but my first time presenting. Though I did not know what to expect, I found my experience to be both enjoyable and educational. The 2012 conference was held at the Omni Parker House located in heart of downtown Boston at the corner of School and Tremont Streets. This hotel has the distinction of being the longest continuously operating hotel in the US and fits right in with the rich history of downtown Boston. The hotel lies along the Freedom Trail a 2.5 mile mostly red brick path that connects Boston's most significant historic sites starting from Boston Commons and ending at the USS Constitution.

My trip to the conference was a red-eye flight from Los Angeles. By the time I arrived in Boston, I was as well rested as anyone could be after such a flight. Getting to the hotel from Logan International was made easy by Dave Walden's clear travel instructions on the TUG website. I took the Silver Line bus to South Station. From there, I took the Red Line subway (the T) to the Park Street station which was a short walk to the hotel.

Downtown Boston and the Omni Parker House

School Street is so named because it was the site of the first public school in the United States, the Boston Latin School. The Boston Latin School is also the oldest existing school in the United States and has moved several times from its original location. About half-a-mile away is the Boston Common, another record for the city of Boston as it is the oldest city park in the US. A first time visitor to the city can't help but be in awe of the history that surrounds them.

Just across from the Omni Parker House and also on the corner of School and Tremont Streets is King's Chapel. In addition to being a historic landmark that lies along the Freedom Trail, King's Chapel is also a place of worship. The Chapel is open to visitors except during times of worship. As it was Sunday, I don't have the opportunity to see inside the chapel but did manage to visit the nearby cemetery — the King's Chapel Burying Ground.

Though the cemetery shares a name with the nearby church, it is in fact property of the city and was founded in 1630 making it Boston's first cemetery. It remained



(a)



(b)

Abbildung 1: Markers at the King's Chapel Burying Ground

Boston's only burial site for thirty years. The church itself didn't come into existence until some time later in 1689 and was built on the burial grounds as no one in Boston wanted to sell land to a non-Puritan church. There are a few notable historic figures who can call the King's Chapel Burying Ground their final resting place. They include, John Winthrop, Massachusetts's first governor and William Dawes, Paul Revere's companion on his famous ride. Though regular burials ceased in 1896, the occasional burial still takes place.

Not far from the King's Chapel Burying Ground is the Granary Burying Ground, another tourist attraction located on Tremont Street along the Freedom Trail. This cemetery is the final resting place of three signers of the Declaration of Independence and all five victims of the Boston Massacre in 1770. Also prominent in the cemetery is an obelisk erected in 1827 to the parents and relatives of Benjamin Franklin. The obelisk was constructed from granite obtained from the Bunker Hill Monument quarry to replace the original family gravestones which had deteriorated over time. One of the notable men buried at the Granary Burial Ground is John Hancock, most remembered not because he served as President of the Second Continental Congress or that he was the first and third Governor of Massachusetts but rather for the prominence of his large and stylish signature on the Declaration of Independence. It is believed that Hancock made his signature so large as an act of defiance to King George III. This story couldn't be further from the truth.

As the President of the Second Continental Congress, Hancock was the first to sign the now historical document. At the time, the Declaration of Independence wasn't a formal declaration of independence from Britain but rather it was intended to be copied and distributed among the colonies to explain the need to declare



(a) The John Hancock Memorial (b) The Paul Revere Memorial (c) The Samuel Adams Marker

Abbildung 2: Markers at the Granary Burying Ground

independence. As a result, Hancock thinking he would be the only person to sign the document along with the Secretary of the Congress, Charles Thompson, made his signature large. As the Declaration of Independence gained support, other delegates stated adding their names over the period of weeks and months.

For the beer drinkers reading this, it may be surprising to note that the face featured on a Sam Adams beer isn't actually Sam Adams but Paul Revere. It is said that Samuel Adams was not a good looking fellow while Paul Revere, on the other hand, was considered a handsome chap. Both men are buried at the Granary Burying Ground.

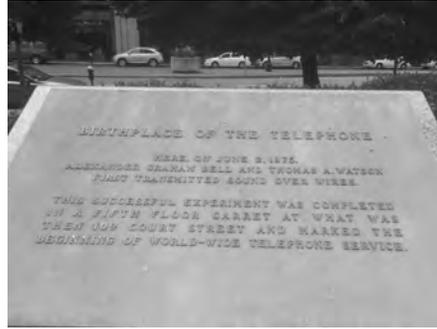
Downtown Boston doesn't just feature historical sites and landmarks. It is also the center of the city's government the section, of which is, is appropriately named the Government Center. Not far from the hotel is both the City Hall Plaza and the John F. Kennedy Federal Building. In front of the JFK Building on Cambridge Street sits a plaque commemorating the first sound transmission over wire, an event that led to the practical invention of the telephone.

The event the plaque commemorates isn't the one we are all familiar with that took place on March 10th, 1876 where Alexander Graham Bell shouted the words, "Mr. Watson, come here, I want to see you". In 1875, Bell under the advice of another scientist, Joseph Henry, stated working on an electrical multi-reed device to transmit the human voice over wires. Bell didn't have the necessary skill or knowledge to work on such a device but a chance meeting with Thomas Watson changed all that.

The principle behind the reed design operated on the same principle as the human ear. The pinna collects sound and concentrates it as it travels down the auditory canal. Vibrations on the tympanic membrane then transmit energy to the ossicles or ear bones which is then converted into electrical impulses by the cochlea to be interpreted by the brain. The ear bones are a system of levers that amplify the force on the cochlea. This allows the tympanic membrane to be relatively small. Bell



(a) Boston City Hall



(b) The First Telephone Marker in front of JFK Building.

Abbildung 3: Boston Government Sites

reasoned that if he had a larger membrane, he would be able to collect more sound and move a steel reed placed in a magnetic field. This would turn sound energy into an electrical current as the reed moves through the magnetic field.

On June 2, 1875, while troubleshooting the multi-reed device, Watson plucked one of the reeds to check the tension believing it to be too tight. Bell, who happened to be listening at the right time in the next room, heard the metallic twang and the first working model of the telephone was born.

The Conference

The Attendees and Presenters

There were 61 participants and 23 presenters (of which I was one) listed in the TUG program with participants coming from Australia, Canada, UK, Germany and Egypt. Attending the conference was a bit of a surreal experience as I saw so many people who were as passionate about \TeX as I was and its future development. Where most of the conferences I have attended in the past tend to be specialized events geared toward one topic and where everyone — more or less — is in the same field, I found this to be quite different. Instead I found people with a wide array of specialties and talents.

Day One

The conference started with an opening from the TUG president, Steve Peter which was then followed by a presentation by a \LaTeX consultant, Amy Hendrickson of

T_EXnology Inc. Amy's talk focused on the use of `\csname` to create L^AT_EX macros and dynamically generate a series of definitions. Amy demonstrated its use to redefine the footnotes command to produce endnotes and to dynamically create online reports. This is something I definitely have to look into and learn about.

After a break, Frank Mittelbach talked about L^AT_EX3 and the direction of the T_EX typesetting engine. When T_EX was started over two decades ago, computing power was for more limited compared to what is available today and, as a result so too was some of the algorithms. Frank compared the limitations of T_EX's algorithms to L^AT_EX3 and how they were overcome, to what degree and what still needs to be done. Given the increase in computing power and the work that has been done in recent years, the present system could be a viable one and remain so for some time.

Next was a presentation on font design by Steve Peter. Steve is a linguist and font designer and showed how font design can be done using the METAFONT package.

Will Robertson next came on to show his work on the fontspec package and uni-code math. Will's talk focused on his experiences in developing these packages while at the same time learning how to program. One of the consequences of which was contributing code to the L^AT_EX3 project.

Will's talk was followed by a lunch and group photo of the conference attendees.

The next group of talks focused on using the T_EX engine to typeset foreign languages. Unfortunately, the talk on using L^AT_EX to typeset Mayan hieroglyphics by Bruno Delprat and Stepan Orevkov was canceled. It was a little disappointed as I have always found it interesting how T_EX typesetting engine can be used in so many ways, from typesetting chess boards to creating Sudoku puzzles. Fortunately, the next talk by Sherif Mansour on typesetting Arabic proved just as interesting.

Sherif is a graduate student from Cairo University and whose research is focused on improving Arabic typesetting using the AlQalam font in LuaT_EX. One of the problems Sherif faces is the typesetting of right-to-left fonts in LuaT_EX. Each written line on a page in the Arabic language is about the same length and this makes line breaking difficult. This makes for a difficult problem to solve as the various shapes can also change in subtle ways for the same characters. Though I don't envy Sherif I am interested in his future work. Maybe I can find a way to use L^AT_EX to help me learn Arabic.

The next presentation by Norbert Preining focused on the problems of typesetting in Japanese. Japanese typesetting differs from English in that there are four different writing systems: Kanji, Haragana, Katakana and Roman letters, the difficulties of which are further compounded as Japanese employs both vertical and horizontal typesetting styles.

Federico Garcia next talked about music typesetting. As a developer working on writing a professional music typesetting system in \TeX , Federico highlighted some of the problems experienced with typesetting music. Though I am not a musician, I could appreciate some of the problems he faced. One of these problems is getting the \TeX engine to automatically (and correctly) place the bars when creating a sheet of music. Federico compared his system with other music typesetting systems. While many problems need to be solved before he has a fully functional system, he believes they can be solved.

Day Two

The first talk of the day by Troy Henderson focused on some of the web based utilities he had developed to generate and plot functions in METAPOST . I haven't paid too much attention in the past where METAPOST or PSTricks are concerned choosing to favor TikZ instead. I may have to reevaluate that position as the online utilities may help me learn and pick up the code quicker.

The next talk by Richard Koch focused on installing \TeX on the Mac using the MacTeX install package. The key feature of this package, like everything Mac, is its ease of use and \LaTeX can be installed with a single click of the button. Richard demonstrated a full \TeX installation during the course of his talk; we were all assured that an actual installation took place and no trickery was involved.

If you are an iPad owner then Bill Cheswick's talk on $i\TeX$ should be of interest. The $i\TeX$ app is a \LaTeX reader for the iPad and shouldn't in any way be confused with Donald Knuth's proposed XML-based successor to \TeX . When typesetting a book, or a page or any document, the final page size is generally known down to the nanometer. \TeX then uses some clever algorithms to optimize the presentation for a high standard of quality output. Unfortunately, this poses some problems for ebook readers. As there are varying screen sizes as well as the different ways one can hold a reader, it would be difficult and time consuming to run \TeX over a document every-time a reader shifts position.

The $i\TeX$ application solves this by using \TeX to create precomputed images for portrait and landscape layouts in both standard and large type versions. These images are then stored in a container file that the app can then read. Besides being a reader, the app also converts text into \LaTeX typeset output. Bill demonstrated this by importing and converting text from Project Gutenberg and the arXiv.org. The app is free on Apple's app store for anyone interested to try.

\LaTeX and Thesis Talks

The next two presentations after the break focused on theses and dissertations. While my talk dealt with the various packages students can use to make their lives

easier, Peter Flynn's talk dealt with the creation of a \LaTeX class file for the University College Cork. Given the number of thesis class files present on the CTAN server, of which we can assume there are far more unofficial class files elsewhere, the question we must ask is, "do we really need another thesis class file?" According to Peter Flynn, we do.

Generally speaking, a thesis class file is supposed to meet the formatting requirements of a particular university by ensuring page dimensions and margins are set correctly. But this is not the only requirement. Students are also required to enter the formal names of their departments and the colleges they belong to. Given that a university might have a complex departmental infrastructure and even stranger naming standards, the best way to ensure that students enter this information correctly is to enter them as options to the `\documentclass` command rather than allow authors to enter this information themselves. The Cork class file was tested on January 2010 over the course of 18 months and so far the response has been favorable — so favorable, in fact, that other institutions are looking to adopt or base their class file on Cork's.

Like most university class files in my experience, the University College Cork class file was designed to be minimalist in nature but at the same time meet the needs of as many students as possible. Any package that is added must be done carefully so as not to break any other existing packages and must be done to meet the needs of as many students as possible. Unfortunately, with the number of disciplines and in some cases, cross-disciplines, this means that all the needs of every student can not be built into a single class file and students will need to add packages as they see fit. This was the subject of my talk.

Typically, when a student looks for a \LaTeX consultant they are under a lot of stress. Generally, any attempts to compile a document freezes and ends up with hundreds of errors. Part of my job as a consultant isn't to just fix these problems and get the document compiling again but to offer solutions and make lives easier. Once a project is concluded, students still need to make edits before the final submission.

\LaTeX has a multitude of packages and, at a quick glance, it almost seems as if you can do anything. By knowing a student's discipline or what their thesis is about, a consultant can often give advice on the best package to optimize the writing process. Students in engineering or the sciences, for example, may find it easy to take advantage of the `siunitx` package which will allow them to enter mathematical units easier rather than having to enter math mode then enter a confusing list of symbols. Science students who deal with chemical equations may also take advantage of the `mhchem` package for the same reason. I talked about some past projects and how proper package use helped my clients.

Post Lunch Talks

The next talk by Boris Veytsman was on using \TeX on the iPad: “ \TeX and friends on a Pad”. Though I can’t imagine actually typing \LaTeX code on a flat screen I am, none the less, excited at the possibility. Maybe I work too much and need to get out once in a while. Who knows? But bottom line, it is possible to have \LaTeX on an iPad.

Bart Childs then talked about the problems of automating the process of converting text from word processors into \LaTeX . The problems that some converters face is that they usually attempt to make the final output look like the original typewritten document as much as possible. This introduces complex and, oft times, terrible \LaTeX code. Bart Child’s goal was to find a conversion process that would produce code that was accurate, clean and maintainable.

Bart used a hybrid process based on the `Writer2 \LaTeX` plugin for the Open Office Writer package as well as macros written in e-lisp. Bart talked about his tests on converting a book on rotodynamics (which was heavy on the mathematical side), a C++ programming text, a memoir written by a friend that contained portions of the Czech language and a novel.

Final Talks for the Day

Frederico Garcia’s talk focused on his `colordoc` package. This package was based on Frank Mittelbach’s `doc` package which highlights braces and other code delimiters and makes for slightly more readable code. This package is sure to come in handy if you have ever need to troubleshoot code and aren’t sure if there is an extra brace or bracket lurking somewhere.

The final talk for the day by Jim Heffron & Michael Doob focused on the `Asymptote` graphics program. Heffron and Doob showed some of the features of the package and its ease of use in generating graphics.

Round Table Discussion

The day concluded with a round table panel discussion moderated by Dave Walden with some of the \TeX consultants of which I took part. The other consultants included Peter Flynn, Amy Hendrickson, Christina Thiele, Steve Peter and Boris Veytsman. Questions asked by the audience focused on the business aspects of \TeX consulting; how consultants got business; how they dealt with problem clients to name a few. After the round table discussion, proceedings concluded for the day. Participants then met for a banquet at the Oceanaire Seafood Room for dinner; the restaurant is located across City Hall Plaza.



Abbildung 4: Conference Room at the Omni Parker House where TUG 2012 took place.

Day Three

The first talk of the day by Pavneet Arora titled “Sleep de(p)ri ν ed typesetting” and focused on the process of typesetting. The second talk by Bob Neveln and Bob Alps looked at a Python program they created to check the syntax of mathematical proofs: ProofCheck. Mathematical proofs were checked against the syntax developed by A. P. Morse in his book, “A Theory of Sets”. The authors presented recent updates to the system that are designed to make proof checking easier to users.

In the next talk Christina Thiele talked about her experiences using \TeX , first to typeset articles and journals in academia to the creation of her company, Carleton Production Center. Christina also highlighted some of the changes in \TeX that took place when she first started using it almost thirty years ago as well as the changes in software and hardware she underwent in her career as a consultant.

The following talk by Will Robertson and Frank Mittelbach looked at the origin, the development and recent changes to \LaTeX 3. The key aspects and ideas behind \LaTeX 3 were developed in the early 1990s but it wasn’t until recently that code came into widespread use. The talk focused on the what \LaTeX 3 can currently do and their plans for the future.

After Lunch

The next talk by Boris Veytsman and Leyla Akhmandeeva focused on the results of a recent study conducted by the two authors to check whether typographic style influenced a reader’s ability to comprehend and remember contents in a passage. It is generally believed among typographers that typography is more than an art as it can influence a person’s comprehension and reading speed. To test this hypothesis, the authors gave university students one page long passage and tested their

comprehension. \LaTeX was used to control various typographic features from fonts, page layout and justification. This preliminary study was intended for textbook designers where comprehension of text is very important. The preliminary study indicated that typography does not play a part in reading speed or comprehension and the human brain may be flexible enough to allow us to read even badly designed pages.

While this first study may indicate that typography does not have an effect, I definitely would like to see more research on the area. Though short passages using bad typography may not have an effect on comprehension, perhaps the effects of typography and longer passages may affect a person's ability to understand a passage.

Norbert Preining followed with a talk about the recent changes and additions to \TeX Live. One of the biggest changes is the extension of the \TeX Live manager to read multiple repositories, something that has been a feature of $\text{Mik}\TeX$ for some time. Norbert also gave an overview of the other changes to \TeX Live 2012.

The final talk of the conference was by Didier Verna and looked at possible modern implementations of \TeX . Didier said his current project came about from a discussion he had with Donald Knuth, the creator of the \TeX typesetting system. \TeX was initially designed to be a simple system as the computer resources of the time were limited and this meant that a full programming language could not be implemented.

Given the time that has passed, we have seen an exponential growth in computing power along with our skills in language design and implementation. Could this be a way to modernize \TeX ? Didier says this is possible and can be done using an old, but very established modern programming language Common Lisp. Didier focused on the features of the language that made it ideally suited to the task of modernizing \TeX and gave some demonstrations.

Concluding Remarks and Observations

I have always known that somewhere in the back of my mind that the development of \LaTeX is community based, much like many open-source, community based projects. Though I am just a user of the system, it was great to meet others like me as well as some of the developers who are going to continue building and contributing to the evolution of \LaTeX . As a first time participant I truly had a wonderful time and attending future conferences is something that I definitely look forward to. Attending gave me the chance to meet others who were just as interested in \LaTeX as I was and also gave me the chance to see the future of \TeX . Attending is definitely something I recommend to any TUG member who hasn't done so.

Klumpen, Kaas & ConTeXt – Die EuroTEX 2012 in Breskens

Uwe Ziegenhagen

Wieder einmal habe ich den Fehler begangen, mich am ersten Abend einer Tagung von DANTE e.V. in die Nähe des DTK-Chefredakteurs Herbert zu begeben. Ich weiß auch nicht, ob es an der guten Stimmung lag oder ob das Mineralwasser bewusstseinsverändernde Substanzen enthielt (man war ja schließlich in den Niederlanden), auf jeden Fall habe ich zu schnell zugesagt, als es darum ging, den Tagungsbericht zu verfassen.

Anreise

Die EuroTEX 2012 fand im niederländischen Breskens statt, in dem Teil des Landes, der nur per Fähre, Tunnel oder über Belgien erreichbar ist. Dank der offenen Grenzen ist die freie Fahrt aber kein Problem, man merkt kaum, in welchem Land man gerade unterwegs ist.

Die Anreise gestaltete sich für mich mehr als einfach, nachdem Thomas angeboten hatte, mich mit dem Auto mitzunehmen. Beladen mit zwei Laptops, diversen »iGeräten« und umfangreicher Fotoausrüstung ging es Sonntagnachmittag los. Dank Navi war die Wegfindung kein Problem, sodass wir kurz nach 20 Uhr im Hotel »Scaldis« empfangen wurden, das für die nächsten Tage unser Zuhause sein sollte.



(a) Frühstück im Hotel Scaldis



(b) Der Tagungsort »de Platte Knoop«

Mit knapp 5000 Einwohnern ist Breskens zwar nicht der Nabel der Welt, der Ort liegt aber sehr schön an der Nordsee und bot genug Möglichkeiten, Nordsee-Flair aufzunehmen. Die vergleichsweise geringe Größe hatte auch den Vorteil, dass man jeden konferenz-relevanten Ort bequem zu Fuß erreichen konnte.

Maandag

Nach dem Frühstück im Hotel ging es dann am Montagmorgen in das »de Platte Knoop«, ein Café ganz in der Nähe des Deichs. Dort warteten ein offenes WLAN – das mit der Zahl der Nutzer so seine Probleme hatte – und Kaffee auf die Teilnehmer. Eine kurze Verzögerung gab es, als innerhalb von fünf Minuten nach dem allgemeinen Laptop-Anstöpseln die Sicherungen aufgaben: Die Mehrfachsteckdosen waren wohl doch etwas einseitig verteilt gewesen!

Die Schar der fast 50 Teilnehmerinnen und Teilnehmer war ähnlich bunt gemischt wie die Marken der zahlreich vorhandenen Laptops: Neben der größten Gruppe aus Deutschland – die Tagung war gleichzeitig die Herbsttagung von DANTE e.V. – waren viele weitere Nationen vertreten. Aus Frankreich, Italien, Finnland, Polen, Tschechien, Norwegen und weiteren Ländern hatten \TeX ies den Weg nach Breskens gefunden. Die bei weitem weiteste Anreise hatten jedoch die drei japanischen Teilnehmer auf sich genommen!

Der erste Vortrag von Kees van der Laan zum Thema »Recreational \TeX & Co – with a serious undertone« führte vor, dass man für verschiedenste Grafiken kein \LaTeX benötigt, da man ja Grafiken direkt in PostScript programmieren kann. Sehr interessant, ich werde aber wohl weiterhin auf PGF/TikZ, Inkscape und CorelDraw setzen. In einem weiteren Vortrag am Nachmittag zeigte Kees uns noch, wie man fraktale Julia-Mengen in PostScript darstellen kann.

Im Anschluss an Kees zeigte uns Jano Kula aus Tschechien, wie er für den Nike Run in Prag personalisierte T-Shirts in ConTEXT erstellt.

Die textile Seite von \LaTeX & Co war auch bei Mari Voipios Vorträgen das Thema. Sie erstellte Flechtmuster mit METAPOST und zeigte auch bei einem Workshop, wie man theoretisch und praktisch \LaTeX und Flechtarbeiten verbinden kann.

Nach einem Vortrag von Taco Hoekwater zum Thema METAPOST hielt dann Patrick Gundlach zwei Vorträge. Im ersten Vortrag zeigte er die aktuelle Version seines »speedata« Publishing-Tools, das anhand von Regeln große Mengen Text auf Papier respektive eine PDF-Datei bringen kann. Im zweiten Vortrag führte er vor, wie mittels »Lua \TeX Callbacks« Abstände in Dokumenten visuell dargestellt werden können, etwas, das mit Plain \TeX nur schwerlich – falls überhaupt – umsetzbar sein dürfte.

Nach den Vorträgen kamen wir dann zum offiziellen Teil des Abends, der Mitgliederversammlung von DANTE e.V. (siehe dazu auch Seite 6).

Dinsdag

Der Dienstag begann mit meinem eigenen Vortrag zum Thema »Reporting mit \LaTeX «, ein Artikel für die DTK folgt. Nach mir präsentierte Leo Arnold von der

Technischen Universität München, wie man mit einem einzigen \LaTeX -Lauf mehrere PDF-Dateien erzeugen kann. Die von ihm selbst entwickelte Klasse erzeugt auf der Basis von `comment.sty` und `writer18` weitere Instanzen von `pdflatex`, die dann die einzelnen Versionen ausgeben. Ich hatte bisher angenommen, dass dies nicht möglich sei und freue mich, wenn diese Technologie in Klassen- oder Paketform auf CTAN kommt.

