

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

35. Jahrgang Heft 1/2023 Februar 2023

1/2023

Impressum

»Die T_EXnische 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-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse dtk erstellt und per E-Mail oder Datenträger (z. B. CD/DVD) an unten stehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, L^AT_EX-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 <https://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit LuaHBTeX, Version 1.15.0 (TeX Live 2022) erstellt. Als Standardschriften kamen Libertinus Serif, Libertinus Sans Serif, Anonymous Pro und Libertinus Math zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2150

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 11 03 61
69072 Heidelberg

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

Druck: Schleunungsdruck GmbH
Eltertstraße 27, 97828 Marktheidenfeld

Redaktion: Luzia Dietsche (verantwortliche Redakteurin)

Mitarbeit: Rudolf Herrmann Ralf Mispelhorn Rolf Niepraschk
Stefan Pinnow Bernd Raichle Christine Römer
Herbert Voß

Redaktionsschluss für Heft 2/2023: 15. April 2023

ISSN 1434-5897

Die T_EXnische Komödie 1/2023

Editorial

Liebe Leserinnen und Leser,

dies ist die erste Ausgabe in diesem Jahr, in dem offiziell Corona beendet wurde – zumindest als Pandemie. Dass sich etwas ändert (hoffentlich bessert), kann man unter anderem an der Anzahl an Terminen sehen: Die Tagung $\text{BachO}\TeX$ 2023 in Polen soll auf jeden Fall stattfinden, nachdem sie zweimal in Folge abgesagt werden musste (siehe auch Seite 62). Die $\text{Bay}\TeX$ findet statt (die Einladung dazu wird im nächsten Heft folgen) und die Tagung der $\TeX$ Users Group wird dieses Jahr in Deutschland abgehalten, auf die ich mich ganz besonders freue. Ja, und natürlich gibt es wieder das $\text{Con}\TeX$ t Meeting, das in jedem Jahr Interessierte anzieht. Viele Möglichkeiten also, andere Textsatzbegeisterte in Persona zu treffen, die genauen Angaben sind auf Seite 62 zu finden.

Auch einige Artikel dieser Ausgabe beschäftigen sich mit dem Blick nach vorne. Das » $\text{L}\TeX$ Tagged PDF Project« wird vorgestellt, ebenso wie die Möglichkeit, XMP-Metadaten in ein PDF hinzuzufügen. $\text{Con}\TeX$ t kommt zu meiner großen Freude nicht zu kurz, wurde es doch nach meinen Gefühl jahrelang sträflich vernachlässigt. Danke an Hraban Ramm, der sich immer wieder die Mühe macht, $\text{Con}\TeX$ t in einen $\text{L}\TeX$ -Rahmen zu zwängen. Von der Artikelreihe zu Schulphysik verspreche ich mir viel, beschäftigt sie sich doch mit einem Thema, das ich enorm wichtig finde – $\TeX$ in Schulen. Nutzen zumindest Teile des Lehrkörpers $\text{L}\TeX$ für ihre Präsentationen, besteht berechnete Hoffnung, dass auch einige Jugendliche sich dafür interessieren.

Mein letztes Anliegen in diesem Editorial bezieht sich auf die Rettung eines Kulturschatzes, des Offizin Haag-Drugulin in Dresden. Ein Handwerksbetrieb, dessen Ziel hohe Qualität im Bereich der Druckkunst ist, dem nun aber die Schließung droht. Daher wird nach Unterstützern für die Rettung via diesem Aufruf gesucht: <https://www.verein-fuer-die-schwarze-kunst.de/die-zukunft-der-offizin-haag-drugulin-ist-ungewiss/>. Ich wünsche dem Offizin auf jeden Fall viel Glück!



Und damit wünsche ich Ihnen und Euch wieder viel Spaß beim Lesen und verbleibe mit $\TeX$ nischen Grüßen

Luzia Dietsche



Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

»Das ist das schönste Amt neben dem Papst – Vorsitzender der SPD zu sein.« So beschrieb einst Franz Müntefering sein Parteiamt. Ich vermag nicht zu sagen, wo der Vorsitz von DANTE e.V. in der »Münste-Skala« rangieren würde – für mich ist es ein Amt, das ich seit nunmehr knapp elf Jahren gerne ausübe. Gleichwohl ich gestehen muss, dass die Zeiten schon angenehmer sowie zeit- und nervenschonender waren.

Zu Beginn des Jahres hat unsere Bürokraft Frank Gerhard gekündigt, so dass unser Büro in Heidelberg seit dem 1. Februar nicht mehr besetzt ist. Was das im Einzelnen für die Mitglieder bedeutet, haben Uwe Ziegenhagen und ich auf S. 5 zusammengefasst. Ich bedauere Franks Entscheidung sehr, weil ich hoffte, dass wir nach dem ersten Jahr der Einarbeitung nun Routinen entwickeln und dabei auch neue Dinge integrieren könnten. Ich danke ihm ausdrücklich für die Bereitschaft, sich unter sicherlich nicht immer leichten Rahmenbedingungen in ein für Außenstehende eher »exotisches« Umfeld einzuarbeiten und wünsche ihm alles Gute.

Franks Weggang hat auch gravierende Auswirkungen auf das Vereinsinnere, auf das Tagesgeschäft im Büro. Vor allem die Post, die Mitgliederverwaltung und die Buchungen werden derzeit vollständig von unserer Schatzmeisterin übernommen. Auch wenn wir versuchen, den Umfang dieser Tätigkeiten zu reduzieren, bleibt dies ein enormer Mehraufwand. Gleichzeitig sind diese Aufgaben für einen gemeinnützigen Verein lebensnotwendig, um z. B. nicht gegenüber dem Finanzamt in Erklärungsnot zu geraten. Daher rechne ich es Doris hoch an, dass sie sich trotz anderer Verpflichtungen dieser Aufgabe ohne zu zögern angenommen hat. Auch die übrigen Vorstandsmitglieder, allen voran Uwe und Klaus, unterstützen an den verschiedenen Baustellen, die sich nun ergeben haben (und noch ergeben werden).

Das kann natürlich kein Dauerzustand sein. Mehrfach bin ich schon gefragt worden, ob es denn eine Idee zur Nachfolge gibt. Ich muss ehrlich sagen: Ich sehe keinen Sinn darin, innerhalb von weniger als zwei Jahren eine weitere – die dritte – Einstellungs- und Einarbeitungsschleife zu drehen (diesmal dann unter nochmals

erschweren Bedingungen). Was aber ist die Alternative? Darüber diskutieren wir im Vorstand intensiv und werden hoffentlich sehr bald ein Gesamtkonzept haben. Schließlich müssen wir ja auch überlegen, wie es mit dem Büro an sich weitergeht. Die Kündigung zum März 2024 ist ausgesprochen. Man sollte also sinnvollerweise Büro (Raum) und Büro (Arbeit) zusammen denken. Wie gesagt, es war schon einmal zeit- und nervenschonender als im Moment, doch wir werden auch diese Situation nicht nur überstehen, sondern aktiv zur Neuausrichtung nutzen.

Die »Fokusverschiebung« hat auch Auswirkungen auf weitere Bereiche des Vereinslebens. Wir werden in diesem Jahr eine »kleine Tagungslösung« wählen und die Mitgliederversammlung an die in Bonn stattfindende TUG-Tagung angliedern. Vorgesehen ist dafür der 13. Juli, 10–13 Uhr. Für die Tagungsteilnahme selbst gewährt DANTE e.V. einen Rabatt von \$100.

Eine offizielle Einladung zur 65. MV folgt in der kommenden Ausgabe der Mitgliederzeitung. In diesem Jahr stehen wieder Wahlen zum Vorstand an. Zudem werden drei Rechnungsprüfer gewählt. Genügend Möglichkeiten also, sich bei DANTE e.V. in einem Amt zu engagieren und hoffentlich gute Gründe, an der Mitgliederversammlung teilzunehmen.

Mit eben solchem Engagement wollte uns Thomas H. Meyer schon im letzten Jahr ins schöne Saarland einladen. Da dies aber leider nicht klappte, haben wir mit ihm nun für 2024 (voraussichtlich im April) eine Tagung in Oßmannstedt (Nähe Weimar) ins Auge gefasst.

Bei all den aktuellen Herausforderungen und dem Blick nach Vorne soll nicht untergehen, dass Herbert Voß von seinem Amt als Beisitzer zurückgetreten ist. 16 Jahre lang hat er in verschiedenen Vorstandsämtern viel Zeit und Herzblut in unseren Verein eingebracht. Ich möchte mich ganz herzlich für sein großes Engagement und die langjährige Zusammenarbeit bedanken. Auch ohne Amt wird Herbert unser Vereinsleben hoffentlich weiterhin mit Vorträgen auf Tagungen, Diskussionsbeiträgen und vielem mehr bereichern.

Am Schluss habe ich die traurige Pflicht, an diejenigen Mitglieder zu erinnern, die im vergangenen Jahr von uns gegangen sind. Wir gedenken der Verstorbenen Uwe Naundorf (Mitgliedsnummer 468), Martin Fink (1226), Andreas Hörstemeier (1966), Gerhard Tischel (2059), Albrecht Mehl (2152), Thomas Schramm (3651), Klaus Volkamer (3889), Günther Kopetzky (3891), Roland Möller (5084), Jürgen Gilg (5629), Andreas Heinze und Joachim Labude.

Trotz allem wünsche ich auch diesmal viel Vergnügen bei der weiteren Lektüre.

Herzlichst Ihr/Euer
Martin Sievers

Veränderungen im Büro

Liebe Mitglieder,

wir haben es bereits über die vereinsinterne Mailingliste kommuniziert, möchten es aber auch auf diesem Weg bekanntgeben.

Frank, unsere Bürokraft in Heidelberg, hat leider zu Beginn des Jahres gekündigt. Dies hat zur Folge, dass wir seit dem 1. Februar auf absehbare Zeit die bisher durch ihn im Büro erbrachten Vereins-Dienstleistungen stark einschränken müssen.

Im Einzelnen bedeutet das für unsere Mitglieder:

- Die Kombi-Mitgliedschaft mit der TUG kann für 2023 nicht über DANTE e.V. gebucht werden. Bitte werden Sie bei Bedarf direkt über die TUG-Seite (<https://tug.org/join.html>) Mitglied.
- Der Versand von Tassen und Büchern wird bis auf weiteres nur in unregelmäßigen Abständen alle paar Wochen erfolgen können.
- Das Büro ist nicht mehr besetzt. Wir werden den Anrufbeantworter regelmäßig abhören bzw. eine Weiterleitung einrichten, bitten aber um Verständnis, falls wir uns nicht zeitnah zurückmelden. Wir verweisen stattdessen auf die Möglichkeit der Kontaktaufnahme per E-Mail, denn
- das Postfach office@dante.de wird regelmäßig gelesen. Wir bemühen uns um eine zeitnahe Antwort.

Vor dem Hintergrund, dass uns zudem das Büro vom Vermieter zum März 2024 gekündigt wurde, werden wir die nächsten Wochen nutzen, um verschiedene Strategien zu betrachten und zu entscheiden, ob, wie und durch wen wir die Büroaufgaben weiter erledigen können.

Wir bitten um Verständnis für diese Maßnahmen. Uns ist bewusst, dass es im Einzelfall zu Unannehmlichkeiten führen wird. Wir sehen aber derzeit keine andere Möglichkeit, um unsere Kräfte bestmöglich auf die wesentlichen Vereinsangelegenheiten zu konzentrieren.

Martin Sievers (Vorsitzender)

Uwe Ziegenhagen (stellv. Vorsitzender)

Bretter, die die Welt bedeuten

L^AT_EX und Schulphysik 1: Größen und Einheiten

Keno Wehr

Der erste Teil der Artikelreihe zur Schulphysik geht auf die typographischen Grundregeln für Formelzeichen und Maßeinheiten ein, erläutert die Anwendung des Pakets `siunitx`, das die korrekte Formatierung von Größen und Einheiten sicherstellt, stellt das Paket `schulma-physik` vor, das Ergänzungen dazu bietet und auch einige Naturkonstanten definiert, und geht schließlich auf die Besonderheiten griechischer Formelzeichen ein.

Die Schulphysik stellt den Lehrer vor spezifische typographische Herausforderungen. Neben die Problematik der korrekten Darstellung physikalischer Größen, Formelzeichen, Maßeinheiten und Teilchensymbole tritt das Erfordernis vielfältiger graphischer Repräsentation von Sachverhalten in Messwertediagrammen, Schaltbildern, Versuchsskizzen und anderen Darstellungen. Diese Artikelreihe will einige Hinweise geben, wie mit Hilfe geeigneter L^AT_EX-Pakete die vielfältigen Anforderungen erfüllt und ansehnliche Unterrichtsmaterialien vorbereitet werden können. Die erste Folge beschäftigt sich mit dem grundlegenden Thema der physikalischen Größen und Einheiten.

Formelzeichen und Einheitensymbole

Wesentlich für die physikalische Typographie ist die Unterscheidung zwischen Formelzeichen und Einheitensymbolen. Formelzeichen bezeichnen eine physikalische Größe und werden nach internationaler Norm und gängiger Praxis wie mathematische Variablen kursiv gesetzt, Einheitensymbole als Kürzel für Maßeinheiten wie mathematische Konstanten aufrecht; siehe z. B. [2] oder [1] für eine genauere Darstellung.

Die Konvention trägt in vielen Fällen dazu bei, die Bedeutung eines Zeichens zu klären. So steht *m* für eine Masse, m dagegen für die Einheit Meter, *s* für einen Weg, s für die Einheit Sekunde, *V* für ein Volumen, V für die Einheit Volt.

Während diese typographische Unterscheidung in Lehrbüchern konsequent zum Einsatz kommt, findet man sie in etlichen Privatdrucken – Anleitungen zu L^AT_EX-Paketen nicht ausgenommen – vernachlässigt. Oft werden Einheitensymbole fälschlich kursiv gesetzt, was wohl daran liegt, dass der Mathematikmodus von T_EX und anderen Satzprogrammen Buchstaben ohne weitere Auszeichnung als Variablen ansieht. Aufgrund der Semantik sollte die Unterscheidung von Aufrecht- und Kursivdruck sorgfältig durchgeführt werden.

Für Formelzeichen genügt der Wechsel in den Mathematikmodus:

Wir messen den Weg s und die Zeit t .
 Der Quotient $\frac{s}{t}$ ist die
 $\hookrightarrow$ Geschwindigkeit v .

Wir messen den Weg s und
 die Zeit t . Der Quotient $\frac{s}{t}$ ist
 die Geschwindigkeit v .

Die angemessene Lösung für Maßeinheiten, physikalische Größen (Maßzahl mit Maßeinheit) und auch dimensionslose Größen ist das Paket `siunitx`, das im nächsten Abschnitt erläutert wird.

Wenn auch die Unterscheidung zwischen Aufrecht- und Kursivdruck für den Leser hilfreich ist, so sollte man sich doch nicht der Illusion hingeben, dass dadurch alle Zweideutigkeiten beseitigt wären. Das Formelzeichen p kann für einen Druck oder einen Impuls stehen, h für das plancksche Wirkungsquantum oder eine Höhe, c für die Ausbreitungsgeschwindigkeit einer Welle (z. B. die Lichtgeschwindigkeit), für eine spezifische Wärmekapazität oder (in der Chemie) für eine Stoffmengenkonzentration, um nur einige Beispiele zu nennen.

Einheitensymbole sind zwar eindeutig, sofern man sich streng auf das SI beschränkt – doch das ist nicht immer durchzuhalten. Die Nicht-SI-Einheit Ar wird wohl in jedem mathematischen Unterstufenlehrbuch beim Thema Flächeninhalte eingeführt. Ebenso wenig wegzudenken ist die Nicht-SI-Einheit Jahr, die bei der Angabe von Halbwertszeiten in der Kernphysik vielfach Verwendung findet. Das eingeführte Einheitensymbol ist in beiden Fällen a , sodass dessen Bedeutung nur aus dem Kontext erschlossen werden kann. Weiter verwenden manche Nuklidkarten für Minuten das Symbol m , was dort natürlich nicht als Meter zu lesen ist.

Zudem sei auf die mögliche Verwechslung von Einheitenvorsätzen und Einheitensymbolen hingewiesen. Die typographische Regel, die hier Eindeutigkeit herstellt, ist die Verwendung eines kleinen Leerzeichens bei Produkten von Einheiten, nicht hingegen nach Einheitenvorsätzen. So steht mN für Millinewton, m N dagegen für »Meter mal Newton«, also Newtonmeter, wofür natürlich praktisch immer die umgekehrte Reihenfolge der Einheitensymbole verwendet wird, bei der keine Verwechslungsgefahr besteht. Entsprechendes gilt für mA (Milliampere) und m A (Meter mal Ampere). Weitere denkbare Beispiele wie THz für Terahertz vs. T Hz

für »Tesla mal Hertz« oder hA für Hektoampere vs. h A für Amperestunden sind von geringer praktischer Relevanz.

Physikalische Größen und Einheiten mit siunitx

Das Paket siunitx ermöglicht eine konsistente und typographisch korrekte Darstellung von Maßeinheiten und Größen. Anderen Paketen mit vergleichbarer Zielsetzung ist es aufgrund seiner umfänglichen Formatierungsfähigkeiten, seiner Anpassbarkeit und der Möglichkeit zur Lokalisierung unbedingt vorzuziehen.

Zwar ist die Schreibweise der Einheiten supranational, doch gibt es bei der Schreibweise der Zahlenwerte lokale Unterschiede. Was anderswo 6.25×10^{12} heißt, ist bei uns $6,25 \cdot 10^{12}$. In der Präambel sollte man daher zunächst global die deutsche Lokalisierung auswählen:

```
\sisetup{locale=DE}
```

Der zentrale Befehl des Pakets, welcher in Version 3.0 umbenannt wurde, ist `\qty` für physikalische Größen. Er kann innerhalb und außerhalb mathematischer Formeln verwendet werden. Als Argumente sind der Zahlenwert und die Einheit anzugeben. Für den Zahlenwert sind Dezimalkomma und Dezimalpunkt gleichermaßen möglich (das Ausgabeformat richtet sich nach der gewählten Lokalisierung), bei Verwendung der wissenschaftlichen Zahlschreibweise ist der Exponent durch e einzuleiten. Für jede SI-Einheit stellt das Paket eine Befehlssequenz zur Verfügung.

<code>\qty{1,254}{\metre}\</code>	1,254 m
<code>\qty{5,81e7}{\kilogram}\</code>	5,81 · 10 ⁷ kg
<code>\qty{86400}{\second}\</code>	86 400 s
<code>\qty{0,00836}{\newton}\</code>	0,008 36 N

Zwischen Zahlenwert und Einheit wird ein kleines, umbruchgeschütztes Leerzeichen gesetzt. Aus dem e in der Eingabe wird eine entsprechende Zehnerpotenz gemacht. Wenn es mehr als vier Stellen vor oder nach dem Komma gibt, wird in Dreiergruppen gegliedert.

Die Eingabe der Einheiten ist in dieser Form allerdings recht umständlich. Daher stellt das Paket Kurzbefehle zur Verfügung, die dem Einheitensymbol mit vorge-setztem Befehlsstrich entsprechen:

<code>\qty{1,254}{\m}\</code>	1,254 m
<code>\qty{5,81e7}{\kg}\</code>	5,81 · 10 ⁷ kg
<code>\qty{86400}{\s}\</code>	86 400 s
<code>\qty{0,00836}{\N}\</code>	0,008 36 N

Wem auch das noch zu aufwendig ist, der kann die Befehlsstriche bei den Einheiten auch weglassen, da `siunitx` für die Einheit auch einen freien Eingabemodus unterstützt. Dann entfällt allerdings die Kontrolle, ob die Einheit existiert, und bei der Formatierung zusammengesetzter Einheiten ist man auf sich gestellt.

