

dante
Deutschsprachige
Anwendervereinigung T_EX e.V.

33. Jahrgang Heft 3/2021 August 2021

Korrekturversion: 2021-07-29 22:54:48

3/2021

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- $\LaTeX$ -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, $\LaTeX$ -Pakete oder Schriften notwendig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden. Weitere Informationen für Autoren findet man auf der Projektseite <https://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit LuaHB $\TeX$, Version 1.13.2 (T ϵ X Live 2021) erstellt. Als Standard-schriften kamen Libertinus Serif, Libertinus Sans Serif, Anonymous Pro und Libertinus Math zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2400

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

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

Druck: Schleunungdruck GmbH
Eltertstraße 27, 97828 Marktheidenfeld

Redaktion: Luzia Dietsche (verantwortliche Redakteurin)

Mitarbeit: Adelheid Bonnetsmüller Rudolf Herrmann Eberhard Lisse
Ralf Mispelhorn Rolf Niepraschk Bernd Raichle
Christine Römer Volker RW Schaa Martin Sievers
Herbert Voß

Redaktionsschluss für Heft 4/2021: 15. Oktober 2021

ISSN 1434-5897

Korrekturversion: 2021-07-29 22:54:48

Die T_EXnische Komödie 3/2021

02-01
02-02
02-03
02-04
02-05
02-06
02-07
02-08
02-09
02-10
02-11
02-12
02-13
02-14
02-15
02-16
02-17
02-18
02-19
02-20
02-21
02-22
02-23
02-24
02-25
02-26
02-27
02-28
02-29
02-30
02-31
02-32
02-33
02-34
02-35
02-36
02-37
02-38
02-39
02-40
02-41
02-42
02-43
02-44
02-45
02-46
02-47
02-48
02-49
02-50
02-51
02-52
02-53
02-54
02-55
02-56

Editorial

Liebe Leserinnen und Leser,

dies ist bereits die dritte Ausgabe der Mitgliederzeitschrift in diesem Jahr, an deren Herstellung ich beteiligt bin, und jede hat mich vor eigene Herausforderungen gestellt. Eine davon ist es, ein ausgeglichenes Verhältnis von internem, externem und technischem Inhalt zu schaffen. Dabei hat mir diesmal ein Mitglied und Leser der Kommunikationsliste `tex-d-1` geholfen. Thomas Demmig hat sich sehr spontan und direkt auf die Anfrage gemeldet, ob jemand Artikel vom $\LaTeX$ -Team übersetzen würde. Vielen Dank Thomas für Deinen Einsatz und Frank Mittelbach für die Zustimmung zur Übersetzung, beiden für ihre problemlose und erfolgreiche Zusammenarbeit. Ich bin mir sicher, dass die Leser unserer Zeitschrift die damit übermittelten Informationen zu schätzen wissen.

Ich freue mich außerdem sehr, dass Adelheid Bonnetsmüller ihre Serie »Having Fun with $\LaTeX$ « fortführen will, habe ich ihren Vortrag bei der diesjährigen Online-Tagung doch sehr genossen. Die Beschreibung, wie man sich einen eigenen Fotokalendar erstellen kann, dürfte passend zur zweiten Jahreshälfte kommen, sind da die Urlaubsbilder doch entwickelt und das neue Jahr und der neue Kalender rücken in greifbare Nähe. Das Arbeiten mit arara stellt für mich eine völlig neue Welt dar, der Artikel darüber ist somit für mich per se spannend.

Ich hoffe, dass der Zufluss von vielfältigen und interessanten Beiträgen für unsere Zeitschrift erhalten bleibt wie bisher. Allerdings habe ich noch eine Bitte: Wenn Sie von einem Ereignis erfahren, das irgendwie mit $\TeX$ und Typographie zu tun hat, schicken Sie mir eine kurze Notiz dazu. Das muss nicht immer eine Tagung sein, es gibt auch Ausstellungen, die von Interesse sind. Oder einen Tag der offenen Tür in einem Museum. Oder vielleicht auch Filmtermine und Podcasts, die den einen oder die andere interessieren könnten. Unser Terminkalender freut sich über Erweiterungen. Ich würde auch gerne unsere Rubrik »Bücher« in »Besprechungen allgemein« wandeln, in der u. a. Tipps zu permanenten Ausstellungen oder Blogs oder Podcasts gegeben werden könnten. Eben allem, was irgendwie mit unserem Lieblingsthema zu tun hat – Fragen zur Typographie.

In diesem Sinne wünsche ich Ihnen viel Spaß beim Lesen und
verbleibe mit $\TeX$ nischen Grüßen
Ihre Luzia Dietsche

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

im letzten Grußwort konnte ich die Neubesetzung unserer Bürostelle nur ankündigen. Heute kann ich mit Freude und auch Erleichterung berichten, dass seit 1. Juni Frau Davina Engelmann die neue Ansprechpartnerin für alle Mitglieder im Büro in Heidelberg ist.

Mit Karin Dornacher haben wir vereinbart, dass sie ihre Arbeitszeit schrittweise reduziert und so bis Oktober eine sorgfältige, geordnete Übergabe stattfinden kann. Für die Zeit danach bis zum Juni 2022 hat sie sich zudem bereiterklärt, weiterhin für Fragen rund ums Büro zur Verfügung zu stehen.

Ich bin froh, dass wir das so wichtige Thema »Nachfolge Karin« abschließen können. Wir leiten als Verein damit auch einen Generationswechsel im Büro ein und ich hoffe natürlich sehr, dass Frau Engelmann uns mindestens so lange kompetent, sympathisch und professionell unterstützen und begleiten wird, wie Karin dies in den vergangenen Jahren getan hat.

Beide werden bei der diesjährigen Herbsttagung am 18. September vor Ort sein (sofern Corona eine Präsenzveranstaltung erlaubt), so dass ich Sie alle auch aus diesem Grund noch einmal ganz herzlich nach Saarbrücken einlade. Sie finden in dieser Ausgabe erneut die offizielle Einladung. Alle wichtigen Informationen rund um die Tagung haben wir unter <https://www.dante.de/veranstaltungen/herbst2021/> zusammengestellt. Dort können Sie sich auch anmelden.

Bei mir privat steht ebenfalls ein Jobwechsel an. Ich verlasse nach mehr als elf Jahren das »Trier Center for Digital Humanities« und werde künftig an der Akademie der Wissenschaft und der Literatur Mainz als »Projektkoordinator Digital Humanities« mein erworbenes Wissen in vielen spannenden Forschungsprojekten anwenden können.

Dieser Schritt geschieht nicht vollkommen freiwillig, sondern ist durch das Wissenschaftszeitvertragsgesetz (WissZeitVG) bedingt. Ich will hier nicht tiefer in die

05-01 politische Diskussion einsteigen, doch wer in den vergangenen Wochen die Diskus-
05-02 sionen rund um #IchBinHanna verfolgt hat, weiß, dass wissenschaftliche Arbeit in
05-03 Deutschland ihre »Eigenheiten« hat.

05-04
05-05 Der Jobwechsel bedeutet leider auch, dass ich den Luxus aufgeben muss, T_EX im
05-06 Berufsleben täglich als Hauptwerkzeug einsetzen zu können. Nicht aufgeben werde
05-07 ich allerdings, T_EX als Lösung für so manche Herausforderung gedruckter Werke
05-08 dort ins Spiel zu bringen, wo dies möglich und sinnvoll ist. Und wer weiß, ob ich
05-09 nicht auch das ein oder andere (kleinere) Dokument mit T_EX setzen kann – T_EX
05-10 Live werde ich in jedem Fall auf dem neuen Bürorechner installieren.

05-11
05-12 Einige Lösungen aus der praktischen Arbeit mit T_EX werden wir sicherlich auch
05-13 wieder bei der Herbsttagung in Saarbrücken sehen können. In diesem Zusammen-
05-14 hang kann ich Sie alle nur immer wieder ermuntern, Tagungseinreichungen oder
05-15 auch Artikel für Die T_EXnische Komödie auch dann einzureichen, wenn es sich
05-16 vermeintlich »nur« um ein (individuelles?) Problem aus Ihrer Praxis handelt. Die
05-17 Anwendungen rund um T_EX sind vielfältig und die Nutzerinnen und Nutzer sind es
05-18 auch. Das wollen wir gerne auf Tagungen und in Beiträgen abbilden. Sollten Sie
05-19 dennoch unsicher sein, fragen Sie gerne vorab beim Vorstand bzw. der Redaktion
05-20 nach.

05-21
05-22 Auch beim 15. internationalen ConT_EXt-Meeting vom 20. bis 25. September sowie
05-23 bei der TUG-Tagung vom 5. bis 8. August, die wie im letzten Jahr vollständig
05-24 virtuell stattfindet, werden T_EX und Co. im Mittelpunkt stehen. Sollten Sie nicht
05-25 teilnehmen können, schauen Sie sich auf jeden Fall die Videos an, die im Nachgang
05-26 voraussichtlich über YouTube verfügbar sein werden.

05-27
05-28 2022 werden dann hoffentlich wieder alle Tagungen, Stammtische und sonstigen
05-29 Möglichkeiten zum T_EXnischen Austausch mindestens teilweise in Präsenz stattfin-
05-30 den können. Die Glaskugel gibt aktuell leider nicht mehr her als diese Hoffnung.

05-31
05-32 Zunächst wünsche ich Ihnen aber viel Vergnügen bei der weiteren Lektüre. Herz-
05-33 lichst Ihr/Euer

05-34
05-35 Martin Sievers
05-36
05-37
05-38
05-39
05-40
05-41
05-42
05-43
05-44
05-45

05-46 Einladung zur Herbsttagung 2021 und 05-47 63. Mitgliederversammlung von DANTE e.V. 05-48 05-49

05-50 Thomas Hilarius Meyer, Martin Sievers
05-51

05-52
05-53 Liebe Mitglieder von DANTE e.V.,
05-54
05-55
05-56

hiermit laden wir Sie ganz herzlich zur Herbsttagung 2021 und 63. Mitgliederversammlung am 18. September 2021 in Saarbrücken ein.

Die genaue Anschrift lautet:

Landesinstitut für Pädagogik und Medien
Beethovenstraße 26
66125 Saarbrücken

Es ist folgender Zeitplan vorgesehen:

Freitag, 17. September, ab 19:00 Uhr: Vorabendtreff

Samstag, 18. September, 9:00 Uhr: Mitgliederversammlung

11:00 Uhr bis etwa 17:30 Uhr: Tagungsprogramm mit Vorträgen

ab 19:00 Uhr: Abendtreff

Sonntag, 19. September, 10:00 Uhr: Stadtrundgang und Möglichkeit des Besuchs des Historischen Museums

Die Tagesordnung der Mitgliederversammlung lautet:

1. Begrüßung und Tagesordnung
2. Bericht des Vorstands
3. Wahl eines/r neuen verantwortlichen Redakteurs/-in für »Die T_EXnische Komödie«
4. Verschiedenes

Ihre Stimmunterlagen erhalten Sie direkt vor Ort. Eine Übertragung des Stimmrechts ist im Rahmen des § 13 Abs. (4) der Vereinssatzung möglich. Wie üblich sind auch Nichtmitglieder als Gäste herzlich willkommen.