Vor der Mittagspause präsentierte Taco Hoekwater, der lokale Organisator der Konferenz, noch zu den Themen »Pfadauflösung in METAPOST« und »Parsen von PDF Content Streams«.

Den Nachmittag nutzte Hans Hagen dann, um verschiedene Themen zu ConTEXt vorzustellen, gefolgt von Luigi Scarso, der zu »MFLua« präsentierte.

Zum Abschluss des Abends hatte man Gelegenheit, mit Willi Egger Konferenzmappen zu basteln.

Woensdag

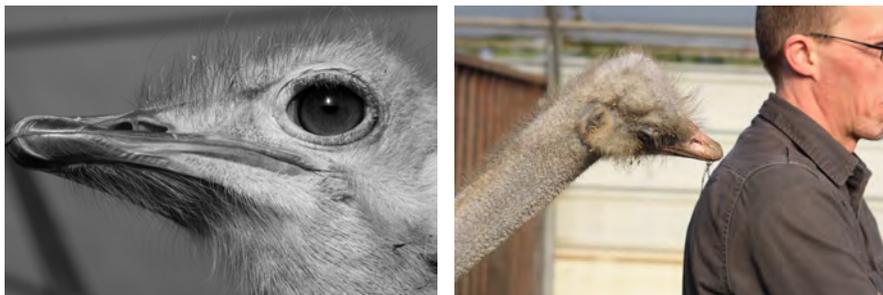


Abbildung 1: Auf der Straußenfarm

Der Mittwoch stand ganz im Zeichen des touristischen Beiprogramms. Morgens um 9 Uhr stand ein original amerikanischer Schulbus abfahrtsbereit, der erste Stopp war dann auf einer Straußenfarm, wo wir mit Kaffee und Kuchen¹ begrüßt wurden. Die Fotoapparate standen bei den meisten dann auf Dauerfeuer, als wir uns die kleinen und großen Strauße anschauten.

Nach dem Mittagessen in einem Landgasthof (der über ein offenes WLAN verfügte) ging die Reise nach Brügge. Brügge, seit dem 2. Jahrhundert besiedelt, war aufgrund der Hanse-Mitgliedschaft und günstiger geografischer Lage einst die reichste Stadt Nordeuropas. Dieser Reichtum ist auch heute noch sichtbar; davon konnten wir uns

¹Mit Kuchen aus Straußeneiern, wobei ein Ei für sieben (!) Kuchen reicht.



(a) Am Minnewater



(b) Am Dijver

Abbildung 2: Brügge

bei der Stadtführung überzeugen. Die verbleibende Zeit bis zur Abfahrt des Busses nutzten dann alle, um sich noch einige Sehenswürdigkeiten anzuschauen oder in den unzähligen Schokoladengeschäften Brügges exzessiv zu shoppen (zumindest ging es mir so).

Bevor es dann zum Konferenzdinner in ein sehr gutes Breskener Restaurant ging, kehrten wir noch in einem Spirituosensladen ein, wo es niederländischen Gin, sogenannten »Geneva«, mit Austern gab.

Donnerdag

Der Donnerstag begann mit mehreren Vorträgen zu Fonts. Bogusław Jackowski, Jerzy Ludwiczowski und Piotr Strzelczyk von der GUST zeigten die Abläufe zur Erstellung der mathematischen Fonts aus dem Projekt \TeX Gyre. Sie präsentierten den aktuellen Stand des Projekts und sprachen über die Abwärtskompatibilität von LM Math und CM Math.

Nach dem Mittagessen erfüllte Hans dann einen Wunsch von mir und zeigte den Anwesenden die absoluten ConTeXt Grundlagen. Da das aktuelle ConTeXt auf Lua-Basis arbeitet, besteht kein Grund mehr, Ruby etc. installiert zu haben, ein aktuelles \TeX Live reicht aus. Nach diesen Grundlagen ging es dann ans Eingemachte, als Hans über das Rechnen in Tabellen mit Lua, das Parsen von XML-Dateien und die Anbindung an MySQL-Datenbanken sprach.

Beendet wurde der Tag mit einem Besuch beim Glasbläser, der aus glühend heißem Glas Schalen und Glastiere formte.

Vrijdag

Am Freitag hatten die Teilnehmer mit der weitesten Anreise, Munehiro und Hironori aus Japan, das Wort. Sie sprachen über die Nutzung von \LaTeX in japanischen Satzhäusern und die Schwierigkeiten, die der japanische Satz mit seiner Vielzahl von Zeichen und unterschiedlichen Satzrichtung hat. Über diese Präsentation hat sich auch ein japanischer Freund von mir gefreut, dem ich die PDF-Datei noch während der Präsentation zukommen ließ und der darin einige seiner \LaTeX -Probleme behandelt sah.

Wer mehr zum japanischen Textsatz mit \LaTeX erfahren möchte, der kann dies übrigens im nächsten Jahr in Tokio tun, wo Ende Oktober 2013 die TUG Konferenz stattfinden wird.

Neben Hans, der noch mehr über ConTeXt berichtete, sprachen am Freitag noch Ivo Geradts und Kai Eigner über den Satz von Sanskrit mit \LuaTeX , Tomás Hála über Unterschiede im Satz von Tschechisch und Slowakisch, bevor Jean-Michel Hufflen über die Fortschritte bei der Entwicklung seines $MLBibTeX$ referierte.

Beschlossen wurde die EuroTeX 2012 dann von Sietse Brouwer, der mit den Anwesenden über mögliche Verbesserungen an der ConTeXt-garden Webseite sprach.

Odyssee – Rückfahrt mit Hindernissen

Die Rückfahrt nach Köln mit Doris und Ulrik war auch mit Spannung und Dramatik gespickt. Den ersten unfreiwilligen Stopp hatten wir in Hoofdplaat, nur wenige Autominuten von Breskens entfernt, als sich Doris' Opel Kadett und ein niederländischer Toyota Yaris auf der Kreuzung ein wenig zu nah kamen. Es gab glücklicherweise nur geringen Blechschaden, daher konnten wir die Fahrt nach kurzer Zeit fortsetzen.

Eine weitere unfreiwillige Unterbrechung gab es dann nur noch in Belgien, als nach kurzer Rast in einem »Heim von Ronald McDonald« das Auto nicht mehr anspringen wollte. Beherztes Anschieben bei eingelegtem Gang war dann angesagt, die restliche Reise lief dann – zwar bei strömendem Regen – problemlos.

Fazit

Was bleibt sind die Erinnerungen an eine interessante Tagung, die dank Tacos guter Organisation für alle Teilnehmer viel Spannendes und Lehrreiches bot. Auch wenn einige Themen jenseits von dem angesiedelt waren, was ich in \LaTeX kann, kenne oder benötige, so bot diese Tagung viele Einblicke in das, was \TeX nisch möglich ist.



Abbildung 3: Konferenzdinner in Breskens





(Fotos: Uwe Ziegenhagen)

Impressionen von der FrOSCon



Urlaubsimpressionen – Dante und Papyrus



(Fotos: Martin Wilhelm Leidig)

Bretter, die die Welt bedeuten

Schulschriften – von Sütterlin bis heute

Walter Entenmann

Dieser Beitrag beschreibt die Implementierung der Schriften einiger Schreibschriften in METAFONT, die als Schulausgangsschriften an deutschen Schulen verwendet werden oder wurden. Dazu gehören die *Sütterlinschrift* (1911), die *Deutsche Normal-schrift* (1941), die *Lateinische Ausgangsschrift* (1953), die *Schulausgangsschrift* (1968) und die *Vereinfachte Ausgangsschrift* (1972). Die METAFONT-Quelldateien sind in dem Paket *schulschriften* [5] zusammengefasst, das auch Dokumentationen, Stildateien und Anwendungsbeispiele enthält. Nach dem Laden des Pakets können beliebige deutsche Texte in den genannten Schriften geschrieben werden.

Nach einer kurzen historischen Übersicht werden die einzelnen Entwurfsschritte vom Handentwurf und dessen systematischer Umsetzung in die METAFONT-Kodierung bis zum Workflow und einer geeigneten Datei-Organisation beschrieben. Zweckmäßige Parameter- und Makrodefinitionen erleichtern den Entwurf und die Pflege des Systems. Die Verbindung der Buchstaben bei einer Schreibschrift wird durch Aneinanderstoßen und die Einführung von Verbindungsniveaus systematisch gelöst. Aufgrund der unterschiedlichen typographischen Besonderheiten der Schriften resultiert daraus für jede Schrift ein individuell angepasstes Entwurfskonzept. Der Zusammenhang zwischen den Abmessungen der Zeichenvorlage und der gewünschten Entwurfsgröße mit den intern in METAFONT verwendeten »scharfen« Variablen und »Pixel«-Größen wird erläutert. Die Schriftdefinitionsdateien, einige Stildateien zum Schreiben auf ein Liniensystem und zur \TeX -Anpassung sowie die Zeichentabellen und Schriftbeispiele unterstützen und zeigen die praktische Anwendung der Fonts.

Historische Entwicklung

Sütterlinschrift (SU)

Die Zeit um 1900 war geprägt von tiefgreifenden Neuerungen und Fortschritten auf allen Gebieten der Wissenschaft, Kunst und Bildung. Da verwundert es nicht, dass das Preußische Schulministerium 1911 den Graphiker und Dozenten an der Kunstgewerbeschule in Berlin, Ludwig Sütterlin (1865–1917) damit beauftragte,



Abbildung 2: Beispiel für Sütterlins Reform der deutschen Schrift.

Auch in der Mathematik und in den Ingenieurwissenschaften wurden die deutschen Schriftzeichen für Vektoren und Matrizen, hyperbolische Funktionen u. a. noch lange Zeit verwendet.

Deutsche Normalschrift (DN)

1941 wurden mit dem Schrifterlass [3], wie bereits erwähnt, die deutschen Schreib- und Frakturschriften verboten und die sogenannte »Deutsche Normalschrift« als lateinische Schreibschrift verordnet. Das Liniensystem wurde bei gleicher Höhe von 15 mm jetzt im Verhältnis 2 : 3 : 2 aufgeteilt. Das Mittelband ist somit etwas höher als die Ober- und Unterlänge. Die Schrift sollte mit einer Rechtsneigung von 75 bis 80 Grad geschrieben werden. Der Erlass enthält eine Musterseite mit den Groß- und Kleinbuchstaben des Alphabets, den Ziffern und einigen Satzzeichen. Diese Schrift war bis 1953 an deutschen Schulen in Gebrauch.

Lateinische Ausgangsschrift (LA)

Nach dem Krieg entwickelte der »Iserlohner Schreibkreis« eine neue lateinische Schreibschrift, die auf der Kultusministerkonferenz 1953 zur allgemeinen Schulausgangsschrift in Deutschland erklärt wurde. Man verwendete wieder das Liniensystem von Sütterlin. Der Neigungswinkel beträgt 72 Grad. [1]

Schulausgangsschrift (SAS)

In der Folgezeit wurde in der damaligen DDR eine neue lateinische Schreibschrift, die sogenannte »Schulausgangsschrift«, entwickelt und dort ab 1968 eingeführt. Eine Besonderheit dieser Schrift stellen die Buchstaben A, F und H dar, die mit den nachfolgenden Kleinbuchstaben knapp unter der Mittellinie fast waagrecht verbunden werden. Das Liniensystem bleibt unverändert, die Neigung beträgt 76 Grad. Heute ist sie vorwiegend in den neuen Bundesländern in Gebrauch. [2]

Vereinfachte Ausgangsschrift (VA)

1969 begannen auch in der BRD die Vorarbeiten für eine neue Schreibschrift, die sogenannte »Vereinfachte Ausgangsschrift«, die ab 1972 nach und nach eingeführt

wurde. Sie basiert auf der Lateinischen Ausgangsschrift, ist aber stark vereinfacht und funktional. Insbesondere drei Eigenschaften sind bemerkenswert:

- Die Vereinfachte Ausgangsschrift verwendet als erste und einzige lateinische Schreibschrift das »Köpfchen-e« anstelle der Schleifchen-e, sodass »alle Buchstaben auf der Mittellinie beginnen und enden«, wenn man den Verbindungsstrich mit einbezieht.
- Sie erleichtert nicht nur das Schreibenlernen, sondern vereinfacht auch die Implementierung in METAFONT entscheidend.
- Die Großbuchstaben B, D, J, N, O, P, S, T, V und W werden nicht verbunden. Das Liniensystem bleibt unverändert und die Neigung beträgt 79 Grad. [1]

Insgesamt stehen heute in Deutschland für den Schreibunterricht drei lateinische Schriften mit einheitlichem Liniensystem zur Verfügung, die »Lateinische Ausgangsschrift«, die »Schulaustrittsschrift« und die »Vereinfachte Ausgangsschrift«. Die Verwendung dieser drei Schriften ist in den einzelnen Bundesländern in unterschiedlicher Weise geregelt. Um möglichst authentische Schriften zu entwickeln, habe ich mich an der Originalveröffentlichung von Sütterlin, dem Schriffterlass von 1941 und den amtlichen Vorgaben in den Lehrplänen der Schulbehörden u. a. von Sachsen und Baden-Württemberg orientiert (siehe [1, 2]).

Handentwurf

Die Entwicklung einer Schrift beginnt mit einem sorgfältigen Studium von handschriftlichen Dokumenten, wie Briefe, Urkunden, Schulhefte usw. und einer eingehenden Analyse des Erscheinungsbilds einer jeden Schrift, um wichtige Eigenschaften wie Lineatur, Neigung, Proportionen, Gemeinsamkeiten bestimmter Buchstabengruppen, usw. herauszufinden. Man benötigt dann eine systematische Methode zur Lösung des Verbindungsproblems bei Schreibschriften. Unter Beachtung der erzielten Ergebnisse erstellt man schließlich eine maßstabsgetreue Zeichnung aller Schriftzeichen als Grundlage für die folgende programmtechnische Umsetzung.

Analyse

Für eine Schreibschrift sind folgende Größen von Interesse:

- Das Teilungsverhältnis des Liniensystems, d. h. das Verhältnis der Höhen von Oberlänge zu Mittelband (ht) zu Unterlänge (dt) = $(Ht - ht) : ht : dt$. Dabei ist Ht die Höhe der Großbuchstaben.
- Der Neigungswinkel der Schrift (β),
- Der Winkel α' der Steigung des Verbindungsstrichs zum jeweils nachfolgenden Buchstaben.

Der Einfachheit halber entwerfen wir alle Schriften senkrecht stehend, weil die Neigung in METAFONT automatisch berücksichtigt werden kann. Für den Neigungsfaktor slant und den Winkel α der senkrechten Schrift gilt dann

$$\text{slant} = \cot \beta, \quad (1)$$

$$m' = \tan \alpha', \quad (2)$$

$$m = \tan \alpha = \frac{1}{\frac{1}{m'} - \text{slant}}. \quad (3)$$

- Die »runden« Buchstaben, wie p, rundes s, z bei der Sütterlinschrift und a, c, d, g usw. bei den lateinischen Schriften, sind geeignet zur Lösung des Verbindungsproblems und zur Bestimmung des Verhältnisses $k = b/a$ der Halbachsen a und $b = \text{ht}/2$ der elliptischen (ovalen) Grundform der genannten lateinischen Buchstaben. Als sehr zweckmäßig erweist sich dabei, wie von Donald Knuth vorgeschlagen, die Form der Superellipse mit dem Superness-Parameter $0 < \sigma < 1$. Für $\sigma = 1/\sqrt{2} = 0.707$ erhält man die normale Ellipse. Experimentell habe ich einen etwas größeren Wert für $\sigma = 0.733$ als besonders geeignet ermittelt, dem ein etwas »dickeres« Oval entspricht.

Die für die einzelnen Schriften ermittelten Werte sind in der Tabelle zusammengefasst.

	SU	DN	LA	SAS	VA
Teilung	1 : 1 : 1	2 : 3 : 2	1 : 1 : 1	1 : 1 : 1	1 : 1 : 1
β (Grad)	90	76	72	76	79
slant	0	0.25	0.325	0.25	0.19
α' (Grad)	–	47.5	45.25	47.5	46.75
α (Grad)	45	56	56	56	53
m	1	3/2	3/2	3/2	4/3
k	–	1.425	1.425	1.5	10/7
	–		$\sigma = 0.73345$		

Verbindungsproblem

Bei $\text{T}_{\text{E}}\text{X}$ werden die Buchstaben durch die Abmessungen der jeweiligen Boundingbox (umschreibendes Rechteck) charakterisiert. Das Setzen eines Wortes erfolgt dann durch lückenloses Aneinanderstoßen dieser Boxen. Dieses Prinzip der *Boundingbox* und des *Verbindens durch Aneinanderstoßen* wollen wir auch auf die Schreibschriften anwenden. Dazu ist aber zwingend erforderlich, dass der Endpunkt E des Linienzugs eines Buchstabens mit dem Anfangspunkt A des Linienzugs irgendeines nachfolgenden Buchstabens übereinstimmt. Wenn der Kurvenverlauf am Verbindungspunkt

außerdem glatt sein soll, müssen auch die Steigungen gleich groß sein. Daraus folgt die Regel: Anfangs- und Endpunkt des Linienzugs eines Buchstabens müssen auf gleicher Höhe und auf dem linken/rechten Rand der Boundingbox liegen. Außerdem muss der Linienzug gegebenenfalls in beiden Punkten A und E die gleiche Steigung m haben. Die so definierte Verbindungshöhe nennen wir *Verbindungsniveau*.

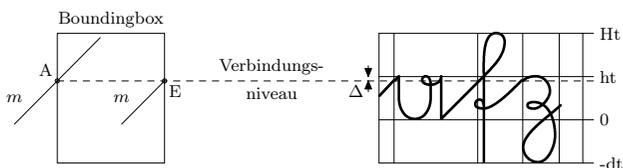


Abbildung 3: Boundingbox und Verbindungsniveau mit Beispiel.

Wegen der runden Buchstaben, die in allen betrachteten Schriften enthalten sind, muss das Verbindungsniveau zwingend um ein Δ unterhalb der Mittellinie liegen. Der Wert ergibt sich aus dem Berührungspunkt $z_0 = (x_0, y_0)$ der Tangente mit der Steigung m an die ovale Grundform der »runden« Buchstaben zu $\Delta = ht - y_0$. Siehe Abbildung 4. Das Zeichen ragt dann um $\delta = a - |x_0|$ nach links aus der Boundingbox heraus. In METAFONT lässt sich der Punkt z_0 mit dem Befehl `directionpoint` leicht berechnen gemäß:

```

1 path p; % Superellipse
2 p:=(a,0){up}..sigma*(a,b){-a,b}..(0,b){left}..
3   sigma*(-a,b){-a,-b}..(-a,0){down}..sigma*(-a,-b){a,-b}..
4   (0,-b){right}..sigma*(a,-b){a,b}..cycle;
5 z0=directionpoint (-1,-m) of (subpath(2,4) of p);

```

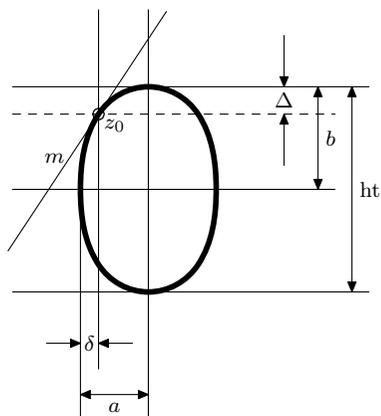


Abbildung 4: Konstruktion des Verbindungsniveaus.

Für die Sütterlinschrift reicht dieses 1. Verbindungsniveau aus. Bei der VA werden wir noch ein 2. Verbindungsniveau und bei der Gruppe der lateinischen Schriften DN, LA und SAS (L-Gruppe) sogar ein 3. Verbindungsniveau benötigen. Dies liegt bei der VA daran, dass viele Buchstaben wie i, j, m, n usw. per Definition auf der Mittellinie senkrecht nach unten (ohne Vorstrich) beginnen, sodass man die Mittellinie als 2. Verbindungsniveau hinzunehmen wird. Zur Verbindung zwischen den beiden Niveaus fügt man gegebenenfalls ein Ergänzungsstück ein.

e-Problem

Die Schriften der L-Gruppe verwenden das sogenannte »Schleifchen-e« ℓ , das zu dem folgenden Problem führt. Der Buchstabe »e« lässt sich bei diesen Schriften als einziges Zeichen nicht mit den bislang besprochenen Methoden verbinden, weil er »unten« in der Nähe der Grundlinie beginnt und auch dort enden muss, um mit einem eventuell nachfolgenden »e« verbunden zu werden. Dies erzwingt die Einführung eines 3. Verbindungsniveaus auf der Grundlinie. Die Steigung ist dort waagrecht und nach Buchstaben mit Unterlänge (z. B. g) gleich m .

Diese ganze Problematik gibt es bei der VA nicht, weil das dort verwendete »Köpfchen-e« ein »runder« Buchstabe ist, der wie beispielsweise »a« tangential auf dem 1. Verbindungsniveau problemlos verbunden werden kann.

Verbundene Buchstaben und Endformen

Die mit den definierten Verbindungsniveaus entworfenen Buchstaben sind »verbundene Buchstaben«, die im Innern eines Wortes stehen. Da derselbe Buchstabe ein etwas anderes Aussehen besitzt, wenn er am Wortanfang oder am Wortende auftritt, muss man mit Ausnahme der VA auch eine »Endform« für alle Buchstaben entwerfen.

Bei der L-Gruppe benötigt man neben der verbundenen Form statt der »Endform« eine entsprechende Sonderform zur Verwendung vor »e«. Diese Formen enden alle auf dem 3. Verbindungsniveau (Grundlinie). Steht ein Zeichen am Wortende, fügen wir diesem ein Sonderzeichen in Form eines kurzen Endstrichs an.

Am Wortanfang muss bestimmten Buchstaben noch ein Vorstrich als Spezialzeichen vorangestellt werden.

Weitere Konsequenzen des e-Problems

1. Bei den Schriften der L-Gruppe benötigt man zur Verbindung der auf der Mittellinie endenden Buchstaben b, o, r, v und w mit einem nachfolgenden »e«, zwei Sonderformen von e, die senkrecht oder schräg auf der Mittellinie (2. Verbindungsniveau) beginnen und auf der Grundlinie (3. Verbindungsniveau) enden.



2. Die Verbindung der Großbuchstaben B, D, F, I, N, O, Ö, P, S, T, V, W mit »e« gestaltet sich bei der L-Gruppe besonders schwierig, weil bei diesen Buchstaben der entsprechende Verbindungspunkt auf dem 3. Niveau weit links vom rechten Rand der Boundingbox liegt. Siehe Abbildung 5. Diesen Abstand nennen wir kern. Für jeden der genannten Großbuchstaben hat kern einen anderen Wert. Er bestimmt, um wieviel das »e« zusätzlich durch Kerning (Unterschneiden) beispielsweise an ein »P« herangeschoben werden muss. Damit sich aber die beiden Buchstaben nicht überdecken, benötigt man Sonderformen von »e« mit unterschiedlichen, relativ langen Aufstrichen (Breite λ).