Natürlich werden auch Einheitenvorsätze unterstützt. Mit entsprechenden Langbefehlen lassen sich beliebige Kombinationen bilden:

<code>\qty{25,4}{\pico\metre}\</code>	25,4 pm
<code>\qty{810,3}{\micro\gram}\</code>	810,3 µg
<code>\qty{3,18}{\milli\second}\</code>	3,18 ms
<code>\qty{733}{\mega\newton}\</code>	733 MN
<code>\qty{100}{\kilo\ohm}\</code>	100 kΩ
<code>\qty{550}{\tera\hertz}</code>	550 THz

Für gängige Kombinationen von Einheitenvorsatz und Einheit gibt es Kurzbefehle, wobei »mikro« durch ein `u` angegeben wird:

<code>\qty{25,4}{\pm}\</code>	25,4 pm
<code>\qty{810,3}{\ug}\</code>	810,3 µg
<code>\qty{3,18}{\ms}\</code>	3,18 ms
<code>\qty{733}{\MN}\</code>	733 MN
<code>\qty{100}{\kohm}\</code>	100 kΩ
<code>\qty{550}{\THz}</code>	550 THz

Für exotische Kombinationen wie »Mikronewton«, »Megameter« oder »Kilosekunden« gibt es keine Kurzbefehle. Falls solche wirklich einmal nötig sind, kann die Langform oder die freie Eingabe genutzt werden. Außerdem ist es möglich, neue Einheitenbefehle zu definieren; dazu unten mehr.

Natürlich braucht man auch Potenzen, Produkte und Quotienten von Einheiten. Für die zweite Potenz kann `\square` vor die Einheit gestellt werden, für die dritte Potenz `\cubic`. Höhere Potenzen erlaubt ein nachgestelltes `\tothe` mit Angabe des Exponenten. Für Produkte reiht man die Einheitenbefehle einfach aneinander, für Quotienten dient der Befehl `\per`.

```

$V=\qty{0,73}{\cubic\m}$\\
$A=\qty{5e-4}{\square\m}$\\
$h=\qty{4,14e-15}{\eV\s}$\\
$v=\qty{5,28e6}{\m\per\s}$\\
$\omega=\qty{6,9}{\per\s}$\\
$p=\qty{1,32e-23}{\kg\m\per\s}$\\
$G=\qty{6,67e-11}{\cubic\m\per\kg\per\square\s}$
↪$\\
$\sigma=\qty{5,67e-8}{\W\per\square\m\per\K}$
↪tothe{4}$

```

$$\begin{aligned}
 V &= 0,73 \text{ m}^3 \\
 A &= 5 \cdot 10^{-4} \text{ m}^2 \\
 h &= 4,14 \cdot 10^{-15} \text{ eV s} \\
 v &= 5,28 \cdot 10^6 \text{ m s}^{-1} \\
 \omega &= 6,9 \text{ s}^{-1} \\
 p &= 1,32 \cdot 10^{-23} \text{ kg m s}^{-1} \\
 G &= 6,67 \cdot 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2} \\
 \sigma &= 5,67 \cdot 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}
 \end{aligned}$$

Bei der Darstellung der Quotienteneinheiten fällt auf, dass die Nenner mit Hilfe negativer Exponenten ausgedrückt werden. Mag dies in der Sekundarstufe II aus wissenschaftspropädeutischen Gründen wünschenswert sein, so sind für Lerngruppen der Sekundarstufe I, denen die entsprechende Potenzdefinition noch nicht bekannt ist, Brüche der Form m/s oder $\frac{\text{m}}{\text{s}}$ erforderlich. Hierfür bietet siunitx die Option `per-mode`, die die drei Werte `symbol` (für einen Schrägstrich), `fraction` (für einen Bruchstrich) und `power` (für die Standarddarstellung mit negativen Exponenten) kennt.

Optionen können einem einzelnen `\qty`-Befehl übergeben werden.

```
\qty[per-mode=fraction]{27,8}{\m\per\s}
```

$$27,8 \frac{\text{m}}{\text{s}}$$

In der Regel wird man sie allerdings dem Befehl `\sisetup` übergeben, um sie global festzulegen.

```

\sisetup{per-mode=fraction}
$v=\qty{5,28e6}{\m\per\s}$\\
$\omega=\qty{6,9}{\per\s}$\\
$p=\qty{1,32e-23}{\kg\m\per\s}$\\
$G=\qty{6,67}{\cubic\m\per\kg\per\square\s}$

```

$$\begin{aligned}
 v &= 5,28 \cdot 10^6 \frac{\text{m}}{\text{s}} \\
 \omega &= 6,9 \frac{1}{\text{s}} \\
 p &= 1,32 \cdot 10^{-23} \frac{\text{kg m}}{\text{s}} \\
 G &= 6,67 \frac{\text{m}^3}{\text{kg s}^2}
 \end{aligned}$$

Auch Messungenauigkeiten können angegeben werden, wobei die Ungenauigkeit der letzten Stellen in Klammern zu setzen ist. Da die Standardausgabe nicht den schulischen Gepflogenheiten entspricht, sollte man die Option `uncertainty-mode=separate` wählen.

```

\sisetup{uncertainty-mode=separate}
$\lambda=\qty{637,2(67)}{\nm}$\\
$h=\qty{6,43(38)e-34}{\J\s}$

```

$$\begin{aligned}
 \lambda &= (637,2 \pm 6,7) \text{ nm} \\
 h &= (6,43 \pm 0,38) \cdot 10^{-34} \text{ J s}
 \end{aligned}$$

Neben dem bisher vorgestellten Befehl `\qty` gibt es `\num` für dimensionslose Größen und `\unit` für Einheiten ohne Zahlenwert, deren Argument dem ersten bzw. zweiten Argument von `\qty` entspricht. Für Winkel in Grad ist der Befehl `\ang` hilfreich.

```
\num{6,24e18} Elektronen\\
$g$ in \unit{\um}\\
\ang{35}
```

$6,24 \cdot 10^{18}$ Elektronen
 g in μm
 35°

Die beschriebenen Optionen für Brucheinheiten erlauben für die Impulseinheit die Darstellungen kg m s^{-1} , kg m/s und $\frac{\text{kg m}}{\text{s}}$. Wenn aus didaktischen Gründen betont werden soll, dass der Impuls ein Produkt aus Masse und Geschwindigkeit ist, ist ein kleiner Kunstgriff nötig:

```
\qty{2,5}{\kg}\,, \unit[per-mode=fraction]{\m\per
\c{s}}
```

$2,5 \text{ kg } \frac{\text{m}}{\text{s}}$

Das `siunitx`-Paket unterstützt von Hause aus nur SI-Einheiten sowie »zur Verwendung mit dem SI zugelassene Einheiten«. Zu letzteren zählen beispielsweise die Minute, die Stunde, der Tag, das Elektronenvolt, das Hektar und der Liter, nicht hingegen das Jahr, das Lichtjahr, das Ar und das Bar – Einheiten, auf die man in bestimmten Zusammenhängen nicht gern verzichten wird und für die die gleichen typographischen Regeln gelten sollten.

Neue Einheiten können mit dem Befehl `\DeclareSIUnit` definiert werden, dessen Name paradox anmutet, da er ja gerade für *Nicht*-SI-Einheiten gebraucht wird. Für die Kernphysik könnte etwa Folgendes nützlich sein:

```
\DeclareSIUnit\{a}
$T_{1/2}=\qty{1600}{\a}$\\
$\lambda=\frac{\ln 2}{T_{1/2}}$
=\qty{4,33e-4}{\per\a}$
```

$T_{1/2} = 1600 \text{ a}$
 $\lambda = \frac{\ln 2}{T_{1/2}} = 4,33 \cdot 10^{-4} \text{ a}^{-1}$

Bei der Darstellung der Einheitensymbole gibt es eigentlich keinen Entscheidungsraum. Eine Ausnahme ist die Einheit Liter, die statt mit *l* auch mit *L* dargestellt werden darf, um Verwechslungen zu vermeiden. `siunitx` sieht die Repräsentation durch einen Großbuchstaben vor, was sich auch auf Milli- und Mikroliter auswirkt. Um stattdessen den regulären Kleinbuchstaben zu erhalten, empfiehlt der Paketautor ebenfalls den Befehl `\DeclareSIUnit`. Dieser kann auch verwendet werden, um die gelegentlich anzutreffende, aber nicht normgerechte Darstellung durch ein geschwungenes *ℓ* zu realisieren.

<code>\qty{3,48}{l}\</code>	3,48 L
<code>\qty{450}{\ml}\</code>	450 mL
<code>\DeclareSIUnit\litre{1}</code>	
<code>\qty{3,48}{l}\</code>	3,48 l
<code>\qty{450}{\ml}\</code>	450 ml
<code>\DeclareSIUnit\litre{\ensuremath{\ell}}</code>	3,48 ℓ
<code>\qty{3,48}{l}\</code>	450 mℓ
<code>\qty{450}{\ml}</code>	

Physikalische Größen und Einheiten mit `schulma-physik`

Das Paket `schulma-physik` aus dem `Schulmathematik`-Bündel stellt keine Alternative zu `siunitx` dar, sondern bietet einige Ergänzungen, die sich in meiner täglichen Praxis als hilfreich herausgestellt haben.

Ein Problem, das damit angegangen wird, ist die Darstellung von Brucheinheiten in eingerückten Formeln. `TEX` stellt bekanntlich gemeine Brüche in eingerückten Formeln (*display style*) größer als im Fließtext (*text style*) dar. Dies überträgt sich auf Maßeinheiten, wenn die `siunitx`-Option `per-mode=fraction` gewählt ist.

<pre>\setup{per-mode=fraction} \[v=\frac{s}{t}=\frac{\qty{20}{m}}{\qty{5}{s}} =\qty{4}{\mper\s}]</pre>	$v = \frac{s}{t} = \frac{20 \text{ m}}{5 \text{ s}} = 4 \frac{\text{m}}{\text{s}}$
--	--

Für die Rechenschritte ist die große Bruchdarstellung gut, für die Einheit am Ende aber zu bombastisch. Auf elementarem Wege ist Abhilfe durch die Umschaltung auf den *text style* an passender Stelle möglich.

<pre>\setup{per-mode=fraction} \[v=\frac{s}{t}=\frac{\qty{20}{m}}{\qty{5}{s}} =\textstyle\qty{4}{\mper\s}]</pre>	$v = \frac{s}{t} = \frac{20 \text{ m}}{5 \text{ s}} = 4 \frac{\text{m}}{\text{s}}$
--	--

Eine sauberere Lösung ist die Verwendung der Option `fraction-command`. Das Paket `amsmath` stellt für das kontextunabhängige Setzen von großen bzw. kleinen Brüchen die Befehle `\dfrac` und `\tfrac` zur Verfügung. Wenn dieses geladen ist, ist Folgendes möglich.

<pre>\setup{per-mode=fraction, fraction-command=\tfrac} \[v=\frac{s}{t}=\frac{\qty{20}{m}}{\qty{5}{s}} =\qty{4}{\mper\s}]</pre>	$v = \frac{s}{t} = \frac{20 \text{ m}}{5 \text{ s}} = 4 \frac{\text{m}}{\text{s}}$
---	--

Als Alternative zu dieser Optionenwahl tritt nun das Paket `schulma-physik` mit den Befehlen `\qty` und `\tunit` auf den Plan. Diese setzen Einheiten immer als kleinen Bruch. Das vorangestellte `t` im Befehlsnamen bezieht sich dabei auf den *text style*.

```
\[ \qty{4}{\m\per\s}\quad
\qty[per-mode=fraction]{4}{\m\per\s}\quad
\tqty{4}{\m\per\s}\]
```

$$4 \text{ m s}^{-1} \quad 4 \frac{\text{m}}{\text{s}} \quad 4 \frac{\text{m}}{\text{s}}$$

```
\[ \unit{\N\per\A\per\m}\quad
\unit[per-mode=fraction]{\N\per\A\per\m}\quad
\tunit{\N\per\A\per\m}\]
```

$$\text{N A}^{-1} \text{ m}^{-1} \quad \frac{\text{N}}{\text{A m}} \quad \frac{\text{N}}{\text{A m}}$$

`schulma-physik` lädt `siunitx` mit den Optionen `locale=DE` und `uncertainty-mode=separate` (seit Version 1.3). Neben den bereits vorgestellten Befehlen `\qty` und `\tunit` gibt es `\Beschl` für Beschleunigungen in $\frac{\text{m}}{\text{s}^2}$.

```
\Beschl{1,83}
```

$$1,83 \frac{\text{m}}{\text{s}^2}$$

Schließlich steht zur bequemen Angabe von Maßstäben noch der Befehl `\Masstab` zur Verfügung, dessen vier Argumente Zahlen bzw. Maßeinheiten in `siunitx`-Syntax sind.

```
\Masstab{1}{\cm}{0,5}{\N}
```

$$1 \text{ cm} \hat{=} 0,5 \text{ N}$$

Naturkonstanten

Naturkonstanten kommen (in angemessen gerundeter Form) häufig im Physikunterricht vor. Daher liegt es nahe, ihre Werte mit Hilfe eines geeigneten Pakets ausgeben zu lassen, sodass die Zahlenwerte und Maßeinheiten nicht jedes Mal neu eingegeben werden müssen. Wünschenswert wäre ein Paket, das die Ausgabe mit einer frei wählbaren Anzahl gültiger Stellen gestattet. Ein solches ist derzeit nicht verfügbar.

Das Paket `physconst` bietet eine umfangreiche Sammlung von Naturkonstanten in unterschiedlichen Einheitensystemen und gibt wahlweise den genauesten bekannten Wert oder einen auf drei gültige Stellen gerundeten Wert aus. Leider ist das Paket für deutschsprachige Dokumente unbrauchbar, da nur die englische Schreibweise unterstützt wird. Eine Lokalisierung wäre unter Rückgriff auf `siunitx` leicht zu haben gewesen, war aber aus Sicht des US-amerikanischen Autors wohl entbehrlich.

Eine Auswahl von acht Naturkonstanten wird vom Paket `schulma-physik` bereitgestellt, das intern auf `siunitx` zurückgreift (Tabelle 1). Die Konstanten werden mit

Befehl	Ausgabe	Befehl	Ausgabe
<code>\Erdb</code>	$9,81 \frac{\text{m}}{\text{s}^2}$	<code>\MFK</code>	$4\pi \cdot 10^{-7} \text{ V s A}^{-1} \text{ m}^{-1}$
<code>\Ortsf</code>	$9,81 \frac{\text{N}}{\text{kg}}$	<code>\MFK*</code>	$1,26 \cdot 10^{-6} \text{ V s A}^{-1} \text{ m}^{-1}$
<code>\Elem</code>	$1,60 \cdot 10^{-19} \text{ C}$	<code>\Lichtg</code>	$3,00 \cdot 10^8 \text{ m s}^{-1}$
<code>\Elekm</code>	$9,11 \cdot 10^{-31} \text{ kg}$	<code>\Planck</code>	$6,63 \cdot 10^{-34} \text{ J s}$
<code>\EFK</code>	$8,85 \cdot 10^{-12} \text{ A s V}^{-1} \text{ m}^{-1}$		

Tab. 1: vom Paket `schulma-physik` definierte Naturkonstanten

drei gültigen Stellen definiert. Für die Erdbeschleunigung und den Ortsfaktor, die bereits in der Sekundarstufe I anzutreffen sind, wird die Einheit als Bruch dargestellt, bei anderen Konstanten der Einstellung von `siunitx` gemäß.

<code>\EFK\</code> <code>\sisetup{per-mode=fraction}</code> <code>\EFK</code>	$8,85 \cdot 10^{-12} \text{ A s V}^{-1} \text{ m}^{-1}$ $8,85 \cdot 10^{-12} \frac{\text{A s}}{\text{V m}}$
---	--

Das Physikmodul des `schule`-Pakets definiert ebenfalls einige Naturkonstanten. Zum Zeitpunkt der Erstellung dieses Artikels war allerdings keine fehlerfreie Übersetzung des entsprechenden Codes möglich.

Griechische Buchstaben als Formelzeichen

Die eingangs erwähnte Konvention des Kursivsatzes von Formelzeichen gilt nicht nur für lateinische, sondern auch für griechische Buchstaben. Bei Kleinbuchstaben ist dies kein Problem, da es dem Standardverhalten von $\TeX$ im Mathematikmodus entspricht.

Griechische Großbuchstaben setzt $\TeX$ hingegen normalerweise auch im Mathematikmodus nicht kursiv. In der Schulphysik betrifft das lediglich den Großbuchstaben Phi, der in der Oberstufe den magnetischen Fluss bezeichnet. Sofern man sich mit den Standard-Schriften von $\TeX$ begnügt, kann man mit `\mathit{\Phi}` sehr einfach ein kursives Groß-Phi erzeugen.

Wenn man eine andere Schrift einstellt, kann es jedoch sein, dass das nicht mehr funktioniert, da das entsprechende Zeichen nicht zur Verfügung gestellt wird. Ein möglicher Ersatz ist der Befehl `\varPhi` aus dem Paket `amsmath`. In der Regel ist man aber mit dem Befehl `\symit` aus dem Paket `unicode-math` besser beraten, da dieser das Zeichen aus der aktuellen Mathematikschrift nimmt.

$$\Phi = 2,8 \cdot 10^{-4} \text{ Wb}$$

$$\Phi = 2,8 \cdot 10^{-4} \text{ Wb}$$

Das pauschale Kursivstellen aller griechischen Großbuchstaben in mathematischen Formeln wäre mit Hilfe des Pakets `isomath` oder der `unicode-math`-Option `math-style=ISO` möglich. Ich halte es aber für die Schulphysik nicht für hilfreich, da dort häufig die Änderung einer Größe durch ein griechisches Delta beschrieben wird, das *nicht* kursiv gesetzt wird.

$$v = \frac{\Delta s}{\Delta t}$$

$$U_{\text{ind}} = N \cdot \frac{\Delta \Phi}{\Delta t}$$

$$v = \frac{\Delta s}{\Delta t}$$

$$U_{\text{ind}} = N \cdot \frac{\Delta \Phi}{\Delta t}$$

Mit `math-style=ISO` könnte man zwar den Befehl `\symbit` einsparen, müsste dann aber `\Delta` durch `\increment` ersetzen, damit das Symbol nicht kursiv gesetzt wird.

Zu den Kleinbuchstaben ist zu bemerken, dass in deutschsprachigen Schulbüchern die von D. Knuth als Varianten qualifizierten Formen von Epsilon, Theta und Phi üblich sind. Daher sollten für diese Buchstaben die Befehle `\varepsilon`, `\vartheta` und `\varphi` verwendet werden.

$$\varepsilon_0 = 8,85 \cdot 10^{-12} \text{ A s V}^{-1} \text{ m}^{-1}$$

$$\vartheta = 20^\circ \text{C}$$

$$\varphi = 35^\circ$$

$$\varepsilon_0 = 8,85 \cdot 10^{-12} \text{ A s V}^{-1} \text{ m}^{-1}$$

$$\vartheta = 20^\circ \text{C}$$

$$\varphi = 35^\circ$$

Literatur

- [1] Walter Entenmann: »ISO-80000 konformer Mathematiksatz mit Lua $\text{\LaTeX}$ «, *DTK*, 31.3 (2019), 28–42.
- [2] Moritz Nadler: ISO-31-konformer Formelsatz in $\text{\LaTeX}$, Version 1.0, 2018, <http://www.moritz-nadler.de/formelsatz.pdf>.