Detaillierte Informationen zur Tagung finden Sie unter <https://www.dante.de/veranstaltungen/herbst2021/>. Die Seite wird fortlaufend aktualisiert.

Die Teilnahme an der Tagung ist kostenfrei, es wird jedoch um frühzeitige Anmeldung gebeten. Bitte nutzen Sie dazu das Formular auf der Tagungsseite.

Schülern und Studenten werden nachgewiesene Reisekosten bis zur Höhe eines Zugtickets der Deutschen Bahn (2. Klasse) von/nach Saarbrücken erstattet, sofern sie sich bis zum 1. September anmelden.

Fragen, Wünsche und Anregungen richten Sie bitte per E-Mail an mv63@dante.de.

Beiträge gesucht (»Call for Presentations«)

Für ein spannendes und vielfältiges Vortragsprogramm sind wir auf Eure/Ihre aktive Unterstützung angewiesen. Mögliche Themen für Einreichungen können sein:

- Beispiele aus der Praxis (Realisierung besonderer Anforderungen); eigene Klassen und Pakete,
- Spezifische Lösungen mit X_YTeX und LuaTeX,
- Typografie und ihre Umsetzung in TeX und Co.,
- Zusammenspiel von TeX mit anderen Dateiformaten (z. B. XML) und anderen (Open-Source-)Werkzeugen,
- die Entwicklung von TeX und Co. in den vergangenen Jahrzehnten,
- Spezielle PDF-Formate, barrierefreie PDF-Dokumente,
- ...

Fühlen Sie sich angesprochen? Dann senden Sie bitte eine E-Mail mit folgenden Angaben an mv63@dante.de:

- Name der Referentin/des Referenten,
- Titel und Art (Vortrag oder Tutorium) des Beitrags,
- Zeitbedarf (Tutorien dauern i. A. 60 bis 90 Minuten; für Vorträge beträgt die übliche Dauer 30 Minuten plus 10 Minuten für die anschließende Diskussion),
- aussagekräftige Zusammenfassung (ca. 0,5 bis 1,5 Seiten),
- evtl. benötigte Hilfsmittel sowie Wünsche bzgl. der Vortragszeit.

Wir freuen uns über alle Einreichungen; es darf auch gerne Ihr erster »Auftritt« bei einer Tagung von DANTE e.V. sein.

Mit freundlichen Grüßen

Thomas Hilarius Meyer (Lokaler Veranstalter)

Martin Sievers (Vorsitzender DANTE e.V.)

Vorstellung des Vorstands

Luzia Dietsche

Im Vorfeld der Mitgliederversammlung, die im Frühjahr online stattfand, bat ein Leser der vereinsinternen Liste dante-ev@dante.de um eine Vorstellung der zur Wahl stehenden Personen. Da diese Informationen nicht nur für die Leser:innen der Liste von Interesse sind, habe ich sie hier nochmal zusammengefasst, auch wenn die Wahlen inzwischen längst vorüber sind.

Martin Sievers – Vorsitzender

Im Rahmen meines Mathematikstudiums in Trier lernte ich $\text{\LaTeX}$ kennen und lieben. Ich suchte Gleichgesinnte vor Ort und gründete zusammen mit zwei Jurastudenten

einen T_EX-Stammtisch. Seit 2004 bin ich Mitglied von DANTE e.V. und besuche seitdem regelmäßig nationale wie internationale Tagungen.

Seit 2008 bin ich als selbstständiger T_EX-Dienstleister im Bereich wissenschaftlicher Publikationen tätig und arbeite seit März 2010 zudem als wissenschaftlicher Mitarbeiter an der Universität Trier im Bereich der »Digital Humanities«. Im Rahmen dieser Tätigkeit bringe ich seit einigen Jahren wissenschaftliche Editionen mit L^AT_EX in druckfertige Form und habe somit das große Glück, T_EX & Co. auch im beruflichen Umfeld täglich nutzen zu können.

Ich bin seit März 2010 im Vorstand von DANTE e.V., zunächst als Beisitzer und seit März 2012 als Vorsitzender.

Uwe Ziegenhagen – stellvertretender Vorsitzender

Ich lebe in Köln und arbeite bei der Hausbank eines japanischen Automobilkonzerns als Analyst im Bereich »Business Intelligence & Reporting«.

Meine T_EXnischen Interessen liegen vor allem im Bereich der Satz-Automatisierung mit und ohne Python.

Von 2010 bis 2021 war ich Beisitzer im Vorstand von DANTE e.V., seit März 2021 bin ich stellvertretender Vorsitzender. Zu meinen Aufgaben im Vorstand gehören die Pflege der Webseite, die Administration der Mailinglisten, die Begrüßung von Neumitgliedern und die Standbetreuung bei Messen wie der FrO_Scon.

Doris Behrendt – Schatzmeisterin

Ich bin seit Mitte der Nullerjahre Mitglied bei DANTE e.V. und seit 2016 Schatzmeisterin.

Ich bin Mathematikerin, deshalb mag ich Zahlen und L^AT_EX. Für den Verein erledige ich viele langweilige Dinge, die aber eben auch jemand tun muss, beispielsweise Rechnungen schreiben. Zudem schätze ich den Austausch auf Tagungen sehr und vertrete den Verein immer wieder auf Veranstaltungen. Beispielsweise halte ich L^AT_EX-Einsteigerworkshops auf den »Chemnitzer Linuxtagen« oder organisiere das L^AT_EX-Village auf dem *Chaos Computer Club Camp* (Zeltlager).

Volker RW Schaa – Schriftführer

Meine erste Aktion für DANTE e.V. war der Transport von unzähligen Kisten des Tagungsbandes der EuroT_EX'92 in Prag im September 1992 nach Heidelberg. Danach habe ich im Redaktions- und Komödienpack-Team mitgearbeitet, war stellvertretender Vorsitzender, Vorsitzender und Beisitzer von DANTE e.V.

09-01 Meine »Laufbahn« mit T_EX et al. begann 1984 mit ersten T_EX-Übungen auf IBM
 09-02 370/MVS, danach die Betreuung der T_EX-Installation auf einem VAX-Cluster bei
 09-03 der Gesellschaft für Schwerionenforschung, der Job als Korrektor für etliche Bücher
 09-04 bei den Verlagen Addison-Wesley (insbesondere die Bücher von Helmut Kopka),
 09-05 dpunkt und für Open-Access-Publikationen von Konferenzbeiträgen (jacow.org).
 09-06
 09-07

09-08 Ich bin in Wilhelmshaven geboren (also ein fast-Ostfriesen), kam 1971 zum Studium
 09-09 nach Darmstadt, war seit 1974 bei der GSI angestellt als Werkstudent, Operateur und
 09-10 wissenschaftlicher Mitarbeiter und von 2002 bis 2020 abgestellt für Open-Access-
 09-11 Publikationen.
 09-12
 09-13
 09-14

09-15 Klaus Höppner – Beisitzer

09-16 Ich bin Physiker, arbeite an einem Beschleuniger für Tumorbestrahlung eines
 09-17 Universitätsklinikums. T_EX und L^AT_EX nutze ich seit Anfang der 1990er Jahre, als ich
 09-18 es während eines Praktikums bei einem Forschungsinstitut kennenlernte.
 09-19
 09-20
 09-21

09-22 Mitglied bei DANTE e.V. bin ich seit etwa 1995, seit 2000 mit kurzer Unterbrechung
 09-23 im Vorstand, inklusive der Tätigkeit als Vorsitzender und Schatzmeister. Mein
 09-24 Hauptinteresse liegt in der Automatisierung von Aufgaben mit Python.
 09-25
 09-26
 09-27

09-28 Harald König – Beisitzer

09-29 Mein erster Kontakt mit T_EX war gleich zu Beginn meines Physik-Studiums die
 09-30 Mithilfe beim Aufsetzen, Betreuen und Unterstützen eines ersten T_EX-Systems in
 09-31 Tübingen auf VAX/VMS. Erste Kontakte zu anderen T_EXies hatte ich 1987 beim
 09-32 Treffen der »Deutschen T_EX-Interessenten« in Münster, 1989 hatte ich das Glück,
 09-33 Gründungsmitglied von DANTE e.V. zu werden (dank hierfür an Uwe Untermarzo-
 09-34 ner!!).
 09-35
 09-36
 09-37
 09-38

09-39 Neben mehreren kleineren Aktivitäten im Verein war ich von 2002 bis 2005 Kassen-
 09-40 prüfer und bin nun seit 2016 im Vorstand als Beisitzer.
 09-41

09-42 Reisen zu T_EX-Konferenzen, Linux-Tagen und regionalen Opensource“-Veranstaltungen
 09-43 sowie einigen Stammtischen in der näheren Umgebung lassen mich immer wieder
 09-44 alte Freunde treffen, manchmal neue Anwender mit T_EX in Kontakt bringen und
 09-45 neue T_EX-Freundschaften knüpfen. T_EX ist für mich ein guter Grund zu reisen und
 09-46 Leute zu treffen. Menschen treffen und gemeinsam mehr über T_EX und Typographie
 09-47 lernen und verstehen, halte ich für einen der ganz großen Vorteile von T_EX und
 09-48 DANTE e.V. – denn: {T_EXies} do it in groups.
 09-49
 09-50
 09-51
 09-52
 09-53
 09-54
 09-55
 09-56

Stephan Lukasczyk – Beisitzer

Mitglied bei DANTE e.V. bin ich seit bald 15 Jahren, »angeworben« wurde ich durch die $\text{\LaTeX}$ -Kurse von Andreas Hirsch am Werdenfels-Gymnasium in Garmisch-Partenkirchen.

Die erste Tagung, an der ich teilgenommen habe, war die Jubiläumstagung 2009 in Wien. Zusammen mit Andreas habe ich die Herbsttagung 2011 in Garmisch-Partenkirchen koorganisiert. 2018 war ich lokaler Organisator der Frühjahrstagung in Passau. Für DANTE e.V. habe ich zusammen mit anderen dreimal die Betreuung des Standes auf den Chemnitzer Linux-Tagen übernommen und war von 2018 bis 2020 Kassenprüfer.

Ich selbst bin in Garmisch-Partenkirchen geboren und aufgewachsen, bin 2012 nach Passau gezogen und habe dort »Bachelor und Master Informatik« studiert. Seit 2018 bin ich als wissenschaftlicher Mitarbeiter und Doktorand an der Uni Passau beschäftigt, wo ich mich mit dem automatisierten Testen und der Analyse von dynamisch getypten Programmiersprachen befasse. Ich bin verheiratet und Vater einer kleinen Tochter.