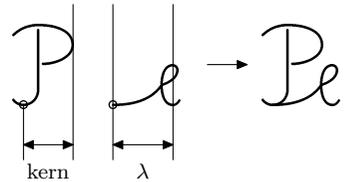


Abbildung 5: Kerning.

3. Eine zusätzliche Besonderheit weist die SAS auf. Hier benötigt man zur Verbindung der Buchstaben A, F und H mit »e« echte Ligaturen als Sonderzeichen.



Spitz verbundene Buchstaben

Die Buchstaben i, j, p, t, u, ü, y der L-Gruppe beginnen auf der Mittellinie (2. Verbindungsniveau) senkrecht nach unten und werden dort mit dem vorangehenden Buchstaben nicht schräg, sondern spitz, d. h. tangential senkrecht nach oben verbunden gemäß Abbildung 6. Dies erreicht man, indem man den normalen schrägen Verbindungsstrich mit der Steigung m durch eine Sonderform ersetzt, die senkrecht auf der Mittellinie endet. Für diesen Aufstrich gilt sinngemäß das Gleiche wie für »e« in Punkt 2.

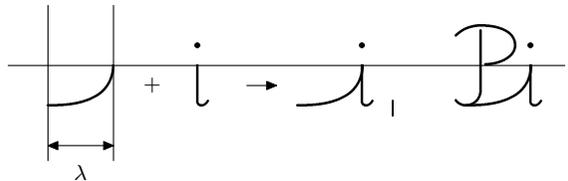


Abbildung 6: Verbindung zu »spitzen« Buchstaben.

Vorzeichnung

Nach diesen Vorbereitungen sind wir nun in der Lage, die einzelnen Groß- und Kleinbuchstaben auf Millimeter- oder Karopapier maßstabsgerecht aufzuzeichnen. Man sollte nicht zu klein zeichnen, um die Graphik gut ausmessen zu können. Andererseits verliert man bei zu großem Maßstab die Übersicht und das Gefühl für Proportionen. Ich habe als Linienabstand 20 mm (DN: 20 mm : 30 mm : 20 mm) gewählt und als Zeicheneinheit 1 mm, d. h. alle Maße werden in mm angegeben. Der Wert für Δ liegt bei ca. 12 % der Höhe des Mittelbandes. Für die SU verwenden wir $\Delta = 2$ ut. Bei den anderen Schriften wird der exakte Wert gemäß Abbildung 4 ermittelt.

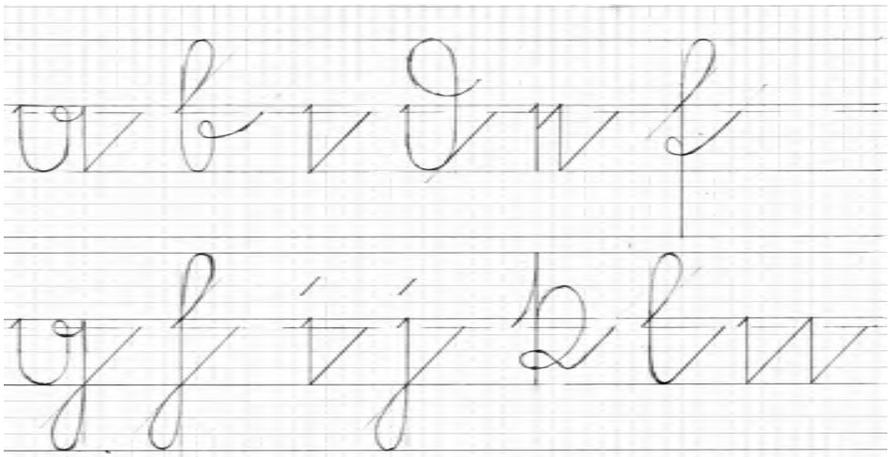


Abbildung 7: Ausschnitt aus meiner Vorzeichnung für die Sütterlinschrift.

METAFONT-Implementierung

METAFONT ist eine Graphik-Programmiersprache zur Beschreibung der Anordnung von Punkten in einem rechtwinkligen Koordinatensystem und deren Verbindung durch gerade oder gekrümmte Linien (Bézier-Kurven), wobei man im letzteren Fall auch in jedem Punkt eine Richtung vorschreiben kann. Geschlossene Linienzüge können gefüllt werden. Man kann Datentypen, Variable und Makros definieren. Es gibt Befehle zur affinen Transformation, zur Manipulation graphischer Elemente, für algebraische Berechnungen, zur Lösung linearer Gleichungen und für Steuerstrukturen. Die vollständige Dokumentation enthält das METAFONTbook von Donald Knuth [7, 8].

Mit diesen Programmanweisungen beschreibt man die Form (Linienzug) der einzelnen Schriftzeichen in einer Quelldatei. Nach der METAFONT-Bearbeitung erhält man als Ausgabe zwei Dateien, die die Abmessungen der Boundingbox bzw. die zugehörige Pixel-Graphik jedes Zeichens enthalten. Auf diese Schriftinformation wird bei der latex/dvips-Bearbeitung eines \TeX -Dokuments zugegriffen (siehe Seite 50).

Eine einfache Entwurfsumgebung

Durch beständigen Wechsel zwischen Editor und METAFONT gemäß der Befehlssequenz `emacs show.mf`, `mf show`, `Ctrl-D` erhält man eine sehr einfache und effiziente Arbeitsumgebung zum METAFONT-Entwurf eines Buchstabens oder Zeichens. Dazu habe ich ein kleines Programm `show.mf` geschrieben, das folgenden Aufbau besitzt:

```

1 screenstrokes;
2 screen_cols:=1024; screen_rows:=768;
3 def openit=openwindow currentwindow from origin to
4   (screen_rows,screen_cols) at (-50,450); enddef;
5 ut:=6.72;
6 pickup pencircle scaled 0.5ut;
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% METAFONT-Code fuer einen Buchstaben
8
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Hilfslinien
10 laenge:=135ut;
11 pickup pencircle scaled 0.2ut;
12 draw (0,-dt)--(0,Ht); draw (0,Ht)--(laenge,Ht);
13 draw (0,ht)--(laenge,ht); draw (0,0)--(laenge,0);
14 draw (0,-dt)--(laenge,-dt);
15 raster:=laenge/100;
16 for i=0 upto 50: draw (2*i*raster,ht-Delta)--
17   ((2i+1)*raster,ht-Delta); endfor;
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF, das File nicht mit 'end' abschliessen!

```

Der Befehl `screenstrokes` in Zeile 1 bewirkt, dass das Ergebnis aller Zeichenbefehle (`draw`) nach der METAFONT-Bearbeitung sofort auf dem Bildschirm angezeigt wird. Mit `Ctrl D` beendet man die Anzeige. Die Zahlenwerte in den Zeilen 2, 4 und 5 beziehen sich auf die Größe und Auflösung des Bildschirms. Sie müssen gegebenenfalls angepasst werden. Für meinen Rechner gilt beispielsweise:

Diagonale $D = 15''$, $\text{cols} \times \text{rows} = 1024 \times 768$. Gewählt: Nulllinie der Zeichnung auf row 450 (von oben), linker Rand 50 cols. Die Zeicheneinheit habe ich gemäß

$$ut = 2\sqrt{\text{cols}^2 + \text{rows}^2} / (D \cdot 25.4) = 6.72 \quad (4)$$

so festgelegt, dass sie auf dem Bildschirm einer Länge von 2 mm entspricht, also Bildschirmbild zu Handzeichnung = 2 : 1. Damit kann man die beiden Darstellungen gut vergleichen und ausmessen.

Nach Zeile 7 fügt man den eigentlichen METAFONT-Code für den zu entwerfenden Buchstaben ein. In den Zeilen 9–17 werden zur besseren Orientierung noch das Liniensystem und das Verbindungsniveau eingezeichnet.

Zeile 18: Die Datei endet ohne abschließenden end-Befehl.

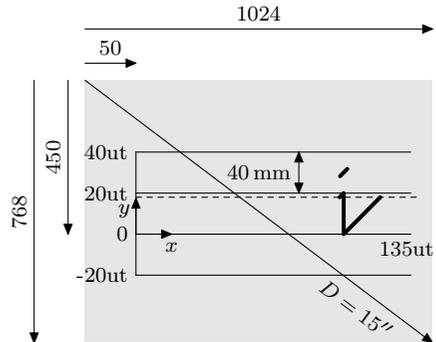


Abbildung 8: Die Abmessungen des Bildschirms und die Anordnung des Liniensystems.

Makros für Formteile und Positionierungen

Anhand eines einfachen Beispiels wollen wir zeigen, wie man mit Hilfe der Entwurfsumgebung `show`, ausgehend von der Vorzeichnung in Abbildung 7, den METAFONT-Code für den Buchstaben »c« der Sütterlinschrift entwerfen kann. Die direkte Umsetzung ergibt

```

1 ht:=20ut; dt:=20ut; Ht:=40ut;
2 draw (0,18ut)--(2ut,20ut)--(2ut,0)--(20ut,18ut);

```

Diese Anweisungen fügen wir mit dem Editor in Zeile 8 der Datei `show.mf` ein. Nach dem Verlassen des Editors geben wir das Shellkommando `mf show` ein. Anschließend erscheint sofort das Ergebnis auf dem Bildschirm.

Diese Lösung ist zwar einfach, und wie man sieht, auch korrekt, aber nicht zweckmäßig, weil z. B. ein anderer Wert für Delta Änderungen der Zahlenwerte an vielen Stellen zur Folge hätte. Außerdem ist es schwierig, bei komplizierteren Buchstaben die x -Koordinate der Punkte relativ zum Nullpunkt korrekt zu bestimmen.

Formteile:

Deshalb zerlegen wir die Buchstaben in elementare *Formteile*, die vielen Buchstaben gemeinsam sind, und benennen sie sinnfällig. In unserem Beispiel sind solche Formteile, der kurze Aufstrich zu Beginn (kurzstrich), der senkrechte Abstrich (stamm) und der Verbindungsstrich (strich). Für jedes Formteil schreiben wir ein Makro. Im Einzelnen:

```

1 def kurzstrich(expr sx)=
2 draw ((0,ht-Delta)--(Delta,ht)) shifted (sx,0);
3 xpos:=sx+Delta;
4 enddef;
5 def stamm(expr ha, hb, sx)=
6 draw ((0,ha)--(0,hb)) shifted (sx,0);
7 xpos:=sx;
8 enddef;
9 def strich(expr sx)=
10 draw ((0,0)--(ht-Delta,ht-Delta)) shifted (sx,0);
11 xpos:=sx+ht-Delta;
12 enddef;

```

Das Programm für das »c« lautet jetzt

```

1 xpos:=0;
2 kurzstrich(xpos);
3 stamm(ht,0,xpos);
4 strich(xpos);
5

```

xpos-Mechanismus:

Jedes Formteil wird innerhalb des Makros relativ zu $x = 0$ entworfen und beim Aufruf an die Stelle xpos geschoben, wobei die Variable xpos am Ende jedes Makros aktualisiert wird, d. h. xpos wird um den Abstand in x -Richtung zwischen Anfangs- und Endpunkt des Linienzugs des entsprechenden Formteils erhöht. Zum Schluss ist der Wert von xpos gleich der Breite w des Buchstabens. Diese Weite w wird im folgenden noch benötigt werden.

Variablen und Parameter:

Im Unterschied zu der ersten Lösung verwenden wir keine expliziten Zahlenwerte in den Zeichenbefehlen sondern Variablennamen für Größen, die für die ganze Schrift gelten. Die Zuweisung der Zahlenwerte an diese globalen Variablen erfolgt an zentraler Stelle am Anfang des Programms gemäß

```

1 ht:=20ut;
2 Delta:=2ut;

```

Beispiel: Buchstabe »h« der SU (siehe Abbildung 9).

Wir definieren zwei weitere sehr häufig auftretende Formteile, Oberschleife (oschl) und Unterschleife (uschl). Als Parameter führen wir die Schleifenbreite sb und den Faktor sbf ein, um die Punkte der Schleife mit waagrechtter und senkrechter Tangente festzulegen.

```

1 dt:=ht;
2 Ht:=ht+dt;
3 sb:=10ut;
4 sbf:=0.4;
5 def oschl(expr sx)=
6   draw ((0,ht){1,1}..(sb,Ht-sb*sbf){up}..(sb*(1-sbf),Ht){left}
7     ..tension1.2..{down}(0,ht)) shifted (sx,0);
8   xpos:=sx;
9   enddef;
10 def uschl(expr sx)=
11   draw ((0,0){down}..tension1.2..(-sb*(1-sbf),-dt){left}..
12     (-sb,-dt+sb*sbf){up}..{1,1}(0,0)) shifted (sx,0);
13   xpos:=sx;
14   enddef;

```

Der Programm-Code für »h« lautet dann:

```

1 xpos:=0;
2 kurzstrich(xpos);
3 oschl(xpos);
4 stamm(ht,0,xpos);
5 uschl(xpos);
6 strich(xpos);

```

Wenn uns die Form der Schleifen nicht gefällt, müssen wir nur an einer einzigen Stelle im Programm die Zahlenwerte für die Parameter *sb* und *sbf* korrigieren, dann sind anschließend bei den Buchstaben *b*, *f*, *h*, *l* und *g*, *j*, *p*, *y*, *z* alle Schleifen einheitlich geändert.

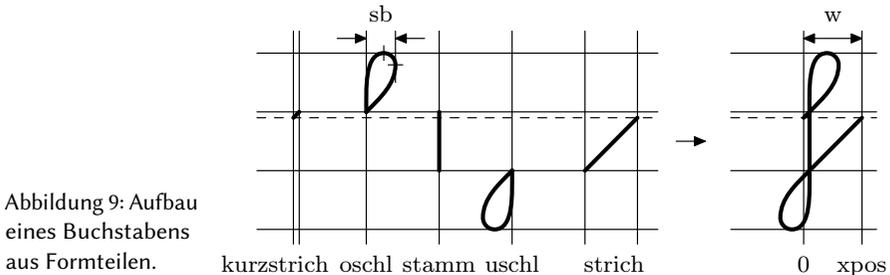


Abbildung 9: Aufbau eines Buchstabens aus Formteilen.

Quelldatei:

Wenn ein Zeichen auf diese Weise zufriedenstellend entworfen ist, kopiert man dessen Code in eine für alle Zeichen gemeinsame Quelldatei und umgibt das Codestück mit einer Umgebung `beginchar`-`endchar`:

```

1 beginchar("h",20ut#,Ht#,dt#);
2 pickup wesu_pen;

```

```

3 <Code>
4 endchar;

```

Die Variablen und Makrodefinitionen sammelt man am Beginn der Quelldatei. Die vier Angaben in `beginchar` beziehen sich auf die Position des Zeichens in der Fonttabelle und die Abmessungen Breite (`width w`), Höhe (`height h`) und Tiefe (`depth d`) des Zeichens in absoluten Maßen.

Zur Bestimmung von `w` ist der `xpos`-Mechanismus sehr hilfreich. Wenn man die Breite nicht kennt, lässt man sich einfach den Wert von `xpos` für jeden Buchstaben mit der Anweisung `zeigen` ausgeben. Das Makro `zeigen` definieren wir dazu wie folgt:

```

1 def zeigen=
2   weite:=20*xpos/ht;
3   showvariable weite;
4 enddef;

```

Anschließend trägt man die ausgegebenen Zahlenwerte als zweiten Parameterwert in die entsprechenden `beginchar`-Anweisungen ein.

Schrifttabelle der T1-Kodierung

Wir verwenden die Schrifttabelle der T1-Kodierung, die 256 Zeichen zu 8 Bit festlegt. Die Positionen sind von 0 bis 255 durchnummeriert. Die untere Hälfte der Tabelle ist der ASCII-Bereich, der auf den Positionen 32 bis 127 die druckbaren Zeichen enthält. Dazu gehören die Großbuchstaben (Pos. 65–90), die Kleinbuchstaben (Pos. 97–122) und die Ziffern (Pos. 48–57). Die restlichen Positionen enthalten ASCII-Symbole wie die Satzzeichen und Klammern. Der Bereich der nicht druckbaren ASCII-Zeichen auf den Pos. 0–31 ist bei der T1-Kodierung belegt u. a. mit Akzenten und Ligaturen. Davon verwenden wir die in der Tabelle angegebenen Positionen mit gleicher Bedeutung auch für die Schreibschriften.

Position	14	15	16	18	19	20	21
Zeichen	<	>	„	“	«	»	–
TeX-Befehl	<code>\flq</code>	<code>\frq</code>	<code>\glqq</code>	<code>\grqq</code>	<code>\flqq</code>	<code>\frqq</code>	--
wesu-Font	⟨	„	“	«	»	–	

Die restlichen Positionen belegen wir nach Bedarf mit Sonderzeichen, die für eine Schrift benötigt werden. Die Umlaute und das »scharfe s« legen wir ebenfalls auf die in der T1-Kodierung dafür vorgesehenen Plätze, sodass diese in TeX wie üblich verwendet werden können (z. B. `\”a` für ä).

Die bei den meisten Schriften erforderlichen Endformen oder Formen-vor-e legen wir in den oberen Bereich, 64 Positionen oberhalb der entsprechenden Positionen

für die verbundenen Formen im ASCII-Bereich. Die Abbildung 10 zeigt als Beispiel die Fonttabelle der Lateinischen Ausgangsschrift (LA).

Endformen:

Da wir jeden Buchstaben auch in einer Endform entwerfen müssen, die sich aber von der verbundenen Form nur in der Gestaltung des letzten Formteils unterscheidet, verzweigen wir anhand der logischen Variablen wortende gemäß

```

1 if not wortende:
2   beginchar("h",2Out#,Ht#,dt#);
3 else:
4   beginchar(oct"250",7ut#,Ht#,dt#);
5 fi
6 pickup wesu_pen;
7 <code>
8 zeigen;
9 endchar;

```

Das Makro `strich` erweitern wir dazu wie folgt

```

1 def strich(expr sx)=
2 if not wortende:
3   draw ((0,0)--(ht-Delta,ht-Delta)) shifted (sx,0);
4   xpos:=sx+ht-Delta;
5 else:
6   draw ((0,0)--(endx,endx)) shifted (sx,0);
7   xpos:=sx+endx;
8 fi
9 enddef;

```

Der Parameter `endx` ist die Breite des Aufstrichs am Ende des Buchstabens, z. B. `endx:=9ut;`

Datei-Hierarchie

In der Praxis wird man die Quelldatei aufteilen in eine *Hauptdatei* (driver file), mehrere *Programmdateien* für Groß- und Kleinbuchstaben sowie Satzzeichen, Ziffern und Sonderzeichen sowie einige *Parameterdateien*, die wichtige Steuergrößen setzen.

Dadurch entsteht folgende Dateihierarchie:

```

1 % Parameterdateien:
2 <font>bsl14.mf % setzt bold=1 und laedt <font>sll4.mf
3 <font>sll4.mf % setzt slant (z.B. SU slant=0.25) und lädt die
   ↪Hauptdatei
4 <font>b14.mf % setzt bold=1 und lädt die Hauptdatei
5
6 % Hauptdatei:
7 <font>14.mf % definiert Font, fontsize, Masse,
8 % sharp variables and Pixel-Groessen,
9 % Federn, Transformation slanted,

```

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	ℓ	ℓ	ℓ	ℓ	ℓ	/	.	.	"0x
'01x	ℓ	ℓ	ℓ				<	>	
'02x	"	"	"	«	»	-	-	~	"1x
'03x	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	~	
'04x		!	"	#	\$	%	&	'	"2x
'05x	()	*	+	.	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	>	=	<	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z						
'14x		a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z						
'20x	/	A	B	C	D	E	F	G	"8x
'21x	H	I	J	K	L	M	N	O	
'22x	P	Q	R	S	T	U	V	W	"9x
'23x	X	Y	Z	Ä	Ö	Ü		§	
'24x	€	a	b	c	d	e	f	g	"Ax
'25x	h	i	j	k	l	m	n	o	
'26x	p	q	r	s	t	u	v	w	"Bx
'27x	x	y	z	ä	ö	ü			
'30x					Ä				"Cx
'31x									
'32x							Ö		"Dx
'33x					Ü			ß	
'34x					ä				"Ex
'35x									
'36x							ö		"Fx
'37x					ü			ß	

Abbildung 10: Fonttabelle der Lateinischen Ausgangsschrift.

```

10      % setzt wortende=false und lädt die Programmdateien.
11
12 % Programmdateien:
13 <font>14_def.mf % Variablen und Makrodefinitionen
14 <font>14_gr.mf  % Grossbuchstaben
15 <font>14_kl.mf  % Kleinbuchstaben
16 <font>14_sz.mf  % Satzzeichen, Ziffern und Sonderzeichen
17 <font>14_end.mf % Parameterdatei, setzt wortende=true
18                % und lädt nochmals die Dateien für
19                % Gross- und Kleinbuchstaben
20 <font>14_lig.mf % Ligatortabellen

```

Hauptdatei

In der Hauptdatei `14.mf` vereinbart man in einer Art Präambel einige METAFONT-spezifische Einstellungen und die für alle Schriftzeichen eines Fonts gemeinsamen Größen. Am Ende werden mit `input` die Programmdateien geladen. Die Datei wird mit dem Befehl `end` abgeschlossen. Das folgende Listing zeigt als Beispiel den Aufbau dieser Datei für die Sütterlinschrift.

```

1 font_identifier "WESU";
2 font_size 14pt#;
3 font_coding_scheme := "WESU text";
4 mode_setup;
5 % Vereinbarungen:
6 boolean wortende;
7 path p, q;
8 % sharp Variables:
9 % unit_width
10 ut#:=0.233pt#;
11 % Linienabstand
12 ht# := 2Out#;
13 dt# := ht#;
14 Ht#:=ht#+dt#;
15 % Umrechnung in Pixel-Groessen:
16 define_pixels(ut,ht,Ht,dt);
17 % Federn:
18 % runde Feder (Redisfeder)
19 if unknown bold:
20   pickup pencircle scaled 1.25ut; else:
21   pickup pencircle xscaled if unknown slant: 3ut else:
22     3.5ut fi yscaled 0.5ut rotated
23     if unknown slant: 45 else: 39 fi; fi
24 wesu_pen:=savepen;
25 % fuer Punkte etc.
26 pickup pencircle scaled 4ut;
27 wesu_pen_thick := savepen;
28 % Current Transformation, slanted:
29 if unknown slant: slant:=0; else:
30   currenttransform:=identity slanted slant; fi
31 % Eingabe der Program-Files:

```

```

32 wortende:=false;
33 % Parameter- und Makrodefinitionen
34 input wesu14_def;
35 % Grossbuchstaben
36 input wesu14_gr;
37 % Kleinbuchstaben
38   input wesu14_kl;
39 % Sonderzeichen (Ziffern, Satzzeichen, ...)
40   input wesu14_sz;
41 % Buchstabenformen am Wortende
42   input wesu14_end;
43 % Abstandsmasse:
44 font_quad := 14pt#;% 1em
45 font_x_height := 4.667pt#; % 1ex, 1/3 von 14pt
46 font_normal_space := 6pt#; % Wortzwischenraum
47 font_normal_stretch := 3pt#;
48 font_normal_shrink := 2pt#;
49 boundarychar:=oct"040"; % space
50 % Ligaturtabellen
51 input wesu14_lig;
52 end

```

Bestimmung der erforderlichen Angaben:

Zeile 2: Die Größe der zu entwerfenden Schrift wird mit `font_size` festgelegt. 14pt entspricht ungefähr der Höhe einer normalen Handschrift.