Buchumschläge mit Con $\text{\TeX}$ t

Henning Hraban Ramm

Ein Buch setzen kann jede*r $\text{\TeX}$ -Anwender*in, der Umschlag dazu ist eine größere Herausforderung. Bei der Gelegenheit lernen wir, mit Dimensionen zu rechnen.

Bisher habe ich die Umschläge für meine in ConT_EXt gesetzten Bücher meist in einem Grafik- oder Bildbearbeitungsprogramm gestaltet. Für Titellentwürfe halte ich das auch immer noch für den sinnvollsten Weg, weil sich Ideen oder Änderungswünsche visuell direkter und schneller umsetzen lassen als mit jeder Programmierung. Wenn es dann aber um den kompletten Umschlag mit Rücken geht, muss ich bei jeder Änderung des Umfangs in der Grafik herumbasteln. Das geht einfacher mit etwas Code.

Es wäre möglich, den kompletten Titellentwurf als Bild einzubinden, und je nach Gestaltung ist das sicherlich sinnvoll, hier bauen wir ihn aber nach.

Fangen wir mal ganz einfach an: Wir definieren eine Papiergröße und eine Ebene (Layer) über die ganze Seite, auf der wir dann unsere Elemente platzieren können:

```
\definepapersize[Cover][width=350mm,height=240mm]
\setuppapersize[Cover]

\definelayer[cover][
  x=0mm,y=0mm,
  width=\paperwidth,
  height=\paperheight,
]
\setupbackgrounds[page][background=cover,state=start]

\starttext
\setlayer[cover][
  x=200mm,
  y=20mm,
]{\ss\bfd Mein Buchtitel}
\strut
\stoptext
```

Den `\strut` am Ende brauchen wir, damit die Seite einen Inhalt hat, sonst gibt ConT_EXt auch den Inhalt der Ebene nicht aus.

Woher kommen jetzt aber die Maße? Ich habe mich an der Bücherreihe von DAN-TE e.V. orientiert, deren Seitengröße 165×240 mm ist. Der Umschlag ist zweimal 165 mm breit, dazu kommen 20 mm Rücken. Vielleicht hat unsere Druckerei uns das so mitgeteilt, aber flexibel ist es nicht. Können wir das nicht von T_EX berechnen lassen?

```
\definepapersize[Cover][
  width={2 * 165mm + 20mm},
  height=240mm,]
```

Schön wär's. ConT_EXt meint »Illegal unit of measure«. So einfach ist es also nicht. Wenn ConT_EXt mit Maßangaben rechnen soll, müssen wir einen »Dimensions-Ausdruck« verwenden, und der muss immer mit einer Dimension beginnen:

```
\definepapersize[Cover][
  width=\dimexpr 165mm * 2 + 20mm\relax,
  height=240mm,]
```

Müssen wir jetzt für jedes Element auf der Titelseite so umständlichen Code angeben? Natürlich nicht. Wir können die wichtigsten Werte vordefinieren. Am einfachsten als Makro:

```
\def\Seitenbreite{165mm}
\def\Rückenbreite{20mm}

\definepapersize[Cover][
  width=\dimexpr \Seitenbreite * 2 + \Rückenbreite\relax,
  height=240mm]
```

Das funktioniert hier sogar, aber es ist »sauberer« und zuverlässiger, wenn wir Maße als Dimensionen deklarieren. Unter ϵ -T_EX ging das so:

```
\newdimen\Seitenbreite
\Seitenbreite=165mm
```

Das funktioniert, aber wenn wir mit ConT_EXt arbeiten, soll es auch nach ConT_EXt aussehen:

```
\definemeasure[Seitenbreite][165mm]
```

Der so definierte Wert lässt sich auf zwei Arten abrufen: mit `\measure` als String für Wertzuweisungen in Setup-Befehlen oder mit `\measured` als Dimension für Berechnungen.

Im Argument von `\definemeasure` können wir Berechnungen ausführen, ohne `\dimexpr` anzugeben – intern wird es trotzdem verwendet, daher haben die Ausdrücke die gleichen Beschränkungen:

```
\definemeasure[Coverbreite][2\measured{Seitenbreite} + 20mm]
```

Oh, der Ausdruck fängt ja gar nicht mit einer Dimension an? Einfache Faktoren wie hier sind möglich, auch mit Kommazahlen (float) – ansonsten sind nur ganze Zahlen (Integer) als Faktoren erlaubt, da muss man manchmal mit Brüchen tricksen: `*1.5` führt zu einem Fehler, aber `*3/2` funktioniert.

Nochmal komplett:

```

\definemeasure[Seitenbreite][165mm]
\definemeasure[Seitenhöhe][240]
\definemeasure[Rückenbreite][20mm]
\definemeasure[Coverbreite][2\measured{Seitenbreite} + \measured{Rückenbreite}]

\definepapersize[Cover][
  width=\measure{Coverbreite},
  height=\measure{Seitenhöhe}]
\setuppapersize[Cover]

\definelayer[Cover][
  x=0mm,y=0mm,
  width=\paperwidth,
  height=\paperheight,]
\setupbackgrounds[page][background=cover,state=start]

\starttext
\setlayer[Cover][
  x=\dimexpr\measured{Seitenbreite} + \measured{Rückenbreite} + 15mm\relax,
  y=20mm,
]{\ss\bfd Mein Buchtitel}
\strut
\stoptext

```

Wenn sich jetzt die Seitenzahl des Inhalts ändert, brauchen wir nur das Maß für die Rückenbreite anpassen. Wir können diese aber auch automatisch berechnen lassen:

```

\useexternalfigure[inhalt][buch.pdf]
\getfiguredimensions[inhalt]
\expanded{\definemeasure[Rückenbreite][2mm + (0.09mm * 3/2 * \nofigurepages/2)]}

```

Zunächst ermitteln wir die Seitenzahl des Buchblocks, indem wir die »Bilddimensionen« des PDFs abfragen. Dadurch steht die Seitenzahl in `\nofigurepages`. Dieser Wert ändert sich, sobald das nächste Bild geladen wird, im Normalfall auf 1. Die Berechnung wird aber erst ausgeführt, wenn sie gebraucht wird. Da wir den momentanen Wert benötigen, müssen wir die Definition sofort »expandieren«.

Wie kommen wir aber auf diese Formel? Wir brauchen die Anzahl der Blätter – meine Berufsschullehrer pflegten uns zu erinnern: »Papier hat zwei Seiten«. Unser Buch wird auf Werkdruckpapier mit einem Flächengewicht (Grammatur) von 90 g/m² und 1,5-fachem Volumen (Füllstoffe) gedruckt. Bei einfachem Volumen ist jedes Blatt 0,09 mm dick, das Volumen ist ein Faktor ohne Maßeinheit. Diese Papiereigenschaften könnten wir natürlich auch noch in Dimensionen ablegen.

Die 2 mm sind ein Falz-Zuschlag, je 1 mm für die beiden Falze am Rücken – pauschal geschätzt; genau genommen müssten wir die Stärke des Umschlagkartons berücksichtigen.

Jetzt passt sich unser Umschlag also automatisch an den Umfang des Inhalts an. Nicht schlecht, oder?

Nun wollen wir aber nicht nur den Buchtitel auf die erste Umschlagseite platzieren, sondern auch noch Untertitel und Autor*in, ebenso auf den Rücken; auf die Rückseite sollen Klappentext und ISBN-Barcode. Müssen wir dabei immer wieder die gleichen umständlichen Berechnungen angeben?

Wir können uns für Vorder- und Rückseite sowie den Rücken eigene Ebenen definieren, dann brauchen wir nur noch den Versatz relativ zum jeweiligen Element angeben:

```
% ...
\definelaye[rU4][ % Rückseite
  hoffset=0mm,
  y=0mm,
  width=\measure{Seitenbreite},
  height=\measure{Seitenhöhe},
]
\definemeasure[StartVorne][\measured{Seitenbreite} + \measured{Rückenbreite}]
\definelaye[rU1][ % Vorderseite
  hoffset=\measure{StartVorne},
  y=0mm,
  width=\measure{Seitenbreite},
  height=\measure{Seitenhöhe},
]
\definelaye[rRücken][
  hoffset=\measure{Seitenbreite},
  y=0mm,
  width=\measure{Rückenbreite},
  height=\measure{Seitenhöhe},
]
\setupbackgrounds[page][background={Cover,U4,U1,Rücken},state=start]
% ...
\setlayer[rU1][
  x=15mm,
  y=20mm,
]{\ss\bfd Mein Buchtitel}
\setlayerframed[rRücken][
  y=12mm,
  offset=overlay,
```

```

frame=off,
align=flushleft,
width=\measure{Rückenbreite},
height=0.66\measured{Seitenhöhe},
]{%
  \rotate[
    rotation=90,
    height=\measure{Rückenbreite},
    width=0.66\measured{Seitenhöhe},
    align={lohi,flushright},
  ]{Autor: Titel}%
}

```

Für den Rückentext habe ich hier `\setlayerframed` verwendet, damit haben wir sämtliche Optionen von `\framed` zur Verfügung – gerade in der Entwurfsphase schalte ich gerne die Rahmen ein, um die Platzierung zu kontrollieren. Um diese Einstellung von der Kommandozeile aus steuern zu können, können wir einen Modus verwenden:

```

\setupframed[offset=overlay] % kein Abstand zum Rahmen
\startnotmode[debug]
  \setupframed[frame=off]
\stopnotmode

```

Genau genommen wird der Rahmen *ausgeschaltet*, wenn wir ConT_EXt *nicht* mit `--mode=debug` aufrufen. Die Option `offset=overlay` möchte ich allgemein nicht bei jedem Rahmen angeben.

Als Nächstes kümmern wir uns um den Klappentext. Wir können ihn mit `\setlayerframed[U4]` platzieren, aber ich finde es unübersichtlich, wenn ein längerer Text mitten zwischen den Platzierungsbefehlen steht. Also definieren wir ihn vorher als Puffer (Buffer):

```

\startbuffer[Klappentext]
\quotation{Ich habe nie ein besseres Buch gelesen!}\wordright{(M. Reich-Radecki)}
\blank[2*line]
Irgendwelches Geschwurbel zum Inhalt...
\stopbuffer

\startsetups[klappentext]
% Einstellungen zu Schrift, Ausrichtung, Einzug usw.
\stopsetups

% ...
\setlayerframed[U4][

```

```
x=.15\measured{Seitenbreite},
y=20mm,
width=.7\measured{Seitenbreite},
height=.8\measured{Seitenhöhe},
setups=klappentext,
]{\getbuffer[Klappentext]}
```

Nun gibt es ein paar Elemente, die wir mehrfach brauchen. Und es ist gar nicht so selten, dass auf Buchrücken ein falscher Titel oder Autor steht, weil das bei einer Änderung übersehen wurde. Definieren wir diese Texte doch einfach als Variablen und verwenden sie mehrmals:

```
\setvariables[buch][
  inhaltPdf={band01},% Name der Inhaltsdatei
  autor={Donald E. Knuth},
  titel={Das \TeX-Buch},
  undertitel={über befehlsbasierten Schriftsatz},
  reihe={Computer & Typographie},
  band={1},
  isbn={978-3-12345-007-Z},
  titelbild={mill},
]
% ...
\useexternalfigure[inhalt][\getvariable{buch}{inhaltPdf}]
\getfiguredimensions[inhalt]
% ...
\setlayer[U1][
  x=15mm,y=20mm,
  setups=haupttitel
]{\getvariable{buch}{titel}}
```

Die Variablen können wir also mit `\getvariable{buch}{titel}` abrufen. Makros wären natürlich ebenfalls möglich.

Da wir die Daten allerdings auch *im* Buch brauchen, lagern wir sie am besten in eine Einstellungsdatei (Environment) aus, die wir für Inhalt und Umschlag aktivieren:

```
\startenvironment settings
\project buchbuch

\setvariables[buch][
%...
]
\stopenvironment
```

Nächster Schritt: Wir bauen ein Hintergrundbild ein. Da es die ganze Seite abdecken soll, muss es »randabfallend« (auch: »angeschnitten«) angelegt sein, d. h. über den Papierrand hinausragen – sonst kann es zu weißen »Blitzern« kommen, wenn das Buch nicht ganz exakt beschnitten wird. (Das hat wenig mit der Arbeitsqualität der Druckerei bzw. Buchbinderei zu tun, sondern vor allem mit dem Papierverzug durch die Feuchtigkeit während der Verarbeitung.)

```
\definemeasure[Beschnitt][3mm]
\definemeasure[MaxHöhe][\measured{Seitenhöhe} + 2\measured{Beschnitt}]
\setlayerframed[U1][
  %x=-\measured{Beschnitt},
  y=-\measured{Beschnitt},
][{\externalfigure[\getvariable{buch}{titelbild}][height=\measure{MaxHöhe}]}
```

Wir könnten natürlich auch ein Hintergrundbild für den gesamten Umschlag (Rückseite, Rücken und Vorderseite) auf die Ebene »Cover« legen; die meisten Motive vertragen ein wenig Verzerrung, wenn sich der Umfang ändert. Typisch für ConT_EXt (und auch für ein Buch in der DANTE-Reihe) wäre eine METAPOST-Grafik als Hintergrund, dabei stehen die Variablen `OverlayWidth` und `OverlayHeight` zur Verfügung.

Nun ist das Bild zwar randabfallend, aber wir sehen nichts davon, denn unser Papier hat nur die beschnittene Größe. Wir könnten die Option `oversized` verwenden, das erweitert das Format ringsum um 7 mm:

```
\setuppapersize[Cover][Cover,oversized]
```

Da wir das Maß aber noch für eigene Berechnungen brauchen, machen wir das Gleiche lieber explizit:

```
\definemeasure[Rand][7.5mm]% Trim
\definemeasure[Coverbreite][2\measured{Seitenbreite} + \measured{Rückenbreite}]
\definemeasure[CoverbreitePlus][2\measured{Seitenbreite} + 2\measured{Rand} + \measured{Rückenbreite}]
\definemeasure[CoverhöhePlus][\measured{Seitenhöhe} + 2\measured{Rand}]

\definepapersize[Cover][
  width=\measure{Coverbreite},
  height=\measure{Seitenhöhe}]
\definepapersize[CoverPlus][
  width=\measure{CoverbreitePlus},
  height=\measure{CoverhöhePlus}]
\setuppapersize[Cover][CoverPlus]
```

Die Ebenen brauchen wir nicht anders zu definieren; Elemente werden an ihren Grenzen nicht abgeschnitten.

Dann wollen wir aber auch Schneidzeichen sehen, und bei der Gelegenheit können wir die unsichtbaren »Boxen« im PDF, die Endformat und Beschnitt definieren, richtig einstellen (kontrollieren lassen sie sich nur mit Acrobat¹).

```
\setuplayout[
  marking=on,% Schneidzeichen
  location=middle,% Seite auf dem Bogen zentrieren
  cropoffset=0mm,
  bleedoffset=\measure{Beschnitt},
  trimoffset=-\measure{Rand},
]
\setupinteractionscreen[width=max,height=max]
```

- Ein positiver Wert von `cropoffset` verkleinert die angezeigte Fläche und wirkt sich auch auf die anderen beiden Werte aus.
- Ein negativer Wert von `trimoffset` gibt den Versatz der TrimBox von der CropBox an.
- Ein positiver Wert von `bleedoffset` gibt den Beschnitt als Versatz der BleedBox von der TrimBox an.
- Ohne `\setupinteractionscreen` sind die Box-Einstellungen wirkungslos.

Durch die Box-Konfiguration brauchen wir die Lage der Ebenen und der Elemente darauf nicht verändern.

Noch ein paar Hinweise zu Dimensionsberechnungen:

Da Dimensionsausdrücke (`dimexpr`) geschachtelt werden dürfen, ist es manchmal sinnvoll, `\dimexpr... \relax` innerhalb von Berechnungen zu verwenden.

Intern rechnet T_EX mit ganzzahligen »scaled point« (sp) von 1/65536 pt. Der Maximalwert für Längen ist 16384 pt, etwa 5,75 m.

Wenn wir Dimensionswerte mit `\measure` ausgeben, werden sie in pt (T_EX-Punkt) angegeben. Eine Umrechnung in andere Maßeinheiten erreichen wir so:

```
\define[2]\Konv{\scratchedimen #1 \the\nodimen #2 \scratchedimen}
% erster Parameter: Dimension, zweiter Parameter: Maßeinheit
\Konv{1pt}{mm}
```

Zum Schluss:

Der Artikel bezieht sich nur auf Broschuren (Softcover). Für den Bezug von Pappbänden (Hardcover) brauchen Sie zusätzlich zum größeren Format mehr Beschnitt

¹ Voreinstellungen/Seitenanzeige: Objekt-, Endformat- und Anschnittrahmen einblenden

(ca. 15 mm), weil das Bezugspapier um die Pappe herumgeschlagen wird. Außerdem braucht der Rücken mehr Falzzuschlag (ca. 4–5 mm) für die »Scharniere«. Wir können aber auf den dargestellten Berechnungen aufbauen; auch für eine Klappenbroschur ist das Beispiel leicht zu erweitern.

ConT_EXt kurz notiert

Henning Hraban Ramm

In der ConT_EXt-Welt ist vieles in Bewegung, das eine Erwähnung wert ist, aber keinen ganzen Artikel begründet. (Stand: 15. 1. 2023)

Schrift-Details

Auf manchen älteren Druckern verschwand die gesamte Schrift beim Ausdruck von PDFs, die von allen PDF-Viewern richtig angezeigt wurden. Leah Neukirchen konnte das Problem auf eine merkwürdige Inkonsistenz in der PDF-Spezifikation bei der Behandlung von OpenType-Schriften zurückführen¹, wodurch der Text auf ein Tausendstel der Größe verkleinert und somit unsichtbar wurde. Nach einer Woche Debugging ließ sich das Problem durch das Weglassen einer Zeile Code beheben.

Die selten verwendeten Varianten `base` und `none` der Schriftbehandlung wurden neu implementiert (`node` ist gleich geblieben); Nebenwirkungen sind keine bekannt. Die Probleme bei der Ansteuerung von bestimmten Designachsen in variablen Fonts sind behoben.

Neu ist die Unterstützung für Farbinformationen nach COLRv1. Ein Beispiel dafür ist die in METAPOST definierte variable Malayalam-Schrift »Nupuram«² (auch ihre lateinischen Buchstaben sind sehr hübsch). Dafür war eine Erweiterung des SVG-Parsers nötig, der jetzt auch besser mit Verläufen umgehen kann. In Schriften definierte Farben sind grundsätzlich etwas fragwürdig – außer ConT_EXt LMTX scheint es kein Programm zu geben, mit dem sie sich »im Betrieb« (d. h. ohne die Schrift selbst zu modifizieren) verändern ließen –, aber »Nupuram« zeigt, was sich damit anstellen lässt.