Herbert Voß – Beisitzer

Ich bin in Niedersachsen geboren und nach dem Studium der Elektrotechnik in Berlin »hängengeblieben«. Dort arbeitete ich als Entwicklungsingenieur mit dem Schwerpunkt Magnetschwebbahn und Elektroauto. Später lehrte ich als Studiendirektor für Naturwissenschaften am Gymnasium (ab 2009 im Ruhestand). Seit 2005 bin ich Lehrbeauftragter für »Publizieren und Präsentieren mit $\text{\LaTeX}$ « und »Objektorientierte Programmierung mit Python« an der FU Berlin. Mit und an $\text{\LaTeX}$ arbeite ich seit 1990.

Bretter, die die Welt bedeuten

Having Fun with L^AT_EX: Jetzt werden Boxen bunt!

Adelheid Bonnetsmüller

Gelegentlich möchte man, besonders in Büchern oder Dokumentationen, bestimmte Inhalte wie beispielsweise Zusammenfassungen hervorgehoben darstellen. Hierfür können sich farbige Boxen eignen. Hat man verschiedene Typen solcher Boxen, kann es schön sein, diese mit einem Logo zu versehen. So möchte man z. B. eine Erklärung in eine Box mit einer Glühbirne in der oberen linken Ecke setzen und eine Zusammenfassung mit einem Stift oben links markieren. Mit dem in diesem Artikel beschriebenen Paket `bclogo` von Patrick Fradin und Maxime Chupin ist dies einfach möglich.

Die Serie »Having Fun with L^AT_EX«, in der kleinere L^AT_EX-Pakete vorgestellt werden, startete vor einigen Jahren mit regelmässigen Vorträgen auf Tagungen von DANTE e.V. und einigen Artikeln in Die T_EXnische Komödie. Nach einigen »ruhigen Jahren« soll sie nun fortgeführt werden.

Generelles

Das Paket `bclogo` wurde von Patrick Fradin und Maxim Chupin geschrieben und ist in der aktuellen Version 3.1 von 2016 auf CTAN zu finden. Es wird über `\usepackage[Optionen]{bclogo}` geladen. Beim Laden stehen zwei Optionen zur Verfügung. Mit dem Wert `pstricks` oder `pdf` wird angegeben, ob für das Gestalten der Boxen PSTricks oder pdf verwendet werden soll, wobei ersteres die Standardeinstellung ist. Entscheidet man sich für PSTricks, kann zusätzlich die Option `blur` angegeben werden, um das Paket `pst-blur` zu laden.

Die Dokumentation des Paketes ist momentan ausschließlich auf französisch verfügbar, zudem sind die möglichen Optionen teilweise französisch benannt.

Etwas gewöhnungsbedürftig ist die Verwendung der Einheiten. Bei manchen Optionen werden die Werte in `pt`, bei anderen wiederum in `cm` angegeben, wobei die Einheit selber entfällt.

Verwendung und Beispiele

Eine erste Box

Das Paket definiert als einzige Umgebung `\begin{bclogo}{Überschrift}` mit einem Argument für die Überschrift der Box.

```
\begin{bclogo}{Eine Überschrift}
```

So sieht eine Standardbox aus. Der Hintergrund ist weiß, die Box hat einen

↪ schwarzen Rahmen der Stärke 0.4pt, der Balken links ist eine gerade Linie mit

↪ 1.5pt Stärke, oben links taucht als Logo eine Blume mit einer Höhe von 17\,pt

↪ auf und die Überschrift ist in fetter Schrift gesetzt.

```
\end{bclogo}
```



Eine Überschrift

So sieht eine Standardbox aus. Der Hintergrund ist weiß, die Box hat einen schwarzen Rahmen der Stärke 0.4pt, der Balken links ist eine gerade Linie mit 1.5pt Stärke, oben links taucht als Logo eine Blume auf mit einer Höhe von 17 pt und die Überschrift ist in fetter Schrift gesetzt.

Im folgenden wird gezeigt, wie über verschiedene Optionen, die dem Paket mitgegeben werden können, das Aussehen der Box und des linken Balkens sowie das Logo oben links gesteuert werden können und welche Möglichkeiten das Paket außerdem bietet.

Einstellungen der Box

Mit der Option `couleur` kann die Hintergrundfarbe der Box angegeben werden, mit `couleurBord` die Farbe des Rahmens und mit `couleurTexte` die Farbe des Textes.



Eine Überschrift

Eine Box mit grünem Hintergrund, grünem Rand sowie blauem Text.

Diese wurde durch Angabe der folgenden Optionen erzeugt:

`[couleur=green!10, couleurTexte=blue,couleur=green].`

Als Logo wurde `\bcbombe` verwendet (siehe weiter unten).

Die Strichstärke des Rahmen kann mit `epBoard=<Wert in pt>` gesteuert werden, über den Wert `arrondi=<Wert in cm>` können die Ecken abgerundet werden. Es können zudem die Abstände innerhalb der Box bzw. ein Abstand vom linken oder rechten Seitenrand angegeben werden (siehe Beispiel).

Mit der Option `noborder=true` wird die Box ohne Rahmen gezeichnet.

Eine Überschrift

Ein Beispiel mit abgerundeten Ecken. Diese wurden über die Option `arrondi=0.5` erzeugt.

Außerdem wurde die Stärke des Randes mit der Option `epBord=5` auf 5pt gesetzt.

Als Logo wurde `\bctrombone` verwendet (siehe weiter unten).



Eine Überschrift

Eine Box mit einem Abstand rund um den Text von 15pt über die Option `[marge=15]`.

Mit der Option `margeD=<Wert in cm>` bzw. `margeG=<Wert in cm>` kann außerdem der Abstand der Box vom rechten (*Droite* = *D*) bzw. linken (*Gauche* = *G*) Textrand des Dokumentes eingestellt werden.

Als Logo wurde `\bctakecare` verwendet (siehe weiter unten).

Zu guter Letzt kann der Box über die Option `ombre` ein Schatten hinzugefügt werden. Mit der Option `epOmbre=<Wert in cm>` lässt sich die Breite, mit `couleurOmbre` die Farbe des Schattens steuern und mit der Option `blur` kann man dessen Rand verschwimmen lassen.



Eine Box mit Schatten

Diese Box hat einen Schatten (Option `ombre`). Außerdem wurden die Optionen `couleurOmbre=red!20`, `epOmbre=1` und `blur` angegeben.

Als Logo wurde `\bcclesol` verwendet (siehe weiter unten).

Einstellungen des linken Balkens

Der rechte Balken in der Box kann über etliche Optionen angepasst werden, so können Form, Stil und Farbe geändert werden. Die Standardeinstellung ist `barre=line`. Die Dicke dieser geraden Linie kann über die Option `epBarre=<Wert in pt>` gesteuert werden, seine Farbe über die Option `couleurBarre`.

Eine Box mit geändertem rechten Balken

In dieser Box wurde die Farbe und Dicke des rechten Balkens verändert.

Angegeben wurden diese Änderungen über die Optionen [`epBarre=5`, `couleurBarre=-blue!75`]. Als Logo wurde `\bcloupe` verwendet (siehe weiter unten).

Neben der geraden Linie stehen die Balkenstile `barre=zigzag` (Zick-Zack-Linie) oder `barre=snake` (Schlangenlinie) zur Verfügung, bei denen über die Option `tailleOndu=<Wert in pt>` die Amplitude gesteuert werden kann.

Eine Überschrift

Eine Box mit dem Balkenstil `snake` mit der Amplitude 1.5pt.

Und hier ein sinnloser Text, damit die Höhe der Box etwas größer wird, um den Effekt besser zeigen zu können.

Als Logo wurde `\bcoeil` verwendet (siehe weiter unten).

Eine Überschrift

Eine Box mit dem Balkenstil `snake` mit der Amplitude 3pt, angeben über die Optionen [`barre=snake`, `tailleOndu=3`].

Und auch hier wieder etwas völlig sinnfreien Text, der nur dazu da ist, die Box etwas höher ausfallen zu lassen, damit man sieht, was eine geänderte Amplitude für ein Bild erzeugt.

Als Logo wurde `\bcdz` verwendet (siehe weiter unten).

Als weitere Möglichkeit können Symbole oder eigene Bilder als Balken verwendet werden. Hierzu wird die Option `barre=imageExt` zusammen mit der Option `imageBarre=DateinameBild` verwendet.

15-01 Das geladene Bild wird dabei so gestreckt, dass es die komplette Höhe der Box
 15-02 einnimmt.
 15-03

15-04 Das kann dazu führen, dass das verwendete Bild in unerwünschtem Maße gestaucht
 15-05 wird, falls seine Originalhöhe von der Höhe der Box abweicht. Um dies zu vermeiden,
 15-06 kann man statt der Option `barre=imageExt` die Option `barre=imageClip` verwenden.
 15-07 Dies bietet sich insbesondere bei Bildern an, die aus sich wiederholenden Elementen
 15-08 bestehen. Im nachfolgenden Beispiel ist dies verdeutlicht.
 15-09
 15-10

15-11
 15-12



Eine Überschrift



Eine Box mit einem geladenen Bild als linken Balken. Durch die Verwen-
 dung der Option `barre=imageClip` behält dies seine Originalform und wird
 abgeschnitten.

Als Logo wurde `\bcinterdit` verwendet (siehe weiter unten).

15-21
 15-22

15-23
 15-24



Eine Überschrift



Eine Box mit einem geladenen Bild als rechten Balken. Durch die Verwendung
 der Option `barre=imageExt` wird das Bild auf die Höhe der Box gestaucht,
 was in diesem Fall unschön aussieht.

Und noch ein wenig Text, der einfach nur dazu da ist, die Box etwas höher
 ausfallen zu lassen.

Als Logo wurde `\bcsoleil` verwendet (siehe weiter unten).

15-35
 15-36

15-37
 15-38

15-39
 15-40

15-41
 15-42



Eine Überschrift



Eine Box mit einem geladenen Bild als rechten Balken. Durch die Verwendung
 der Option `barre=imageExt` wird das Bild auf die Höhe der Box angepasst.

Und noch ein wenig sinnloser Text zum Füllen.

Als Logo wurde `\bcplume` verwendet (siehe weiter unten).

15-50
 15-51

15-52
 15-53

15-54
 15-55

15-56

★ Eine Überschrift

Eine Verwendung der Option `barre=imageClip` ist hier nicht empfehlenswert, da die Klammer in der Originalgröße beibehalten und in diesem Fall abgeschnitten wird.

Als Logo wurde `\bcetoile` verwendet (siehe weiter unten).