Zeile 10: Das Liniensystem der Handzeichnung ist 60 mm (DN: 70 mm) hoch, die Zeicheneinheit ist 1 mm, also 60 Längeneinheiten. Dies soll in METAFONT einer Entwurfsgröße (`font_size`) von 14 pt bei einer Einheitslänge von 1 ut entsprechen. Daraus folgt $60 \text{ ut} = 14 \text{ pt}$ oder allgemein

$$\text{ut} = \frac{\text{font_size in pt}}{\text{Längeneinheiten der Höhe des Liniensystems}}, \quad (5)$$

daraus folgt $\text{ut}\# = 0.233 \text{ pt}\#$ (DN: $\text{ut}\# = 0.23 \text{ pt}\#$). Das Zeichen # gibt an, dass diese Größen absolute Werte darstellen, sogenannte *sharp variables*.

Zeilen 12–14: betreffen das Liniensystem. Das Mittelband ist in der Handzeichnung $20 \text{ mm} = 20$ Zeicheneinheiten hoch. Folglich ist die entsprechende Variable $\text{ht}\# = 20 \text{ ut}\#$ (DN: $\text{ht}\# = 30 \text{ ut}\#$). Bei einer Teilung von 1 : 1 : 1 (DN: 2 : 3 : 2) erhält man mit $\text{dt}\# = \text{ht}\#$ (DN: $\text{dt}\# = 20 \text{ ut}\#$) für die Höhe der Großbuchstaben $\text{Ht}\# = \text{ht}\# + \text{dt}\#$.

Zeile 16: Der Befehl `define_pixels` rechnet die absoluten Werte in Pixelgrößen um, die im Folgenden zum Zeichnen verwendet werden. Aus $\text{ut}\#$ wird `ut`, usw.

Zeilen 19–24: Definition der Federn (`pen`). Wenn die Variable `bold` definiert ist, verwenden wir zum Schreiben eine Bandzugfeder anstelle der Redisfeder. Entsprechend bestimmt die Variable `slant` die Neigung der Schrift.

Redisfeder: Als Strichstärke verwendet man ungefähr 2 % von `font_size`, also $0.02 \cdot 14 \text{ pt} = 0.28 \text{ pt} = 0.28/0.233 = 1.2 \text{ ut}$ (gewählt: 1.25 ut), dann erhält man im Vergleich zur `cmr`-Schrift fast den gleichen Schwärzungsgrad.

Bandzugfeder: Als Form der Bandzugfeder wählen wir eine flache Ellipse mit einem Achsenverhältnis $dx : dy$ von 6 : 1. Damit bleiben auch die feinen Linien noch gut sichtbar. Der Anstellwinkel sollte nach einer alten Druckerregel in Richtung der feinsten Linien gewählt werden. Bei der SU sind die feinsten Linien die Verbindungsstriche, sodass der Anstellwinkel gleich $\alpha = 45^\circ$ ist. Verwenden wir als `slant`-Faktor 0.25 ($\beta = 76^\circ$), ergibt sich nach der Gleichung auf Seite 33 $\alpha' = 39^\circ$. Die Strichstärke d sollte bei senkrechter und geneigter Schrift gleich groß sein und ungefähr das 2-fache des bei der Redisfeder verwendeten Wertes betragen, um einen merklich dekorativeren Schriftzug zu erhalten, gewählt $d = 2.2 \text{ ut}$. Gibt man die Strichstärke $d = d'$ vor, berechnet sich die Breite dx und dx' der Bandzugfeder für senkrechte und geneigte Schrift gemäß Abbildung 11 aus

$$d = dx \cdot \sin(90^\circ - \alpha) \quad (\text{senkrecht}), \quad d' = dx' \cdot \sin(\beta - \alpha') \quad (\text{geneigt}) \quad (6)$$

zu $dx = 3.1 \text{ ut}$ bzw. $dx' = 3.6 \text{ ut}$, gewählt 3 ut bzw. 3.5 ut.

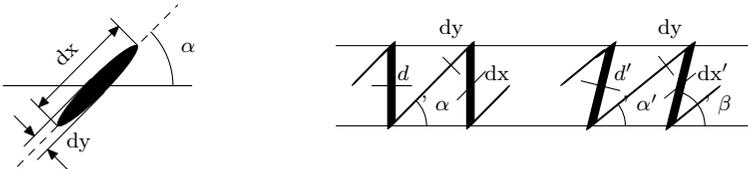


Abbildung 11: Bandzugfeder: Querschnitt. Schrift senkrecht und geneigt: Strichstärke d , Federbreite dx , Federhöhe dy .

Zeilen 25–27: Als Durchmesser für Punkte verwenden wir ungefähr den 3-fachen Wert der Strichstärke der Redisfeder: gewählt 4 ut.

Zeilen 28–30: METAFONT verwaltet eine im Hintergrund mitlaufende Transformation, die implizit auf alle Zeichenbefehle (`draw`) wirkt. Auf diese Weise lässt sich die Neigung einer Schrift automatisch mit der Transformation `slanted` berücksichtigen.

Zeilen 32–42, 50–51: Die Treiberdatei lädt mit `input` die Programmdateien, die den eigentlichen METAFONT-Code für die Schriftzeichen enthalten. Da jeder Buchstabe in zwei verschiedenen Formen entworfen werden muss (»verbundene Form« und »Endform/Form-vor-e«), setzt man die logische Variable `wortende` zunächst auf `false` und erzeugt die verbundenen Buchstaben. Die Parameterdatei `14_end.mf` setzt dann `wortende` auf `true` und lädt nochmals die Dateien

`14_kl.mf` und `14_gr.mf`, sodass dann die Endformen der Groß- und Kleinbuchstaben erzeugt werden.

Zeilen 44–48: Hier werden die Abmessungen der \TeX -Größen `1 em`, `1 ex` und `\space` definiert, die man wie folgt bemisst: `1 em` ist die Breite und Höhe eines Drucker-Gevierths, wobei die Höhe gleich der `font_size` ist, also `font_quad=14pt`. `1 ex` ist die Höhe des Mittelbandes (`font_x_height`), also $1/3 \cdot 14 \text{ pt} = 4.667 \text{ pt}$ (DN: $3/7 \cdot 14 \text{ pt} = 6 \text{ pt}$). Der Wortzwischenraum entspricht ungefähr der Breite des Kleinbuchstabens »n«. Diese beträgt ca. $25 \text{ ut} = 25 \cdot 0.233 \text{ pt} = 5.82 \text{ pt}$, gewählt `6 pt (+3 pt, -2 pt)`.

Parameterdateien

Parameterdateien haben einen sehr einfachen Aufbau. Die Datei `wesubs14.mf` für die mit Bandzugfeder geschriebene, geneigte Sütterlinschrift lautet:

```
1 bold:=1;
2 input wesus14.mf;
```

Die Datei `wesus14.mf` für die geneigte Sütterlinschrift lautet:

```
1 slant:=0.25;
2 input wesu14.mf;
```

Die Datei `wesub14.mf` für die mit Bandzugfeder geschriebene, senkrechte Sütterlinschrift lautet:

```
1 bold:=1;
2 input wesu14.mf;
```

und die Datei `wesu14_end.mf` zur Generierung der Endformen für die Groß- und Kleinbuchstaben lautet:

```
1 wortende:=true;
2 input wesu14_gr.mf;
3 input wesu14_kl.mf;
```

Ligaturtabellen

Die Dateien `14_lig.mf` enthalten die Ligaturanweisungen für jede Schrift. Eine Ligaturanweisung verändert entweder den Abstand zwischen zwei aufeinanderfolgenden Zeichen (Kerning) oder fügt ein Zeichen s_3 zwischen zwei aufeinanderfolgende Zeichen s_1 und s_2 ein. Die Zeichen s_1 , s_2 werden anschließend entfernt, wenn sie nicht einzeln oder beide entsprechend markiert werden.

Beispielsweise benötigt man bei der LA folgende Ligaturanweisung für die Zeichenfolge »Pe« und für »P« mit Verbindung zu spitzen Buchstaben

```

1 ligtable oct"220": oct"004" kern Pkern#,
2 oct"034" kern Pkern#;

```

Der Wert für Pkern# wurde zuvor beim Entwurf des Buchstabens »P« ermittelt und in die Datei wela14_def.mf eingetragen gemäß Pkern#:=–25ut#;.

Bei allen Schriften benützen wir für den Gedankenstrich die folgende Ligaturanweisung

```

1 ligtable "-": "-" =: oct"025";

```

Wie man anhand der beiden Beispiele sieht, sind die Ligaturanweisungen sehr einfach, aber nicht besonders mächtig, sodass die Ligaturdatei im allgemeinen sehr viele dieser elementaren Anweisungen enthalten wird.

Da sich bei unseren Schreibschriften die Buchstaben am Wortanfang und Wortende von den verbundenen Buchstaben im Wortinneren unterscheiden, benötigen wir u. a. Kriterien, um Wortanfang und -ende zu erkennen und danach entsprechende Aktionen auszuführen:

1. Wortende erkennen, Endformen verwenden:

Ein Buchstabe steht am Wortende, wenn nach ihm eines der Zeichen □ ! “ . , ;) ’ ? usw. folgt.

2. Wortanfang erkennen, Aufstrich voranstellen:

Wenn einem Buchstaben eines der Zeichen || 0 1 2 3 4 5 6 7 8 9 □ „ (‘ - usw. vorangeht, steht er am Anfang eines Wortes.

3. »Spitze« Buchstaben erkennen, Verbindungsstück einfügen:

Diese Situation liegt vor, wenn auf einen verbundenen Buchstaben ein »spitzer« Buchstabe (bei der VA z. B. b, f, h, i, ...) folgt.

4. Echte Ligaturen erkennen, durch Sonderzeichen ersetzen:

Bei der VA werden beispielsweise die Buchstabenfolgen st, ßt, tz durch spezielle Zeichen ersetzt, z. B.

```

1 ligtable "s": "t" =: oct"034";

```

5. Bestimmte Schriften erfordern gegebenenfalls weitere Ligaturanweisungen. Beispielsweise gibt es bei der Sütterlinschrift ein spitzes und ein rundes »s«.

Das runde s bei der Sütterlinschrift

Eine Besonderheit der deutschen Schriften ist die unterschiedliche Form des Buchstabens »s«. Im Prinzip wird stets die spitze Form »ſ« verwendet, lediglich am Wortende steht die runde Form »ſ«, das sogenannte »Schluss-s«. Dies wird in den Ligaturtabellen automatisch berücksichtigt. Soll jedoch im Wortinneren ein rundes s geschrieben werden, z. B. bei zusammengesetzten Wörtern wie »Hausaufgabe«, muss man das »s« durch einen nachgestellten Doppelpunkt »:« markieren gemäß

Haus : aufgabe. Dies ergibt *haußaufgabe*. Diese Markierung hat aber zur Folge, dass ein echter Doppelpunkt nach einem Wort, das auf »s« endet, verdoppelt werden muss. Beispielsweise »\dots\ besteht aus:: Holz, Stein, \dots« ergibt ... *bnstft vrb: Holz, Stein, ...*

Workflow

Datei-Organisation

Es empfiehlt sich, die Dateien der Schriften in einem lokalen ~/texmf-Pfad abzulegen, der bei jedem \LaTeX - oder METAFONT-Lauf automatisch durchsucht wird.

Dazu legt man im Heimatverzeichnis folgendes Directory an:

```

1 ~/texmf/tex/latex/<font>/...      % fd sty tex eps pdf
2   /fonts/source/<font>/...      % mf
3   /tfm/<font>/...                % tfm
4   /pk/ljfour/<font>/...         % <pixel>pk

```

Dabei steht für wesu, wedn, wela, wesa und weva, d. h. für jede Schrift legt man jeweils ein eigenes Unterverzeichnis an.

Die Aktualisierung des Kpathsea-Dateisystems von \TeX erfolgt mit dem Shell-Kommando `sudo mktexlsr`.

Dann müssen noch die Besitzerrechte für die Datei `ls-R` geändert werden gemäß `sudo chown <user>:<user> ~/texmf/ls-R`, damit bei der \LaTeX -Bearbeitung die Datei `ls-R` ohne root-Rechte automatisch aktualisiert werden kann.

METAFONT-Bearbeitung

Die METAFONT-Bearbeitung der Schriftdateien erfolgt mit den Shell-Kommandos

```

1 mf '\mode=localfont;' input <font>14
2 mf '\mode=localfont;' input <font>s114
3 mf '\mode=localfont;' input <font>b14 % nur SU
4 mf '\mode=localfont;' input <font>bs114 % nur SU

```

Als Ausgabe erhält man die \TeX -Fontmetrikdateien `.tfm`, die man im lokalen texmf-Baum abspeichert. Die ebenfalls erzeugten `.<pixel>gf`-Dateien werden im weiteren nicht benötigt und können gelöscht werden.

Die METAFONT-Bearbeitung von Hand wird aber im allgemeinen nicht erforderlich sein, weil die `.tfm`-Dateien bei der Bearbeitung eines \LaTeX -Dokuments automatisch aus den `.mf`-Quelldateien erzeugt werden. Die Pixeldateien `.<pixel>pk` werden beim `dvips`-Lauf ebenfalls automatisch generiert.

Schriftdefinitionsdateien

Damit die Schriften in einem \LaTeX -Dokument verwendet werden können, muss man noch eine *Schriftdefinitionsdatei* schreiben, welche die Schriftfamilie und die Schnitte definiert und mit den Dateien verknüpft. Der Name der Schriftdefinitionsdatei ist bei Verwendung der T1-Kodierung von der vorgeschriebenen Form `t1.fd`, also z. B. für die Sütterlinschrift `t1wesu.fd`:

```

1 \DeclareFontFamily{T1}{wesu}{}
2 \DeclareFontShape{T1}{wesu}{m}{n}{<->wesu14.mf}{}
3 \DeclareFontShape{T1}{wesu}{m}{sl}{<->wesusl14.mf}{}
4 \DeclareFontShape{T1}{wesu}{b}{n}{<->wesub14.mf}{}
5 \DeclareFontShape{T1}{wesu}{b}{sl}{<->wesubsl14.mf}{}
6 \endinput

```

Der dritte und vierte Parameter gibt den Schnitt an: m medium (hier Redisfeder), n normal (senkrecht), sl slanted (geneigt), b boldface (fett, hier Bandzugfeder). Der fünfte Parameter ist der Name der METAFONT-Datei, mit dem die Schrift erzeugt wurde. Die Zeichen davor geben den Skalierungsbereich an. Das Zeichen `<->` bedeutet, dass die Schrift beliebig skaliert werden kann.

Schriftbefehlsdefinition (optional)

Zur Vereinfachung der Schriftauswahl in einem \LaTeX -Dokument kann man zusätzlich in einer Stildatei `.sty` entsprechende \LaTeX -Kommandos definieren. Zum Beispiel für die Vereinfachte Ausgangsschrift (Standardschrift)

```

1 \providecommand{\weva}{\usefont{T1}{weva}{m}{sl}}
2 \endinput

```

Fonttabelle ausdrucken

Mit dem »New-Font-Selection-Scheme«-Programm `nfssfont` kann man allgemein zu jedem Font die zugehörige Fonttabelle generieren, betrachten, abspeichern und ausdrucken.

```

1 latex nfssfont
2 ...
3 *<fontname eingeben, z.B. wesusl14>
4 *\table
5 *\bye
6
7 dvips nfssfont
8 gv nfssfont.ps

```

Um die korrekte Tabelle für unsere Schriften zu generieren, muss man zuvor in der Treiberdatei `14.mf` den Befehl `\input` für die Ligaturtabellen vorübergehend

deaktivieren. Abbildung 10 zeigt als Beispiel die Zeichentabelle für die Lateinische Ausgangsschrift.

Anwendung der Fonts in L^AT_EX-Dokumenten

Einfaches Beispiel

Das folgende Minimalbeispiel eines L^AT_EX-Dokuments `wela_xpl.tex` zeigt die prinzipielle Anwendung der Schrift LA:

```

1 \documentclass{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[german]{babel}
4 \begin{document}
5 \usefont{T1}{wela}{m}{sl}\fontsize{16pt}{18pt}\selectfont
6 Dieser Text ist in Lateinischer Ausgangsschrift geschrieben.
7 \end{document}

```

Nach der L^AT_EX-Bearbeitung erhält man

Dieser Text ist in Lateinischer Ausgangsschrift geschrieben.

Alternativ kann man mit `\usepackage{wela}` die Stildatei `wela.sty` laden und die Schriftauswahl mit dem dort definierten Befehl `\wela` vornehmen.

Die Schriftgröße wird mit `\fontsize{...}{...}\selectfont` festgelegt oder wie üblich mit den Schriftgrößenbefehlen `\Large` etc.

Damit der *Einzug* zu Beginn eines jeden Abschnitts den für die gewählte Schrift maßgeblichen Wert aufweist, muss man diesen unmittelbar nach der Schriftauswahl einstellen mit

```

1 \par\parindent1em

```

Textbeispiel

Als weiteres Beispiel formatieren wir einen Text in den vier Schnitten der Sütterlin-schrift. Dazu ersetzen wir in der obigen L^AT_EX-Datei die Zeilen 5–6 durch folgende Anweisungen

```

1 \fontsize{16pt}{18pt}\usefont{T1}{wesu}{<series>}{<shape>}
2 Es war ein Thema, das damals in der Luft

```

mit `<series>` `<shape>` nacheinander gleich `m n`, `m sl`, `b n` und `b sl`.

Es war ein Thema, das damals in der Luft

Es war ein Thema, das damals in der Luft

Es war ein Thema, das damals in der Luft
 Es war ein Thema, das damals in der Luft

Lineaturen

Die Stildatei `stil.sty` definiert einige praktische Befehle, um ein Liniensystem zu erzeugen und auf dieses zu schreiben. Wenn man davon Gebrauch machen will, muss man es mit `\usepackage{stil}` laden.

Befehle:

```
1 \Linien[Breite]{Text}
2 \Zeile[Breite]{Text}
3 \Linienblatt[Text]{Schriftgroesse}{Zeilenabstand}{Zeilenzahl}{Breite}
```

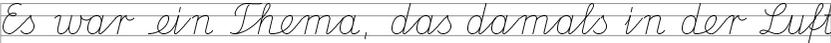
Hierbei gilt hellgrau als Hintergrundfarbe und dunkelgrau für die Linien.

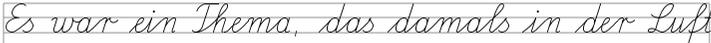
Der Befehl `\Zeile` unterscheidet sich von `\Linien` nur hinsichtlich der Farbe der Linien (dunkelgrau statt schwarz). Ohne explizite Angabe der `breite` richtet sich diese nach der `Breite` des Textes `text`. Der Befehl `\Linienblatt` beginnt eine neue Seite, wählt als Hintergrundfarbe hellgrau, stellt den Font ein und schreibt den Text schwarz in ein weißes Schriftfeld. Mit dem Befehl `\color{farbe}` unmittelbar vor `text` kann auch eine andere Farbe für die Schrift gewählt werden, z. B. `{\color{red}text}`.

Text auf Linien:

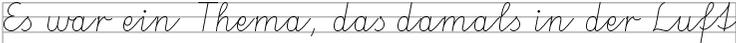
```
1 \fontsize{16pt}{18pt}\usefont{T1}{<font>}{m}{sl}
2 \Zeile{Es war ein Thema, das damals in der Luft}
```

mit `` nacheinander gleich `wedn`, `wela`, `wesa` und `weva`.

DN: 

LA: 

SAS: 

VA: 

Linienblatt

Formatierung eines Linienblatts für Schriftgröße 43 pt und Zeilenabstand 17 mm bestehend aus 11 Zeilen der Breite 12.5 cm.

```
1 \Linienblatt{43pt}{17mm}{11}{12.5cm}
```



Abbildung 12: Linienblatt (Ausschnitt, verkleinert).

LaTeX-Anpassung

Wenn man in einem Dokument, das die Standard-Layoutbefehle `section`, ..., `itemize`, `enumerate`, `description`, `theorem`, `table`, `figure`, `equation` usw. verwendet, als Schrift eine der Schreibschriften wählt, ist man zunächst enttäuscht, weil diese Teile nicht in Schreibschrift sondern in Computer Modern gesetzt werden. Dies betrifft auch die Seitenzahlen und die Nummerierung von Gleichungen, etc.

Die beigegefügte Stildatei ist ein Versuch, Abhilfe zu schaffen. Sie ist experimentell und vielleicht gibt es eine einfachere Lösung. Bei Schreibschriften ist *Unterstreichen* die einzige Auszeichnungsmöglichkeit, da man ja nicht **bold**, *italic* und verschieden groß und schräg schreiben kann.

Die Stildatei `_ltx.sty` ermöglicht mit den Befehlen `\normalschrift` und `\schreibschrift` in einem LaTeX-Dokument das Umschalten von der LaTeX-Standardformatierung auf die LaTeX-Anpassung für die verwendete Schreibschrift und umgekehrt. Das Kommando `\schreibschrift` bewirkt:

- Einstellung der Schrift.
- Verwendung dieser Schrift in Überschriften (sectioning).
- Setzen der Nummern für Seitenzahlen, Gleichungen, Tabellen, Bilder in der eingestellten Schrift.
- In Theoremen etc. wird Fettdruck durch *Unterstreichen* ersetzt.
- Außerdem stehen zum ein- oder zweifachen Unterstreichen von Text die Kommandos `\uline{text}` und `\uuline{text}` aus dem geladenen Paket `ulem` zur Verfügung.

Die praktische Anwendung zeigt das folgende Beispiel.

```

1 \schreibschrift %%%
2 \fontsize{14pt}{18pt}\selectfont\par\parindent1em
3 \section{\uline{Lesebuch, VA-Font}}
4 \subsection{\uline{Rationalität im Wandel der Zeit}}
5 Aber worin besteht diese wohl schon so oft
6 diskutierte Rationalität der Vorsokratiker?
7 \begin{enumerate}

```

```

8 \item Nach der Pause, wenn's geht, wird die Besprechung
9 mit den Gästen im Seminarraum stattfinden
10 \end{enumerate}
11 \begin{description}
12 \item\uline{Vorwort} Das Buch beginnt mit einem Vorwort,
13 in welchem die Zielsetzung und der wesentliche Inhalt
14 \end{description}
15
16 Und jetzt schalten wir wieder auf Normalschrift um.
17 \normalschrift %%%
18 \section{Normalfont}
19 Und wieder ein Text in Standard-\LaTeX.

```

1 Lesebuch, VA-Font

1.1 Rationalität im Wandel der Zeit

Aber worin besteht diese wohl schon so oft diskutierte Rationalität der Vorsokratiker?