¹<https://leahneukirchen.org/blog/archive/2022/10/50-blank-pages-or-black-box-debugging-of-pdf-rendering-in-printers.html>

²<https://gitlab.com/smc/fonts/Nupuram/>

Mathematiksatz

Hans Hagen und Mikael Sundqvist arbeiten weiterhin an der Optimierung des Mathematiksatzes. Die bisherigen Fortschritte sind in einem lesenswerten Artikel³ im TUGboat 2/2022 dargestellt. Seitdem wurde die Darstellung von Wurzeln und Ableitungsstrichen (*primes*) sowie der Weißraum-Ausgleich bei der Verwendung von `\unit` weiter verbessert.

Die »Anker« an Boxen eröffnen neue Möglichkeiten für die Kommentierung von Formeln mit Hilfe von METAPOST-Grafikelementen. Sogar seitenübergreifende Verbindungen sind möglich, wenn auch noch nicht ausgereift.

Anwender-Ebene

Leah Neukirchen hat die traditionellen L^AT_EX-Anführungszeichen als Ligatur-Feature germantextlig umgesetzt.

Seiten aus anderen ConTeXt-PDFs können jetzt auch anhand ihrer Abschnittsreferenz platziert werden – dazu wird allerdings auch deren `.tuc`-Datei benötigt⁴.

Docker

Ben Frank von *Island of T_EX* demonstrierte uns in einer Online-Sitzung, wie sich Docker-Images⁵ von ConTeXt in Continuous-Integration-Projekten einsetzen lassen.

Gegenüber den L^AT_EX-Images sind sie so viel kleiner, so dass er sich bisher noch nicht einmal die Mühe machte, die Dokumentation wegzulassen. Der Unterschied erklärt sich unter anderem dadurch, dass der ConTeXt-Kern weitgehend monolithisch ist, während die allermeisten L^AT_EX-Dokumente auf das Laden von umfangreichen Paketen angewiesen sind. Insbesondere die Ausgabe von Projekten, die T_EX nur im Backend einsetzen, z. B. via Pandoc, ließe sich durch den Wechsel zu ConTeXt verschlanken und beschleunigen.

Ein Artikel⁶ zur Verwendung der Docker-Images für L^AT_EX und ConTeXt erschien bereits im TUGboat 3/2019.

³ <https://tug.org/TUGboat/tb43-2/tb134hagen-math.pdf>

⁴ https://wiki.contextgarden.net/Including_pages_from_PDF_documents\#Include_a_section_from_other_PDF_document

⁵ <https://gitlab.com/islandoftex/images/context>

⁶ <https://www.tug.org/TUGboat/tb40-3/tb126island-docker.pdf>

COW: Online-IDE für ConT_EXt

Zdeněk Svoboda von der Uni Brno hat als Masterarbeit eine Web-IDE mit dem passenden Namen COW⁷ erstellt. Sie besteht aus einem Texteditor mit Syntax-highlighting, Befehlsvervollständigung und -referenz, dazu kommen »Wizards« für Layout, Tabellen und Listen.

Wie bei anderen ConT_EXt-Projekten aus Brno ist etwas Skepsis in Bezug auf die längerfristige Weiterentwicklung und Pflege angebracht, auch der Quellcode ist noch nicht veröffentlicht, das Projekt ist aber sicherlich eine Bereicherung für die Editoren-Landschaft, zumal die ConT_EXt-Unterstützung für mehrere andere Editoren ausgelaufen ist (bei SciTE und Textadept wurde die Plugin-API inkompatibel geändert, Atom ist abgekündigt, für manche andere wurde das Syntaxhighlighting seit MkII nicht mehr aktualisiert) und nicht jede* sich auf vim oder Emacs einlassen möchte.

Spaß mit Unicode

Bei Vergleichen, z. B. mit `\ifnum`, versteht ConT_EXt jetzt Unicode-Operatoren wie $\neq$, $\geq$, $\neq$, $\in$.

⁷ <https://context-on-web.eu>

Von fremden Bühnen

Das L^AT_EX Tagged PDF Project – Status und Fortschritte¹

Frank Mittelbach, Ulrike Fischer

Das »L^AT_EX Tagged PDF Project« wurde im Frühjahr 2020 gestartet und der T_EX-Community auf der (Online-)TUG-Konferenz des gleichen Jahres vom L^AT_EX-Team vorgestellt. Dieser kurze Bericht beschreibt die Fortschritte und den Status dieses über mehrere Jahre laufenden Projekts.

Projektüberblick

Ein Tagged PDF ist ein PDF mit einer zusätzlichen semantischen Struktur, die die Barrierefreiheit und Wiederverwendbarkeit verbessert. Um es L^AT_EX zu ermöglichen, solche Tagged PDFs zu erstellen, wurde im vierten Quartal 2019 das »L^AT_EX Tagged PDF Project« mit einer Machbarkeitsstudie für Adobe ins Leben gerufen [20]. Das führte zu der Zusage von Adobe, das in dieser Studie vorgeschlagene Projekt finanziell zu unterstützen. Leider wurde diese Zusage durch den Beginn der COVID-19-Pandemie erst ab dem dritten Quartal 2020 umgesetzt. Trotz dieser Verzögerung begann das L^AT_EX Project Team im Spätfrühling 2020 mit den ersten Arbeiten (mit zunächst begrenzten Ressourcen) und das Projekt wurde der T_EX-Community auf der (Online-)TUG-Konferenz 2020 vorgestellt, wo das Team zudem die ersten Ergebnisse aus Phase I des Projekts zeigen konnte [21].

Das »L^AT_EX Tagged PDF Project« ist in sechs Phasen unterteilt, die jeweils zu direkt nutzbaren Anwendungen führen. Damit ist sichergestellt, das frühe Benutzer Vorteile davon haben und das Projektteam gleichzeitig schnelles Feedback erhält. Die Phasen sind grob auf die halbjährlichen Releasezyklen von L^AT_EX abgestimmt, wobei jede Phase ein bis drei L^AT_EX-Releases umfassen soll.

In der Machbarkeitsstudie haben alle ermittelten Projektaufgaben eine eindeutige Nummer, um leicht auf sie verweisen, ihre Abhängigkeiten beschreiben und sie im Projektzeitplan darstellen zu können, der in der Studie umrissen wurde.

¹ Dieser Artikel wurde zuvor im TUB 42:3 [19] veröffentlicht und von Thomas Demmig übersetzt.

Neben der Benennung der Aufgaben sind in dem hier vorliegenden Report auch diese Nummern angegeben, um Leserinnen und Leser dabei zu unterstützen, mehr Details über eine bestimmte Aufgabe in der Studie nachschlagen zu können [20].

Phase I – Das Fundament vorbereiten

Zweck der ersten Phase war das Implementieren der Kernfunktionalität in L^AT_EX, die die Basis für alle folgenden Arbeiten für das Erstellen sauber getaggtter PDFs legt. Diese Phase wurde 2021 abgeschlossen und sie enthielt drei wichtige Meilensteine:

- »Hook-Management-System« (Aufgabe 2.2.5)
- »PDF Object Support« (Aufgabe 2.2.6)
- »Automated Testing Environment« (Aufgabe 2.1.2)

Als Teil der Arbeit an Phase I haben sich auch zwei neue Aufgaben ergeben, die nicht in der Machbarkeitsstudie aufgeführt waren:

- »Generische Befehls-Hooks hinzufügen« (Aufgabe 2.2.5(b))
- »Ein allgemeines Configuration Point Management bereitstellen« (Aufgabe 2.2.5(c))

Hook-Management-System

Das »Hook-Management-System« wurde der T_EX-Gemeinde mit dem L^AT_EX-Release im Herbst 2020 zur Verfügung gestellt. Es erweitert L^AT_EX um eine generische Hook-Schnittstelle und eine Vielzahl von Dokument-, Shipout-, Datei- und Umgebungs-Hooks [9, 14, 16].

Nach einem halben Jahr im allgemeinen Einsatz (durch das Team und viele Entwickler von dritter Seite) hat sich gezeigt, dass das Hook-Management-System einige Erweiterungen und Anpassungen benötigt. Dies führte zu der zusätzlichen Aufgabe 2.2.5(b), um das Hook-Management-System bei Bedarf durch eine generische Methode zum automatischen Hinzufügen von Hooks zu Befehlen von dritter Seite zu erweitern.

Diese generischen Befehls-Hooks werden es uns ermöglichen, fremden Code von außen zu patchen (also ohne die Wartung von nicht mehr betreuten, aber durchaus funktionierenden Paketen zu übernehmen) und so die Umsetzung von Tagged PDF zu vereinfachen. Die generischen Befehls-Hooks wurden mit dem L^AT_EX-Release im Frühling 2021 implementiert und zur Verfügung gestellt [13].

Wir haben auch gesehen, dass ein »General Configuration Point Management« notwendig ist – analog zum Hook-Management, aber für Konfigurationen, bei denen nur ein einzelnes Paket oder eine einzelne Klasse verantwortlich sein kann.

Das ist erforderlich, um zu verhindern, dass Pakete in internem L^AT_EX-Code patchen und sich gegenseitig (teilweise) überschreiben oder den Code verändern, der für das Tagging gebraucht wird. So versucht beispielsweise aktuell eine Reihe von Paketen, die gleichen internen Befehle für die Ausgaberroutine zu verändern, um Spezial-Code für das Fußnoten-Handling einzufügen.

Für diese neue Aufgabe 2.2.5(c) sind bereits konzeptionelle Arbeiten vorgenommen worden – Package-Writer-Interfaces, die denen des Hook-Management ähneln, werden 2023 bereitgestellt werden.

PDF Object Support

»PDF Object Support«, Code zur Unterstützung einer Reihe von für PDFs notwendigen Aufgaben, ist teilweise schon durch das Modul l3pdf der L3-Programmier-schicht im Kernel vorhanden [12]. Zusätzliche Funktionalität wird durch das externe Bundle pdfmanagement-testphase [17] bereitgestellt, welches zu Beginn des Jahres 2021 releast wurde. Zu einem späteren Zeitpunkt wird es in den Kernel integriert werden.

Als Teil dieser Aufgabe war es notwendig, mit der Engine-Entwicklung von T_EX zusammenzuarbeiten, um Engine-Patches für LuaT_EX und pdfT_EX zu programmieren, damit diese Engines PDF 2.0 Structure-Destinations vollständig unterstützen können (X_YT_EX war dazu schon von Anfang an in der Lage). Solche Structure-Destinations bieten die gleichen Anzeigemechanismen wie eine Destination, verweisen aber auf ein Strukturelement statt auf eine Seite und sorgen so für eine direkte Verbindung von einem Link zu Inhalten. Mit T_EX Live 2022 und einem aktuellen MiK_TE_X werden Structure-Destinations nun automatisch erstellt, wenn der Befehl \DocumentMetadata mit passenden Schlüsselwerten benutzt wird, um ein PDF 2.0-Dokument zu erstellen.

Leider gibt es aktuell keine einfache Möglichkeit, lokal zu prüfen, ob ein Link auf eine Struktur zeigt (und wenn ja, auf welche). Die Tag-Ansicht von Adobe Pro zeigt sie nicht und dessen HTML-Export ignoriert die Strukturen. Um die neuen Features zu testen, muss man die internen PDF-Strukturen prüfen oder einen Online-Service wie ngPDF [1] verwenden – eine Demo-Site für neue Technologie, um HTML von Tagged PDF vorhersagbar ableiten zu können (entwickelt durch die PDF Association). Beim Einsatz von ngPDF enthält der HTML-Export eines Tagged PDF mit Structure-Destinations Links auf die ID einer Struktur:

```
<h1 id="6d4ff5-1">1 abc</h1>
<a href="#6d4ff5-1">1</a>
```

Ohne die Structure-Destination würde der Link nur auf ein mit einer Seite verbundenes Ziel verweisen:

```
<a href="#page-0">1</a>
```

Automated Testing Environment

Die »Automated Testing Environment« ist ein wichtiges Element, um das Projekt erfolgreich abschließen zu können. Es ist entscheidend, eine große Testsuite aufzubauen, in der alle Tests automatisch ausgeführt und überprüft werden. Das liegt daran, dass wir den zentralen L^AT_EX-Code deutlich verändern müssen, ohne dass die Millionen von Benutzern negative Auswirkungen verspüren – sie wollen ihre Dokumente weiterhin erstellen können, ohne sich unerwarteten visuellen Änderungen gegenüberzusehen. Auch wenn dieser Code direkt nur vom L^AT_EX-Team genutzt wird, ist das Vorhandensein und die Stabilität der Testumgebung für den Erfolg des Projekts ausgesprochen wichtig.

Phase II – Tagging einfacher Dokumente

Hauptziel von Phase II ist das Ermöglichen eines automatischen Taggings einfacher Dokumente – kompliziertere Strukturen wie Formeln, Tabellen und so weiter sind ausgeschlossen. In dieser Phase wird der zentrale Code bereitgestellt, der allgemeine Mechanismen für das automatische Erkennen von Absatztext und seinem korrekten Tagging bietet. Zudem wird eine Untermenge der Standardelemente von L^AT_EX-Dokumenten unterstützt, die so die erforderlichen Tags erzeugen.

Die Arbeit an diesen beiden elementaren Grundlagen für Phase II begannen schon parallel zum Abschluss von Phase I:

- »Core Tagging Support« (Aufgabe 2.3.1); und
- Unterstützung für »Automated Paragraph Tagging« (Aufgabe 2.3.2).

Zentrales Ziel dieser Aufgaben ist, die notwendige Infrastruktur für das automatische Tagging relativ einfacher Dokumente bereitzustellen. Der aktuelle Fokus liegt daher auf der verbleibenden großen Aufgabe für Phase II:

- »Tagging für die grundlegenden Dokumentelemente von L^AT_EX implementieren« (Aufgabe 2.3.3).

Die Arbeit an Phase II braucht etwas mehr Zeit als zunächst geschätzt wurde, weil zusätzliche Aufgaben erkannt wurden, die entweder Voraussetzung für einen erfolgreichen Abschluss von Phase II oder für spätere Phasen notwendig waren und – aus dem einen oder anderen Grund – am besten jetzt parallel erledigt wurden.

Wir gehen davon aus, Ende 2022 mit Phase II fertig zu sein. Dann werden wir außer der Reihe ein Release des latex-lab-Bundles anbieten, welches im Juni 2022 hinzugefügt wurde, um mit neuen Projektcode experimentieren zu können, ohne die Abläufe mit dem Standard-L^AT_EX zu stören [10].

Core Tagging Support

Für das Taggen eines PDFs ist das Schreiben und Managen einer Reihe von Objekten und Literalen im PDF erforderlich. Der notwendige Code für den »Core Tagging Support« steht aktuell als Addon-Paket zur Verfügung (das Paket tagpdf [3]), da so diejenigen sicher experimentieren können, die jetzt schon gerne eine Tagged PDF-Ausgabe haben wollen, während andere Abläufe aber nicht aus dem Tritt gebracht werden. Ist alles umfassend getestet worden, wird der Code in den Kernel integriert (das ist eine eigene Aufgabe in einer späteren Phase). Die Verwendung des Codes zum Erstellen einfacher Tagged PDFs wurde von Ulrike Fischer auf der (Online-)TUG-Konferenz 2021 beschrieben [2].

Automated Paragraph Tagging

Große Teile von Standarddokumenten bestehen aus normalen Absätzen. Für den Erfolg des Projekts ist es außerordentlich wichtig, dass solche Absätze automatisch getaggt werden und dass Absätze, die über mehrere Seiten verteilt sind, korrekt behandelt werden. Die für solch ein »Automated Paragraph Tagging« notwendigen Kernel-Erweiterungen wurden rechtzeitig für das L^AT_EX-Release Frühling 2021 fertig und sie wurden auf der (Online-)TUG-Konferenz 2021 von Frank Mittelbach angekündigt [18]. Dabei kommen Markierungen und neue Hooks am Anfang und Ende von Absätzen zum Einsatz [15].

Tagging für grundlegende Dokumentelemente

Ziel dieser Aufgabe ist, Standardelemente von L^AT_EX-Dokumenten für das Taggen empfänglich zu machen, so dass alle Strukturinformationen, die sie kapseln, automatisch in eine entsprechende Tagstruktur (einschließlich der Attribute) im erzeugten PDF-Dokument übertragen werden. Im Projektzeitplan ist diese Aufgabe auf die Phasen II und III aufgeteilt. Dabei wird in Phase II damit begonnen, sich auf die High-Level-Strukturelemente wie Links, Kopf- und Fußzeilen, Überschriften, Listen, Fußnoten und Inhaltsverzeichnisse zu konzentrieren. Das Paket tagpdf und der Code zum Managen des PDFs implementiert bereits das automatische Taggen von

Hyperlinks und das Taggen von Kopf- und Fußzeilen (als Artefakte). Das automatische Taggen von Fußnoten (mit Links) einschließlich von Sonderfällen – wie zum Beispiel Fußnoten, die über Spalten oder Seiten hinweg umbrochen werden, und das Taggen von Listen und Inhaltsverzeichnissen – wird mit dem Release von latex-lab Ende 2022 ermöglicht werden. In Phase III werden (im Jahr 2023) die restlichen grundlegenden Dokumentstrukturen ergänzt, während komplexere Strukturen, wie zum Beispiel Tabellen, für spätere Phasen verbleiben.

In diesem Kontext bedeutet ein »automatisches Tagging«, dass »erkannte Dokumentelemente mit einem Standard-Mapping auf PDF-Tags ohne manuelle Anpassungen oder eine feingranularere Flexibilität abgebildet werden.« Solch eine Flexibilität wird natürlich irgendwann notwendig werden, um Tagged PDFs mit höchster Qualität erzeugen zu können – daher musste diese Flexibilität im zugrundeliegenden Code bereits vorgesehen werden, was zu einer neuen Aufgabe führte:

- »Eine allgemeine Schlüssel/Wert-Schnittstelle entwerfen und implementieren« (Aufgabe 2.3.3b).

Dies wurde als zusätzliche Voraussetzung für das erfolgreiche Abschließen der Aufgaben 2.3.3 und 2.3.5 erkannt. Es muss möglich gemacht werden, das optionale Argument von Standard-Befehlen und -Umgebungen – zum Beispiel für Abschnitte und Beschriftungen – zu erweitern, um Schlüssel/Wert-Argumente für alternativen Text übergeben zu können.

Zu der Aufgabe gehört zudem das Entwerfen und Implementieren eines allgemeinen Template-Mechanismus für Befehle und Umgebungen, um die Schlüssel/Wert-Methode zur Konfiguration nutzen zu können. Allerdings wird das Bereitstellen dieser Konzepte auf Benutzer- und Paketentwickler-Ebene das Designen von Schnittstellen für deren Konfiguration und ein manuelles Überschreiben erfordern, was beides Teil späterer Phasen ist.

Vorbereitende Arbeiten für Aufgaben in den Phasen III und IV

Aus einer Reihe technischer und praktischer Gründe sind wir vom ursprünglichen Zeitplan abgewichen und haben schon mit der Arbeit an Aufgaben begonnen, die für spätere Phasen eingeplant waren. Dazu gehören:

- »Designen und Implementieren eines erweiterten Mechanismus für Querverweise in L^AT_EX« (Aufgabe 2.2.2)
- »Bereitstellen einer Schnittstelle für das Spezifizieren aller Arten von Dokument-Metadaten« (Aufgabe 2.3.4)
- »Designen und Implementieren von Hyperlinks und Übertragen in den L^AT_EX-Kernel« (Aufgabe 2.2.3)
- »Erfüllen von Standards« (Aufgabe 2.3.9)

Schnittstelle zu Dokument-Metadaten

Status und Ausblick auf die Implementierung von Dokument-Metadaten (Aufgabe 2.3.4) sind in einem eigenen Artikel im TUB beschrieben [4] und für die T_EXnische Komödie übersetzt [5].