Einstellungen des verwendeten Logos

Das Logo in der oberen linken Ecke kann auf vielfältige Art verändert werden. So stehen zum einen über 60 im Paket definierte Logos zur Verfügung. Ein paar davon wurden in den Beispielen bereits gezeigt, weitere finden sich in der Paketdokumentation.

Zum anderen können sämtliche Symbole anderer $\text{\LaTeX}$ -Pakete, eigene mit `PSTricks`, `TikZ` oder `Metapost` erstellte Logos oder sogar beliebige Bilder verwendet werden. Möchte man ein solches Logo verwenden, definiert man sich entsprechende Befehle, die man später mit der Option `\logo=` der Umgebung mitgeben kann.

Über `\setlength{\logowidth}{Wert in pt}` kann die Größe des Logos verändert werden.

```
\newcommand\einPSTricksLogo{\begingroup\input{logo.pst}\endgroup}
\newcommand\einPGFLogo{\begingroup\input{logo.pgf}\endgroup}
\newcommand\einfacheinBild{\includegraphics[width=17pt]{logo}}
\newcommand\bcheartsuit{\centering \huge $\heartsuit$}
```

♡ Das ♡ Symbol als Logo

Im diesem Beispiel wurde das ♡-Symbol aus dem Mathematiksatz als Logo mit dem Namen `\bcheartsuit` definiert (siehe oben).



Ein beliebiges Bild als Logo

Im diesem Beispiel wurde ein Bild geladen, als neues Kommando `\dahoam` definiert und über die Option `[logo=\dahoam]` der Umgebung mitgegeben.

Ausserdem wurde die Höhe des Logos auf 100pt gesetzt und der Balken links über die Option `barre=none` ausgeblendet.

Sonstiges

Es gibt ausserdem die Möglichkeit, den Titel zu umrahmen (Option `[cadreTitre]`) und einen Untertitel anzugeben (Option `sousTitre`). Die Option einen Untertitel anzugeben, steht nur zur Verfügung, wenn man den Titel nicht in einen Rahmen gesetzt hat. Mit den Befehlen `\renewcommand\bcStyleTitre[1]{Anpassungen}` und `\renewcommand\bcStyleSousTitre[1]{Anpassungen}` kann man den Stil des Titels und des Untertitels anpassen.

```
\renewcommand\bcStyleTitre[1]{\color{orange}{#1}}
\renewcommand\bcStyleSousTitre[1]{\hfill\itshape\color{red}\small{#1}}
```



eine Box mit Überschrift und Untertitel

ein Untertitel

Es wurde der Befehl `\bcStyleSousTitre` neu definiert (siehe oben) und über die Option `sousTitre=ein Untertitel` angegeben.

Der Rand der Box wurde außerdem mit `[noborder]` unterdrückt. Als Logo wurde `\bcinfo` verwendet (siehe weiter oben).



Eine Überschrift mit Rahmen

Ein Beispiel mit Rahmen um den Titel. Dies kann erreicht werden mit der Option `[cadreTitre]`. Der Farbverlauf wie im Bild ist standardmäßig definiert.

Als Logo wurde `\bcclampe` verwendet (siehe weiter oben).



Titelstiländerung

Ein Beispiel mit geändertem Titel und ohne Box (Option=`[noborder]`). Es wurde der Befehl `\bcStyleTitre` neu definiert.

Als Logo wurde `\bccrayon` verwendet (siehe weiter oben).

Das Paket erlaubt standardmäßig einen Seitenumbruch innerhalb der Boxen. Möchte man dies unterbinden, muss die Option `nobreak` auf `true` gesetzt werden.

Zu guter Letzt ist es möglich, ein Verzeichnis aller `bclogo`-Boxen mit dem Befehl `\listofbclogo` auszugeben. Vergibt man sprechende Titel, mag so eine Übersicht am Ende eines Dokuments sinnvoll sein. Der Titel des Verzeichnisses ist standardmäßig auf französisch (*Liste des bclogo*), möchte man ihn auf deutsch haben, muss der Befehl `\titrebclogo` beispielsweise mit `\renewcommand{\titrebclogo}{Liste der Zusammenfassungsboxen}` umdefiniert werden.

Sämtliche Einstellungen können im Fall, dass alle Boxen im Dokument gleich aussehen sollen, auch in der Präambel mit Hilfe von `\presetkeys{bclogo}{Optionen, durch Komma getrennt}` gesetzt werden.

Mit *arara* externe $\text{\LaTeX}$ -Dokumente erzeugen und einfügen

Rainer-M. Fritsch

Mit *arara* lässt sich die Produktion von $\text{\LaTeX}$ -Dokumenten auf vielfältige Weise automatisieren. Dieser Beitrag zeigt, wie ein Hauptdokument mit mehreren in sich abgeschlossenen $\text{\LaTeX}$ -Filialdokumenten ohne Inkonsistenzen zur Laufzeit erzeugt werden kann. Alle Befehle dazu stehen im Kopf des Hauptdokuments.

Für eine Broschüre[3] wollte ich drei externe, von einander unabhängige Dokumente einbinden, die jeweils die gleichen personenbezogenen Stammdaten enthalten sollen. Diese persönlichen Daten werden über eine Parameter-Datei in jedes externe Dokument eingebunden, sodass die Broschüre einfach für verschiedene Personen erstellt werden kann. Die drei externen Dokumente sollen mit den korrekten Seitenangaben im Inhaltsverzeichnis der Broschüre als Hauptdatei aufgeführt werden.

Nun könnte man die drei externen Dokumente von Hand jeweils selbständig erzeugen und anschließend diese drei PDFs mit dem Paket *pdfpages* einbinden. Abgesehen davon, dass sich hier schnell Fehler einschleichen, wenn nicht die aktuellste Version der externen PDF eingebunden wird, ist es doch ein recht hoher Aufwand.

Mit *arara*[1] lässt sich das wunderbar automatisieren; sich in die Syntax von *arara* einzuarbeiten, ist nicht gerade trivial. Dafür gibt es hier gleich den fertigen Code.

Die Broschüre erstellen

Die Broschüre besteht aus einer Hauptdatei *main.tex* und drei externen Dokumenten *dok-ext-01.tex*, *dok-ext-02.tex* und *dok-ext-03.tex*. Die drei externen Dokumente wer-

den in einem Unterverzeichnis (hier im Beispiel `subdir` genannt) des Verzeichnisses vorgehalten, in dem sich die Datei `main.tex` befindet.

Der Arbeitsablauf:

- `dok-ext-01.pdf` mit allen notwendigen L^AT_EX-Läufen aus `dok-ext-01.tex` erzeugen
- `dok-ext-02.pdf` mit allen notwendigen L^AT_EX-Läufen aus `dok-ext-02.tex` erzeugen
- `dok-ext-03.pdf` mit allen notwendigen L^AT_EX-Läufen aus `dok-ext-03.tex` erzeugen
- Das Hauptdokument `main.pdf` aus `main.tex` erzeugen und dabei die drei PDFs `dok-ext-01` bis `dok-ext-03.pdf` einbinden und im Inhaltsverzeichnis mit den korrekten Seitenangaben aufführen.

Die Lösung mit arara und dem Paket pdfpages

arara ist in der neuesten Version Teil der T_EX-Distribution. Möglicherweise muss Java nachinstalliert werden. Beim ersten Aufruf von arara bekommt man ggf. einen entsprechenden Hinweis.

In den Kopf von der Datei `main.tex` schreibt man folgenden Code (hier für xelatex):

Listing 1: arara-Code im Kopf main.tex

```
% arara: xelatex: {
% arara: --> shell: yes,
% arara: --> syntex: yes,
% arara: --> directory: subdir,
% arara: --> files: [ dok-ext-01.tex,dok-ext-01.tex,dok-ext-01.tex,
% arara: --> dok-ext-02.tex, dok-ext-02.tex,dok-ext-02.tex,
% arara: --> dok-ext-03.tex, dok-ext-03.tex, dok-ext-03.tex
% arara: --> ]
% arara: --> }
% arara: xelatex: {
% arara: --> shell: yes,
% arara: --> syntex: yes,
% arara: --> files: [main.tex, main.tex, main.tex] }
```

Die jeweiligen *.tex-Dateien werden jeweils dreimal aufgerufen, um alle internen Referenzen aufzulösen. Einzelne Anweisungen lassen sich mit einem Ausrufungszeichen auskommentieren, z. B.:

Listing 2: arara-Code auskommentieren

```
% !arara: --> dok-ext-03.tex, dok-ext-03.tex, dok-ext-03.tex
```

In die Datei `main.tex` werden die drei PDFs wie folgt eingebunden:

Listing 3: Code-Schnippssel main.tex

```

\begin{document}
\tableofcontents
\include{kap_a}
\include{kap_b}
\include{kap_c}
\includepdf[pages=-, link, addtotoc={1,chapter,1,Titel ext-dok-01,cha:ext-dok
↪01}]{subdir/ext-dok-01}
\includepdf[pages=-, link, addtotoc={1,chapter,1,Titel ext-dok-02,cha:ext-dok
↪02}]{subdir/ext-dok-02}
\includepdf[pages=-, link, addtotoc={1,chapter,1,Titel ext-dok-03,cha:ext-dok
↪03}]{subdir/ext-dok-03}
\include{kap_n}
\end{document}

```

arara anpassen

Datei araraconfig.yaml

Eine Datei araraconfig.yaml anlegen.

Listing 4: Inhalt araraconfig.yaml

```

!config
paths:
- '.'

```

Wer mag, kann gleich noch die Ausgaben von arara auf Deutsch umstellen, eine log-Datei mit allen Ausgaben erstellen lassen und die Bildschirmausgaben unterdrücken. Die Datei araraconfig.yaml hat dann folgenden Inhalt:

Listing 5: Erweiterte araraconfig.yaml

```

!config
language: de
logging: true
verbose: false
paths:
- '.'

```

Durch die Anweisung paths wird arara angewiesen, lokale rules – hier *xelatex.yaml* – im Verzeichnis der Master-T_EX-Datei (*main.tex*) aufzurufen.

Datei xelatex.yaml

Eine Datei *xelatex.yaml* mit folgendem Inhalt anlegen:

Listing 6: Datei xelatex.yaml

```
!config
identifizier: xelatex
name: XeLaTeX
authors:
- Paulo
commands:
- name: XeLaTeX engine
  command: >
    @{
      if (isEmpty(directory)) {
        return getCommand('xelatex', interaction, shell,
          synctex, options, reference.getName());
      }
      else {
        return getCommandWithWorkingDirectory(directory, 'xelatex',
          interaction, shell, synctex, options, reference.getName());
      }
    }
arguments:
- identifizier: interaction
  flag: >
    @{
      if ([ 'batchmode', 'nonstopmode', 'scrollmode',
        'errorstopmode' ].contains(parameters.interaction)) {
        return '--interaction='.concat(parameters.interaction);
      }
      else {
        throwError('The provided interaction value is not valid.');
```

```

    }
- identifier: options
  flag: >
    @{
      if (isList(parameters.options)) {
        return parameters.options;
      }
      else {
        throwError('I was expecting a list of options.');
```

Broschüre erstellen

Wenn diese Dateien fertiggestellt sind, auf der Kommandozeile in das Verzeichnis der Datei *main.tex* wechseln.