1. Nach der Pause, wenn's geht, wird die Besprechung mit den Gästen im Seminarraum stattfinden

Vorwort Das Buch beginnt mit einem Vorwort, in welchem die Zielsetzung und der wesentliche Inhalt

Und jetzt schalten wir wieder auf Normalschrift um.

2 Normalfont

Und wieder ein Text in Standard-L^AT_EX.

Anmerkungen

Wenn man die Sütterlinschrift verwendet, sollte man das Paket `babel` mit der Option `german` laden und die alten Rechtschreib- und Trennregeln verwenden. Außer Umlauten gibt es bei dieser Schrift keine diakritischen Zeichen (beispielsweise Akzente). Fremdsprachliche Wörter und Texte wurden stets in lateinischer Schrift geschrieben.

Ich habe aber bislang auch für die lateinischen Schriften keine diakritischen Zeichen entworfen, weil man bei Schulausgangsschriften annehmen kann, dass sie von Schulanfängern im Schreibunterricht verwendet und nur deutsche Texte damit geschrieben werden.

Bei Bedarf können den Schriften aber weitere gewünschte Zeichen hinzugefügt werden. Für Fehlermeldungen und Verbesserungsvorschläge, die sich aus der praktischen Anwendung ergeben, wäre ich daher sehr dankbar.

Literatur

- [1] *Bildungsplan Grundschule 2004*, S. 52; Baden-Württemberg Ministerium für Kultus, Jugend und Sport.
- [2] *Lehrplan Grundschule Deutsch 2004, 2009*, S. 46; Sächsisches Staatsministerium für Kultus.
- [3] *Schrifterlass 1941*; www.hbf.dipf.de/cgi-shl/digibert.pl?id=BBF0833246.
- [4] M. F. Eisenlohr: *Fibel – Deutsches Kinderbüchlein*; Ludwig Auer Verlag; Donauwörth; 1998; 2. Aufl., Erstausgabe 1920.
- [5] Walter Entenmann: *Paket schulschriften*; CTAN: /fonts/schulschriften/; 2012.
- [6] Heinrich Hoffmann und Walter Sauer: *Der Struwwelpeter – (Ausgabe in Sütterlinschrift)*; Edition Tintenfaß; Neckarsteinach; 2009.
- [7] Donald E. Knuth: *The METAFONT book*; Addison-Wesley; Reading, MA; 1986.
- [8] Helmut Kopka: *TEX Band 2: Ergänzungen – mit einer Einführung in METAFONT*; Addison-Wesley; Bonn; 1995.
- [9] Ludwig Sütterlin: *Neuer Leitfaden für den Schreibunterricht*; Albrecht-Dürer-Haus; Berlin; 1926.

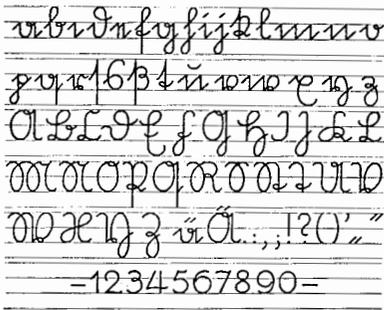


Abbildung 13: Sütterlinschrift [9].

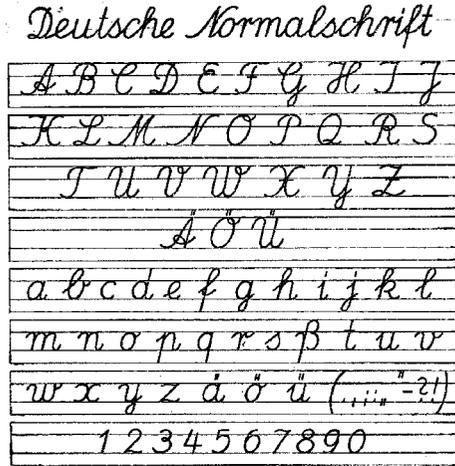


Abbildung 14: Deutsche Normalschrift [3].



Abbildung 15: Lat. Ausgangsschrift [1].



Abbildung 16: Vereinf. Ausgangsschrift [1].



Abbildung 17: Schulausgangsschrift [2].

Briefpapier mit KOMA-Script nachbauen

Markus Kohm

Seit meinem Artikel über moderne Briefe mit KOMA-Script [5] ist viel Zeit vergangen. Die Briefklasse hat seither einige Verbesserungen und Erweiterungen erfahren. Noch immer gehören Fragen zur Anpassung von `scr1ttr2` an ein bestimmtes, gewünschtes Brieflayout zu den häufigsten im Support. In diesem Artikel soll deshalb am Beispiel des Briefpapiers der *Washington State University* gezeigt werden, wie man einen solchen Briefbogen mit Hilfe von KOMA-Script und `scr1ttr2` nachbauen kann.

Vorbereitung

Zunächst einmal muss man sich für ein solches Vorhaben alle benötigten Informationen und Daten besorgen. An erster Stelle stehen dabei Layoutvorschriften. Im konkreten Fall habe ich diese [7] entnommen.

Vorgefertigte Briefbögen tragen häufig ein Logo. Je nachdem, ob \LaTeX , $\PDF\LaTeX$ oder $X\LaTeX$ zum Einsatz kommen soll, benötigt man das Logo in unterschiedlichen Formaten. Im konkreten Fall kann man sich ein Logo im EPS-Format von [9] herunterladen. Leider hat das Logo im EPS-Format, `wsuSig4cW.eps`, das man dort bekommen kann, eine Menge störenden Rand. Also habe ich das Logo zunächst mit `epstopdf` in eine PDF-Datei umgewandelt, dann mit `pdfcrop` beschnitten und schließlich mit `pdftops` wieder in eine EPS-Datei zurück gewandelt. Der gesamte Vorgang ist in der Eingabeaufforderung mit fünf Zeilen erledigt, wenn man über die notwendigen Programme `wget`, `unzip`, `epstopdf`, `pdfcrop` und `pdftops` verfügt:

```
wget -c http://identity.wsu.edu/downloads/files/signatures/  
↳PullmanSigsWindows/wsuSig4cW.zip  
unzip wsuSig4cW.zip  
epstopdf wsuSig4cW.eps  
pdfcrop wsuSig4cW.eps  
pdftops -eps wsuSig4cW-crop.pdf
```

Die Programme `epstopdf` und `pdfcrop` sind Bestandteil jeder aktuellen \TeX -Distribution. Windows-Anwender finden das Unix-Standard-Netzwerk-Werkzeug `wget` unter [2], können die Datei aber natürlich auch mit jedem Internet-Browser herunterladen. Ebenso können Sie die Datei `wsuSig4cW.eps`, statt sie mit `unzip` von [1] auszupacken, auch einfach mit dem Explorer in das Arbeitsverzeichnis kopieren.

Das Programm `pdftops` ist Bestandteil von `xpdf`, und steht auch für Windows-Anwender unter [10] zur Verfügung.

Es sei an dieser Stelle darauf hingewiesen, dass `TEXLive` bereits seit einigen Versionen für die Verwendung von `pdfTEX` nicht mehr auf das Vorhandensein von PDF- oder PNG-Dateien angewiesen ist. Dank entsprechender Konfigurationsdateien kann es in der Voreinstellung die notwendige Konvertierung bei Bedarf *on-the-fly*, also zur Laufzeit, vornehmen.

Darüber hinaus sei darauf hingewiesen, dass für die Beispiele, abweichend von obiger Anleitung zur Verwendung des Original-Logos in diesem Artikel aus rechtlichen Gründen nicht das Original-Logo verwendet wurde. Stattdessen wurde ein verfremdetes Logo erstellt und verwendet. Dadurch kommt es bei den Linien und den Farben zu minimalen Abweichungen, die jedoch als vernachlässigbar betrachtet wurden.

Umsetzung des Layouts

Zur Umsetzung seien mir einige Vorbemerkungen erlaubt. Zunächst sei darauf hingewiesen, dass alle hier verwendeten KOMA-Script-Anweisungen und die Theorie zu `LCO`-Dateien, Pseudo-Längen und Variablen sowohl im KOMA-Script-Buch [6] als auch in der KOMA-Script-Anleitung zu finden sind. Die KOMA-Script-Anleitung ist in deutscher und englischer Sprache Bestandteil der KOMA-Script-Sammlung [3]. KOMA-Script kann bei Bedarf bei jeder aktuellen `TEX`-Distribution über den jeweiligen Paketmanager nachinstalliert werden. In der Standardinstallation von `TEXLive` ist es normalerweise bereits installiert. Weder auf die Installation von KOMA-Script noch auf die erwähnten Grundlagen-Themen aus der Anleitung soll im Rahmen dieses Artikels näher eingegangen werden.

Identifikation

Die Umsetzung der Layout-Vorgaben erfolgt in einer eigenen `LCO`-Datei. Diese bekommt den passenden Namen `wsu.lco`:

```
1 \ProvidesFile{wsu.lco}[2012/06/18 v1.01
2   unsupported letter class option for Washington State University]
```

Mit dieser Anweisung erklären wir `LATEX`, worum es sich dabei handelt. Die entsprechenden Informationen sind beispielsweise über die Anweisung `\listfiles` für eine Übersichtstabelle am Ende der `log`-Datei verfügbar. Damit bei Verwendung der `LCO`-Datei mit der falschen Papiergröße eine Warnung ausgegeben wird, sollte die geplante Größe zusammen mit dem `LCO`-Namen vermerkt werden:

```
3 \LetterOptionNeedsPapersize{wsu}{letter}
```

Im Falle einer amerikanischen Universität gehe ich davon aus, dass das letter-Format gewünscht wird. Außerdem werden gleich zu Anfang noch zwei Befehle für Warnungen und Fehlermeldungen definiert.

```

4 \providecommand*\LCOWarning[2]{%
5   \GenericWarning{%
6     (#1)\@spaces\@spaces\@spaces\@spaces\@spaces\@spaces
7   }{%
8     Letter class option #1 Warning: #2%
9   }%
10 }
11 \providecommand*\LCOError[3]{%
12   \GenericError{%
13     (#1)\@spaces\@spaces\@spaces\@spaces\@spaces\@spaces\@spaces
14   }{%
15     Letter class option #1 Error: #2%
16   }{%
17     See the #1 package documentation for explanation.%
18   }{#3}%
19 }

```

Die beiden Befehle arbeiten wie `\PackageWarning` und `\PackageError`, die in [8] dokumentiert sind. Für die Definition wurde `\providecommand` verwendet für den Fall dass die Anweisungen bereits in `scr1ttr2` oder einer anderen `LCO`-Datei definiert werden.

Seitenspiegel

Bereits unabhängig vom Briefbogen selbst, muss für alle Seiten ein Seitenspiegel, bestehend aus Satzspiegel und Stegen festgelegt werden. Kopf- und Fußzeilen sind je nach ihrer Wirkung als Teil des Satzspiegels oder der Stege zu betrachten. \LaTeX selbst unterscheidet vertikal zwischen oberem Rand, Kopfzeilen, Abstand zum Textbereich, Textbereich und Abstand zur Grundlinie der Fußzeilen (siehe Abbildung 1). Der untere Rand ergibt sich implizit. Eine definierte Höhe für den Fuß existiert im Gegensatz zur definierten Höhe des Kopfes nicht. Horizontal wird zwischen dem linken Rand von ungeraden Seiten, dem linken Rand von geraden Seiten und dem Textbereich unterschieden. Der rechte Rand sowohl von ungeraden als auch von geraden Seiten ergibt sich implizit. Bei einseitigen Dokumenten wird nur der linke Rand von ungeraden Seiten verwendet – auch für gerade Seiten.

Die Vorschriften für die Seitenränder und den Textbereich sind in [7] wie folgt angegeben:

All text in one-and-onehalf inches from the left [...] Begin the letter two inches from the top [...] The maximum line length should not exceed six inches [...] If additional pages are needed, the left-hand margin of the letter should be maintained, but the text may be started one inch from the top.

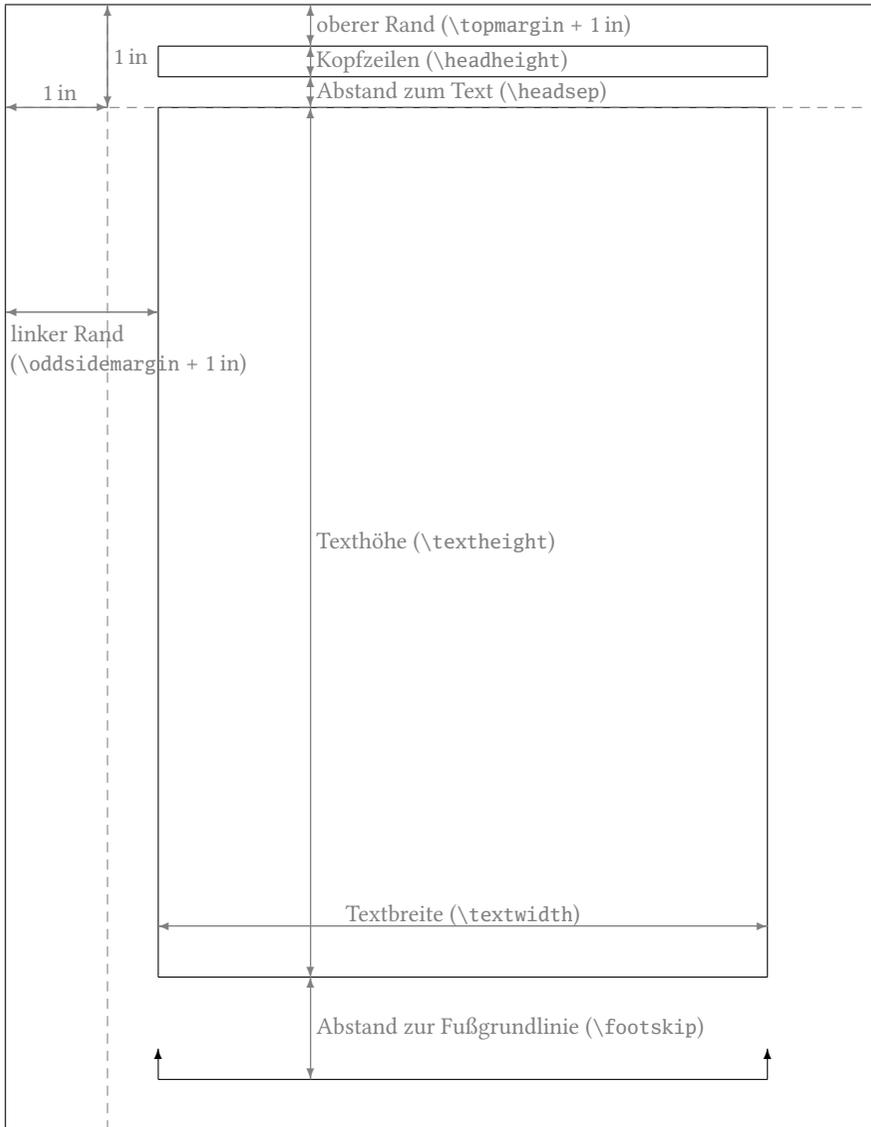


Abbildung 1: Seitenspiegel einer ungeraden Seite bei LATEX

Die zugehörige Abbildung gibt außerdem einen rechten Rand von fünf achtel Zoll und einen unteren Rand von einem halben Zoll an.

Diese Angaben können verwendet werden, um Ränder und Satzspiegel einzustellen. Dabei gehe ich aufgrund der Vorschrift, dass auf Extraseiten der Text abweichend vom Briefbogen bereits ein Zoll unter der Oberkante beginnen soll, davon aus, dass auf diesen weiteren Seiten kein Logo mehr verwendet werden soll. Trotzdem schaffe ich Platz, um das Logo notfalls unter zu bringen, jedoch deutlich höher als auf dem Briefbogen:

```

20 \topmargin=-.75in% plus the lin from the default offset
21 \headheight=.5in
22 \headsep=.25in
23 \oddsidemargin=.5in% plus the lin from the default offset
24 \textwidth=\dimexpr\paperwidth-1.5in-.625in\relax
25 \ifdim\textwidth>6in
26   \textwidth=6in
27 \fi
28 \evensidemargin=-.375in% plus the lin from default offset
29 \textheight=\dimexpr\paperheight-1in-1.5in\relax
30 \footskip=1in
31 \topskip=8.4bp% 0.7 * 12bp (should be \ht\strutbox)

```

Die Einstellungen werden hier direkt vorgenommen. Damit ist die LCO nicht vom Paket `geometry` abhängig. Gleichzeitig kann so die Textbreite einfach zunächst berechnet werden und dann für den Fall, dass die berechnete Breite größer als die angegebene Grenze von sechs Zoll ist, diese als neuer Wert gesetzt werden. Zu beachten ist allerdings, dass von den verlangten Werten für den oberen und den linken Rand jeweils ein Zoll abgezogen werden muss. Diese Verschiebung des Ursprungs, die in Abbildung 1 durch gestrichelte Linien verdeutlicht wird, hat historische Gründe. Es sei an dieser Stelle auch darauf hingewiesen, dass `\evensidemargin` nicht der rechte Rand ist, sondern der linke Rand von geraden Seiten im doppelseitigen Satz.

Falls Sie aber lieber mit `geometry` arbeiten möchten, so wäre auch das möglich. Allerdings sollte man in einer LCO-Datei nicht einfach Pakete laden, da es dafür bereits zu spät sein könnte. Wir werden auf dieses Problem im weiteren Verlauf noch zurück kommen.

Die Werte für `\textheight` und `\footskip` sind geschätzt. Für den Briefbogen sind diese Werte kaum von Bedeutung und für die weiteren Seiten fehlen konkrete Angaben in den Vorschriften. Im Endeffekt resultieren die angegebenen Werte in einem Abstand von eineinhalb Zoll zwischen dem Ende des Textbereichs und dem Papierende und einem Rand von einem halben Zoll unterhalb der Fußzeilen. Dies erscheint mir ein tauglicher Kompromiss zwischen typographisch optimierten Werten und dem häufigen Wunsch, möglichst viele Zeilen auf eine Seite zu bekommen. Bei

einer Vergrößerung von `\textheight` sollte `\footskip` entsprechend verkleinert werden und umgekehrt, damit der Rand unter den Fußzeilen gleich bleibt.

Der Wert für `\topskip` definiert sozusagen, wie hoch die oberste Zeile des Textbereichs über der Grundlinie maximal sein darf, damit die Grundlinie nicht nach unten verschoben werden muss. Der angegebene Wert sollte für einen Grundlinienabstand von 12 bp genügen.

Üblicherweise sollte `\textheight` eigentlich ein ganzzahliges Vielfaches von `\baselineskip` zuzüglich `\topskip` sein. Dadurch wird verhindert, dass der vertikale Seitenausgleich der Einstellung `\flushbottom` die Absätze auseinander ziehen muss. Dieser ist bei doppelseitigen Dokumenten normalerweise obligatorisch. Da Briefe selten als doppelseitiges Werk gebunden werden, kann man aber selbst bei doppelseitigen Briefen in der Regel getrost mit `\raggedbottom` arbeiten, also den vertikalen Ausgleich deaktivieren. Daher wurde hier diesbezüglich kein Aufwand getrieben. Da man eine LCO-Datei auch noch nach `\begin{document}` laden kann, müssen diese Seitenspiegeleinstellungen gegebenenfalls noch zu internen \LaTeX -Längen verrechnet werden. Dazu dient bei KOMA-Script die Anweisung:

```
32 \activateareas
```

Schrift

Bezüglich der Schrift ist in [7] lapidar vermerkt:

The format shown here is typeset in ITC Stone Serif, 9.5 on 12, the commended typeface for University correspondence.

Ich interpretiere diese Angabe so, dass die Schrift ITC Stone Serif in der Größe 9,5 Punkt, mit einem Grundlinienabstand von 12 Punkt verwendet werden soll. Dazu ist zu erwähnen, dass ITC Stone Serif eine kommerzielle Schrift ist, die sich inzwischen einiger Beliebtheit erfreut. Meines Wissens verwendet beispielsweise auch die TU Clausthal diese Schrift.

Die gut ausgebaute OpenType-Schriftfamilie ist für einen durchaus angemessenen Preis zu bekommen. Mitarbeiter der Washington State University können eine Lizenz über einen internen Bestellvorgang erwerben. Dennoch wollte ich die LCO-Datei nicht zwingend auf eine bestimmte, kommerzielle Schrift festlegen. Deshalb wird eine Option definiert, die es ermöglicht, die Schriftvorgabe zu ignorieren:

```
33 \DefineFamily{wsu}
34 \DefineFamilyMember[.wsu.lco]{wsu}
35 \FamilyBoolKey[.wsu.lco]{wsu}{nostone}{@nostone}
36 \FamilyProcessOptions[.wsu.lco]{wsu}
```

Zunächst wird mit Hilfe der Anweisung `\DefineFamily` aus dem KOMA-Script-Paket `scrbase` eine neue Familie für Schlüssel definiert. Die Verwendung des Paketes

scrbase bietet sich für die Definition der Option an, da es von allen KOMA-Script-Klassen ohnehin geladen wird.

Dann wird mit `\DefineFamilyMember` in der Familie ein neues Mitglied geschaffen. Mit Hilfe von `\FamilyBoolKey` wird die neue Option `nostone` definiert. Deren Einstellung kann später über die Verzweigung `\if@nostone` abgerufen werden.

Da dies vorerst die einzige Option ist, werden zum Schluss noch die Optionen des neuen Familienmitglieds abgearbeitet. Die Anweisung `\LoadLetterOption`, mit der LCO-Dateien geladen werden, kennt keine Optionenübergabe. Deshalb bleibt als einzige Möglichkeit, diese Option bei `\documentclass` anzugeben. Das kann allerdings dazu führen, dass die Option als »*unused global option*«, also als nicht verwendete globale Option gemeldet wird, wenn `wsu.lco` erst nach `\begin{document}` geladen wird. Ich betrachte das lediglich als verzeihlichen Schönheitsfehler.

Falls die Option nicht gesetzt wurde, wenn also bei `\if@nostone` lediglich der `\else`-Zweig gültig wäre, soll auch gleich die Schrift ITC Stone Serif als Standardschrift eingestellt werden:

```

37 \if@nostone\else
38   \if@atdocument
39     \scr@ifundefinedorrelax{setmainfont}{%
40       \LCOWarning{wsu}{%
41         You've not loaded package 'fontspec', \MessageBreak
42         and now it's to late to do so.\MessageBreak
43         Because of this option 'nostone' has\MessageBreak
44         been activated%
45       }%
46       \Family@Options{wsu}{nostone}{}%
47     }%
48   \else
49     \RequirePackage{fontspec}%
50     \KOMAOptions{fontsize=10bp}%
51     \setmainfont[Scale=0.95,
52 %       BoldFont=(* Semibold),% change this to use Bold instead of
↪ Semibold
53 %       ItalicFont=(* Italic),
54 %       BoldItalicFont=(* Semibold Italic),% change this to use Bold
↪ instead of
55 %                                     % Semibold
56 %       SlantedFont=(* Italic),% we use italic instead of slanted
57 %       BoldSlantedFont=(* Semibold Italic)
58     ]{ITC Stone Serif LT}% gives us 9.5bp on 12bp
59     \selectfont
60   \fi
61 \fi

```

Da die Einbindung der Schrift mit PDF \LaTeX recht aufwändig sein kann, habe ich mich für die ausschließliche Verwendung von X \LaTeX oder Lua \LaTeX entschieden.