Verbesserungen bei Hyperlinks

Benutzerschnittstellen (und Backend-Code) für das Ermöglichen von Hyperlinks werden aktuell vor allem durch das Paket `hyperref` bereitgestellt. Mit dem neuen, weiter oben beschriebenen PDF-Object-Support wurden schon große Teile des Backend-Codes in den L^AT_EX-Kernel oder in das Paket `pdfmanagement-testphase` verschoben. Werden aktuell Strukturen wie Fußnoten, Überschriften und Inhaltsverzeichnisse für das Tagging angepasst, wird auch gleich eine native Unterstützung für Hyperlinks eingebaut und die nicht mehr länger erforderlichen Patches in `hyperref` werden unterbunden.

Erfüllen von Standards

Die Unterstützung diverser PDF/A-Standards wird durch das `pdfmanagement-testphase`-Paket (Modul `l3pdfmeta`) bereitgestellt. Der Code wird die üblicherweise erforderlichen Farbprofile und PDF-Objekte hinzufügen und verbotene Aktionen wie JavaScript-Code unterdrücken.

Es sei darauf hingewiesen, dass L^AT_EX nicht alle Anforderungen eines Standard abprüfen kann und dass ein externer Validator wie `veraPDF` dazu verwendet werden sollte. Die Unterstützung von PDF/X-Standards geschieht aktuell nur in Form von XMP-Metadaten-Einträgen.

Phasenübergreifende Aufgaben

Eine Reihe von Aufgaben erfordert Aufmerksamkeit und Arbeiten in allen Phasen. Bisher sind das:

- »Eine Änderungsstrategie definieren, um L^AT_EX sicher zu erweitern, ohne der weltweiten Anwenderbasis ernsthafte Probleme zu bereiten« (Aufgabe 2.1.1)
- »Akzeptanztests durch Entwickler für abgeschlossene Aufgaben« (Aufgabe 2.4.1)
- »Koordinierte Aktualisierungen externer Pakete« (Aufgabe 2.4.3)

Das Verändern und Erweitern von L^AT_EX, ohne bestehende Dokumente und Abläufe durcheinanderzubringen, ist ein wichtiges Ziel des Projekts. Dafür wurde eine Reihe von Tools implementiert:

- Die `\DocumentMetadata`-Schnittstelle erlaubt es, getaggte und nicht getaggte Dokumente vom gleichen L^AT_EX-Format verarbeiten zu lassen, indem nur eine einzige Zeile verändert wird.
- Experimenteller Code wird zuerst in externen Paketen wie `pdfmanagement-testphase` und `tagpdf` oder im `latex-lab`-Bundle untergebracht.
- Das Paket `firstaid` erlaubt es uns, externe Pakete temporär zu patchen, wenn sich herausstellt, dass sie zu einer Änderung inkompatibel sind.
- Die `latex-dev`-Releases verschaffen Paketautoren Zeit und Gelegenheit zum Testen von Änderungen und zum Melden von Problemen, und dem L^AT_EX-Team Zeit, Paketautoren zu kontaktieren und Updates zu koordinieren.
- Das Paket `tagpdf-base` stellt Dummy-Versionen der zentralen Tagging-Befehle bereit, so dass das Schreiben von Befehlen und Umgebungen möglich wird, die sauber taggen, wenn der Benutzer das Tagging aktiviert, die aber auch ohne Tagging funktionieren.
- Der Kernel-Befehl `\IfDocumentMetadataTF` erlaubt die Prüfung, ob die neue Schnittstelle in einem Dokument verwendet wird.
- Der Kernel-Befehl `\MakeLinkTarget` stellt eine Dummy-Version des von `hyperref` genutzten Befehls zum Erstellen von Ankern bereit, so dass Befehle und Umgebungen mit eingebauten Hyperlink-Fähigkeiten geschrieben werden können, die aktiviert werden, wenn der Benutzer `hyperref` lädt.

Schnittstelle zum Projektcode für Anwender

Als Teil der Metadaten-Aufgabe (2.3.4) haben wir einen Befehl `\DocumentMetadata` im L^AT_EX-Kernel bereitgestellt, der als allererste Deklaration in einem Dokument genutzt werden kann (also noch vor `\documentclass`). So können wir den PDF-Management-Code laden und Tagging sowie anderen projektbezogenen Code aktivieren. Kurz gesagt gibt der Benutzer durch diese Deklaration an, dass dies ein Dokument ist, für welches Tagging angewendet werden soll. So erfüllt er eine ähnliche Rolle wie der Wechsel von `\documentstyle` (altes L^AT_EX 2.09 vor 1994) nach `\documentclass` (modernes L^AT_EX). Das erleichtert die Umstellung und erlaubt es, alten und neuen Code parallel existieren zu lassen.

Außerdem vermeiden wir so, dass die Anwender spezielle Pakete laden müssen, um neue Features zu testen – es reicht aus, einfach eine Zeile

```
\DocumentMetadata{testphase=phase-II,...}
```

zu Beginn des Dokuments anzugeben, wodurch dann alles Notwendige geladen wird, um die aktuell verfügbaren Ergebnisse aus einer bestimmten Phase zu aktivieren.

Diese Schnittstelle wurde mit dem L^AT_EX-Release vom Juni 2022 bereitgestellt und steht seitdem jedem L^AT_EX-Anwender zur Verfügung.

Zusammenfassung des aktuellen Status

Phase I des Projekts wurde schon abgeschlossen und die Ergebnisse sind in L^AT_EX integriert und werden aktuell allgemein genutzt.

Phase II ist fast fertig – die meisten Aufgaben sind implementiert und bereitgestellt; mit einem weiteren Release des latex-lab-Bundles Ende 2022/Anfang 2023 werden sie abgeschlossen sein. Damit wird dann ein automatisches Tagging von L^AT_EX-Dokumenten mit einer (immer noch recht einfachen) Elementstruktur möglich, indem man das schon erwähnte

```
\DocumentMetadata{testphase=phase-II, ...}
```

an den Anfang eines Dokuments stellt.² Das Hauptziel von Phase II ist daher, viele bestehende L^AT_EX-Dokumente verarbeiten zu können und eine getaggte und damit barrierefreie PDF-Ausgabe zu erhalten, ohne den Quellcode abgesehen von der einen Zeile verändern zu müssen.

Ziel späterer Phasen ist, dies auf mehr und mehr Dokumentelemente auszuweiten und Anpassungen am Tagging vornehmen zu können, so dass die Qualität der Tagged PDF-Ausgabe steigt. Schließlich wird auch die temporäre Schnittstelle in \DocumentMetadata, die für das Laden des Support-Pakets tagpdf verantwortlich ist, nicht mehr länger notwendig sein.

Insgesamt können wir zuversichtlich davon ausgehen, dass das Projekt gut voranschreitet, sich innerhalb des groben Plans bewegt und das auch in Zukunft tun wird.

Software-Releases

In der Zeit zwischen dem Q4/2020 und Q4/2022 hat das Team fünf Haupt-Releases von L^AT_EX implementiert und bereitgestellt, die einen direkten Einfluss auf den Fortschritt des Projekts haben. Wichtige Features der Releases sind im folgenden zusammengefasst – die Releases enthalten auch andere Verbesserungen, die nicht direkt mit dem Tagged PDF Project in Verbindung stehen.

² Die im Argument von \DocumentMetadata anzugebenden Daten sind aktuell vorläufig und werden sich im Laufe des Projekts noch ändern. So wird beispielsweise mit testphase=phase-II der Code für Phase II aktiviert – etwas, das nicht mehr erforderlich sein wird, wenn der Code finalisiert und in den L^AT_EX-Kernel integriert wurde.

L^AT_EX Release 2020-10, siehe [6]

- Das neue Hook-Management für L^AT_EX bereitstellen (Aufgabe 2.2.5)
- Die xparse-Schnittstelle in den Kernel verschieben (erforderlich für eine Reihe späterer Aufgaben, um Schnittstellen auf Dokumentenebene anbieten zu können)

L^AT_EX Release 2021-06, siehe [7]

- Erweitertes Hook-Management für Absätze (erforderlich für Aufgabe 2.3.2)
- Erweitertes Hook-Management für Befehle (erforderlich für Aufgaben in den Phasen II & III)

L^AT_EX Release 2021-11, siehe [8]

- Konsolidierungs-Release
- Korrekturen und Verbesserungen am Hook-Management-System, nachdem dieses vom Projektteam und anderen Entwicklern ausführlich verwendet wurde

L^AT_EX Release 2022-06, siehe [10]

- \DocumentMetadata-Schnittstelle (erforderlich für Aufgabe 2.3.4)
- Einführung des latex-lab-Bundles (erforderlich für das sichere Experimentieren von Anwendern mit neuer Funktionalität aus dem Projekt)
- Erster Teil der neuen Schlüssel/Wert-Funktionalität im Kernel (erforderlich für viele Aufgaben)

L^AT_EX Release 2022-11, siehe [11]

- Automatisches Erkennen neuer Schlüssel/Wert-Argumente, zum Beispiel in \section oder \caption (implementiert Aufgabe 2.3.3(b); erforderlich für viele andere Aufgaben)

Literatur

- [1] DualLab: Next Generation PDF, <https://ngpdf.com> (besucht am 11. 10. 2022).
- [2] Ulrike Fischer: »On the road to Tagged PDF: About StructElem, Marked Content, PDF/A and Squeezed Bärs«, *TUGboat*, 42.2 (131 2021), 170–173, ISSN: 0896-3207, DOI 10.47397/tb/42-2/tb131fischer-tagpdf, <https://tug.org/TUGboat/tb42-2/tb131fischer-tagpdf.pdf>.

- [3] Ulrike Fischer: The tagpdf package, Tools for experimenting with tagging using pdfL^AT_EX and LuaL^AT_EX, Version 0.97, 2022, <https://ctan.org/pkg/tagpdf> (besucht am 15. 10. 2022).
- [4] Ulrike Fischer, Frank Mittelbach: »Adding XMP metadata in L^AT_EX«, *TUG-boat*, 43.3 (135 2022), 263–267, ISSN: 0896-3207, DOI 10. 47397/tb/43-3/tb135fischer-xmp.
- [5] — »XMP-Metadaten in L^AT_EX einfügen«, *DTK*, 35.1 (2023), 39–49.
- [6] L^AT_EX Project Team: L^AT_EX news, issue 32, 2020, <https://www.latex-project.org/news/latex2e-news/ltnews32.pdf>.
- [7] — L^AT_EX news, issue 33, 2021, <https://www.latex-project.org/news/latex2e-news/ltnews33.pdf>.
- [8] — L^AT_EX news, issue 34, 2021, <https://www.latex-project.org/news/latex2e-news/ltnews34.pdf>.
- [9] — L^AT_EX’s hook management, 2022, <https://mirrors.ctan.org/macros/latex/base/lthooks-doc.pdf>.
- [10] — L^AT_EX news, issue 35, 2022, <https://www.latex-project.org/news/latex2e-news/ltnews35.pdf>.
- [11] — L^AT_EX news, issue 36—Draft, 2022, <https://www.latex-project.org/news/latex2e-news/ltnews36.pdf>.
- [12] — The L3kernel package, L^AT_EX3 programming conventions, 2022, <https://ctan.org/pkg/l3kernel> (besucht am 7. 10. 2022).
- [13] — The ltcmdhooks module, 2022, <https://mirrors.ctan.org/macros/latex/base/ltcmdhooks-doc.pdf>.
- [14] — The ltfilehook documentation, 2022, <https://mirrors.ctan.org/macros/latex/base/ltfilehook-doc.pdf>.
- [15] — The ltpara.dtx code, 2022, <https://mirrors.ctan.org/macros/latex/base/ltpara-doc.pdf>.
- [16] — The ltshipout package, 2022, <https://mirrors.ctan.org/macros/latex/base/ltshipout-doc.pdf>.
- [17] — The pdfmanagement-testphase package, L^AT_EX PDF management test-phase bundle, Version 0.95s, 2022, <https://ctan.org/pkg/pdfmanagement-testphase> (besucht am 7. 10. 2022).
- [18] Frank Mittelbach: Taming the beast — Advances in paragraph tagging with pdfT_EX and X_ƎT_EX (2021): Automatic paragraph tagging with the pdfT_EX and X_ƎT_EX engine now possible, <https://www.latex-project.org/news/2022/09/06/TUG-online-talks-21-22/> (besucht am 11. 10. 2022).

- [19] Frank Mittelbach, Ulrike Fischer: »The L^AT_EX Tagged PDF project—A status and progress report«, *TUGboat*, 43.3 (135 2022), 268–272, ISSN: 0896-3207, DOI 10.47397/tb/43-3/tb135mitt-tagged.
- [20] Frank Mittelbach, Ulrike Fischer, Chris Rowley: L^AT_EX Tagged PDF Feasibility Evaluation, L^AT_EX Project, 2020, <https://latex-project.org/publications/2020-tagged-pdf-feasibility.pdf> (besucht am 8. 12. 2020).
- [21] Frank Mittelbach, Chris Rowley: »L^AT_EX Tagged PDF—A blueprint for a large project«, *TUGboat*, 41.3 (129 2020), 292–298, ISSN: 0896-3207, <https://latex-project.org/publications/2020-FMi-TUB-tb129mitt-tagpdf.pdf>.

XMP-Metadaten in L^AT_EX einfügen¹

Ulrike Fischer, Frank Mittelbach

Eine Aufgabe des »L^AT_EX Tagged PDF Project« [5] ist die Evaluation bestehender Lösungen, mit denen sich XMP-Metadaten einem PDF hinzufügen lassen, und bei Bedarf das Entwerfen und Implementieren einer neuen Standardschnittstelle dafür. In diesem Artikel werden wir den aktuellen Status dieser Aufgabe beschreiben.

Einleitung

Das PDF-Format bietet zwei Stellen an, um Metadaten abzulegen. Eine ist das *Info Dictionary*. Es befindet sich direkt an der Wurzel der PDF-Struktur und enthält Schlüssel/Wert-Paare, die Dokumentdaten enthalten, wie das PDF-Erstellungsdatum (beispielsweise `/CreationDate (D:20221005153151+02'00')`) und den Titel (zum Beispiel `/Title (Bearwear)`). (Denken Sie daran, dass String-Konstanten in PDF und PostScript in Klammern stehen.) Es gibt eine ganze Reihe von Standard-Schlüsseln, wie zum Beispiel `/Title`, `/Author` oder `/CreationDate`, aber das Dictionary kann auch private Schlüssel enthalten. pdfT_EX fügt beispielsweise seinen Banner hinzu: `/PTeX.Fullbanner (This is pdfTEX, ...)`. PDF-Viewer zeigen normalerweise eine Auswahl der Standard-Schlüssel in den Eigenschaften eines PDF an. Alle T_EX-Engines bieten Tools an, mit denen sich Werte in dieses Dictionary eintragen und die von den Engines automatisch hinzugefügten Werte anpassen lassen. In L^AT_EX werden normalerweise Titel und Autor mit Hilfe des Pakets `hyperref` und seinen Schlüsseln `pdftitle` und `pdfauthor` hinzugefügt.

¹ Dieser Artikel wurde zuvor im TUB 43:3 [1] veröffentlicht und von Thomas Demmig übersetzt.

Während das Info-Dictionary beliebige Daten enthalten kann, handelt es sich nichtsdestotrotz nur um eine ungeordnete Liste, die als ernsthafter Datencontainer nur bedingt taugt. Daher wurde mit PDF 1.4 dem Format eine zweite Möglichkeit hinzugefügt: Man kann eine XML-Datei mit den Daten einbetten und diese Datei aus dem PDF-Katalog über den `/Metadata`-Schlüssel referenzieren.² Das Format der XML-Datei ist als Teil eines Frameworks namens *Extensible Metadata Platform* (XMP) definiert, das erstmals in einem Adobe-Dokument beschrieben wurde und bei dem es sich mittlerweile um einen ISO-Standard handelt [4], daher werden diese Metadaten allgemein als *XMP-Metadaten* bezeichnet.

In PDF 2.0 ersetzen die XMP-Metadaten das als veraltet gekennzeichnete Info-Dictionary. Auch in früheren PDF-Versionen waren die Metadaten schon für Standards wie die diversen PDF/A- und PDF/X-Versionen, PDF/UA (der Standard für barrierefreies PDF) und Standards für den Austausch elektronischer Rechnungen (wie ZUGFeRD 2.2/Factur-X 1.0 [2]) erforderlich. Es ist daher ziemlich wichtig, Tools und Schnittstellen zur Verfügung zu haben, mit denen diese Daten zu einem PDF hinzugefügt werden können. In den folgenden Abschnitten beschreiben wir verschiedene bestehende Möglichkeiten und stellen unseren Ausblick auf zukünftige Pläne in diesem Bereich vor.

Wir setzen voraus, dass alle Quellcode-Dateien UTF-8-kodiert sind – Stellen, an denen 8-Bit-kodierte Dateien eventuell zusätzliche Aufmerksamkeit erfordern, werden nicht erwähnt (XMP-Metadaten in einer PDF-Datei sind immer UTF-8-kodiert).

Die XMP-Metadaten erstellen

Es ist ziemlich einfach, einem PDF eine XML-Datei hinzuzufügen und sie im Katalog zu referenzieren. In allen Engines lässt sich das mit wenigen Zeilen Code erreichen – das kleine Paket `xmptool` [10] demonstriert das für `pdf $\text{\LaTeX}$` . Die Herausforderung liegt darin, den Inhalt der XML-Datei korrekt zu erstellen.

- Zunächst einmal gibt es – wie immer bei XML – ziemlich viel formale Syntax, die man verstehen und der man folgen muss.
- PDF-Standards enthalten weitere Anforderungen an die Inhalte und die Struktur des XML. So müssen beispielsweise Eigenschaften, die nicht als vordefiniert zählen [7], in Erweiterungsschemata deklariert werden, und wenn Daten (wie zum Beispiel ein Titel) sowohl im Info-Dictionary als auch in den XMP-Metadaten vorhanden sind, müssen diese übereinstimmen – etwas, das nicht leicht sicherzustellen ist, wenn unterschiedliche Kodierungen und Formatierungen

² Es ist möglich, mehrere XML-Dateien einzubetten und sie aus anderen Teilen des PDF zu referenzieren, aber in diesem Artikel beschränken wir uns auf den dokumentweiten Datencontainer.

Listing 1: Eine Auswahl von automatisch durch hyperxmp hinzugefügten XMP-Metadaten

```
<pdf:PDFVersion>1.5</pdf:PDFVersion>
<dc:format>application/pdf</dc:format>
<xmp:CreateDate>2022-10-06T10:27:33+02:00</xmp:CreateDate>
<xmp:CreatorTool>LaTeX with hyperref</xmp:CreatorTool>
<xmpMM:DocumentID>uuid:aef2b675-9b18-4d18-97f7-a3339b139000</xmpMM:DocumentID>
```

zum Einsatz kommen. Das erfordert konkrete Tests mit PDF-Viewern und -Validatoren.

- Dann muss man entscheiden, welche XMP-Metadaten unterstützt werden sollen (es gibt hier mehrere mehr oder weniger standardisierte Namensräume) und man muss Benutzerschnittstellen für die Daten entwickeln, die nicht automatisch detektiert werden können.
- Die Benutzereingaben müssen bereinigt, korrekt für die Verwendung in einer XML-Datei maskiert und nach UTF-8 konvertiert werden.