Mit

```
arara main.tex
```

werden die Anweisungen von arara im Kopf der Datei *main.tex* abgearbeitet. Fertig!

Auf [https://www.texwelt.de/\[5\]](https://www.texwelt.de/[5]) gibt es noch ein paar Informationen, wenn man nicht auf der Kommandozeile arbeiten möchte, um arara in einige der T_EX-Editoren einzubinden.

Mein Dank gilt Ulrike Fischer[2] und Paulo Cereda[1] – Autor von arara – für ihre hilfreiche Unterstützung bei dieser Lösung. Alle Beispiel-Dateien können unter [4] heruntergeladen werden.

Literatur

- [1] Paulo Roberto Massa Cereda: arara – Automation of L^AT_EX compilation, 2021, <https://ctan.org/pkg/arara?lang=de> (besucht am 31. 5. 2021).
- [2] Ulrike Fischer: Troubleshooting T_EX, 2021, <https://www.troubleshooting-tex.de/> (besucht am 31. 5. 2021).
- [3] R-M Fritsch: Bin ich bereit?, 2021, <https://bin-ich-bereit.de> (besucht am 31. 5. 2021).

23-01
23-02
23-03
23-04
23-05
23-06
23-07
23-08
23-09
23-10
23-11
23-12
23-13
23-14
23-15
23-16
23-17
23-18
23-19
23-20
23-21
23-22
23-23
23-24
23-25
23-26
23-27
23-28
23-29
23-30
23-31
23-32
23-33
23-34
23-35
23-36
23-37
23-38
23-39
23-40
23-41
23-42
23-43
23-44
23-45
23-46
23-47
23-48
23-49
23-50
23-51
23-52
23-53
23-54
23-55
23-56

- [4] Rainer-Maria Fritsch: Beispiel-Dateien für DTK, 2021, <https://github.com/rmfberlin/TeXLaTeX-Usergroup-Berlin/tree/master/arara> (besucht am 7. 6. 2021).
- [5] TexWelt: Wie integriere ich arara in meinen Editor?, 2021, <https://texwelt.de/fragen/5571/wie-integriere-ich-arara-in-meinen-editor> (besucht am 31. 5. 2021).

Wochenkalender

Harald Lichtenstein

Ich hatte bisher immer den Aufwand gescheut, selbst einen Kalender zu entwerfen und zu gestalten, aber meine Frau hat sich für nächstes Jahr einen Wochenkalender gewünscht, der so ähnlich aussehen sollte, wie die vom Harenberg Verlag.

Den verschiedenen großen und kleinen Anbietern von Fotokalendern fehlt allerdings so ein Wochenkalender. Ich habe dort nur Kalender gefunden, die einen ganzen Monat oder ein ganzes Jahr darstellen.

Hier also mein Wochenkalender, erzeugt mit $\TeX$.

Idee

Meine Idee für einen Wochenkalender erfordert neben Kenntnissen in $\TeX$ auch etwas handwerkliches Geschick und man braucht Fotos für jede Woche, die möglichst alle die gleiche Größe haben sollten. Mein Telefon macht Fotos in der Größe 4032px×1908px. Man muss mit dem Skalierungsfaktor des `\includegraphics`-Makros und der Knotenposition `at (x,y)` etwas experimentieren, wenn die Fotos eine andere Größe haben, siehe Zeile 10 im großen Listing.

Der Herstellungsprozess schrittweise:

- Fotos der gleichen Größe aussuchen und im gleichen Ordner sammeln, in dem auch die $\TeX$ -Datei liegt.
- Alle Fotos umbenennen, das erste Foto beginnt bei 0, also zum Beispiel 0.jpg, dann 1.jpg usw. bis 52.jpg, je nach Anzahl der Wochen im Jahr.
- Das PDF erzeugen und bei einem Fotodienstleister hochladen.
Bei <https://www.saal-digital.de/> kann man Dateien im *Portable Data Format* hochladen und jedes Blatt wird als einzelnes Bild übernommen.
- Alle Fotos selbst mit Fotokleber zusammenkleben und den Kalender aufstellen oder verschenken. :-)



Überblick

Wie man auf der Abbildung sehen kann, enthält mein Wochenkalender im oberen Teil ein Bild und das Kalendarium besteht aus dem Monatsnamen, der Jahreszahl, den Wochentagskürzeln und den Tagen. In den Kalendern von Harenberg sind noch die Sternzeichen und Mondphasen angegeben. Außerdem werden beide Monatsnamen angegeben, wenn der Monatswechsel innerhalb der Woche ist. Diese Besonderheiten habe ich weggelassen.

Das folgende Listing zeigt nicht alle Details, sondern soll einen Überblick geben.

```

1 \documentclass[german]{scrartcl} % KOMA-Script-Klasse Artikel
2 \usepackage{pgfcalendar} % Paket fuer Datumsberechnungen
3 \begin{document}
4 \loop
5   <Bild einfuegen>
6   <Monatsname und Jahr anzeigen>
7   {\loop
8     <Tagesnamen anzeigen, Sa und So in rot>
9     \repeat}
10  {\loop
11    <Wochentage anzeigen, Wochenendtage in rot>
12    \repeat}
13    <neues Kalenderwochenblatt>
14  \repeat
15 \end{document}

```

Ich habe für den Kalender die KOMA-Script-Klasse `scrartcl` verwendet, siehe Zeile 1. Das Paket `pgfcalendar` liefert die notwendigen Makros, um Berechnungen mit Kalenderdaten durchzuführen, siehe mehr dazu im PGF-Manual [1] in Kapitel 84. Für den Wochenkalender brauche ich ein Anfangsdatum und dann mehrere Zähler, die alle Wochen und Tage im Jahr durchzählen.

In Zeile 4 beginnt die äußere Schleife, die die Wochen des Jahres durchläuft. Für jede Woche soll ein neues Wochenblatt angelegt werden, Zeile 13. Auf jedes Wochenblatt wird oben ein Bild eingefügt, Zeile 5, und darunter folgt das Kalendarium mit dem Monatsnamen links und der Jahreszahl rechts, Zeile 6.

Anschließend folgen zwei innere Schleifen, Zeilen 7–9 und 10–12. Diese inneren Schleifen müssen in geschweiften Klammern eingeschlossen werden, andernfalls gibt es seltsame Fehlermeldungen. In der ersten inneren Schleife werden die Wochentagskürzel gesetzt, Zeile 8. In der zweiten inneren Schleife folgen in einer weiteren Zeile die Wochentage, Zeile 11.

`\loop` ist ein $\TeX$ -Makro und funktioniert wie das folgende Beispiel zeigt:

```
1 % TeX Schleife
2 \newcount\i           % Definition des TeX-Zaehlers \i
3 \loop                % Schleifenbeginn
4   <some code>         % was wiederholt werden soll
5   \advance\i by 1     % TeX-Zaehler inkrementieren
6   \ifnum\i < 10       % Abbruchbedingung
7   \repeat             % Wiederholung bis zur Abbruchbedingung
```

Dabei ist zu beachten, dass das Makro `\newcount` in Zeile 2 einen $\TeX$ -Zähler `\i` initialisiert und nicht zu verwechseln ist mit dem $\LaTeX$ -Makro `\newcounter`. Das Paket `pgfcalendar` funktioniert nur mit $\TeX$ -Zählern. In Zeile 3 beginnt die Schleife mit dem $\TeX$ -Makro `\loop`. Der Code für diese Art von Schleifen ist ziemlich verkürzt, weil das Ende der Schleife `\repeat` in Zeile 7 gleichzeitig auch die Bedingung `\ifnum` in Zeile 6 schließt. Mit dem $\TeX$ -Makro `\advance` wird der Zähler inkrementiert¹. Hinter `by` können auch größere oder negative Zahlen stehen.

Details

Doch jetzt der ganze Code; die Hinweise folgen danach. Zunächst der Vorspann:

```
1 \documentclass[ngerman]{scrartcl} % KOMA-Script-Klasse Artikel
2
3 \usepackage[utf8]{inputenc}       % Kodierung UTF-8
4 \usepackage{babel}                % Ersetzung deutscher Namen
```

¹ Neben `\advance` gibt es noch `\multiply` und `\divide`.

5	<code>\usepackage[T1]{fontenc}</code>	% Eingabe von Umlauten und ss	26-01
6	<code>\usepackage{translator}</code>	% Uebersetzt ins Deutsche	26-02
7	<code>\usepackage{tikz}</code>	% fuer Hintergrundbild	26-03
8	<code>\usepackage{pgfcalendar}</code>	% Paket fuer Kalenderberechnungen	26-04
9	<code>\usepackage{xcolor}</code>	% Paket um Text zu faerben	26-05
10	<code>\usepackage{graphicx}</code>	% Paket fuer Bilder, jpeg	26-06
11			26-07
12	<code>\newcount\woche</code>	% TeX Zaehler fuer die Woche	26-08
13	<code>\newcount\tag</code>	% TeX Zaehler fuer den Tag	26-09
14	<code>\newcount\wochentag</code>	% TeX Zaehler fuer den Wochentag	26-10
15			26-11
16	<code>\pgfcalendarstartdate{2021-12-27}{\tag}</code>	% setzt \tag auf das Datum	26-12
17	<code>\pagestyle{empty}</code>	% Unterdrueckung der Seitenzahlen	26-13
			26-14
			26-15
			26-16
			26-17
			26-18
			26-19
			26-20
			26-21
			26-22
			26-23
			26-24
			26-25
			26-26
			26-27
			26-28
			26-29
			26-30
			26-31
			26-32
			26-33
			26-34
			26-35
			26-36
			26-37
			26-38
			26-39
			26-40
			26-41
			26-42
			26-43
			26-44
			26-45
			26-46
			26-47
			26-48
			26-49
			26-50
			26-51
			26-52
			26-53
			26-54
			26-55
			26-56

In Zeile 1 steht – wie schon erwähnt – die Dokumentklasse. Dann folgen in den Zeilen 3–10 die Pakete, die für das Layout gebraucht werden. Die Daten für den Wochenkalender werden mit dem Paket `pgfcalendar` berechnet. Für eine deutsche Ausgabe sorgen die Pakete `babel` und `translator` und zwar in dieser Reihenfolge. Dann brauche ich noch das Pakt `color`, damit ich die Wochenenden **rot** einfärben kann und das Paket `graphicx`, damit ich Bilder mit dem Befehl `\includegraphics` einbinden kann.