Hier kann die systemweit installierte Schrift über das Paket `fontspec` geladen werden.

Leider ergibt sich dadurch bereits wieder ein Problem: Pakete können nur in der Dokumentpräambel geladen werden. `LCO`-Dateien können jedoch auch noch zu einem späteren Zeitpunkt geladen werden. Deshalb wird mit der Verzweigung `\if@atdocument`, die ebenfalls von `scrbase` bereitgestellt wird, zunächst getestet, ob wir uns bereits nach `\begin{document}` befinden. Falls das der Fall ist und der normalerweise von `fontspec` bereitgestellte Befehl `\setmainfont` nicht zur Verfügung steht, wird eine entsprechende Warnung ausgegeben.

Befinden wir uns hingegen noch in der Präambel, so wird mit `\RequirePackage` zunächst sichergestellt, dass das Paket `fontspec` geladen ist. Danach stellen wir als Schriftgröße 10 bp ein. Das steht zwar im Widerspruch zur geforderten Schriftgröße von 9,5 Punkt, der Grundlinienabstand passt dann aber grundsätzlich schon einmal. Die Einheit bp, also *big point*, wird deshalb gewählt, weil die Vorgaben für Word-Benutzer definiert sind und der dortige Punkt dieser Einheit entspricht.

Als nächstes wird die gewünschte Schrift geladen. Leider stand mir statt der OpenType-Familie ITC Stone Serif nur die TrueType-Version ITC Stone Serif LT zur Verfügung. Die Optionen zur Auswahl anderer Schnitte, die bei OpenType angegeben werden können, sind in obigem Code deshalb auskommentiert.

Beim Laden wird die Schrift per Option mit dem Faktor 0,95 skaliert, also geringfügig verkleinert. Damit wird erreicht, dass statt einer 10 bp-Schrift tatsächlich nur 9,5 bp verwendet werden. Die Vorgabe ist also eingehalten.

Absatzformatierung

Bezüglich der Absatzformatierung ist unter [7] zum einen die Angabe:

Paragraphs are indicated by one line space, with no indentation.

zu finden. Darüber hinaus zeigt die Abbildung, dass mit linksbündigem Flattersatz ohne Trennung gearbeitet werden soll. Beides ist rasch eingestellt:

```
62 \KOMAoption{parskip}{full}
63 \raggedright
64 \let\raggedsignature\raggedright
```

Hier wurde die Variante `full` für Option `parskip` gewählt. Weitere Varianten mit unterschiedlichen Auswirkungen auf die letzte Zeile eines Absatzes stehen bei KOMA-Script zur Verfügung und können in der Anleitung nachgelesen werden.

Bei der Gelegenheit wurde nicht nur der Text selbst linksbündig gesetzt. Die letzte Zeile in obigem Code sorgt außerdem dafür, dass auch Schlussgruß und Signatur zueinander linksbündig angeordnet werden. Mir erscheint das bei einem Brief im Flattersatz die einzig, sinnvolle Form.

Logo und Farbe

Wie das Logo erzeugt wird, wurde bereits in den Vorbereitungen erklärt. Auf dem Briefbogen soll außerdem eine farbige Linie zum Einsatz kommen. In der EPS-Datei meines mit Inkscape abgewandelten Logos sind für die Farbwerte des Logos folgende Zeilen zu finden:

```
/rg { setrgbcolor } bind def
0.803922 0.462745 0.541176 rg
```

Die erste dieser beiden Zeilen definiert eine PostScript-Anweisung zur Auswahl einer Farbe im RGB-Modell. In der zweiten Zeile wird diese dann verwendet, um eine konkrete Farbe auszuwählen. Dabei ist die Abfolge der Werte wie gewohnt Rot, Grün, Blau und die Werte sind im Bereich zwischen Null und Eins angegeben. Das passt also zur Definition mit der Anweisung `\definecolor` aus einem der Pakete `color` oder `xcolor`. Wird das Original-Logo verwendet, könnte die Farbe leicht abweichen und muss ggf. mit einem geeigneten Werkzeug ermittelt werden.

Wir benötigen also eines der Pakete `color` oder `xcolor` und zusätzlich das Paket `graphicx`. Hier ergibt sich dasselbe Problem wie bereits für im Abschnitt über die Schrift für `fontspec` erklärt.

```
65 \if@atdocument
66   \scr@ifundefinedorrelax{ver@graphicx.sty}{%
67     \LCOError{wsu}{%
68       package 'graphicx' missing%
69     }}%
70     This LCO needs package 'graphicx', but it's to late to load it
71     myself.\MessageBreak
72     You either should load this LCO before
73     \string\begin{document},\MessageBreak
74     or load package 'graphicx' yourself.%
75   }%
76 }{%
77 \scr@ifundefinedorrelax{definecolor}{%
78   \LCOError{wsu}{%
79     command '\string\definecolor' missing%
80   }}%
81   This LCO needs command \string\definecolor, but it's to late to
82   load a package myself.\MessageBreak
83   You either should load this LCO before
84   \string\begin{document},\MessageBreak
85   or load package 'xcolor' or 'color' yourself.%
86 }%
87 }%
88 \else
89   \RequirePackage{graphicx}%
90   \IfFileExists{xcolor.sty}{\RequirePackage{xcolor}}{\RequirePackage{
91     ↪color}}%
92 \fi
```

Hauptunterschied ist hier, dass Farbe und Logo unverzichtbar sind und deshalb ggf. nicht nur gewarnt, sondern ein Fehler gemeldet wird.

Sobald sichergestellt ist, dass die Pakete geladen sind, können Farbe und Logo definiert werden:

```

92 \definecolor{wsuSig4cW}{rgb}{0.67451,0.682353,0.686275}
93 \setkomavar{fromlogo}{%
94   \includegraphics[height=.5in]{wsuSig4cW-crop}%
95 }

```

Die gewünschte Größe des Logos wurde dabei über den Daumen gepeilt.

Definition des Briefbogens

Zunächst fällt auf, dass weder Faltmarken noch Rücksendeadresse verwendet werden:

```

96 \KOMAoptions{foldmarks=false,backaddress=False}

```

Beide sind per Option rasch abgeschaltet.

Als nächstes ist in [7] für den Briefkopf angegeben:

All text in one-and-onehalf inches from the left, flush left with the crimson line in the Washington State University signature. Begin the letter two inches from the top, with the date followed by two line spaces, then the address, another line space, then the salutation.

Das sind eine Menge Informationen.

Beginnen wir mit dem Briefkopf:

```

97 \@setlength{firstheadvpos}{1in}
98 \@setlength{firstheadhpos}{\dimexpr 2in-1.03in\relax}
99 \@setlength{firstheadwidth}{%
100   \dimexpr\paperwidth-\useplength{firstheadhpos}-.625in\relax
101 }

```

Die vertikale Position des Logos lässt sich aus der Abbildung auf [7] relativ leicht ermitteln. Der obere Rand, über dem Logo wird von mir auf ein Zoll geschätzt. Die Höhe des Logos selbst und der Abstand zwischen dem Logo und dem Textbereich scheinen ungefähr gleich zu sein. Für die horizontale Position habe ich jedoch zum Lineal gegriffen und ermittelt, dass die horizontale Linie, die in der Erklärung als *crimson line* angegeben ist, 1,03 Zoll rechts von der linken Kante des Logos beginnt. Entsprechend wurde `firstheadhpos` berechnet. Der Kopf selbst soll sich gemäß der Bemaßung derselben Abbildung bis fünf achtel Zoll vor der rechten Papierkante erstrecken. Auch dies ist einfach zu berechnen.

Als nächstes folgt in der Erklärung die Position des Datums:

```

102 \KOMAoption{refline}{dateleft}
103 \@setlength{refvpos}{2in}
104 \@setlength{refaftervskip}{8\baselineskip}
105 \@setlength{refhpos}{\dimexpr\oddsidemargin+1in\relax}

```

Das Datum wird bei `scr1ttr2` automatisch in der Geschäftszeile gesetzt. Über die Option `refline` kann dabei gewählt werden, dass das Datum links gesetzt werden soll. Der Abstand zwischen Geschäftszeile und Anrede ergibt sich aus zwei Zeilen Abstand zur Anschrift, fünf Zeilen Anschrift in der Beispielabbildung und einer weiteren Zeile Abstand zur Anrede.

Damit ist aber auch fest definiert, dass die Anschrift immer mindestens fünf Zeilen zu umfassen hat:

```

106 \@setlength{toaddrhpos}{\uselength{refhpos}}
107 \@setlength{toaddrvpos}{\dimexpr\uselength{refvpos}+2\baselineskip\relax}
108 \@setlength{toaddrheight}{5\baselineskip}
109 \@setlength{backaddrheight}{0pt}

```

Die Höhe der Rücksendeadresse begründet sich daraus, dass eine solche offenbar nicht verwendet werden soll. Alle anderen Werte ergeben sich ebenfalls aus den vorgenannten Angaben und Überlegungen.

Definieren wir nun den Inhalt des Briefkopfs:

```

110 \setkomavar{firsthead}{%
111   \color{wsuSig4cW}\footnotesize
112   \makebox[0pt][l]{%
113     \makebox[\uselength{firstheadwidth}][r]{%
114       \raisebox{\baselineskip}{%
115         \rule{\dimexpr\paperwidth-1.5in-.625in\relax}{.5bp}%
116       }%
117     }%
118   }%
119   \raisebox{\dimexpr \baselineskip-.31in+.5bp\relax}{%
120     \usekomavar{fromlogo}
121   }%
122   \hfill
123   \begin{tabular}[t]{@{}r@{}}
124     \usekomavar{office}
125   \end{tabular}
126   \par
127 }

```

Zunächst wird die Farbe für die Linie und den Text rechts unter der Linie eingestellt. Ich verwende hier für beides dieselbe Farbe. Man könnte aber später für den Text rechts unter der Linie auch noch einmal die Farbe wechseln. Die Schriftgröße für den Text erscheint mir mit `\footnotesize` passend.

Dann geht es darum eben diese Linie zu setzen. Breite und Höhe der Linie sind aus der Abbildung zu bestimmen. Das sind die Argumente von `\rule`. Die Linie soll über dem rechtsbündig zu setzenden Text erscheinen. Daher wird sie mit `\raisebox` entsprechend nach oben verschoben. Sie soll außerdem rechtsbündig innerhalb der Breite des Kopfes gesetzt werden. Dazu dient die innere `\makebox`-Anweisung. Anschließend wollen wir das Logo über dieser Linie platzieren. Das Setzen der Linie darf also die horizontale Position nicht ändern. Dazu verwenden wir die äußere `\makebox`-Anweisung, bei der als Breite 0 pt angegeben wurde, wobei innerhalb dieser Breite linksbündig gesetzt werden soll.

Als nächstes wird das Logo ausgegeben. Auch dieses muss nach oben verschoben werden und zwar so weit, dass die Linie innerhalb des Logos genau deckungsgleich mit der soeben gezeichneten Linie ist. Bei der Berechnung der Verschiebung wurde die Verschiebung `\baselineskip` der Linie und die Dicke 0,5 bp der Linie übernommen. Die restliche Verschiebung um 0,31 in wurden im Logo ausgemessen.

Als letztes wird rechtsbündig der Text gesetzt. Damit der Text austauschbar ist, wurde hierfür die Variable `office` verwendet. Damit diese auch mehrzeilig sein darf, wird sie in einer Tabelle gesetzt, deren oberste Zeile mit der Grundlinie ausgerichtet ist. Mehrzeilige Textangaben werden dabei rechtsbündig gesetzt. Man könnte sie auch innerhalb des Textblocks linksbündig oder zentriert setzen, indem man die Spaltendefinition `@{}l@{}` oder `@{}c@{}` verwendet.

Die Variable, die in der Kopfdefinition verwendet wurde, ist keine von KOMA-Script vordefinierte Variable:

```
128 \newkomavar{office}
```

Die neue Variable wurde hier ohne druckbaren Bezeichner definiert, da ein solcher auch nicht verwendet wird.

Bevor wir zum Fuß kommen, beachten wir noch die Angabe

After the complimentary close, allow five line spaces for the signature.

bezüglich der Signatur:

```
129 \@setlength{sigbeforevskip}{5\baselineskip}
```

Hier, wie bei allen anderen Werten, die zuvor bereits von `\baselineskip` abhängen, wird davon ausgegangen, dass die Grundschriftgröße nach dem Laden der LCO-Datei nicht mehr geändert wird.

Es geht weiter mit dem Fuß des Briefbogens:

```
130 \@setlength{firstfootvpos}{\dimexpr\paperheight-.5in-\dp\strutbox\relax}
    ↵
131 \@setlength{firstfoothpos}{\uselength{refhpos}}
132 \@setlength{firstfootwidth}{\textwidth}
```

Die Platzierung ergibt sich allein aus der Abbildung von [7]. Nähere textuelle Angaben dazu finden sich nicht.

Der Inhalt besteht aus Adresse und weiteren Informationen zum Absender:

```

133 \setkomavar{firstfoot}{%
134 \parbox[b][Opt][b]{\textwidth}{\raggedright\footnotesize\color{wsuSig4
↪cW}%
135 \ifkomavareempty{fromaddress}{}{%
136 {\renewcommand*{\}[1][]{, \ignorespaces}\usekomavar{fromaddress
↪}}\
137 }%
138 \newcommand*{\elementsep}{}%
139 \newcommand*{\xelementsep}{\ \textbullet\ }%
140 \ifkomavareempty{fromphone}{}{%
141 \elementsep
142 \usekomavar{fromphone}%
143 \let\elementsep\xelementsep}%
144 \ifkomavareempty{fromfax}{}{%
145 \elementsep
146 \usekomavar*{fromfax}: \usekomavar{fromfax}%
147 \let\elementsep\xelementsep}%
148 \ifkomavareempty{fromemail}{}{%
149 \elementsep
150 \usekomavar{fromemail}%
151 \let\elementsep\xelementsep}%
152 \ifkomavareempty{fromurl}{}{%
153 \elementsep
154 \usekomavar{fromurl}}}%
155 }%
156 }

```

Die einzelnen Angaben sind in Standard-Variablen von `scrLtr2` abgelegt. Sie werden nur dann ausgegeben, wenn sie nicht leer sind. Dies wird mit `\ifkomavareempty` erreicht. Trickreich ist dabei die Entscheidung, ob ein Trennzeichen vor einem Feld auszugeben ist. Dies wird dadurch gelöst, dass eine zunächst leer definierte Anweisung zu dem Trennzeichen undefiniert wird, wenn ein Feld ausgegeben wurde.

Damit ist die Form des Briefbogens aber auch das Layout der Folgeseiten vollständig definiert.

Personalisierung

Bevor der erste Brief mit dem neuen, nachgebauten Layout erstellt wird, sollte der Absender noch eine weitere *LCO*-Datei mit seinen persönlichen Daten erstellen.

```

1 \LoadLetterOption{wsu}
2
3 \setkomavar{date}{\today}
4 \setkomavar{signature}{John Doe\

```

```

5 Vice President}
6 \setkomavar{office}{Office of University Publications and Printing}
7 \setkomavar{fromaddress}{PO Box 123456\ Pullman, WA 12345-6789}
8 \setkomavar{fromphone}{555-335-3518}
9 \setkomavar{fromfax}{555-335-8568}
10 \setkomavar{fromemail}{publications@wsu.invalid}
11 \setkomavar{fromurl}{www.wsu-pub.invalid/publications}

```

Abwandlung: Verwendung von geometry

Wie bereits erwähnt, könnte man für die Einstellung des Seitenspiegels in `wsu.lco` auch das Paket `geometry` verwenden. Der Code dazu könnte beispielsweise so aussehen:

```

\if@atdocument
  \scr@ifundefinedorrelax{newgeometry}{%
    \LCOError{wsu}{%
      command '\string\newgeometry' missing%
    }{%
      This LCO needs package 'geometry', but it's too late to load it
      myself.\MessageBreak
      You either should load this LCO before
      \string\begin{document},\MessageBreak
      or load package 'geometry' yourself.%
    }%
  }%
  \newgeometry{left=1.5in,right=.625in,top=1in,includefoot,bottom=.5in}%
  \ifdim \textwidth>6in
    \newgeometry{left=1.5in,width=6in,top=1in,includefoot,bottom=.5in}%
  \fi
}%
\else
  \RequirePackage{geometry}%
  \geometry{left=1.5in,right=.625in,top=1in,includefoot,bottom=.5in}%
  \AtBeginDocument{%
    \ifdim \textwidth>6in
      \newgeometry{left=1.5in,width=6in,top=1in,includefoot,bottom=.5in}%
    \fi
  }%
\fi

```

Nachteilig dabei ist, dass für doppelseitige Dokumente nur entweder der linke Rand von geraden Seiten auf fünf achtel Zoll festgelegt werden kann oder die Breite des Textes auf maximal sechs Zoll. Eine Kombination beider Möglichkeiten wäre nur mit Tricks möglich. Entweder müsste man Vorabberechnungen oder nachträgliche Vergleiche anstellen und gegebenenfalls `geometry` sogar mehrfach die Ränder einstellen lassen. Der Aufwand dafür wäre kaum geringer als die direkte Festlegung der Ränder und des Textbereichs, wie sie bereits gezeigt wurde. Mir

erscheint daher im konkreten Fall die zuvor gewählte Ausnahme von der Regel, dass für konkrete Randangaben `geometry` verwendet werden sollte, angebracht.

Zusammenfassung

In diesem Artikel wurde gezeigt, wie man mit den Mitteln der `LCO`-Datei ein gegebenes Brieflayout nachbilden kann. Insbesondere wurde der Umgang mit Pseudolängen und Variablen vorgeführt, wie man neue Optionen in `LCO`-Dateien definieren kann, und wie das Nachladen von Paketen zu handhaben ist. Beispielfhaft wurde auch die Verbesserung von extern verfügbaren, suboptimalen Grafikdateien erwähnt.

Abschließend sei in Abbildung 2 ein Beispielbrief gezeigt. Dieser wurde zur Verdeutlichung der vorgegebenen Ränder mit einigen Hilfslinien versehen und bemaßt. `LCO`-Dateien, die im Wesentlichen den in diesem Artikel vorgestellten entsprechen, sowie ein Beispielbrief sind auf der Projektseite von KOMA-Script im Beitrag [4] zu finden.

Literatur

- [1] *UnZip for Windows*; <http://gnuwin32.sourceforge.net/packages/unzip.htm>.
- [2] *Wget for Windows*; <http://gnuwin32.sourceforge.net/packages/wget.htm>.
- [3] Markus Kohm: *KOMA-Script: A bundle of versatile classes and packages*; <http://www.ctan.org/pkg/koma-script>.
- [4] Markus Kohm: *Reproducing the official header of the Washington State University*; <http://www.komascript.de/node/1599>.
- [5] Markus Kohm: *Moderne Briefe mit KOMA-Script; Die T_EXnische Komödie*; 2/03, S. 32–51; 2003; DANTE e. V.
- [6] Markus Kohm und Jens-Uwe Morawski: *KOMA-Script*; Lehmanns Media; Berlin; 4. Aufl.; 2012.
- [7] Ali Mehrizi: *Reproducing an official letterhead*; <http://tex.stackexchange.com/questions/59932/reproducing-an-official-letterhead>.
- [8] L^AT_EX3 Project Team: *L^AT_EX 2_ε for class and package writers*; Febr. 2006; <http://www.ctan.org/pkg/clsguide>.
- [9] *Pullman Academic Signatures*; <http://identity.wsu.edu/downloads/pullman/academic-signatures.aspx>.
- [10] *xpdf Download*; <http://www.foolabs.com/xpdf/download.html>.

		2" Office of University Publications and Printing Vice President
	Date Adresse Name Company Name Department or PO Box Number Street Address City, State 00000-0000	
1.5"	Dear Addressee,	.625"
	<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. </p> <p> Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. </p> <p>Complimentary close</p> <p>Sender's Name Title</p> <p> <small> PO Box 123456, Pullman, WA 12345-6789 555-335-3518 • Fax: . 555-335-8568 • publications@wsu.invalid • www.wsu-pub.invalid/publications </small> </p>	
	.5"	

Abbildung 2: Beispielbrief mit Bemaßung der Ränder

Das Erstellen eines Glossars

Christine Römer

Für das Erstellen eines Glossars bietet \LaTeX nur geringe Unterstützung. Erweiterungen in Dokumentklassen, zusätzliche Tools und Pakete stellen jedoch zahlreiche Möglichkeiten bereit. Sie unterscheiden sich hinsichtlich ihrer Variationsmöglichkeit und Komplexität. Je mehr Möglichkeiten zur Variation bestehen, um so schwieriger ist ihre Handhabung bzw. um so mehr Einarbeitungszeit ist für ihre Benutzung erforderlich.

Einführung

Glossare sind alphabetisch geordnete, registerförmige Listen von Erläuterungen für Fachausdrücke, die in wissenschaftlichen Texten mit Lehrcharakter verwendet werden. Sie werden in der Regel am Dokumentende im Anhang in Form von meist kurzen und treffenden Erklärungen bzw. Definitionen, öfters zweispaltig ausgegeben. Ihre Funktion besteht darin, den Leser/innen eine Verständnishilfe bei der Textrezeption zu geben. Das zu erklärende Wort wird meist fett gedruckt und der nachfolgende Erläuterungstext etwas eingerückt. Da oft noch ein Wortindex beigefügt wird, werden zumindest in den Geisteswissenschaften in die Glossareinträge keine Verweise auf Seiten, wo das Wort im Text vorkommt, eingefügt. [1]

Für das Erstellen eines Glossars bietet Standard- \LaTeX nur minimale Unterstützung. Es hält jedoch den Befehl `\glossary` bereit. Mit `\makeglossary` wird eine `.glo`-Datei erzeugt, die analog zu `\makeindex` Glossareinträge enthält. Dabei gibt es mit Sonderzeichen ähnliche Probleme wie bei der Indexerzeugung. Mit `\printglossary` kann das Glossar ausgedruckt werden. Die Einträge werden in eine Umgebung `theglossary` eingefügt.

Die Universaltextklasse `memoir` hat einige Modifizierungen und Erweiterungen zur Glossarerzeugung vorgenommen (siehe [5, 17.3]).