In der Vergangenheit haben zwei L^AT_EX-Pakete diese Aufgabe übernommen. Sie erzeugen nicht genau die gleichen XMP-Metadaten, aber die Unterschiede sind gering – letztendlich arbeiten sie mit mehr oder weniger den gleichen Daten.

Das Paket hyperxmp

Alleine durch das Laden des Pakets hyperxmp von Scott Pakin [6] werden schon XMP-Metadaten hinzugefügt. In Listing 1 sehen Sie ein paar ausgewählte Zeilen davon.

hyperxmp unterstützt alle wichtigen Kompilierungswege: pdfL^AT_EX, LuaL^AT_EX, XeL^AT_EX, L^AT_EX mit dvips,³ (u)(p)L^AT_EX mit DVIPDFMx.

Für die Benutzerschnittstelle hängt sich das Paket in den Befehl \hypersetup von hyperref ein. Es liest die Werte nativer hyperref-Schlüssel wie pdftitle und pdfauthor und definiert neue Schlüssel für eine Reihe zusätzlicher XMP-Metadaten. Listing 2 zeigt ein paar Beispiele. Das Paket hyperxmp unterstützt einen ziemlich großen Satz an Metadaten-Tags, besitzt aber keine Schnittstelle, um diesen Satz zu erweitern.

hyperref wird von hyperxmp nicht nur für die Benutzerschnittstelle verwendet, sondern auch, um die Benutzereingaben zu bereinigen: Alle Eingaben werden zuerst von \pdfstringdef in das Format umgewandelt, das in Lesezeichen verwendet wird, und dann zurück nach UTF-8 konvertiert – sie können daher beliebige Unicode-

³Leider besitzt Ghostscript keine Möglichkeit, die XMP-Metadaten als unkomprimierten Stream hinzuzufügen.

Listing 2: Beispiel für hyperxmp mit zusätzlichen \hypersetup-Schlüsseln

```

\documentclass{article}
\usepackage{hyperxmp}
\usepackage{hyperref}
\hypersetup
{ pdftitle = {Über einen die Erzeugung und Verwandlung des Lichtes
              betreffenden heuristischen Gesichtspunkt},
  pdfauthor={Albert Einstein}, pdflang = {de},
  pdfmetalang={de}, pdfdate={1905-03-17}, pdfcontactcity={Bern},
  pdfcontactcountry={Switzerland}, pdfissn={0003-3804},
  pdfdoi={10.1002/andp.19053220607}, }
\begin{document}
Text
\end{document}

```

Zeichen und alle Befehle, die von hyperref in den Lesezeichen unterstützt werden, enthalten. Fehlt hyperref, wird es automatisch von hyperxmp am Ende der Präambel geladen. Soll das Dokument XMP-Metadaten enthalten, aber keine Links oder Lesezeichen, laden Sie hyperref mit der Paketoption draft.

Individuelle Elemente in der Autoren- und Schlüsselwort-Liste sollten durch Kommas getrennt werden – wird ein echtes Komma gebraucht, muss \xmpcomma verwendet werden.

Manche Schlüssel erlauben das Hinzufügen von Sprachvarianten mit dem Befehl \XMLLangAlt:

```

\hypersetup{pdflang=de,pdftitle=Mein Titel}
\XMLLangAlt{en}{pdftitle={My title}}

```

Das führt zu Metadaten, die das Attribut xml:lang nutzen:

```

<dc:title>
  <rdf:Alt>
    <rdf:li xml:lang="x-default">Mein Titel </rdf:li>
    <rdf:li
xml:lang="de">Mein Titel</rdf:li>
    <rdf:li xml:lang="en">My title</rdf:li>
  </rdf:Alt>
</dc:title>

```

Listing 3: Beispieleingabe mit dem Paket pdfx

```

\begin{filecontents}[force]{\jobname.xmpdata}
  \Title{Baking through the ages}
  \Author{A. Baker\sep C. Kneader}
  \Language{en-GB}
  \Keywords{cookies\sep muffins\sep cakes} \Publisher{Baking International}
\end{filecontents}
\documentclass{article}
\usepackage[a-1b]{pdfx}
\begin{document}
  some text
\end{document}

```

Das Paket pdfx

Ziel des pdfx-Pakets [3], aktuell betreut durch Ross Moore, ist das Unterstützen der Generierung von PDF/X-, PDF/A- und PDF/E-konformen Dokumenten. Da XMP-Metadaten von den Standards verlangt werden, werden sie vom Paket erzeugt, aber es wird auch (teils abhängig vom angeforderten Standard) ein Farbprofil eingebettet, ToUnicode-Werte geändert, mathematische Akzente neu definiert und vieles mehr.

pdfx lässt sich nur mit pdfL^AT_EX, LuaL^AT_EX und XeL^AT_EX verwenden. (Es nutzt das für pdfL^AT_EX geschriebene Paket xmpincl für das eigentliche Einbetten der XMP-Daten und passt es ein bisschen an, um es kompatibel zu den beiden anderen Engines zu machen.) Für das Kompilieren mit XeL^AT_EX ist `--shell-escape` erforderlich, da pdfx LuaL^AT_EX aufruft, um das Erstellungsdatum der Datei zu erhalten – das lässt sich vermeiden, indem man `\pdfcreationdate` manuell erstellt, bevor pdfx geladen wird,⁴ zum Beispiel:

```

\def\pdfcreationdate{\string D:20221002224824+10'00'}

```

Beim Einsatz von pdfx muss der Anwender Metadaten in einer externen Datei mit der Erweiterung `.xmpdata` bereitstellen. Das Paket empfiehlt, diese Datei am Anfang des Dokuments mit einer `filecontents`-Umgebung zu erstellen. In dieser `.xmpdata`-Datei werden die Daten dann als Argumente einer Reihe von Befehlen angegeben – Listing 3 zeigt ein Beispiel. Die Argumente können beliebige Unicode-Zeichen enthalten und müssen nicht im Dokument druckbar sein. Befehle für Zeichen außerhalb des Latin-Zeichensatzes (wie `\CYRD` für Kyrillisch, `\hebzayin` für Hebräisch oder `\textarabicfa`) können ebenfalls verwendet werden, wenn sie über Paketoptionen wie `cyrxmp` oder `hebxmp` aktiviert wurden.

⁴ Das `\string` ist erforderlich, da pdfx ein D mit dem Catcode »other« erwartet.

pdfx lädt hyperref, da es wie das hyperxmp-Paket `\pdfstringdef` nutzt, um manche der Texteingaben zu bereinigen. Zudem verwendet es hyperref, um die PDF-Version zu setzen. Da hyperref normalerweise spät geladen werden sollte, die PDF-Version aber früh gesetzt werden muss, kann es knifflig sein, die richtige Ladereihenfolge zu ermitteln.

Die Standardvorgaben für die XMP-Metadaten sind in externen Templates abgelegt:⁵ `pdfa.xmp`, `pdfx.xmp`, `pdfe.xmp`, Durch Kopieren und Anpassen dieser Templates, ist es möglich, eigene Daten hinzuzufügen.

Als ein Nebeneffekt der Implementierung werden die XMP-Daten beim Verarbeiten in eine Datei `pdfa.xmpi` oder `pdfx.xmpi` geschrieben, was es erlaubt, sie an einen RDF-Validator [12] zu übergeben.

Listen werden erzeugt, indem man die Elemente durch den Befehl `\sep` trennt. Eine Unterstützung mehrerer Sprachen lässt sich durch optionale Argumente erreichen:

```
\Title[en]{Baking through the ages}
\Keywords[de]{Kekse\sep[en]Cookies\sep[fr]Biscuits}
```

Das führt zu

```
<dc:title>
<rdf:Alt>
  <rdf:li xml:lang="en">
    Baking through the ages
  </rdf:li>
</rdf:Alt>
</dc:title>
<dc:subject>
<rdf:Bag>
  <rdf:li xml:lang="de">Kekse</rdf:li>
  <rdf:li xml:lang="en">Cookies</rdf:li>
  <rdf:li xml:lang="fr">Biscuits</rdf:li>
</rdf:Bag>
</dc:subject>
```

Konflikte zwischen hyperxmp und pdfx

Die zwei Pakete lassen sich nicht gemeinsam in einem Dokument verwenden. Da beide primitive Low-Level-Befehle verwenden, um ihren XMP-Stream zu referenzieren, würde der PDF-Katalog zwei `/Metadata`-Einträge enthalten, wenn beide Pakete zum Einsatz kämen – was in einem PDF nicht erlaubt wäre:

⁵ Abhängig vom Standard erzeugt pdfx leicht unterschiedliche Metadaten.

```
/Metadata 3 0 R /Metadata 6 0 R
```

XMP-Metadaten mit dem L^AT_EX PDF Management Support

In den letzten Jahren hat das L^AT_EX Project Team im Rahmen des Tagged-PDF-Projektes [5] Code geschrieben und veröffentlicht, der diverse PDF-bezogene Aufgaben angeht. Ein Teil dieses Codes ist schon über das Modul l3pdf der L3-Programmierschicht im Kernel enthalten [8], während der Rest derzeit durch das externe Bundle pdfmanagement-testphase bereitgestellt wird [9].

Dieser Code hat zum einem das Ziel, abstrahierende, vom Backend unabhängige Befehle bereitzustellen. Anstatt beispielsweise die PDF-Version abhängig von der Engine mit `\pdfminorversion`, `\pdfvariable minorversion` oder mit einem `\special` zu setzen, können Sie `\pdf_gset_version:n{1.7}` bei allen unterstützten Backends nutzen. `hyperref` macht sich das schon zunutze: Ist das PDF-Management aktiv, wird der gleiche generische Treiber für alle Backends geladen.

Ein zweites Ziel des neuen Codes (und der Grund, warum er *PDF-Management* genannt wird) ist es, Konflikte wie den oben erwähnten zwischen `hyperxmp` und `pdfx` zu vermeiden, indem Schnittstellen für einen *gemanagten* Zugriff auf zentrale PDF-Ressourcen angeboten werden. Versuchen beispielsweise zwei Pakete, eine `/Metadata`-Referenz oder andere Ressourcen zum Katalog hinzuzufügen, wird der Code sicherstellen, dass der Konflikt korrekt aufgelöst wird – entweder, indem die Ressourcen verschmolzen werden, oder indem eine von beiden abgelehnt wird.

Das PDF-Management funktioniert nicht mit Paketen, die die neuen Schnittstellen durch den Einsatz von primitiven Befehlen umgehen. Daher sind weder `hyperxmp` noch `pdfx` kompatibel und es war ein Ersatz für das Hinzufügen der wichtigen XMP-Metadaten erforderlich. Als Zwischenlösung wurde `hyperxmp` gepatcht, aber im Rahmen von Aufgabe 2.3.4 der Machbarkeitsstudie [5] wurde dieser Patch nun durch eine saubere Unterstützung im `l3pdfmeta`-Paket des `pdfmanagement-testphase`-Bundles ersetzt.

Designziele für die Unterstützung von XMP-Metadaten

Nach dem Begutachten der bestehenden Pakete wurden die folgenden Designziele für die neue Unterstützung von XMP-Metadaten ermittelt:

- Die Abhängigkeit zu `hyperref` sollte verschwinden. XMP-Metadaten hängen nicht direkt mit Links und anderen interaktiven Features zusammen und sollten daher auch in Dokumenten funktionieren, die diese nicht einsetzen.

Listing 4: Minimaleingabe für XMP-Metadaten mit dem PDF-Management

```

\DocumentMetadata {
  %xmp=false, % kein XMP
  xmp=true    % optional, da Default
}
\documentclass{article}
\begin{document}  Hello World!  \end{document}

```

- Die Eingabe von Benutzerdaten sollte über ein Schlüssel/Wert-System mit `\DocumentMetadata` als Standardschnittstellenbefehl erfolgen. Um den Übergang von `hyperxmp` zu erleichtern, sollten die `\hypersetup`-Schlüssel wenn möglich weiter funktionieren.
- Die Eingabe von Benutzerdaten sollte den vollständigen Unicode-Umfang und Standard-Befehle unterstützen.
- Der Standardsatz unterstützter XMP-Tags sollte dem der bestehenden Pakete ähneln.
- Es sollte in der Benutzereingabe nicht notwendig sein, Spezialbefehle wie `\xmpcomma`, `\XMLlangalt` oder `\sep` verwenden zu müssen. Listeneinträge sollten als kommaseparierte Liste übergeben werden, wobei »echte Kommas« wie üblich durch geschweifte Klammern geschützt werden. Die Sprachvarianten lassen sich über optionale Argumente setzen.
- Wie bei `pdfx` sollte es möglich sein, die XMP-Metadaten in ein externes Dokument zu exportieren, das sollte aber nur eine Debug-Option sein.
- Die XMP-Metadaten sollten erweiterbar sein. Als *Proof of Concept* soll ein Beispieldokument entwickelt werden, das zeigt, wie man die für ein ZUGFeRD-Dokument erforderlichen Metadaten hinzufügt.

Implementieren der Designziele

Die neue Unterstützung für XMP-Metadaten wurde im Modul `l3pdfmeta` implementiert und sie wird zusammen mit dem Code zum PDF-Management geladen. Das geschieht durch die Verwendung des Befehls `\DocumentMetadata` mit Schlüssel/Wert-Paaren am Anfang des Dokuments. XMP-Metadaten werden dann automatisch zum PDF hinzugefügt – das lässt sich verhindern, indem man den Schlüssel `xmp` auf `false` setzt (siehe Listing 4). Das funktioniert mit allen Engines, die das L3 Layer Backend unterstützen.

Entsprechend der weiter oben angegebenen Ziele ist für das Minimaldokument das Laden von `hyperref` nicht erforderlich: Der Code verlässt sich auf `\text_purify:n` und andere Funktionen aus der L3-Schicht, um die Eingabedaten zu bereinigen.

Die für einen PDF-Standard⁶ nötigen XMP-Metadaten, die PDF-Version und die Sprache werden schon automatisch aus Schlüsseln übernommen, die in `\DocumentMetadata` gesetzt werden:

```
\DocumentMetadata { pdfstandard = a-2b, pdfversion = 1.7, lang = de }
```

Für andere Metadaten ist aktuell noch `hyperref` und die `\hypersetup`-Schnittstelle mit den `hyperxmp`-Schlüsseln erforderlich, die aus Listing 2 bekannt sind – wir haben noch nicht entschieden, wie passende Schlüssel im Befehl `\DocumentMetadata` benannt und organisiert werden sollen.

Wie schon in den Designzielen erwähnt, werden Listen durch Kommas getrennt übergeben, wobei echte Kommas wie üblich durch geschweifte Klammern geschützt werden. Wo es sinnvoll ist, ist es möglich, ein Sprach-Tag durch ein optionales Argument vor dem Eintrag festzulegen. Das nächste Listing zeigt beides für den Titel:

```
\hypersetup {pdftitle = {[en]Baking,[de]{Kekse, Kuchen und Torten backen}}} }
```

Das führt zu diesen Metadaten:

```
<dc:title>
  <rdf:Alt>
    <rdf:li xml:lang="en">Baking</rdf:li>
    <rdf:li xml:lang="de">Kekse, Kuchen und Torten backen</rdf:li>
  </rdf:Alt>
</dc:title>
```

Die XMP-Metadaten lassen sich mit einer `debug`-Einstellung in `\DocumentMetadata` in eine externe Datei exportieren (der Standardname ist `\jobname.xmpi`):

```
\DocumentMetadata { debug = { xmp-export, mehr Debug-Optionen... } }
```

Schließlich wurden die ersten Schritte unternommen, Schnittstellen für die Erweiterung der XMP-Metadaten einzubauen. Um beispielsweise den ZUGFeRD-Standard zu unterstützen, müssen die XMP-Metadaten an drei Stellen ergänzt werden:

- Es muss ein neuer XML-Namensraum mit einem passenden Präfix deklariert werden.
- Es muss ein neues Schema mit Deklarationen für die neuen Tags zum Abschnitt `pdfaExtension:schemas` hinzugefügt werden.
- Und die Daten selbst dürfen natürlich nicht vergessen werden.

⁶ Wie bei pdfx kann das Deklarieren eines Standards weitere Effekte haben – zum Beispiel wird ein Farbprofil eingebettet oder es werden Validierungen vorgenommen. Beachten Sie aber, dass L^AT_EX nicht alle Anforderungen eines Standards sicherstellen oder auch nur überprüfen kann. Dafür sollte ein externer Validator wie veraPDF [11] genutzt werden.

Für all diese Aufgaben wurden interne Funktionen definiert und es gibt einen ersten Prototyp, der den ZUGFeRD 2.2-Standard implementiert, aber es ist noch nicht klar, wie genau die öffentlichen Funktionen dazu aussehen sollen.

Stand und Ausblick

Der neue Code unterstützt XMP-Metadaten auf einer Stufe, die mit den bestehenden Paketen vergleichbar ist. So gut wie alle oben angegebenen Designziele sind schon implementiert. Offen sind noch einige Arbeiten zu öffentlichen Schnittstellen. Zudem sind mehr Tests mit PDF-Viewern und -Validatoren notwendig, um zu prüfen, ob deren Interpretation der Standards mit dem Code übereinstimmen. Feedback und Kommentare sind erwünscht!

Literatur

- [1] Ulrike Fischer, Frank Mittelbach: »Adding XMP metadata in L^AT_EX«, *TUGboat*, 43.3 (135 2022), 263–267, ISSN: 0896-3207, DOI 10.47397/tb/43-3/tb135fischer-xmp.
- [2] Forum for Electronic Invoicing Germany: What is ZUGFeRD?, <https://www.ferd-net.de/standards/what-is-zugferd>.
- [3] Hàn Thế Thành u. a.: The pdfx package, PDF/X and PDF/A support for pdfT_EX, LuaT_EX and X_YT_EX, Version 1.6.3, <https://ctan.org/pkg/pdfx> (besucht am 7. 10. 2022).
- [4] International Organization for Standardization: ISO 16684-1:2019: Graphic technology — Extensible metadata platform (XMP) specification — Part 1: Data model, serialization and core properties, 2nd, 2019, <https://www.iso.org/obp/ui/#!iso:std:75163:en> (besucht am 5. 1. 2020).
- [5] Frank Mittelbach, Ulrike Fischer, Chris Rowley: L^AT_EX Tagged PDF Feasibility Evaluation, L^AT_EX Project, 2020, <https://latex-project.org/publications/2020-tagged-pdf-feasibility.pdf> (besucht am 8. 12. 2020).
- [6] Scott Pakin: The hyperxmp package, Embed XMP metadata within a L^AT_EX document, Version 5.9, 2020, <https://ctan.org/pkg/hyperxmp> (besucht am 7. 10. 2022).
- [7] PDF competence center: TechNote 0008: Predefined XMP Properties in PDF/A-1, Techn. Ber., 2008, <https://www.pdfa.org/resource/technical-note-tn0008-predefined-xmp-properties-in-pdf-a-1/>.
- [8] L^AT_EX Project Team: The l3kernel package, L^AT_EX3 programming conventions, 2022, <https://ctan.org/pkg/l3kernel> (besucht am 7. 10. 2022).

- [9] — The pdfmanagement-testphase package, L^AT_EX PDF management test-phase bundle, Version 0.95s, 2022, <https://ctan.org/pkg/pdfmanagement-testphase> (besucht am 7. 10. 2022).
- [10] Maarten Sneepe: The xmpincl package, Include eXtensible Metadata Platform data in pdfL^AT_EX, Version 2.4, <https://ctan.org/pkg/xmpincl> (besucht am 7. 10. 2022).
- [11] veraPDF consortium: veraPDF—Industry Supported PDF/A Validation, <https://verapdf.org> (besucht am 11. 10. 2022).
- [12] W3C: W3C RDF validation service, <https://www.w3.org/RDF/Validator/> (besucht am 11. 10. 2022).