In den Zeilen 12–14 werden drei $\TeX$ -Zähler definiert, ein Wochenzähler für alle Kalenderwochen im Jahr, ein Tageszähler für alle Tage im Jahr und ein Wochentag-zähler, der immer wieder von Montag bis Sonntag zählt. $\TeX$ initialisiert die Zähler mit 0.

Bevor das Dokument beginnt, wird in Zeile 16 das Startdatum festgelegt. Da immer auf jedem Wochenblatt eine ganze Woche dargestellt ist, muss man das Startdatum so wählen, dass der erste Tag auch ein Montag ist. Im Jahr 2022 ist der erste Januar ein Samstag, also startet der Kalender schon am 27.12.2021, weil das ein Montag ist.

Das Makro `\pgfcalendarstartdate` in Zeile 16 lädt in den $\TeX$ -Zähler `\tag` den Wert 2 459 576, der für den 27.12.2021 steht. Dieser Wert wird dann in den Schleifen inkrementiert.

Zeile 17 unterdrückt die Seitenzahlen, die im Wochenkalender stören würden.

Und hier nun das eigentliche Dokument:

1	<code>\begin{document}</code>		26-48
2	<code>\KOMAOPTIONS{</code>	% Parameter fuer die Klasse scrartcl	26-49
3	<code>paper=13cm:18cm,</code>	% Fotopapierformat fuer Abzuege 13x18	26-50
4	<code>paper=landscape,</code>	% Ausrichtung Querformat	26-51
5	<code>DIV=22}</code>	% Verhaeltnis der Seitenraender	26-52
6	<code>\loop</code>		26-53
			26-54
			26-55
			26-56

```

27-01 7 \pgfcalendarjuliantodate{\tag}{\theyear}{\themonth}{\theday}
27-02 8
27-03 9
27-04 10 \tikz [remember picture, overlay, shift=(current page.center)]
27-05 11 \node [inner sep=0] at (0,1.5){\includegraphics[scale=0.16]{\the\woche}};
27-06 12 \vfill
27-07 13 \textbf{\pgfcalendarmonthname{\themonth}}\hfill\textbf{\theyear}
27-08 14 \par
27-09 15 {\loop
27-10 16 \hfill
27-11 17 \pgfcalendarjuliantodate{\tag}{\theyear}{\themonth}{\theday}
27-12 18 \pgfcalendarifdate
27-13 19 {\theyear-\themonth-\theday}
27-14 20 {\weekend}
27-15 21 {\makebox[2.2em]{\textcolor{red}
27-16 22 {\pgfcalendarweekdaysshortname{\wochentag}}}}
27-17 23 {\makebox[2.2em]
27-18 24 {\pgfcalendarweekdaysshortname{\wochentag}}}
27-19 25 \hfill
27-20 26 \advance\tag by 1
27-21 27 \advance\wochentag by 1
27-22 28 \ifnum\wochentag < 7
27-23 29 \repeat}
27-24 30 \par
27-25 31 \wochentag = 0
27-26 32 {\loop
27-27 33 \hfill
27-28 34 \pgfcalendarjuliantodate{\tag}{\theyear}{\themonth}{\theday}
27-29 35 \pgfcalendarifdate
27-30 36 {\theyear-\themonth-\theday}
27-31 37 {\weekend}
27-32 38 {\makebox[2.2em]{\textcolor{red}{\number\theday}}}
27-33 39 {\makebox[2.2em]{\number\theday}}
27-34 40 \hfill
27-35 41 \advance\tag by 1
27-36 42 \advance\wochentag by 1
27-37 43 \ifnum\wochentag < 7
27-38 44 \repeat}
27-39 45 \newpage
27-40 46 \ifnum\woche < 52
27-41 47 \advance\woche by 1
27-42 48 \advance\tag by 7
27-43 49 \repeat
27-44 \end{document}
27-45
27-46
27-47
27-48
27-49
27-50
27-51
27-52
27-53
27-54
27-55
27-56

```

In den Zeilen 2–5 wird die Geometrie des Wochenkalenders festgelegt. Der Fotoservice, den ich beauftragen werde, hat Fotopapier in der Größe $13\text{ cm} \times 18\text{ cm}$, Zeile 3. Die Ausrichtung ist im Querformat, Zeile 4 und die Seitenränder berechnet die KOMA-Script-Klasse über den DIV-Wert. Da habe ich etwas experimentiert und 22 hat mir am besten gefallen, Zeile 5.

Die äußere Schleife über die Wochen im Jahr beginnt in Zeile 6 und setzt sich in den Zeilen 44–48 fort. Zeile 44 erzeugt jeweils eine neue Seite pro Kalenderwoche. Zeile 45 ist die Abbruchbedingung. Das Jahr 2022 hat 52 Wochen, wobei der 1.1.2022 und der 2.1.2022 noch zur letzten Woche des Jahres 2021 gezählt werden. Die erste Woche im Jahr muss immer einen Donnerstag enthalten. Der Wochenkalender hat also 53 Blätter, gezählt von 0–52. In Zeile 46 wird der Wochenzähler inkrementiert, in Zeile 47 der Tageszähler. Der Tageszähler startet wie gesagt bei 2 459 576 und endet bei $2459576 + 364 = 2459940$ für das Jahr 2022.

Machen wir weiter mit Zeile 7. `\pgfcalendarjuliantodate` ist die Umkehrfunktion zu `\pgfcalendaratetotjulian` und ermittelt das Datum aus dem $\text{\TeX}$ -Zähler `\tag`. Das Datum wird in den Registern `\theyear`, `\themonth` und `\theday` abgelegt, damit man je nach Belieben das Datum formatieren kann. Ich brauche für den Wochenkalender nur das Jahr und den Monat, Zeile 12.

In den Zeilen 9 und 10 füge ich das Bild der Woche ein. `TikZ` ermöglicht mit den Parametern `remember picture`, `overlay` und `shift`, dass das Bild über die Seitenränder hinaus das Blatt füllt. Mehr dazu findet man im `PGF-Manual` [1] in Kapitel 17.13.

Zeile 11 fügt einen variablen vertikalen Abstand ein `\vfill`, damit rutscht das Kalendarium unter das Bild. Vor den Schleifen habe ich in Zeile 13 einen Absatz `\par` eingefügt, damit die Wochenkürzel und Tage unter dem Monat und Jahr stehen.

Die erste innere Schleife beginnt in Zeile 14 und endet in Zeile 28. In dieser Schleife werden auf jedes Blatt die Wochentagskürzel geschrieben. Das hätte man auch einfacher haben können, aber so lernt man noch etwas mehr über die Möglichkeiten des Pakets `pgfcalendar`. Die Wochentagskürzel sollen zentriert auf dem Blatt erscheinen. Das erreiche ich durch zwei variable horizontale Abstände `\hfill` in den Zeilen 15 und 24. In Zeile 16 wird das Datum zum aktuellen Tag `\tag` ermittelt. Das Datum wird in Zeile 17 bewertet und ich kann die Ausgabe in Abhängigkeit vom Wochenende steuern. Das Wochenende soll rot gefärbt sein. Das Makro `\pgfcalendarifdate` hat folgende Syntax, siehe im `PGF-Manual` [1] in Abschnitt 84.1.2:

```

1 % Syntax
2 \pgfcalendarifdate
3   {<Datum im ISO-Format>}           % ISO-Format ist YYYY-MM-DD
4   {<Bedingung>}                     % Test auf Daten oder Tage
5   {<Code fuer true>}                 % Code bei ja
6   {<Code fuer false>}               % Code sonst

```

29-01	7	
29-02	8	% Beispiel 1
29-03	9	\pgfcalendarifdate
29-04	10	{2021-09-21} % ein Datum im ISO-Format
29-05	11	{Tuesday} % Test auf Dienstag
29-06	12	{Ja, ist ein Dienstag} % Ausgabe, wenn ja.
29-07	13	% Der 21.9.2021 ist ein Dienstag.
29-08	14	{Nein, ist ein anderer Wochentag} % Ausgabe sonst
29-09	15	
29-10	16	% Beispiel 2
29-11	17	\pgfcalendarifdate
29-12	18	{\theyear-\themonth-\theday} % ein anderes Datum im ISO-Format
29-13	19	{weekend} % Test auf Wochenende, Sa oder So
29-14	20	{Hurra, Wochenende!} % Ausgabe, wenn ja
29-15	21	{Arbeitsstag} % Ausgabe sonst
29-16		
29-17		
29-18		
29-19		
29-20		

29-21 Damit immer der gleiche Platz für die Weekenkürzel und Tage verwendet wird,
29-22 werden alle Ausgaben in eine Box \makebox fester Größe [2.2em] geschrieben. So
29-23 stehen die Weekenkürzel sauber über den einstelligen oder zweistelligen Tages-
29-24 zahlen. Wenn also der Tag ein Wochenende ist, dann wird das Weekenkürzel
29-25 \pgfcalendarweekdaysshortname rot \textcolor{red} gesetzt, sonst schwarz. Das Ma-
29-26 kro \pgfcalendarweekdaysshortname kennt nur die Werte 0–6 und ermittelt nicht aus
29-27 dem Datum direkt das Wochentagskürzel, deshalb gibt es den Zähler \wochentag.
29-28 Schließlich werden in den Zeilen 25 und 26 die Zähler inkrementiert und nach den
29-29 sieben Tagen einer Woche in den Zeilen 27 und 28 die erste innere Schleife beendet.

29-30 Es folgt ein neuer Absatz \par und der Wochentagszähler \wochentag wird zurück-
29-31 gesetzt, Zeile 30. Die zweite innere Schleife in den Zeilen 31 bis 43 ist genauso
29-32 aufgebaut wie die erste, daher sollen nur die Unterschiede erläutert werden.

29-33 Unter den Wochentagskürzeln sollen die Tage stehen, Werktage schwarz und Wo-
29-34 chenenden rot. Da das Makro \pgfcalendarjuliantodate in Zeile 33 bereits das
29-35 Register für den Tag im Monat \theday belegt, muss der nur ausgegeben werden.
29-36 Die Makros \number\theday liefern den Tag des Monats ohne führende Null.

29-37 Viel Spaß beim Basteln :-)

29-38 Literatur

- 29-39 [1] Till Tantau: The TikZ and PGF Packages, Manual for version 3.1.8b, Versi-
29-40 on 3.1.8b, 2020, CTAN:/pgf/pgf (besucht am 13. 4. 2021).

Von fremden Bühnen

LaTeX Tagged PDF – Ein Entwurf für ein großes Projekt¹

Frank Mittelbach, Chris Rowley

In Franks Vortrag auf der Online-Konferenz TUG 2020 haben wir den Beginn eines mehrjährigen Projekts zum Verbessern von LaTeX angekündigt, mit dem das Erstellen strukturierter Dokumentenformate vollständig und ganz natürlich unterstützt werden soll – insbesondere das »Tagged PDF«-Format, wie es für Barrierefreiheit-Standards wie PDF/UA erforderlich ist.