Es stehen für die Glossarerzeugung Zusatzstildateien (wie `glossar`) und spezielle Pakete wie `glosstex` (1997 von Volkan Yavuz), `nomencl` (2005 von Boris Veytsman und Bernd Schandl) oder `glossaries` (2012 von Nicola L.C. Talbot) zur Verfügung. Sie unterscheiden sich hinsichtlich ihrer Komplexität und sollen nachfolgend einführungsvorgestellt werden. Dabei wird nur auf ihre Glossarerzeugungsfunktion eingegangen.

glosstex

Dieses Tool ermöglicht neben dem Erstellen von Glossaren auch das Anfertigen von Abkürzungslisten und das generelle Listensortieren. Besonders erwähnenswert ist, dass man die Worterklärungen in einer externen Datenbank (<Datei>.gdf) ähnlich wie bei bib(1a)tex speichern kann. Über die Markierung der zu erklärenden Wörter im Text mit `\gls{<...>}` werden dann beim Kompilieren die ausgewählten Worterklärungen im Glossar erscheinen. Das sehr leicht zu handhabende Tool ist nicht Teil des aktuellen \TeX Live und muss deshalb zusätzlich installiert werden. [6]

Die Kompilierung erfolgt in Zusammenarbeit mit `makeindex` für das Sortieren der Eintrágeliste mittels folgender Befehle:

```
pdflatex <Dateiname>
makeindex <Dateiname>
glosstex <Dateiname> <Dateiname>
makeindex <Dateiname> <Dateiname>.gxs -o <Dateiname>.glx -s glosstex.ist
pdflatex <Dateiname>
```

Es folgt ein Beispiel, dessen Ausgabe Abbildung 1 zeigt.

```
1 %% Glossar1.tex
2 \documentclass[a4paper,12pt,ngerman]{scrreprt}
3 \usepackage[T1]{fontenc}
4 \usepackage[utf8]{inputenc}
5 \usepackage{babel}
6 \usepackage{makeidx}
7 \makeindex
8 \usepackage{glosstex}
9 \makeglossary
10
11 \begin{document}
12
13 Ein \gls{Lemma} ist ein spezielles \gls{Stichwort}.
14
15 \clearpage
16 \def\glossaryname{Glossar}
17 \printglosstex(glo)
18
19 \end{document}
```

```
1 %% Glossar1.gdf
2 @entry{Lemma} einleitendes Kennwort, Stichwort eines Wörterbuchartikels.
3
4 @entry{Stichwort} siehe Lemma.
```

nomencl

Dieses Paket, das über einen längeren Zeitraum hinweg kontinuierlich verbessert wurde, bildete eine Basis von `glosstex`. Allerdings ist die Entwicklung von

Glossar

Lemma einleitendes Kennwort, Stichwort eines Wörterbuchartikels.

Stichwort siehe Lemma.

Abbildung 1: glosstex-Beispiel

nomenc1 mittlerweile eingestellt worden. Das Paket ermöglicht neben Nomenklaturen und Indexen auch das Setzen von Glossaren in einem Dokument. Es nutzt ebenfalls makeindex und nimmt automatisch eine alphabetische Ordnung vor. Eine Kompatibilität mit zahlreichen Sprachen besteht und ohne eine große Einführung kann damit ein Glossar erzeugt werden. Der Terminus, der im Glossar erscheinen soll, wird im Text, am besten nach dem Stichwort, mit folgendem Befehl `\nomenclature[<Präfix>]{<Symbol>}{<Beschreibung>}` eingefügt. Dem Paket können zusätzlich verschiedene Optionen übergeben werden. Dies sind beispielsweise `refpage` (Angabe der Seitenzahl) oder `intoc` (Aufnahme ins Inhaltsverzeichnis); für weitere Optionen siehe [4]. In der Symbolposition von `\nomenclature` kann der Glossareintrag formatiert werden. Ein Beispiel sehen Sie nachfolgend:

```

1 \documentclass[a4paper,12pt,ngerman]{scrreprt}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage{babel}
5 \usepackage[german]{nomencl}
6 \renewcommand{\nomname}{Glossar}
7 \makenomenclature
8
9 \begin{document}
10 Eine Abkürzung\nomenclature{\textbf{Abkürzung:}}{Graphische Kürzung
11 eines Wortes, die in der mündlichen Sprache nicht beibehalten wird (wie
12 \textit{u.\,a.}).} ist kein Kurzwort\nomenclature{\textbf{Kurzwort:}}{%
13 Kürzung eines Wortes, ist gleichbedeutend mit der Vollform (wie
14 \textit{Akku} vs. \textit{Akkumulator}).}
15
16 \printnomenclature
17 \end{document}

```

Die Kompilierungsbefehle sind folgende:

```

pdflatex <Dateiname>
makeindex <Dateiname>.nlo -s nomencl.ist -o <Dateiname>.nls
pdflatex <Dateiname>

```

Der erste pdf_latex-Lauf erzeugt eine Datei <Dateiname>.nlo und der darauf folgende makeindex-Lauf eine Datei <Dateiname>.nls, welche dann in einem weiteren pdf_latex-Lauf automatisch durch \printglossary eingelesen wird. Das Resultat zeigt die nachfolgende Abbildung:

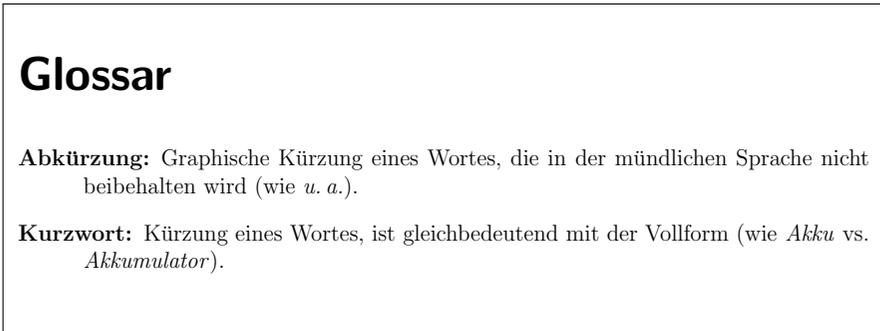


Abbildung 2: nomencl-Beispiel

Wenn man Folgendes in die Präambel einfügt, kann man das Glossar auch zweispaltig ausgeben (Abbildung 3).

```

1 \usepackage{multicol}
2 \renewcommand*\nompreamble{\begin{multicols}{2}}
3 \renewcommand*\nompostamble{\end{multicols}}

```

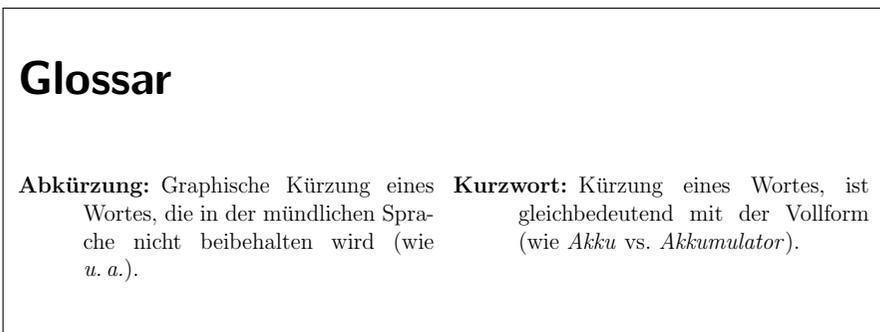


Abbildung 3: nomencl-Beispiel zweispaltig

glossaries

Das komplexe Paket `glossaries` ist der Nachfolger des `glossary`-Pakets und erlaubt die Erstellung eines Glossars, sowie eines Symbol- und/oder Abkürzungsverzeichnisses. Neben einer umfangreichen Einführung in das Paket [3] hat die Autorin auch eine Kurzeinführung für Anfänger [2] bereitgestellt. Letzteres wäre m. E. auch für andere komplexe Pakete wie `hyperref` oder `biblatex` wünschenswert. Neben vielen Zusatzfunktionen sind besonders die implementierten Möglichkeiten für Modifizierungen des Ausgabestils hervorhebenswert. Zum Sortieren und Konfigurieren der Einträge ist das Paket für das Verwenden von `makeindex` oder `xindy` konfiguriert. Außerdem besteht die Möglichkeit, dafür ein Perl-Script (`makeglossaries`) oder Java-GUI (`makeglossariesgui`) zu verwenden. Wenn man beispielsweise mit `xindy` arbeiten möchte, muss man dies über das optionale Argument `kundtun` (`\usepackage[xindy]{glossaries}`).

Am einfachsten ist die Kompilierung mit `makeglossaries`, beziehungsweise über die grafische Umgebung `makeglossariesgui`:

```
pdflatex <Dateiname>
makeglossaries <Dateiname>
pdflatex <Dateiname>
```

Die Glossareinträge werden mit dem nachfolgenden Befehl definiert:

```
\newglossaryentry[<Optionen>]{<Etikett>}{<Definitionsrahmen>}
```

Im Definitionsrahmen wird zum einen ein Name für den Eintrag und zum anderen eine Beschreibung verlangt. Weitere Ergänzungen sind möglich (wie `plural`, legt den Text für die mögliche Pluralausgabe mit `\gls{<Etikett>}` fest, oder `symbol`, bestimmt ein Symbol, das auf den Eintrag verweist). Der Glossareintrag erscheint dann im Glossar, wenn mit ihm beim Auftreten des Terminus im Text auf den Glossareintrag referiert wird. Es erscheint dann auch im Glossar die Nummer der Seite, wo die Referenz(en) gesetzt wurde(n). Dafür stehen verschiedene Befehle bereit; der Hauptbefehl ist `\gls{<Etikett>}`. Mit `hyperref` ist das referierte Lexem mit dem relevanten Glossareintrag verlinkt.

Einen Standardbeispieleintrag ohne spezielle Optionen zeigt die nächste Abbildung:

```
1 \documentclass[a4paper,12pt,ngerman]{scrreprt}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage{babel}
5 \usepackage{hyperref}
6 \usepackage{glossaries}
7 \makeglossaries
8
9 \begin{document}
10
```

```

11 \newglossaryentry{Lexem}{name=Lexem,
12 description={Basiseinheit des Lexikons, die in verschiedenen
13 grammatischen Formen realisiert werden kann}} \gls{Lexem}
14 und Wort werden in der Linguistik unterschieden.
15
16 \printglossary[title={Glossar}]
17
18 \end{document}

```

Lexem und Wort werden in der Linguistik unterschieden.

Glossar

Lexem Basiseinheit des Lexikons, die in verschiedenen grammatischen Formen realisiert werden kann. 1

Abbildung 4: glossaries-Standardeintrag

Wenn man, wie bei einem Glossar üblich, keine Seitenzahlen im Eintrag haben möchte, kann man die Paketoption `nonumberlist` verwenden. Möchte man im Text nicht vorkommende Termini einfügen, kann man die Einträge ohne Textbezug erstellen: Mit dem Befehl `\glsadd[<Optionen>]{<Etikett>}` kann je ein Eintrag in das Glossar hinzugeschrieben werden, mit `\glsaddall[<Optionen>]` werden alle definierten Einträge eingefügt.

Wie bei `glosstex` kann man die Glossarstichwörter auch aus einer externen Datei über das Makro `\loadglsentries` laden (beschrieben in [3, Kap. 4.4]). Dies empfiehlt sich für die Übernahme von Einträgen ohne Textbezug. Vordefiniert sind acht Glossarstile. Dabei sind auch »Multicols Styles« mit zwei Spalten als Standardfall (vgl. [3, Kap. 15]). Außerdem wird eine Anleitung für die Definition einer eigenen Stildatei gegeben [3, Kap. 16].

Zusammenfassend kann festgestellt werden, dass die besprochenen Pakete es ermöglichen, ein usuelles Glossar zu erstellen. Das Paket `glossaries` bietet sogar mehr Möglichkeiten, als die Verfasser von Glossaren bisher benötigt haben.

Literatur

- [1] Michael Schlafer: *Lexikologie und Lexikographie*; Erich Schmidt Verlag; Berlin; 2002.
- [2] Nicola L.C. Talbot: *The glossaries package: a guide for beginners*; 2012; CTAN: macros/latex/contrib/glossaries/glossariesbegin.pdf; zuletzt besucht: 2012-09-17.
- [3] Nicola L.C. Talbot: *User Manual for glossaries.sty*; 2012; CTAN: macros/latex/contrib/glossaries/glossaries-user.pdf; zuletzt besucht: 2012-08-26.
- [4] Boris Veytsmann, Bernd Schandl et al.: *nomencl 3.1a*; 2005; CTAN: macros/latex/contrib/nomencl/nomencl.pdf; zuletzt besucht: 2012-09-16.
- [5] Peter Wilson und Lars Madsen: *The Memoir Class*; 2011; CTAN: macros/latex/contrib/memoir/memman.pdf; zuletzt besucht: 2012-08-26.
- [6] Vokan Yavuz: *GlossTeX 0.4*; 1997; CTAN: support/glosstex/glosstex.pdf; zuletzt besucht: 2012-09-16.

Klausur »Steuerlehre« – \LaTeX -Einsatz für Sehbehinderte

Heiner Richter

Tja, was soll man denn schon antworten, viel mehr kann man ja eigentlich nicht dagegen vorbringen, wenn man von einer stets hilfreichen Person, zumal einem bekannten \TeX -Guru, freundlich gebeten wird, für »Die \TeX nische Komödie« einen kurzen Artikel, ein, zwei Sätze, zu schreiben, wo man doch die Vorteile von \LaTeX im Allgemeinen und die der Nachrichtengruppe `de.comp.text.tex` und der dort helfenden Fachleute im Speziellen über Jahre hinweg in Anspruch nimmt. Sei es, um ein Gesetz in der Entwurfsphase farblich unterscheidbar zu machen, sei es, um mit Randnotizen Arbeiten für einen Loseblatt-Kommentar zum Einkommensteuergesetz in \LaTeX fertigen zu können. Und wenn einem eben gerade aktuell wieder einmal gleich von mehreren Personen geholfen wurde, die für einen Nicht-Word-Nutzer doch recht schwierige Vorgabe – aufgrund eines genehmigten Antrags auf besondere Ausfertigung der Prüfung bei einer anstehenden Steuerlehreklausur – die **Aufgaben in Schriftgröße 14, Schriftart Arial, Zeilenabstand 1,5** zu drucken. Damit sollen sehbehinderte Personen, die krankheitsbedingt nicht in der Lage sind, Prüfungsaufgaben in normaler Ausfertigung in der vorgeschriebenen

Zeit zu bearbeiten, einen Nachteilsausgleich erhalten. In einem Gespräch mit der betroffenen Person, die von meiner Steuerlehreprüfung betroffen war, berichtete diese, dass entgegen der »öffentlichen« Meinung, dass Serifen-Schriften einfacher zu lesen seien, bei ihr gerade wegen der Serifen die Zeilen ineinander verschwömmen. So startete ich kurz einen Hilferuf in der Nachrichtengruppe `de.comp.text.tex`,

- ob mal jemand helfen könne und
- was für Pakete ich verwenden solle.

Nach Auskunft der sehbehinderten Person wäre für jede serifenlose Schrift, nicht nur für den Helvetica-Klon Arial, ein größerer Wert für Schriftgröße und Zeilenabstand bei der vorliegenden Sehbehinderung sehr willkommen. Damit würde der Lesefluss erleichtert und damit zugleich das intellektuelle Verständnis gefördert. Dieses wird aber ohnehin bei meinen im Allgemeinen kurzen, manchmal mathematisch-knappen Aufgabenstellungen mit zuweilen nur einer Zeile Aufgabentext auf einer Seite – der Rest ist freigehalten für die individuelle Lösung der Aufgabe – nicht vor allzu große Hürden gestellt. Schon Minuten nach meiner Anfrage in der Nachrichtengruppe, in der ich meine Frage gestellt hatte, kam der erste Vorschlag. Ich sollte für die serifenlose Schrift die TG-Heros aus dem Projekt »T_EX Gyre« verwenden, wozu das Paket `tgheros` einzubinden war:

```
\documentclass[fontsize=14pt]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{tgheros} \renewcommand{\familydefault}{\sfdefault}

\usepackage[latin1]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{setspace} \onehalfspacing
\usepackage{blindtext}
\begin{document}
\blinddocument
\end{document}
```

Nach einem kurzen Testlauf vermisste ich allerdings passende Mathematikschriften. Auf erneute Nachfrage wurde ich auf die passenden Mathematikschriften `hvmath` verwiesen. Diese sind aber kostenpflichtig, sodass ich die freien Schriften der `kpfonts` empfohlen bekam. Letztere gefielen mir recht gut und waren für den beabsichtigten Einsatz durchaus brauchbar. Durch eine weitere Antwort auf meine Frage bekam ich eine Gegenüberstellung von Text- und Mathematikschriften, die zudem die benötigten Pakete angab:

Textschrift	Paket	Formelsatzpaket
Arev	arev	enthalten
Helvetica	helvet	cmbright, lxfonts
LX	lxfonts	enthalten
Tex Gyre Adventor	tgadventor	arev, lxfonts
Tex Gyre Heros	tgheros	cmbright, lxfonts

Diese Tabelle veranlasste mich, die LX-Schriften zusammen mit den Textschriften arev zu testen. Zumindest für meinen Geschmack erschienen mir die Zeichen aus dem Paket lxfonts irgendwie in die Breite gezogen, als seien sie für Präsentationen gedacht. Es besteht auch die Möglichkeit, dass ich schlicht etwas falsch gemacht habe. Das Paket arev entsprach dagegen meinen Idealvorstellungen, wie man einer Sehbehinderung mit ineinander verschwimmenden Zeilen zu begegnen hat. Für die Steuerklausur wählte ich daher letztlich diese Schrift. Ein Beispieldokument sieht dann wie folgt aus:

```
\documentclass[fontsize=14pt]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{arev} \renewcommand{\familydefault}{\sfdefault}
\usepackage[latin1]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{setspace} \onehalfspacing
\usepackage{blindtext}
\begin{document}
\blinddocument
\end{document}
```

Beispielhaft soll in der Schrift Arev hier der obige Absatz wiederholt werden, wobei die Schrift auf 80 % der natürlichen Größe verkleinert wurde.

Diese Tabelle veranlasste mich, die LX-Schriften zusammen mit den Textschriften Arev zu testen. Zumindest für meinen Geschmack erschienen mir die Zeichen aus dem Paket lxfonts irgendwie in die Breite gezogen, als seien sie für Präsentationen gedacht. Es besteht auch die Möglichkeit, dass ich schlicht etwas falsch gemacht habe. Das Paket arev entsprach dagegen meinen Idealvorstellungen, wie man einer Sehbehinderung mit ineinander verschwimmenden Zeilen zu begegnen hat. Für die Steuerklausur wählte ich daher letztlich diese Schrift.

Spezielle Schriften

Herbert Voß

In dieser Zeitschrift wurde bereits mehrfach darauf hingewiesen, dass mit den Programmen Lua \TeX und Xe \TeX jede Schrift im Format OpenType benutzt werden kann. In diesem Beitrag soll eine ungewöhnliche Schrift vorgestellt werden, die zudem auf einfache Weise in die Definition von Ligaturen bei einer OpenType-Schrift einführt.

Einführung

Auf der Webseite <http://www.ingofonts.de/ingofonts/index.html> von Ingo Zimmermann kann man einige »freie« Schriften herunterladen. Diese haben zwar gegenüber den kommerziellen Varianten wesentliche Einschränkungen, die aber für die hier vorgestellte Schrift unerheblich sind. Es sei darauf hingewiesen, dass hier »frei« nur die freie Nutzung meint. Die spezielle »Schrift« Countries_of_Europe (http://www.ingofonts.de/ingofonts/fonts/schriften/countries_of_europe.html) enthält neben anderen Zeichen die Umrisse der europäischen Länder, die mit der aktuellen Schriftfarbe ausgefüllt sind.

OpenType-Variante

Bei aktuellen Distributionen von \TeX Live oder MiK \TeX kann man davon ausgehen, dass eine entsprechende Type-1-Variante der Schrift bereits installiert ist. Eine OpenType-Variante installiert man mit der auf der angegebenen Webseite vorhandenen zip-Datei `iF_Countries_of_Europe.zip`. Diese enthält ein Mac-typisches Archiv, in welchem man in einem weiteren Unterverzeichnis gleichen Namens die Schriftdatei im Format OpenType findet: `CountriesofEurope.otf`.

Für Linux speichert man diese Datei mit der Endung `.otf` entweder im persönlichen Ordner `.fonts/`, im lokalen Ordner `/usr/local/share/fonts/` oder systemweit in `/usr/share/fonts/`, wobei für die letzten beiden Varianten im Allgemeinen Administratorrechte nötig sind. Unter Windows speichert man die Datei im Schriftenordner `C:\Windows\Fonts\`. Der eigentliche Schriftname lässt sich mit dem Programm `otfinfo` angeben:

```

1 voss@shania:~/fonts> otfinfo -i CountriesofEurope.otf
2 Family:                Countries of Europe
3 Subfamily:             Regular
4 Full name:             Countries of Europe
5 PostScript name:      CountriesofEurope
6 Preferred family:     Countries of Europe
7 Preferred subfamily:  Book
8 Version:              Version 1.000

```

```

9 Unique ID:          ingoFontsIngoZimmermann: Countries of Europe: 2008
10 Description:       Copyright (c) 2008 by Ingo Zimmermann. All rights
    ↳reserved.
11 Designer:          Ingo Zimmermann
12 Designer URL:      www.ingofonts.com
13 Manufacturer:      ingoFonts Ingo Zimmermann
14 Vendor URL:        www.ingofonts.com
15 Trademark:         Countries of Europe is a trademark of ingoFonts
    ↳Ingo Zimmermann.
16 Copyright:         Copyright (c) 2008 by ingoFonts Ingo Zimmermann.
    ↳All rights reserved.
17 License URL:       www.ingofonts.com
18 Vendor ID:         ifon

```

Man findet die Länder innerhalb der OpenType-Schrift im Bereich F8C6–F8EC. Die Bundesrepublik Deutschland liegt auf der Position F8DF und kann durch das Makro `\symbol` auch direkt dargestellt werden, wenn die richtige Schriftart aktiv ist. Sinnigerweise packt man die Schriftumschaltung in eine eigene Definition, die zudem auch gleich eine Skalierung bestimmt:

```

1 \newfontface\Europe[Scale=5]{Countries Of Europe}
2 \newcommand\UniCountry[1]{\Europe\char"#1}}

```

Somit kann die Bundesrepublik Deutschland dann durch `\UniCountry{F8DF}` dargestellt werden:



```

1 {\fontspec{CountriesofEurope.otf} GE }
2 \UniCountry{F8DF}
3 \textcolor{red}{\UniCountry{F8DF}}

```

Das Land selbst ist als normales Zeichen definiert, wobei lediglich die Zahl der Kontrollpunkte erheblich größer ist. Abbildung 1 zeigt dies am Beispiel Deutschlands unter Anwendung der Software `fontforge` (<http://fontforge.sourceforge.net/>).

Für Zeichen im OpenType-Format können mehrfache Buchstabenkombinationen definiert werden, die alle die gleiche Ligatur ergeben. Im Fall von Deutschland zeigt Abbildung 2 die Definitionen. Es werden die sechs Zeichenkombinationen GE, Ge, ge, DEU, Deu und deu definiert, die alle auf die »Ligatur« F8DF abgebildet werden.