L^AT_EX News – Issue 36, November 2022¹

Frank Mittelbach

Einleitung

Beim L^AT_EX-Release 2022-11 handelt es sich größtenteils um ein Konsolidierungs-Release, bei dem wir eine Reihe kleinerer Änderungen vorgenommen haben, um Fehler zu beheben und die eine oder andere Schnittstelle zu verbessern.

Die einzig wirklich wichtige neue Funktionalität ist im nächsten Abschnitt beschrieben – die Fähigkeit, Befehle und Umgebungen auf Dokumentenebene zu definieren, die eine Schlüssel/Wert-Liste in einem (meist optionalen) Argument akzeptieren, einschließlich der Möglichkeit, zu prüfen, ob das Argument tatsächlich aus solch einer Schlüssel/Wert-Liste oder aus einem »klassischen« Wert besteht.

Für das »L^AT_EX Tagged PDF Project« ist diese Funktionalität sehr wichtig, weil viele Befehle auf Dokumentenebene solche Schlüssel/Wert-Listen nutzen können müssen – zum Beispiel, um alternativen Text anzugeben oder ein Standard-Tagging zu überschreiben, wenn das in einem Dokument wichtig ist.

Schlüssel/Wert-Argumente automatisch erkennen

Um ein Erweitern der zentralen L^AT_EX-Syntax zu ermöglichen, unterstützt `ltxcmd` nun einen `=...`-Modifikator beim Interpretieren von Argumenten. Dieser Modifikator

¹ Der Newsletter wurde zuvor im *TUGboat* 43-3 [2] veröffentlicht und von Thomas Demmig übersetzt.

weist L^AT_EX an, dass das Argument als eine Liste mit Schlüssel/Wert-Paaren an den nachfolgenden Code übergeben werden soll. Sieht das Argument nicht wie eine derartige Liste aus (enthält z. B. keine =-Zeichen), wird es in ein einzelnes Schlüssel/Wert-Paar umgewandelt, wobei der Name des Schlüssels durch das Argument des =-Modifikators festgelegt ist. So könnte beispielsweise der Befehl `\caption` wie folgt definiert werden:

```
\DeclareDocumentCommand\caption {s ={short-text}+0{#3} +m}
{...}
```

Das würde dazu führen, dass das optionale Argument, wenn es *keine* Schlüssel/Wert-Paare enthält, in ein einzelnes Schlüssel/Wert-Paar mit dem Schlüsselnamen `short-text` umgewandelt wird. Der nachfolgende Code kann sich also immer darauf verlassen, eine Schlüssel/Werte-Liste zu erhalten.

Argumente, die mit = beginnen, werden immer als Schlüssel/Wert-Paare interpretiert, auch wenn sie keine weiteren Gleichheitszeichen enthalten. Alle Gleichheitszeichen, die durch `$...$` oder `\(...\)` umschlossen sind – also im Inline-Mathematikmodus – werden ignoriert, so dass im Allgemeinen nur = außerhalb des Mathematikmodus für eine Interpretation als Schlüssel/Wert-Material sorgen.

Enthält das Argument ein Gleichheitszeichen, das fälschlicherweise als Schlüssel/Wert-Indikator betrachtet wird, können Sie dies durch eine Geschweifte-Klammer-Gruppe verbergen, so wie Sie es auch an anderer Stelle machen würden, beispielsweise:

```
\caption[{Verwendung von ==Zeichen}]
{Verwendung von ==Zeichen in optionalen Argumenten}
```

Aber weil Gleichheitszeichen im Mathematikmodus schon ignoriert werden, sollte das selten notwendig sein.

Ein Hinweis für Entwickler von Font-Paketen

Encoding-Subsets für per TS1-kodierte Fonts

Die Text-Companion-Kodierung TS1 wird leider von Fonts, die keine enge Beziehung zu Computer-Modern-Fonts haben, nur schwach unterstützt. Es war daher erforderlich, die Idee von »Sub-Encodings« pro Font einzuführen. Diese Sub-Encodings werden für eine Schriftfamilie mit Hilfe einer Deklaration `\DeclareEncodingSubset` deklariert – Details dazu finden Sie in [3].

Maintainer von Font-Bundles, die Font-Dateien mit per TS1 kodierten Fonts enthalten, sollten eine passende Deklaration in der zugehörigen Datei `ts1family.fd` ergänzen, da ansonsten das Standard-Subencoding angenommen wird, das wahr-

scheinlich zu viele Glyphen deaktiviert, die im Font eigentlich vorhanden sind.²
(*Github Issue 905*)

Neue oder verbesserte Befehle

Bessere Berücksichtigung der Sprache bei Befehlen zum Verändern von Groß- und Kleinschreibung

Die Befehle `\MakeUppercase`, `\MakeLowercase` und `\MakeTitlecase` erkennen nun automatisch das verwendete Locale, wenn `babel` geladen wird. Das ermöglicht ein automatisches Anpassen des Buchstaben-Mappings, sofern das erforderlich ist. Außerdem ist nun ein führendes optionales Argument möglich. Dieses erwartet eine Schlüssel/Wert-Liste mit Steuerungsparametern. Aktuell gibt es einen Schlüssel: `locale`, der sich auch über den Alias `lang` nutzen lässt. Er ist dazu gedacht, lokal eine Sprache zu setzen, wobei ein BCP-47-Deskriptor zum Einsatz kommt. So ließe sich beispielsweise eine türkischsprachige Umwandlung in einer ansonsten deutschsprachigen Umgebung erzwingen:

```
\MakeUppercase[lang = tr]{Ragıp Hülûsi Özdem} % oder locale = tr
```

liefert RAGIP HULÛSÎ ÖZDEM, während man ohne das optionale Argument die inkorrekte Ausgabe RAGIP HULÛSİ ÖZDEM erhält.

Code-Verbesserungen

Unterstützung für schräggestellte Kapitälchen in den EC-Fonts

Seit einiger Zeit unterstützt L^AT_EX die Kombination aus Kapitälchen und kursiv/schräg. Die EC-Fonts enthalten schräggestellte Kapitälchen, aber für deren Einsatz musste ein externes Paket geladen werden. Es wurden nun entsprechende Font-Definitionen in `t1cmd.fd` aufgenommen, so dass folgender Code:

```
\usepackage[T1]{fontenc}
...
\textsc{\textsl{Schräge Kapitälchen}} und
\textsc{\textit{Kursive Kapitälchen}};
sowie \bfseries
\textsc{\textsl{Fette schräge Kapitälchen}} und
\textsc{\textit{Fette kursive Kapitälchen}}.
```

² Das L^AT_EX-Format enthält schon Deklarationen für viele Schriftfamilien. Das geschah im Jahr 2020, um diese Symbole direkt nutzbar zu machen, ohne dass zuerst externe Font-Pakete angepasst werden mussten, aber eigentlich ist das der falsche Ort für solche Deklarationen. Daher sollten diese Deklarationen für neue Schriften in den entsprechenden `.fd`-Dateien untergebracht werden.

das erwartete Ergebnis liefert: *SCHRÄGE KAPITÄLCHEN* und *KURSIVE KAPITÄLCHEN*; sowie **FETTE SCHRÄGE KAPITÄLCHEN** und **FETTE KURSIVE KAPITÄLCHEN**.

Da die Computer-Modern-Fonts in T1 keine echten kursiven Kapitälchen, sondern nur schräge Kapitälchen haben, werden die kursiven automatisch durch schräge Kapitälchen ersetzt. Darum funktioniert obiger Code, aber es gibt keinen Unterschied zwischen beiden Formen (und Sie erhalten eine Substitution-Warning für die Kombination `\textit\textsc`). (Github Issue 782)

EC Sans Serif bei kleinen Größen

Die EC-³-Sans-Serif-Fonts enthalten Fehler bei kleinen Größen: Die Medium-Schriftstärke ist fetter und weiter als Bold Extended. Das sorgt dafür, dass die Fonts in diesen kleinen Größen unbenutzbar sind. Daher wurde die Standard-`.fd`-Datei angepasst, damit stattdessen ein herunterskalierter 8-Punkt-Font zum Einsatz kommt. (Github Issue 879)

Font-Series-Handling bei falschen `.fd`-Dateien verbessern

In Fontdefinitionsdateien (`.fd`) gilt die Konvention, dass der Wert für die Font Series kein `m` enthält, sofern man sich nicht auf die »medium«-Series bezieht (was durch ein einzelnes `m` dargestellt wird). So sollte man `c` für »medium weight, condensed width« und nicht `mc` schreiben. Das war in den frühen Tagen von L^AT_EX 2_ε eine der vielen (wichtigen) Methoden zum Platz sparen.

Manche ältere `.fd`-Dateien folgen dieser Konvention nicht und verwenden in ihren Deklarationen `mc`, `bm` usw. Deshalb hat mancher Code zum Auswählen eines Font-Schriftschnitts nicht korrekt funktioniert, wenn er sich mit solchen `.fd`-Dateien konfrontiert sah. Wir haben daher `\DeclareSymbolFont` und `\SetSymbolFont` so erweitert, dass alle überzähligen `m` aus ihrem Series-Argument entfernt werden, damit sie keine unnötige Font-Substitution auslösen. Unabhängig davon sollten solche `.fd`-Dateien durch ihre Maintainer korrigiert werden. (Github Issue 918)

Verschachtelte `minipage`-Umgebungen erkennen

Das Verschachteln von `minipage`-Umgebungen wird in L^AT_EX nur teilweise unterstützt und es kann zu einer fehlerhaften Ausgabe führen, wie zum Beispiel übervollen Boxes oder Fußnoten, die an der falschen Stelle erscheinen – siehe [1, S. 106].

³ T1-kodierte Computer Modern

Aber bisher gab es keine Warnung, wenn das geschah. Dies wurde nun geändert und die Umgebung gibt eine Warnung aus, wenn Sie sie in einer anderen minipage-Umgebung verpacken, die schon Fußnoten enthält. *(Github Issue 168)*

Robuste Befehle in Paketoptionen

Mit dem im letzten Release hinzugefügten Schlüssel-basierten Options-Handler oder mit Fremd-Paketen, die ähnliche Features ermöglichen, gehen Benutzer eventuell davon aus, dass sie eine Paketoption wie `[font=\bfseries]` nutzen können. Das ging bisher mit internen Fehlern schief, da die Optionsliste durch `\edef` expandiert wurde. Dies wurde nun geändert und der schon vorhandene Befehl `\protected@edef` kommt nun zum Einsatz, so dass man jeden robusten L^AT_EX-Befehl im Wert eines Schlüssels verwenden kann. *(Github Issue 932)*

l3docstrip-Integration in docstrip verbessern

2020 haben wir die Funktionalität von `l3docstrip.tex` in `docstrip.tex` integriert, um auch die Syntax `%<@=<module>` von `expl3` zu unterstützen – siehe [4]. Allerdings war diese Unterstützung unvollständig, weil sie keine `docstrip`-Zeilen der Form `%<+...>` oder `%<-...>` abgedeckt hat. Das wurde erst kürzlich erkannt, weil normalerweise `%<*...>`-Blöcke verwendet werden. Jetzt werden alle Zeilen in einer `.dtx`-Datei beim Ersetzen durch `@@` berücksichtigt. *(Github Issue 903)*

Verbessern der Effizienz von LuaT_EX-Callbacks

Der Mechanismus zum Bereitstellen der `pre/post_mlist_to_hlist_filter`-Callbacks in LuaT_EX wurde verbessert, um ihn besser wiederverwenden zu können und einen Overhead zu vermeiden, wenn diese Callbacks nicht genutzt werden. *(Github Issue 830)*

Regelbasierte Reihenfolgen für LuaT_EX-Callback-Handler

In LuaL^AT_EX wurden die Callback-Handler in der Reihenfolge aufgerufen, in der sie registriert wurden, aber das war oft eine eher fragile Sache. Es hing viel von der Ladereihenfolge ab und alle Versuche, eine andere Reihenfolge zu erzwingen, erforderten ein De- und Re-Registrieren der umzuordnenden Handler. Und selbst wenn Einschränkungen bei der Reihenfolge auf diese Art und Weise erzwungen werden sollten, konnte ein anderer Paket-Loader diese später unabsichtlich überschreiben.

Zum Verbessern haben wir die Callback-Handler nun anhand von Sortierregeln angeordnet, die denen der Hook-Regeln ähnlich sind.

Beim Registrieren eines Callbacks, der vor oder nach einem anderen Callback laufen soll, lässt sich nun `luatexbase.declare_callback_rule` verwenden, um diese Reihenfolge festzulegen. Zum Beispiel stellt

```
luatexbase.add_to_callback ('pre_shaping_filter', my_handler, 'my_name')
luatexbase.declare_callback_rule
    ('pre_shaping_filter', 'my_name', 'before', 'other_name')
```

sicher, dass `my_handler` immer vor dem Handler aufgerufen wird, der als `other_name` registriert wird.

Das bedeutet auch, dass die Reihenfolge, in der Callbacks registriert werden, nicht mehr länger implizit eine Ordnung definieren. Code, der sich auf diese implizite Reihenfolge verlässt, sollte nun die Reihenfolgeregeln explizit definieren.

Fehlerkorrekturen

T_EX davon abhalten, ein `\smash` zu verlieren

Setzt T_EX einen Bruch, wird es das Material im Zähler oder Nenner neu in einer Box verpacken – abhängig davon, welcher breiter ist. Besteht der neu verpackte Teil aus einer einzelnen Box, erhält diese neue Dimensionen, wurde er mit einem `\smash` erstellt, löste sich dessen Effekt in Wohlgefallen auf (weil ein `Smash` nichts anderes macht, als die Dimensionen einer Box auf Null zu setzen, was nun nicht mehr wirksam war). So befanden sich beispielsweise in

```
\frac{1}{2} = \frac{1}{\smash{2^X}} \neq \frac{100}{\smash{2^X}}
```

die 2 im Nenner nicht immer an der gleichen vertikalen Position, weil das zweite `\smash` aufgrund des Reboxing ignoriert wurde:

$$\frac{1}{2} = \frac{1}{2^X} \neq \frac{100}{2^X}$$

Die Unterschiede sind subtil, aber bemerkbar. Das wurde nun korrigiert und das `\smash` wird immer berücksichtigt. Sie erhalten daher diese Ausgabe:

$$\frac{1}{2} = \frac{1}{2^X} \neq \frac{100}{2^X}$$

(Github Issue 517)

Ein Problem mit `\mathchoice` und `localalphabets` lösen

Der Code, mit dem sich eine Reihe von Mathematik-Alphabeten lokal halten ließ (eingeführt im Jahr 2021, siehe [5]), hat `\aftergroup` genutzt, um aufzuräumen, nachdem eine Formel abgeschlossen wurde. Leider lässt sich `\aftergroup` nicht in einem Argument des Primitivs `\mathchoice` verwenden, weshalb es Low-Level-Fehler gab, wenn das Freezing an solch einer Stelle geschah. Die Implementierung wurde daher überarbeitet, um das `\aftergroup` komplett zu vermeiden. *(Github Issue 921)*

Ungenutzte globale Optionen beim Verarbeiten von Schlüssel/Wert-Paaren melden

Beim Verwenden des neuen Schlüssel/Wert-Prozessors wurden ungenutzte globale Optionen nicht ordentlich gemeldet, wenn sie für Klassenoptionen zum Einsatz kamen. Das wurde jetzt korrigiert. *(Github Issue 938)*

Änderungen an Paketen in der `graphics`-Kategorie

Einen Fehler bei `\mathcolor` beheben

Der Befehl `\mathcolor` (eingeführt in [6]) muss auf folgende Hoch- und Tiefstellungen scannen, aber wenn er das am Ende einer Alignment-Zelle tat, zum Beispiel in einer array-Umgebung, wurde das `&` zu früh ausgewertet, was zu internen Fehlern führte. Das wird nun korrekt abgefangen. *(Github Issue 901)*

Änderungen an Paketen der `tools`-Kategorie

`array`: Einzeilige m-Zellen korrekt erkennen

Zellen in m-Spalten, die nur eine einzelne Zeile enthalten, sollen sich wie einzeilige p-Zellen verhalten und an der Grundlinie ausgerichtet werden. Um das zu prüfen, hat `array` die Höhe der Zelle mit der Höhe der für die Tabellenzeilen genutzten `Strut` verglichen. Aber die Höhe dieser `Strut` hängt von den Einstellungen für `\arraystretch` ab und wenn Sie diese negativ (oder sehr groß) gemacht haben, war der Vergleich fehlerhaft. Daher prüfen wir nun gegen die Höhe einer normalen `Strut`, um sicherzustellen, dass einzeilige Zellen korrekt als solche erkannt werden (sofern ihr Inhalt nicht wirklich sehr hoch ist – dann ist ein Ausrichten aber sowieso sinnlos). *(Github Issue 766)*

Literatur

- [1] Leslie Lamport: *L^AT_EX: A Document Preparation System: User's Guide and Reference Manual*, Reprinted with Corrections in 1996, Addison-Wesley, Reading, MA, USA, 2nd Edition, 1994, ISBN: 0-201-52983-1.
- [2] »L^AT_EX News 36«, *TUGboat*, 43.3 (2022), 273–275, ISSN: 0896-3207, <https://tug.org/TUGboat/tb43-3/tb1351tnews36.pdf>.
- [3] L^AT_EX Project Team: L^AT_EX 2_ε Font Selection, <https://latex-project.org/help/documentation/>.
- [4] L^AT_EX Project Team: L^AT_EX 2_ε News 32, <https://latex-project.org/news/latex2e-news/1tnews32.pdf>.
- [5] — L^AT_EX 2_ε News 34, <https://latex-project.org/news/latex2e-news/1tnews34.pdf>.
- [6] — L^AT_EX 2_ε News 35, <https://latex-project.org/news/latex2e-news/1tnews35.pdf>.

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss in umgekehrter chronologischer Reihenfolge vor. Bloße Updates können auf der moderierten *CTAN-ann*-Mailingliste oder als RSS-Feed auf <https://ctan.org/> verfolgt werden.

writeongrid von *Cédric Pierquet* ermöglicht es, Text mit L^AT_EX auf ein Raster zu platzieren.

CTAN:macros/latex/contrib/writeongrid

zennote von *Yifan Liang* stellt einfache Befehle bereit, um Notizen zu setzen.

CTAN:macros/latex/contrib/zennote

scrabble von *Cédric Pierquet* dient zum Setzen von Scrabble-Spielen. Die Angabe von Begriffen im Klartext beim Spiel ist optional, und es gibt eine sparsame Lokalisierung für Deutsch, Englisch, Französisch und Spanisch.

CTAN:graphics/pgf/contrib/scrabble

jwjournal von *Jinwen Xu* ist eine Klasse, die der Autor entwickelt hat, um sein privates Tagebuch zu schreiben. Notizen in einer einfachen Syntax in reinem Text werden in L^AT_EX-Befehle gewandelt. Überschriften werden in einer bestimmtem

Reihenfolge nach dem Wochentag farbig hinterlegt.

CTAN:macros/latex/contrib/jwjournal

tlc3-examples von *Frank Mittelbach* enthält die PDF-Dateien und die dazugehörigen Quelltexte der Beispiele aus der dritten Auflage des *L^AT_EX Companion*, der im April/Mai bei Addison-Wesley erscheinen soll.