In diesem kurzen Artikel umreißen wir den Hintergrund dieses Projekts und seiner bisherigen Historie. Dann beschreiben wir die wichtigsten Features und die Aufgaben, wobei mehr Details in der Machbarkeitsstudie [9] zu finden sind, die als erster Teil unserer Kooperation mit Adobe erstellt wurde.

Weiter geht es mit einer Beschreibung, wie wir die Studie als Grundlage für unsere Arbeit am Projekt nutzen wollen, und ein paar Details zu unserer geplanten Arbeitsweise – illustriert durch das, was wir bisher erreicht haben und abgeschlossen durch eine Diskussion der möglichen Hürden, die vor uns liegen.

Schließlich gibt es noch eine Zusammenfassung der bisherigen, aktuellen und kommenden Aktivitäten im und rund um das Projekt.

Eine kurze Geschichte des Projekts

Als TeX in den frühen 1980er Jahren von Don Knuth entwickelt wurde, war es als automatisches System für qualitativ hochwertigen Satz gedacht, das sich nur mit der Produktion eines »gedruckten Ergebnisses« aus der Texteingabe befasse und für das Papier das Ziel-Ausgabemedium war. Jegliche andere Art von Satzausgabe wurde entweder gar nicht oder nicht direkt unterstützt.

Teilweise parallel zu Knuths Arbeit entwickelte Leslie Lamport LaTeX als komplexe und hochintegrierte Sammlung von TeX-Makros, die auf TeX aufsetzten. Einige Zeit später kam in den 1990er Jahren eine gewisse Unterstützung für Grafik und

¹ Dieser Artikel wurde zuvor im *TUGboat* 41:3 [10] veröffentlicht und von Thomas Demmig übersetzt.

30-01
30-02
30-03
30-04
30-05
30-06
30-07
30-08
30-09
30-10
30-11
30-12
30-13
30-14
30-15
30-16
30-17
30-18
30-19
30-20
30-21
30-22
30-23
30-24
30-25
30-26
30-27
30-28
30-29
30-30
30-31
30-32
30-33
30-34
30-35
30-36
30-37
30-38
30-39
30-40
30-41
30-42
30-43
30-44
30-45
30-46
30-47
30-48
30-49
30-50
30-51
30-52
30-53
30-54
30-55
30-56

Farben hinzu, während weiterhin das Paradigma verfolgt wurde, auf realen Medien zu drucken – jetzt allerdings auch auf Overhead-Folien.

Kurz danach ging es über Papier hinaus – mit Hyperlinks und Unterstützung für weitere Aspekte des Web-Publishing, die in einer Schicht über dem zugrundeliegenden LaTeX lagen. Aber diese (und viele weitere folgende Erweiterungen) wurden nie als integrale Teile des Designs angesehen. Diese stückweise Entwicklung führte dazu, dass die Erweiterungen sehr fragil sind, so dass sie in komplexeren Dokumenten oft nicht mehr funktionieren – insbesondere wenn man sie kombiniert einsetzt.

Selbst heute noch gibt es viele Bereiche in LaTeX, die nur wenige APIs (*Application Programming Interface* bzw. Programmierschnittstellen) bereitstellen, welche robust genug sind, um sie sicher zu verwenden und auf die man in der Entwicklung guten Gewissens aufbauen kann. Daher sind die meisten solcher Erweiterungen dazu gezwungen, viele interne LaTeX-Befehle zu überschreiben: Solche unglücklichen und wenig nachhaltigen Praktiken sind momentan notwendig, um einen Großteil der zusätzlichen Funktionalität implementieren zu können. Diese Vorgehensweise führt naturgemäß zu vielen Problemen, wobei die Inkompatibilität zwischen Erweiterungen besonders verbreitet und frustrierend ist.

Eine weitere ernsthafte Einschränkung des aktuellen LaTeX besteht darin, dass es jedes Mal beim Erstellen einer Ausgabeseite sorgfältig all die strukturellen Informationen verwirft, die es beim Erzeugen dieser Seite verwendet und angesammelt hat.² Daher ist eine von einfachem LaTeX erzeugte PDF- oder DVI-Datei nicht mehr als ein Stream aus positionierten Glyphen und sie enthält nur sehr wenig strukturelle Informationen.

Solange es die Intention ist, ein Dokument nur auf einem physischen Medium auszudrucken, ist auch nicht mehr erforderlich. Aber schon seit einer ganzen Weile werden andere Verwendungsarten zunehmend wichtiger, so dass viele Dokumente heutzutage entweder gar nicht mehr oder nur noch im Nachgang gedruckt werden.

Im 21. Jahrhundert ist nun aus einer Reihe von Gründen das Erstellen von PDF-Dokumenten wichtig geworden, die »barrierefrei« sind – also Informationen enthalten, die Screenreader-Software und ähnliches unterstützen, und sich formaler gesehen an den PDF/UA (Universal Accessibility)-Standard [4] halten, der genau

² Dieses Entsorgen eines Großteils der nützlichen Informationen war in der Frühzeit der LaTeX-Entwicklung unabdingbar, weil das System sonst niemals in dem begrenzten Speicher damaliger Computersysteme gelaufen wäre. Daher wurden alle Informationen, die nicht länger für die Ausgabe einer »gedruckten Seite« erforderlich waren, sofort verworfen, um den Speicherbedarf in vernünftigen Grenzen zu halten.

Aber selbst das war nicht immer ausreichend: Auf seinem ersten PC sah sich einer der Autoren einem »Out of Memory« gegenüber, wenn er die Klasse `article` lud.

er in [2] beschrieben wird. Ross Moore hat vor noch nicht allzu langer Zeit eine umfassende Einführung zu diesem Thema geschrieben [11].

Eine wichtige Anforderung für solche PDF-Dokumente ist, dass sie eine stattliche Menge eingebetteter struktureller Informationen und andere Daten enthalten müssen. Aktuell erfordern alle Methoden zum Erstellen solcher »barrierefreier PDFs«, auch beim Einsatz von L^AT_EX, umfangreiche manuelle Arbeiten zum Vorbereiten der Quellen oder bei einer Nachbearbeitung des PDFs (eventuell auch beides) – und diese Arbeiten müssen schon nach minimalen Änderungen an der Quelle (egal ob L^AT_EX oder etwas anderes) wiederholt werden. Solche Methoden und ihre inhärenten Probleme wurden kürzlich von Entwicklerinnen und Entwicklern wie Ross Moore beschrieben und demonstriert, dessen Arbeiten in diesem Bereich zu einem Großteil über seine Website erreichbar sind [12].

Dem L^AT_EX Project Team ist schon seit Jahren bewusst, dass diese neuen Einsatzzwecke nicht ausreichend durch die aktuelle Systemarchitektur von L^AT_EX unterstützt werden und dass daher in diesem Bereich größere Arbeiten dringend erforderlich sind, um sicherzustellen, dass L^AT_EX ein wichtiges und relevantes Dokumenten-Quellformat bleibt. Aber der Umfang der Arbeiten, die für solch eine große Änderung an der Systemarchitektur von L^AT_EX erforderlich sind, wird enorm sein, und definitiv die Ressourcen unseres kleinen Teams an Freiwilligen deutlich sprengen, die in ihrer Freizeit im Einsatz sind! Vielleicht wäre es möglich, würde dann aber eine sehr lange – und wahrscheinlich zu lange – Zeit brauchen.

Auf der TUG-Konferenz 2019 in Palo Alto wurde unsere zuvor pessimistische Sicht auf dieses Thema vorsichtig optimistisch, weil wir von zwei Senior Engineers von Adobe angesprochen wurden, um die Möglichkeit des Erzeugens strukturierter PDFs aus L^AT_EX-Quellen ohne die üblicherweise erforderliche manuelle Nachbearbeitung zu diskutieren. Zudem wollten sie von uns wissen, was erforderlich wäre, um das umzusetzen. Sie deuteten an, dass sie – wenn es sich als umsetzbares Projekt erweisen würde, für das wir einen überzeugenden Projektplan beisteuern könnten – Adobe liebend gerne auf der passenden Managementebene vorschlagen würden, diese Arbeit zu unterstützen – sowohl finanziell als auch auf anderen Wegen, und ganz ohne Bedingungen.³

³ Adobes Gründe für diese Unterstützung sind, dass es sich bei L^AT_EX aktuell um ein sehr wichtiges Dokumentenformat in den MINT-Disziplinen handelt und dass es eines der Formate ist, das für das Erzeugen wohlstrukturierter PDFs »out of the box« besonders geeignet ist, da die Quellen schon so viele Informationen über die Dokumentstruktur enthalten.

Wäre L^AT_EX also dazu in der Lage, strukturierte PDF-Dokumente ohne großen Aufwand erzeugen zu können, würden sehr viele solche Dokumente im Internet verfügbar werden. Zudem könnten viele ältere Dokumente halbwegs einfach neu verarbeitet werden, um ihre PDF-Form um strukturierte Informationen zu ergänzen. Für Adobe ist all das von großem Interesse, da die Existenz sehr vieler solcher Dokumente ein wichtiges und strategisches Argument für ihre Document-Cloud-Angebote [1] und zukünftige zugehörige Services ist.

Als ein Ergebnis dieser Diskussionen erstellten die Autoren zusammen mit Ulrike Fischer Ende 2019 eine erweiterte Machbarkeitsstudie für das Projekt, deren Zielgruppe vor allem die Entwicklung und das Management bei Adobe war. Diese Studie [9] beschreibt detailliert die verschiedenen im Projekt anstehenden Aufgaben und die Zusammenhänge zwischen ihnen. Auch enthält sie einen Plan, der zeigt, wie und in welcher Reihenfolge diese Aufgaben angegangen werden sollten, um das abschließende Ziel zu erreichen und gleichzeitig konkrete Zwischenergebnisse zu liefern, die für die Anwendungs-Community von Relevanz sind (sowohl für LaTeX wie auch für PDF) – diese Zwischenschritte werden dabei helfen, Feedback zu bekommen, das für die erfolgreiche Umsetzung späterer Aufgaben sehr wichtig sein wird.