Die Tabelle 1 zeigt eine Zusammenstellung der Länder mit ihrem zugehörigen Unicode. Anstelle der Eingabe der Unicodenummer kann man auch die erwähnten Textbezeichnungen verwenden. Eine Erweiterung oder Änderung der möglichen Abkürzungen kann über das Menü von Abbildung 2 erfolgen. Die Größe der Länder ist jeweils relativ zueinander korrekt, sodass man bei den sehr kleinen Staaten schon genauer hinsehen muss, um sie zu erkennen, beispielsweise bei Liechtenstein. In solchen Fällen empfiehlt es sich, über die Option `Scale` einen größeren Wert vorzusehen oder das Makro `\scalebox` aus dem Paket `graphicx` zu benutzen. Da die einzelnen Zeichen der Schrift trotz der vielen Stützpunkte als Kurvenzug definiert

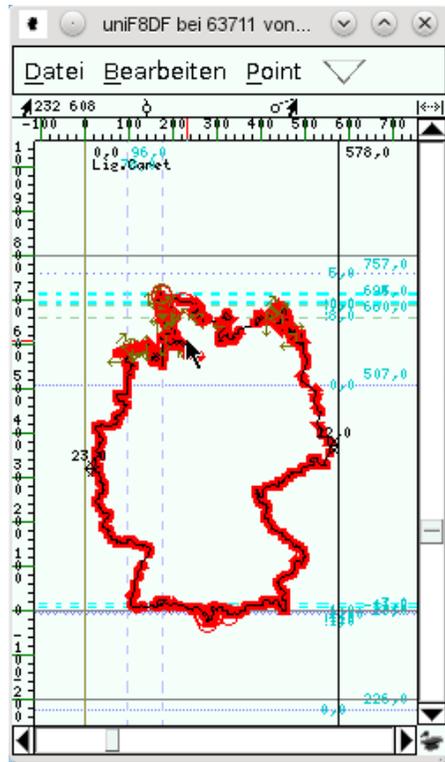


Abbildung 1: Definition des »Zeichens« F8DF (Deutschland) mit fontforge.

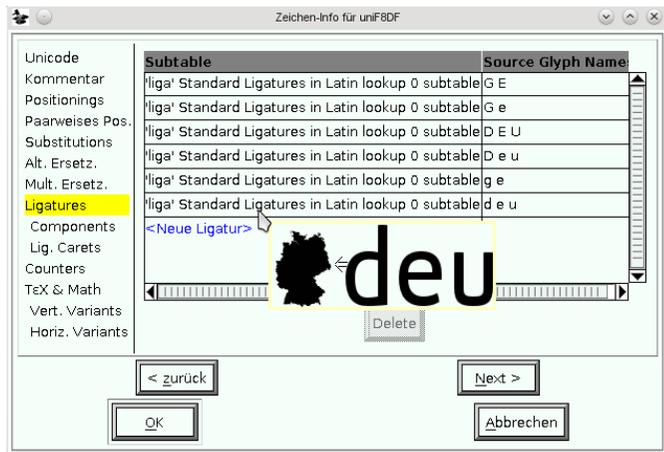


Abbildung 2: Definition von Ligaturen innerhalb des Programms fontforge.

sind (Vektorschrift), kann die Skalierung zwar ohne Probleme durchgeführt werden, jedoch bleibt die Zahl der Stützpunkte gleich. Wie im folgenden Beispiel deutlich zu sehen ist, macht sich dies bei sehr starker Vergrößerung durch längere Linienzüge negativ bemerkbar, indem der Grenzverlauf »eckiger« wird. Der Abstand zum letzten Zeichen kommt durch den ungewöhnlich großen Wert der Boxbreite des Zeichens zustande. So kann selbst Liechtenstein zu einem großen Land werden, wenn auch ziemlich eckig aussehend:



```

1 \UniCountry{F8D7}% 5fach
2 \begingroup
3 \setmainfont[Scale=20]{CountriesofEurope.otf}%
4 LIC\scalebox{5}{LIC}% entspricht dem 100fachen
5 \endgroup

```

Der Verwendung dieses Zeichensatzes sind somit automatisch Grenzen gesetzt, wenn kleinere Staaten stark vergrößert werden müssen.

Tabelle 1: Zusammenstellung der verfügbaren Länder und möglicher Zugriff über Textsequenzen oder Unicode-Nummer (sortiert nach der Unicode-Nummer).

<i>Land</i>	<i>Unicode</i>	<i>Text</i>	<i>Alternativen</i>	<i>Ausgabe</i>
Albanien	F8EC	AL	Al, al	
Andorra	F8EB	AN	An, an	
Österreich	F8EA	AU	Au, au	
Weißrussland	F8E9	WE	We, we, BELA, Bela, bela	
Belgien	F8E8	BELG	Belg, belg	
Bosnien/ Herzegowina	F8E7	BO	Bo, bo, BH, Bh, bh, BYH, Byh, byh, BIH, Bih, bih	
Bulgarien	F8E6	BU	Bu, bu	
Kroatien	F8E5	CR	Cr, cr, HR, Hr, hr, KR, Kr, kr	
Tschechien	F8E4	CZ	Cz, cz, TS, Ts, ts, TC, Tc, tc	
Dänemark	F8E3	DAN	Dan, dan, DEN, Den, den, DAE, Dae, dae	
Estland	F8E2	EST	Est, est	
Finnland	F8E1	FI	Fi, fi	

Fortsetzung ...

... Fortsetzung

<i>Land</i>	<i>Unicode</i>	<i>Text</i>	<i>Alternativen</i>	<i>Ausgabe</i>
Frankreich	F8E0	FR	Fr, fr	
Deutschland	F8DF	GE	Ge, ge, DEU, Deu, deu	
Großbritannien	F8DE	BR	Br, br, GRO, Gro, gro, GREA, Grea, grea, EN, En, en	
Griechenland	F8DD	GRI	Gri, gri, GREE, Gree, gree, HE, He, he	
Ungarn	F8DC	UN	Un, un, HU, Hu, hu	
Island	F8DB	IS	Is, is, IC, Ic, ic	
Irland	F8DA	IR	Ir, ir	
Italien	F8D9	IT	It, it	
Lettland	F8D8	LA	La, la, LE, Le, le	
Liechtenstein	F8D7	LIC	Lic, lic, LIE, Lie, lie	
Litauen	F8D6	LIT	Lit, lit	
Luxemburg	F8D5	LU	Lu, lu	
Mazedonien	F8D4	MAZ	Maz, maz, MAC, Mac, mac, MAK, Mak, mak	
Malta	F8D3	MAL	Mal, mal	
Moldawien	F8D2	MOL	Mol, mol	
Montenegro	F8D1	MON	Mon, mon	
Niederlande	F8D0	NE	Ne, ne, NI, Ni, ni, HO, Ho, ho	

Fortsetzung ...

... Fortsetzung

<i>Land</i>	<i>Unicode</i>	<i>Text</i>	<i>Alternativen</i>	<i>Ausgabe</i>
Norwegen	F8CF	NO	No, no	
Polen	F8CE	POL	Pol, pol	
Portugal	F8CD	POR	Por, por	
Rumänien	F8CC	RO	Ro, ro, RUM, Rum, rum	
Serbien	F8CB	SE	Se, se	
Slowakei	F8CA	SLOWA	Slowa, slowa, SLOVA, Slova, slova	
Slowenien	F8C9	SLOVE	Slove, slove, SLOWE, Slowe, slowe	
Spanien	F8C8	SP	Sp, sp	
Schweden	F8C7	SV	Sv, sv, Swe, swe, SWE, SCHWED, Schwed, schwed	
Schweiz	F8C6	SWI	Swi, swi, Schwei, SCHWEI, schwei	

Type-1-Variante

Für die Anwendung mit pdf_{La}TeX ist eine spezielle Anpassung notwendig, da hier der Speicherbereich nicht in der Weise zur Verfügung steht, wie es für OpenType möglich ist. Auf CTAN und auch auf einem aktuellen TeXLive sollte das Paket `CountriesofEurope` mit allen notwendigen Dateien zur Verfügung stehen, sodass die Schrift auch als Type 1 verwendet werden kann. Da es nur eine einzige Variante

der Schrift gibt, fehlen die so genannten .fd-Dateien; das Laden des Paketes ist völlig ausreichend:

```
1 \usepackage[scaled]{CountriesofEurope}
```

Das optionale Argument [scaled] vergrößert die Zeichen um den Faktor 10. Durch Zuweisung eines anderen Wertes, beispielsweise [scaled=5], kann eine andere Skalierung gewählt werden. Alternativ kann mit den Makros \scalebox oder \resizebox, beide aus dem Paket graphicx, die Größe des Zeichens jederzeit beeinflusst werden.

Das Paket definiert die Zuordnung der einzelnen Länder zu entsprechenden Makros, auf die dann normal zugegriffen werden kann.

\Albania, \Andorra, \Austria, \Belarus, \Belgium, \Bosnia, \Bulgaria, \Croatia, \Czechia, \Denmark, \Estonia, \Finland, \France, \Germany, \GreatBritain, \Greece, \Hungary, \Iceland, \Ireland, \Italy, \Latvia, \Liechtenstein, \Lithuania, \Luxembourg, \Macedonia, \Malta, \Moldova, \Montenegro, \Netherlands, \Norway, \Poland, \Portugal, \Romania, \Serbia, \Slovakia, \Slovenia, \Spain, \Sweden, und \Switzerland.

Intern liegen die Zeichen auf den Positionen 128–166 (in der hier angegebenen Reihenfolge) und sind auch über das Makro \char oder \symbol zugänglich, wobei in diesen Fällen vorher auf die Schriftkodierung U umgeschaltet werden muss. Die durch das Paket CountriesofEurope definierten Makros \CountriesofEuropeFamily und \EUCountry erleichtern die Anwendung. Im ersten Fall wird auf die entsprechende Kodierung umgeschaltet und im zweiten Fall kann dem Makro eine Zahl zwischen 128–166 übergeben werden.

Hinweise

Die Zeichen haben im Wesentlichen nur eine Höhe, sodass sie mit ihrer Unterkante



fast immer auf der Höhe der aktuellen Basislinie gesetzt werden: Dies ist zu beachten, wenn eine Ausgabe in der laufenden Zeile erfolgen soll und es sich dabei um ein relativ großes Land handelt, wie beispielsweise Norwegen, welches eine definierte Höhe von 64.95 pt, aber nur eine Tiefe von 1.45 pt hat. In der Tabelle 1 wurden die Zeichen daher mit dem Makro \raisebox um die Hälfte ihrer aktuellen Höhe nach unten verschoben:

```
1 \newcommand\RUniCountry[1]{\raisebox{-0.5\height}{\Europe\char"#1}}
```


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.

filedate von *Uwe Lück* erlaubt den Zugriff auf die Infodaten von \LaTeX -Paketen.

CTAN:macros/latex/contrib/filedate

mkbib von *Rudra Banerjee* ist ein gtk+-3-basiertes Projekt zur Erzeugung von \BibTeX -Dateien.

CTAN:biblio/bibtex/utills/mkbib

francais-bst von *Vincent Goulet* ist ein \BibTeX -Stil für französische Bibliografien. Er ist mit *natbib* kompatibel.

CTAN:biblio/bibtex/contrib/francais-bst

actuarialangle von *Vincent Goulet* erzeugt ein Winkelsymbol im Mathematikmodus.

CTAN:macros/latex/contrib/actuarialangle

setdeck von *Gwyneth Whieldon* dient zur Wiedergabe von Spielstellungen des Kartenspiels »Set!«.

CTAN:graphics/pgf/contrib/setdeck

adhocfilelist von *Uwe Lück* ist ein Shellskript, das eine Liste der `\Provides<...>`-Befehle eines Pakets auf dem Bildschirm ausgibt.

CTAN:support/adhocfilelist

physics von *Sergio C. de la Barrera* stellt Makros zum Setzen von Dirac-Gleichungen zur Vektoranalysis und zur linearen Algebra bereit.

CTAN:macros/latex/contrib/physics

pkuthss von *Casper Ti. Vector* ist eine Klasse für Doktorarbeiten an der Universität Peking.

CTAN:macros/latex/contrib/pkuthss

biblatex-casparvector von *Casper Ti. Vector* ist ein \BibTeX -Stil für chinesische Texte.

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

exsheets von *Clemens Niederberger* dient zum Erstellen von Arbeitsblättern für den Unterricht.

CTAN:macros/latex/contrib/exsheets

ulthese von *Pierre Lasou* ist eine Klasse für Abschlussarbeiten der Université Laval, Québec.

CTAN:macros/latex/contrib/ulthese

latex-sciences-humaines von *Maïeul Rouquette* ist ein E-Book in französischer Sprache über den Einsatz von $\text{Xe}_{\text{L}}\text{TeX}$ für geistes- und sozialwissenschaftliche Texte: *TeX appliqué aux sciences humaines*. Der Quelltext liegt auf <https://github.com/maieul/latexhumain>. Eine gedruckte Ausgabe kann über atramenta.net bestellt werden.

CTAN:info/latex-sciences-humaines.pdf

eledform von *Maïeul Rouquette* ergänzt das Paket *eledform* und erlaubt das Auszeichnen von Textvarianten in kritischen Textausgaben.

CTAN:macros/latex/contrib/eledform

ocgx von *Paul Gaborit* erlaubt es, *Optional content groups* in PDFs ohne JavaScript zu verwenden.

CTAN:macros/latex/contrib/ocgx

loops von *Ahmed Musa* stellt Makros für Schleifen bereit.

CTAN:macros/latex/contrib/loops

scrjrn1 von *Raphaël Pinson* ist eine Klasse, die auf *scrbook* aufsetzt, mit der man Tagebücher setzen kann.

CTAN:macros/latex/contrib/scrjrn1

bohr von *Clemens Niederberger* erlaubt es, einfache Bohr-Atommodelle bis zur Ordnungszahl 112 zu erstellen.

CTAN:macros/latex/contrib/bohr

jmakepdfx von *Nicola Talbot* ist ein Java-Interface für Ghostscript, um PDF nach PDF/X zu wandeln.

CTAN:graphics/jmakepdfx

pxcjkcat von *Takayuki Yato* ändert die CJK-Kategorieeinstellungen in $\text{up}_{\text{E}}\text{TeX}$.

CTAN:macros/latex/contrib/pxcjkcat

langcode von *Uwe Lück* erlaubt es, alle sprachabhängigen Einstellungen mit dem Befehl `\uselangcode` zu setzen, so dass diese einheitlich für alle Sprachen verwendet werden können.

CTAN:macros/generic/langcode

catcodes von *Uwe Lück* ist ein Bundle, das mehrere Pakete enthält, die *catcodes* für alle $\text{T}_{\text{E}}\text{X}$ -Formate umsetzen können.

CTAN:macros/generic/catcodes

plainpkg von *Uwe Lück* stellt eine minimale Methode zum Erstellen von »generischen« Paketen bereit.

CTAN:macros/generic/plainpkg

libertine-type1 und *biolinum-type1* von *Bob Tennent* enthält die Unterstützung für die *Type1*-Fonts *Linux Libertine* und *Biolinum*.

CTAN:fonts/libertine-type1

CTAN:fonts/biolinum-type1

designcon von *Peter J. Pupaiaikis* enthält Klassen und Vorlagen für \LaTeX und \LyX zum Verfassen von Beiträgen für die Tagung *DesignCon*.

CTAN:macros/latex/contrib/designcon

resphilosophica von *Boris Veytsman* ist eine Klasse zum Setzen von Beiträgen für die Zeitschrift *Res Philosophica*.

CTAN:macros/latex/contrib/resphilosophica

eledmac und *eledpar* von *Maïeul Rouquette* sind die Nachfolger der Pakete *ledmac* und *ledpar* zum Setzen textkritischer Ausgaben.

CTAN:macros/latex/contrib/eledmac

calcage von *Robin Schneider* berechnet das Alter einer Person in Jahren.

CTAN:macros/latex/contrib/calcage

fnumprint von *Robin Schneider* setzt mithilfe des Pakets *zahl2string* automatisch Zahlen zwischen null und zwölf als Wörter. Zahlen größer als zwölf werden automatisch mit dem Paket *numprint* gesetzt.

CTAN:macros/latex/contrib/fnumprint

uspatent von *Peter J. Pupaiaikis* ist ein Paket zum Verfassen eines US-amerikanischen Patentantrags, auch mit \LyX einsetzbar.

CTAN:macros/latex/contrib/uspatent

biblatex-trad von *Marco Daniel* implementiert die traditionellen Zitationsstile *plain*, *abbrv*, *unstr* und *alpha* für \BibTeX .

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

svg von *Philip Ilten* erlaubt es, auf unixoiden Plattformen SVG-Grafiken mit dem neuen Befehl \includegraphics in Dokumente einzubinden. Benötigt werden *Inkscape* und die Programme *pdftops* sowie *convert* aus *ImageMagick*.

CTAN:graphics/svg

hacm von *Kazuaki Miyatani* dient zum Setzen des Alphabets für die konstruierte Sprache *Arka*.

CTAN:fonts/hacm

hobby von *Andrew Stacey* definiert eine Pfadfunktion für *TikZ/PGF* in \LaTeX , die den *Hobby*-Algorithmus implementiert

CTAN:graphics/pgf/contrib/hobby

abraces von *Werner Grundlingh* stellt ein Interface für neue \overbrace / \underbrace -Klammerpaare bereit.

CTAN:macros/latex/contrib/abraces

counttexruns von *Robin Schneider* zählt, wie oft ein \LaTeX -Dokument kompiliert worden ist (kann auch ausgegeben werden).

CTAN:macros/latex/contrib/counttexruns

plipsum von *Sergio Spina* erzeugt *Lorem-ipsum*-Texte für *Plain TeX*.

CTAN:macros/plain/contrib/plipsum

Presentations_en von *Herbert Voß* enthält die Beispiele aus dem Buch »Presentations with \LaTeX «.

CTAN:info/examples/Presentations_en

chkfloat von *Tomas Hejda* gibt eine Warnung aus, wenn ein Gleitobjekt »zu weit« von seinem Ursprung entfernt platziert wird.

CTAN:macros/latex/contrib/chkfloat

oscola von *Paul Stanley* ist ein BibLaTeX-Stil der die Vorgaben des *Oxford Standard for the Citation of Legal Authorities (OSCOLA)* umsetzt. Es erleichtert auch das Erstellen von Tabellen mit Gerichtsurteilen und Gesetzen aus den zitierten Fundstellen.

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

hf-tikz von *Claudio Fiandrino* hebt mathematische Formeln bzw. Teile davon in Dokumenten und in Präsentationen mithilfe von TikZ hervor.

CTAN:graphics/pgf/contrib/hf-tikz

pst-ode von *Alexander Grahn* erlaubt es, Anfangswertprobleme gewöhnlicher Differentialgleichungen mit dem Runge-Kutta-Fehlberg-Algorithmus zu lösen und die Lösung als 2D- oder 3D-Grafik zu plotten. Optional können die Zustandsvektoren in eine Textdatei ausgegeben werden.¹

CTAN:graphics/pstricks/contrib/pst-ode

gs1 von *Markus Kohm* dient dazu, GS1-Codes und Strichcodes mit *exp13* zu erzeugen. Derzeit werden nur EAN-8 und EAN-13 unterstützt.

CTAN:macros/latex/contrib/gsl

imprintmtshadow von *Boris Veytsman* enthält die Unterstützung für den Font Monotype Imprint Shadow.

CTAN:fonts/monotype/imprintmtshadow

lsabon von *Boris Veytsman* enthält die Unterstützung für die Schriftart Sabon von Linotype.

CTAN:fonts/psfonts/linotype/lsabon

cellwise von *Matthias Borck-Elsner* erlaubt das zellweise Aufbauen und Formatieren von Tabellen.

CTAN:macros/latex/contrib/cellwise

¹Der Verf. dankt Moritz Brinkmann und Oliver Jennrich für diesen sachdienlichen Vorschlag zur Beschreibung des Pakets auf der Vereinsmailingliste.

Spielplan

Termine

2013

6. 3. – 8. 3. **DANTE 2013**
und 48. Mitgliederversammlung von DANTE e.V.
Justus-Liebig-Universität
Interdisziplinäres Forschungszentrum (IFZ)
Heinrich-Buff-Ring 26
35392 Gießen
<http://www.dante.de/events/dante2013.html>
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/>
9. 11. – 10. 11. **OpenRheinRuhr**
Freie Software und Netzpolitik
Rheinisches Industriemuseum (RIM) Oberhausen
<http://www.openrheinruhr.de/>

2014

- April **DANTE 2014**
und 50. Mitgliederversammlung von DANTE e.V.
Universität Heidelberg

Stammtische

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

Aachen

Torsten Bronger, bronger@physik.rwth-aachen.de

Gaststätte Knossos, Templergraben 28, 52062 Aachen

Zweiter Donnerstag im Monat, 19.00 Uhr

Berlin

Michael-E. Voges, Tel.: (03362) 50 18 35,

mevoges@t-online.de

Ort derzeit wechselnd

Zweiter Donnerstag im Monat, 19.00 Uhr

Bielefeld

Jürgen Schwarze, Tel.: (0521) 5 57 39 06,

juergen.schwarze@bitel.net

Hackerspace Bielefeld, Sudbrackstraße 42, 33611 Bielefeld, Zweiter Montag im Monat, 19.30 Uhr

Bremen

Winfried Neugebauer, Tel.: 0176 60 85 43 05, tex@wphn.de

Wechselnder Ort

Erster Donnerstag im Monat, 18.30 Uhr

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	22.03.1990	Klaus Thull †	22.03.1990
Yannis Haralambous	05.09.1991	Barbara Beeton	27.02.1997
Luzia Dietsche	27.02.1997	Donald E. Knuth	27.02.1997
Eberhard Mattes	27.02.1997	Hermann Zapf	19.02.1999

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

- Walter Entenmann** [29] **Heiner Richter** [80]
 Habichtweg 10
 82152 Krailling
 Walter.Entenmann@t-online.de
 Fachhochschule Stralsund
 Fachbereich Wirtschaft
 Zur Schwedenschanze 15
 18435 Stralsund
- Jürgen Fenn** [91] **heiner.richter@fh-stralsund.de**
 Friedensallee 174/20
 63263 Neu-Isenburg
 juergen.fenn@gmx.de
- Markus Kohm** [58] **Christine Römer** [74]
 Freiherr-von-Drais-Straße 66
 68535 Edingen-Neckarhausen
 komascript@gmx.info
 Institut für germanistische
 Sprachwissenschaft
 FSU Jena
 Christine.Roemer@uni-jena.de
- David Latchman** [12] **Martin Sievers** [4, 10]
 David.Latchman@gmail.com
 siehe Seite 98
- Manfred Lotz** [6] **Herbert Voß** [3, 82]
 Schriftführer von DANTE e.V.
 manfred@dante.de
 Wasgenstraße 21
 14129 Berlin
 herbert@dante.de
- Günter Partosch** [10] **U. Ziegenhagen** [21]
 Guenter.Partosch@hrz.uni-giessen.de
 Köln
 Uwe@Ziegenhagen.info

Die T_EXnische Komödie

24. Jahrgang Heft 4/2012 November 2012

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 6 Beschlüsse der 47. Mitgliederversammlung
- 10 DANTE 2013 – Einladung zur MV und »Call for Papers«
- 12 TUG 2012: A First-Time Attendee
- 22 Klompen, Kaas & ConT_EXt – Die EuroT_EX 2012 in Breskens

Bretter, die die Welt bedeuten

- 29 Schulschriften – von Sütterlin bis heute
- 58 Briefpapier mit KOMA-Script nachbauen
- 74 Das Erstellen eines Glossars
- 80 Klausur »Steuerlehre« – L^AT_EX-Einsatz für Sehbehinderte
- 83 Spezielle Schriften

Von fremden Bühnen

- 91 Neue Pakete auf CTAN

Spielplan

- 95 Termine
- 96 Stammtische

Adressen

- 99 Autoren/Organisatoren