CTAN:info/examples/tlc3

gfdl von »*niranjana*« soll dabei helfen, ein Dokument oder ein Paket korrekt mit der *GNU Free Documentation License* (GFDL) zu versehen.

CTAN:macros/latex/contrib/gfdl

luagcd von *Chetan Shirore* und *Ajit Kumar* bestimmt den größten gemeinsamen Teiler zweier ganzer Zahlen nach dem Euklidischen Algorithmus mithilfe von Lua^L_AT_EX.

CTAN:macros/luatex/latex/luagcd

blopentye von *Luis Rivera* ist der bisher bei Pi^T_EX fehlende OpenType-Font-Loader. Er sollte zumindest mit Plain T_EX funktionieren.

CTAN:macros/luatex/generic/blopentye

luacomplex von *Chetan Shirore* und *Ajit Kumar* setzt auf dem Paket *luamaths* auf und dient dazu, komplexe Zahlen zu definieren und grundlegende Rechenoperationen mit ihnen durchzuführen.

CTAN:macros/luatex/latex/luacomplex

maze von *Sicheng Du* erstellt quadratische Labyrinth mit zufälligen Mustern, die immer von unten links nach oben rechts zu durchlaufen sind. Die Darstellung erfolgt mit einer *picture*-Umgebung.

CTAN:macros/latex/contrib/maze

resmes von *Paolo De Donato* erzeugt ein mathematisches Maßbeschränkungszeichen.

CTAN:macros/latex/contrib/resmes

luaset von *Chetan Shirore* und *Ajit Kumar* stellt einige Mengen-Operationen bereit, die mit Lua^L_AT_EX in einem Dokument ausgeführt werden können.

CTAN:macros/luatex/latex/luaset

luamaths von *Chetan Shirore* und *Ajit Kumar* stellt einige mathematische Operationen bereit, die mit Lua^L_AT_EX in einem Dokument ausgeführt werden können.

CTAN:macros/luatex/latex/luamaths

luamodulartables von *Chetan Shirore* und *Ajit Kumar* erstellt konfigurierbare Tabellen zur modularen Multiplikation und Addition positiver ganzer Zahlen mit Lua^L_AT_EX.

CTAN:macros/luatex/latex/luamodulartables

tikz-mirror-lens von *Fernando H. G. Zucatelli* berechnet und zeichnet den Strahlengang bei sphärischen Linsen und Spiegeln.

CTAN:graphics/pgf/contrib/tikz-mirror-lens

gradient-text von *Sicheng Du* ermöglicht es, Text linear verlaufend farbig auszugeben. Ausgangs- und Endfarbe können mit RGB-Werten angegeben werden, der Wert dazwischen wird automatisch gesetzt.

CTAN:macros/latex/contrib/gradient-text

pythonimmediate von »*user202729*« ist eine weitere Bibliothek, um Python auszuführen. Auch Fehler werden aufgefangen.

CTAN:macros/latex/contrib/pythonimmediate

tsvtemplate von *Esgert Renkema* liest TSV-Dateien (Tabellen mit *tab-separated values*) mithilfe von Lua $\text{\LaTeX}$ und Plain Lua $\text{\LaTeX}$ ein.

CTAN:macros/luatex/generic/tsvtemplate

xput von *Friedemann Bartels* ist ein Paket, mit dem man Dokumente mit einem hohem Bildanteil besser gestalten kann. Dazu wird der Befehl `\put` neu definiert, so dass er ähnlich wie beim Desktop Publishing verwendet werden kann.

CTAN:macros/latex/contrib/xput

ibrackets von *Antoine Missier* definiert eckige Klammern im Mathematikmodus neu, um den sie umgebenden Weißraum bei offenen Intervallen anzupassen.

CTAN:macros/latex/contrib/ibrackets

pst-flags von *Amit M. Manthanwar* zeichnet die Flaggen vieler Staaten mittels PSTricks.

CTAN:graphics/pstricks/contrib/pst-flags

namedtensor von *David Chiang* setzt die *named tensor notation* in $\text{\LaTeX}$ um, die in dem gleichnamigen Aufsatz von *David Chiang*, *Alexander M. Rush* und *Boaz Barak*, <https://doi.org/10.48550/arXiv.2102.13196> vorgestellt wurde.

CTAN:macros/latex/contrib/namedtensor

pdfmsym von »*Slurp*« verbessert und erweitert die mathematischen Symbole, die in $(\text{\LaTeX})$ schon vorhanden sind, und greift dabei auf die Tools zurück, die zum Zeichnen bereits in PDF integriert wurden. Es funktioniert mit pdf $\text{\TeX}$ und Lua $\text{\TeX}$, teilweise auch mit X $\text{\TeX}$.

CTAN:macros/generic/pdfmsym

bibcop von *Yegor Bugayenko* ist ein Validator für Bib $\text{\TeX}$ -Dateien, der aus dem $\text{\LaTeX}$ -Dokument heraus gestartet werden kann. Das Perl-Skript kann zwar auch separat aufgerufen werden, das Paket funktioniert aber nicht unter Windows.

CTAN:macros/latex/contrib/bibcop

scikgtex von *Christof Bless* übernimmt mithilfe von Lua $\text{\LaTeX}$ semantisch ausgezeichnete Textteile aus einem Dokument und fügt sie dem PDF-Dokument im

XML-Format XMP hinzu. Diese strukturierten Daten können maschinell aus dem PDF extrahiert und in den Open Research Knowledge Graph (ORKG) übernommen werden, der im Rahmen der Nationalen Forschungsdateninfrastruktur (NFDI) in Deutschland aufgebaut wird. Das Projekt dient zur verteilten inhaltlichen Erschließung von grauer Literatur und Aufsätzen. Dabei werden Metadaten erzeugt, die die Titeldaten und die Normdaten, die parallel dazu bei ORCID und (redaktionell gepflegt) in der Gemeinsamen Normdatei GND vorliegen, ergänzen. Dadurch kann das Retrieval und damit die Auffindbarkeit von wissenschaftlicher Literatur in Katalogen, Fachportalen und Suchmaschinen verbessert werden.

CTAN:macros/luatex/latex/sciimgtex

jourcl von *Rok Kukovec* dient zum Gestalten eines Deckblatts für die Einreichung eines Aufsatzes bei einer wissenschaftlichen Zeitschrift.

CTAN:macros/latex/contrib/jourcl

hfutexam von *Shenxing Zhang* ist eine Klasse für Abschlussarbeiten an der (chinesischen) Technischen Universität Hefei.

CTAN:macros/latex/contrib/hfutexam

digestif von *Augusto Stoffel* ist ein Style Checker und Language Server für $\text{\LaTeX}$, plain $\text{\TeX}$, Con $\text{\TeX}$ t und Texinfo. Es funktioniert in jedem Editor, der mit dem Language Server Protocol zurechtkommt.

CTAN:support/digestif

ukbill von *Elijah Z. Granet* ist eine Klasse, die die Typografie nachahmt, die vom britischen Parlament für die Veröffentlichung von Gesetzen verwendet wird. Um das Schriftbild zu vervollständigen, wird ergänzend die freie Schriftart Palatine Parliamentary benötigt.

CTAN:macros/latex/contrib/ukbill

lgrmath von *Jean-François Burnol* sorgt dafür, dass im Mathematikmodus nur die LGR-kodierten Fonts für griechische Buchstaben verwendet werden, die der Benutzer ausdrücklich angegeben hat.

CTAN:macros/latex/contrib/lgrmath

crossrefenum von *Bastien Dumont* ist ein weiteres Paket, um die Standardbefehle von $\text{\LaTeX}$ und Con $\text{\TeX}$ t für Querverweise zu ergänzen und zu vereinfachen. Das Paket ist in Plain $\text{\TeX}$ geschrieben, so dass es auch an andere Formate angepasst werden kann.

CTAN:macros/generic/crossrefenum

hwemoji von »hwy« ergänzt Lua $\text{\LaTeX}$ um eine Unterstützung für Emojis einschließlich Emoji-Sequenzen.

CTAN:macros/latex/contrib/hwemoji

cvss von *Pierre Vivegnis* setzt das »Common Vulnerability Scoring System (CVSS)« zur »Bewertung des Schweregrades von möglichen oder tatsächlichen Sicher-

heitslücken in Computer-Systemen« (so die Definition in der deutschsprachigen Wikipedia) in $\text{\LaTeX}$ um.

CTAN:macros/latex/contrib/cvss

songproj von *Tanguy Ortoló* wurde ursprünglich für den Gottesdienst entwickelt, es ermöglicht aber allgemein das Projizieren von Liedertexten bei Veranstaltungen mit einer beamer-Präsentation. Beigefügt ist ein Python-Skript, um Liedertexte aus reinem Text mit $\text{\LaTeX}$ -Markup zu versehen.

CTAN:macros/latex/contrib/songproj

euler-math von *Daniel Flipo* enthält den Font Euler Math, eine OpenType-Version der Schriftart Euler von *Hermann Zapf*, die den Font Neo-Euler ersetzt. Das Projekt wurde ursprünglich von *Khaled Hosny* 2009 bis 2016 begonnen.

CTAN:fonts/euler-math

unimath-plain-xetex von *Zhang Tingxuan* enthält eine Unterstützung für OpenType-Mathematikschriften mit Plain $\text{\XeTeX}$.

CTAN:macros/xetex/plain/unimath-plain-xetex

luacas von *Timothy All* stellt ein Computeralgebra-System für $\text{\LuaTeX}$ bereit.

CTAN:macros/luatex/latex/luacas

hershey-mp von *Esger Renkema* enthält die METAPOST-Unterstützung für die Vektorschriften im Format jhf, die, soweit ersichtlich, nur von den Hershey Fonts in den späten 1960er-Jahren verwendet wurden.

CTAN:graphics/metapost/contrib/macros/hershey-mp

luaoptions von *Jacques Peron* und *Urs Liska* stellt Befehle für Optionen in $\text{\LuaTeX}$ -Paketen zur Verfügung. Optionen können auf mehreren Ebenen sowohl in einem Paket als auch lokal definiert und geändert werden.

CTAN:macros/luatex/latex/luaoptions

hep-reference von *Jan Hajer* enthält Anpassungen für das Einfügen von Literaturfundstellen und Fußnoten in Veröffentlichungen auf dem Gebiet der Teilchenphysik.

CTAN:macros/latex/contrib/hep-reference

pangram von *Tingxuan Zhang* ist ein Paket, mit dem man Schriften testen kann. Standardmäßig wird der bekannte Satz »The quick brown fox jumps over the lazy dog. 0123456789« in verschiedenen Schriftgrößen ausgegeben. Man kann aber auch ein deutsches Pangramm verwenden, das (außer dem ß und den Umlauten) alle Buchstaben der deutschen Sprache enthält, zum Beispiel: »Franz jagt im komplett verwahrlosten Taxi quer durch Bayern«.

CTAN:macros/latex/contrib/pangram

beamertheme-simpleplus von *Pin-Yen Huang* ist ein minimalistisches Theme für beamer-Präsentationen.

CTAN:macros/latex/contrib/beamer-contrib/themes/beamertheme-simpleplus

eolang von *Yegor Bugayenko* unterstützt das Setzen des φ -Kalküls und Surging-Object-DiGraph-Graphen (SODG) für die Programmiersprache EO.

CTAN:macros/latex/contrib/eolang

Spielplan

The 28th GUST T_EX conference

Jerzy Ludwichowski

Dear T_EX users and friends,

good news – come what may, there will be BachoT_EX 2023 as indeed I promised when announcing BachoT_EX 2022 cancellation.

The SARS-CoV-2 virus did not disappear but seems to have become less of a danger as far as one can tell. The war in Ukraine is raging still, but none the less we decided to organize BachoT_EX 2023 at the usual dates and venue.

We hope that for the fall and to make up for the lost time by keeping the same theme as the cancelled 2020 conference.

Please find at <https://gust.org.pl/bachotex/2023-en> the



*Call for Papers for BachoT_EX 2023
from 29th of April until 3rd of May, 2023
in Bachotek, Poland*

with the same theme as planned for BachoT_EX 2020 and 2022. Registrations will be announced separately as soon as possible.

Jerzy Ludwichowski
(for the Organizing Committee)

29. 4. – 3. 5. 2023 28th GUST T_EX conference

Bachotek (nahe Brodnica), Polen

<https://gust.org.pl/bachotex/2023-en>



13. 7. 2023 65. Mitgliederversammlung von DANTE e.V.

Hotel Collegium Leoninum

Noeggerathstraße 34

53111 Bonn

<https://www.dante.de/veranstaltungen/dante2023/>



14. 7. – 16. 7. 2023 TUG 2023 – The 44th Annual Meeting of the T_EX Users Group

Presentations covering the T_EX world Typography & typesetting, fonts & design, publishing and more

Hotel Collegium Leoninum

Noeggerathstraße 34

53111 Bonn

<https://www.tug.org/tug2023/>



28. 7. – 29. 7. 2023 BayT_EX 2023

Genua GmbH

Domagkstraße 7

85551 Kirchheim bei München

<https://www.ks-ingenieurconsult.de/TeX/Stammtisch.html>



10. 9. – 16. 9. 2023 ConT_EXt Meeting 2023

Prag-Sibřina, Tschechien

<https://meeting.contextgarden.net/2023/>



April 2024 (geplant) DANTE 2024

Wielandgut Oßmannstedt

Wielandstraße 16

99510 Ilmtal-Weinstraße, OT Oßmannstedt

Stammtische



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

Aachen

Torsten Bronger

bronger@physik.rwth-aachen.de

<https://lists.rwth-aachen.de/postorius/lists/tex-stammtisch.lists.rwth-aachen.de>

»Anvers«, Kockerellstr. 20, 52062 Aachen

Erster Donnerstag im Monat, 20:00 Uhr

Bad Doberan

Carsten Vogel

texnicer@web.de

zur Zeit inaktiv, Interessenten bitte per Mail melden

Berlin

Michael-E. Voges, Tel.: 0 33 62/ 50 18 35,

mevoges@t-online.de

»Mantée« – Café Restaurant, Chausseestr. 131, 10115 Berlin

Zweiter Donnerstag im Monat, 19:00 Uhr

Darmstadt

Karlheinz Geyer

geyerk@posteo.de

ab Frühjahr 2023 Neustart geplant. Interessenten können sich bei Karlheinz melden

Erlangen

Peter Seitz

p.seitz@KplusS-Ing.de

<https://www.ks-ingenieurconsult.de/TeX/Stammtisch.html>

Gaststätte »Deutsches Haus«, Luitpoldstr. 25, 91052 Erlangen

Dritter Dienstag im Monat, 19:00 Uhr

Frankfurt a. Main

Harald Vajkonny

vajkonny@t-online.de

zur Zeit inaktiv, Interessenten bitte per Mail melden

Göttingen

Holger Nobach

holger.nobach@nambis.de





<http://goetex.nambis.de/>

*Restaurant »Mazzoni Cucina Italiana«, Hermann-Rein-Straße 2, 37075 Göttingen
Dritter Donnerstag im Monat, 18:00 Uhr*

Hamburg

Günther Zander

guenther.zander@lug-balista.de

z. Zt. inaktiv. Bei Fragen steht Günther gern per Mail zur Verfügung.

Hannover

Reiko Kaps

kaps@luis.uni-hannover.de

<http://tex-hannover.de/>

Zweiter Donnerstag im Monat, 18:30 Uhr

online über <https://vc.sonia.de/TeXStammtischHannover>



Heidelberg

Martin Wilhelm Leidig, Tel.: 01 70 41 83 32 9,

moss@moss.in-berlin.de

Anmeldeseite zur Mailingliste: <https://tinyurl.com/stammtisch-HD>

Der Stammtisch bleibt bis auf Weiteres ausgesetzt.



Köln

Uwe Ziegenhagen

uwe@dante.de

zur Zeit inaktiv, Interessenten bitte per Mail melden

Leipzig

Erhard Pross

Erhard.Pross@gmx.de

nächstes Treffen am 11. Mai 2023 18:00 Uhr

le-tex publishing services GmbH, Weissenfelder Str. 84, 04229 Leipzig

München

Leah Neukirchen

leah@vuxu.org

1. Woche in geraden Monaten, wechselnde Wochentage und Orte, 19:00 Uhr

Info folgt über Mailingliste dante-ev@dante.de

Stuttgart

Bernd Raichle

bernd.raichle@gmx.de

Zweiter Dienstag im Monat, 19:30 Uhr

»Wichtel«, Bahnhofstr. 30, 70372 Stuttgart-Bad Cannstatt

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.

Postfach 11 03 61

69072 Heidelberg

Tel.: (0 62 21) 2 97 66

Fax: (0 62 21) 16 79 06

E-Mail: info@dante.de

Konto: VR Bank Rhein-Neckar eG

IBAN DE67 6709 0000 0002 3100 07 SWIFT-BIC GENODE61MA2

Vorstand

Vorsitzender:	Martin Sievers	president@dante.de
stv. Vorsitzender:	Uwe Ziegenhagen	vice-president@dante.de
Schatzmeisterin:	Doris Behrendt	treasurer@dante.de
Schriftführer:	Volker RW Schaa	secretary@dante.de
Beisitzer:	Klaus Höppner	
	Harald König	
	Stephan Lukasczyk	

Ehrenmitglieder

Peter Sandner	22.03.1990	Klaus Thull († 2012)	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 († 2015)	19.02.1999
Joachim Lammarsch	12.04.2014	Rainer Schöpf	12.04.2014

Webserver und Mailingliste

DANTE: <https://www.dante.de/> (Erik Braun)
CTAN: <https://mirror.ctan.org/> (Gerd Neugebauer)
DANTE-EV: <https://lists.dante.de/mailman/listinfo/dante-ev>

FAQ

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

T_EX: <https://projekte.dante.de/DanteFAQ/WebHome>

T_EXnische Fragen

beraterkreis@dante.de

ak-schule@dante.de

Autoren/Organisatoren

Luzia Dietsche 71394 Kernen dtkred@dante.de	[3]	Frank Mittelbach L ^A T _E X Project Team Mainz frank.mittelbach@latex-project.org	[28,38,47]
Jürgen Fenn Neu-Isenburg juergen.fenn@gmx.de	[53]	Henning Hraban Ramm hraban@fieee.net	[16,25]
Ulrike Fischer L ^A T _E X Project Team Bonn ulrike.fischer@latex-project.org	[28,38]	Martin Sievers siehe Seite 66	[4,6]
Jerzy Ludwichowski Jerzy.Ludwichowski@gust.org.pl	[59]	Keno Wehr wehr@abgol.de	[7]
		Uwe Ziegenhagen siehe Seite 66	[6]

Die T_EXnische Komödie

35. Jahrgang Heft 1/2023 Februar 2023

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 6 Veränderungen im Büro

Bretter, die die Welt bedeuten

- 7 L^AT_EX und Schulphysik 1: Größen und Einheiten
- 16 Buchumschläge mit ConT_EXt
- 25 ConT_EXt kurz notiert

Von fremden Bühnen

- 28 Das L^AT_EX Tagged PDF Project – Status und Fortschritte
- 39 XMP-Metadaten in L^AT_EX einfügen
- 49 L^AT_EX News – Issue 36, November 2022
- 56 Neue Pakete auf CTAN

Spielplan

- 62 The 28th GUST T_EX conference
- 62 Termine
- 64 Stammtische

Adressen

- 67 Autoren/Organisatoren