Der Projektplan erhielt die Zustimmung des Adobe-Managements und wir einigten uns Anfang 2020 darauf, das Projekt anzugehen, wobei Adobe einen wesentlichen Teil der zu erwartenden Projektkosten trägt. Aber das war 2020 und Corona grätschte dazwischen! Daher kamen alle Aktivitäten zum Erliegen und Adobe war dazu gezwungen, solche externen finanziellen Zusagen neu zu bewerten. Aber uns wurde ziemlich schnell signalisiert, dass das nur ein zeitweiliger Rückschlag sein würde, daher begannen wir mit der Arbeit am Projekt, wenn auch mit etwas reduzierter Geschwindigkeit, und konnten unsere ersten Ergebnisse auf der Online-Konferenz TUG 2020 vorstellen – das neue Hook Management System für LaTeX.⁴

Was das Projekt abdeckt

Dieses Projekt hat den Titel »LaTeX Tagged PDF«. Bevor wir also genauer darüber reden, was es abdeckt, sollten wir die Verwendung des Wortes »Tagged« erläutern. Ein guter Ausgangspunkt ist dafür das folgende Zitat aus dem PDF-Standard [17, 6], der Tagging im PDF-Kontext wie folgt definiert:

Tagged PDF ([eingeführt in] PDF 1.4) ist eine strukturierte und stilbehaltende Verwendung von PDF, die auf dem in Abschnitt [...] »Logical Structure« beschriebenen Framework für logische Strukturen aufbaut. Es definiert einen Satz von Standard-Strukturtypen und -Attributen, die es Seiteninhalten (Text, Grafik und Bildern) erlaubt, extrahiert und für andere Zwecke wiederverwendet zu werden. Es ist dazu gedacht, von Tools genutzt zu werden, die die folgenden Arten von Operationen ausführen:

- Einfache Extraktion von Text und Grafiken, um sie in andere Anwendungen einzufügen.

⁴ Die Projektaufgabe dazu ist in Abschnitt 2.2.5 der Machbarkeitsstudie [9] beschrieben.

- Automatisierte Neuordnung von Text und zugehörigen Grafiken, um auf eine Seite zu passen, die eine andere Größe als im ursprünglichen Layout besitzt.
- Verarbeiten von Text zum Suchen, Indizieren, zur Rechtschreibprüfung und ähnlichem.
- Umwandeln in andere gebräuchliche Dateiformate (wie HTML, XML und RTF), wobei die Dokumentenstrukturen und grundlegende Stilinformationen erhalten bleiben.
- Inhalte für Anwenderinnen und Anwender mit Sehbehinderung verfügbar machen (siehe Abschnitt [...], »Accessibility Support«).

Ausgehend von dieser Definition können wir nun einen Überblick über dieses Projekt geben, indem wir den zitierten Text aus Sicht der aktuellen Vorgehensweise von L^AT_EX analysieren und erläutern, wie wir diese ändern müssen. Der zitierte Text macht deutlich, dass wir zum Erstellen eines Tagged-PDF-Dokuments aus einer L^AT_EX-Quelle wie folgt vorzugehen haben:

Zuerst müssen wir ausnutzen, dass L^AT_EX weiß, wie sich die Komponenten eines Dokuments in der L^AT_EX-Quelldatei darstellen und wie sie dessen Struktur beschreiben – bestehend aus »Strukturtypen und -attributen«, wie zum Beispiel dem Absatztext, Überschriften, Listen, Listenelementen, Grafiken, Tabellen, Tabellenfeldern, Formeln und so weiter.

Dann müssen wir dieses Wissen über das Quelldokument einsetzen, um die notwendigen Komponenten zum PDF hinzuzufügen, das wir erstellen. Es gibt zwei grundlegende Zutaten dafür: Eine Baumstruktur, deren Knoten die »logischen Komponenten« des Dokuments repräsentieren, und ein Mechanismus zum Identifizieren dieser Elemente des »Seiteninhalts«, die mit jedem (Blatt-)Knoten in diesem Baum korrespondieren.

Diese »getaggten« Elemente des Inhalts können dann zusammen mit den korrespondierenden Strukturknoten von anderen (konsumierenden) Anwendungen verwendet werden, um Operationen wie die am Ende des oben zitierten Textes auszuführen.

All das klingt nicht allzu komplex angesichts der Tatsache, dass L^AT_EX weiß, was für Strukturen sich in der Quelldatei befinden, die es verarbeitet. Aber wie schon erläutert, werden diese Strukturinformationen aktuell aus Performancegründen (notwendig zu einer Zeit, als L^AT_EX designed und implementiert wurde) so schnell wie möglich wieder verworfen – und daher stehen sie beim Schreiben des Inhalts einer Seite in die PDF-Datei nicht mehr zur Verfügung.

Zudem gibt es ein paar problematische Details innerhalb dieser Prozesse. So handelt es sich beispielsweise bei dem Hauptinhalt eines PDF-Dokuments um eine

35-01 Zusammenstellung von »Seitenobjekten« und die »getaggten Elemente« des Textes
 35-02 müssen in jedes Seitenobjekt eingebettet werden – aber der asynchrone Paginie-
 35-03 rungsprozess von TeX ist für diese Situation nicht sehr hilfreich. Das Lösen dieses
 35-04 Problems ist daher Teil einer der Low-Level-Aufgaben im Projekt und eines, dass
 35-05 sich eventuell am besten durch Erweiterungen in der Engine lösen lässt.
 35-06

35-07
 35-08 Aus diesem Grund besteht ein Großteil der notwendigen Arbeiten im Projekt daraus,
 35-09 den Kern von LaTeX um all die Funktionalität, Datenstrukturen und Schnittstellen
 35-10 zu erweitern, die notwendig sind, um die vielen Details dieser Strukturinformatio-
 35-11 nen aufzubewahren und zu verarbeiten. Das führt zu einem deutlich verbesserten
 35-12 LaTeX-System, das Strukturinformationen in einer Quelle verwalten und in ein
 35-13 Ausgabeformat übertragen kann.
 35-14

35-15
 35-16 Ist all das ordentlich in LaTeX eingebettet, wird das Projekt weiter fortschreiten
 35-17 können und beispielsweise passende Schnittstellen auf Dokumentenebene für Daten
 35-18 anbieten, die aktuell – zumindest nicht standardmäßig – in der Quelldatei nicht
 35-19 explizit gemacht wurden. In diesem Bereich gibt es schon Lösungen für Aspekte
 35-20 wie Metadaten, PDF-Lesezeichen und so weiter, die von Ross Moore, Scott Pakin,
 35-21 Heiko Oberdiek und anderen erstellt wurden [13, 14, 16]. Wann immer sinnvoll
 35-22 möglich, sollten wir solche bestehenden nützlichen Arbeiten vereinheitlichen und
 35-23 einbinden. Die Unterstützung für einige (aber nicht alle) der allgemeineren (über
 35-24 das Tagging hinausgehenden) Anforderungen von »Accessible PDF« wird ebenfalls
 35-25 im Projekt angegangen.
 35-26

35-27
 35-28 Wir wollen hier auch ein paar (zumindest für uns) wichtige Beobachtungen be-
 35-29 züglich der Verarbeitung von Strukturinformationen mit LaTeX aufnehmen. Alles,
 35-30 was wir in der allgemeinen Beschreibung weiter oben beschrieben haben, ist von
 35-31 der (Frontend-)Verarbeitung auf Dokumentenebene bis hin zu den internen Da-
 35-32 tenstrukturen und Schnittstellen mehr oder weniger unabhängig vom anvisierten
 35-33 Ausgabeformat (in diesem Projekt Tagged PDF). Für jedes Ausgabeformat, das
 35-34 solche Strukturinformationen benötigt, unter anderem HTML, XML und andere
 35-35 »getaggte/strukturierte Formate«, ist eine identische Verarbeitung erforderlich.
 35-36

35-37
 35-38 Daher wird sich dieses Projekt zwar vor allem auf die PDF-Ausgabe konzentrieren
 35-39 (entweder direkt von der TeX-Engine oder über einen DVI-basierten Workflow gene-
 35-40 riert), es aber als Bonus auch leichter machen, andere solche Ausgabeformate zum
 35-41 Workflow hinzuzufügen, indem einfach ein Ausgabe-(Backend-)Modul ersetzt wird.
 35-42 Statt einer PDF-Ausgabe wird dann HTML5 oder ein anderes Format geschrieben
 35-43 werden. Aktuell sind solche alternativen Backends nicht vom Projekt abgedeckt,
 35-44 aber ist LaTeX erst einmal dazu in der Lage, Strukturinformationen über wohldefi-
 35-45 nierte Schnittstellen an ein Backend zu übergeben, gehen wir davon aus, dass die
 35-46 Unterstützung für andere strukturierte Ausgabeformate folgen wird. Solche Arbeit
 35-47
 35-48
 35-49
 35-50
 35-51
 35-52
 35-53
 35-54
 35-55
 35-56

könnte von uns oder von anderen Teams übernommen werden – möglicherweise parallel zu den späteren Phasen des in diesem Artikel vorgestellten Projekts.

Erläuterung des Entwurfs

Bei dem Dokument mit dem Titel »L^AT_EX Tagged PDF Feasibility Evaluation« [9], verfügbar auf der Website des L^AT_EX Project [18], handelt es sich um eine 40-seitige Studie, die detailliert sowohl die Projektziele wie auch die zu erledigenden Aufgaben beschreibt und mit einem Plan endet, wie das Projekt unserer Meinung nach angegangen werden sollte. Daher besteht es aus drei Teilen, beginnend mit einer »Introduction«, die einen Überblick über die Vorteile des Projekts enthält und dann erläutert, warum L^AT_EX-Dokumente ein guter Ausgangspunkt für das Erstellen von Tagged PDF sind.

Beim Lesen dieser Studie ist es sinnvoll, im Hinterkopf zu behalten, dass sie vor allem für ein Publikum bei Adobe gedacht ist,⁵ welches aus Entwicklerinnen und Entwicklern, Managerinnen und Managern mit einem breiten Wissen über digitale Typographie und elektronisches Publizieren besteht, die aber nicht notwendigerweise einen Hintergrund in der spezialisierten Welt von T_EX, L^AT_EX und dem Drumherum haben.

Die Hauptaufgaben

Nach dem einleitenden Überblick dokumentiert der nächste Teil der Studie (»Project Overview«) all die Hauptaufgaben, die wir als notwendig erachten, um die Projektziele zu erreichen. Diese Aufgaben sind unter fünf Überschriften kategorisiert, von denen die beiden wichtigsten »General L^AT_EX Extension Tasks« und »Structured PDF Tasks« sind.

Für jede Aufgabe beschreibt die Studie die Gründe für die Aufnahme und die Arbeiten, die zu erledigen sind. Bei den größeren Aufgaben werden auch sinnvolle Unteraufgaben herausgearbeitet. Um zudem die Zeitplanung des Projekts zu unterstützen, werden alle Abhängigkeiten der Aufgaben voneinander dokumentiert. Jeder solcher Abschnitt endet mit einer Liste von zu erwartenden Ergebnissen, die uns dabei helfen, unseren Fortschritt im Projekt zu messen und damit zu überwachen.

Es gibt insgesamt 20 Entwicklungsaufgaben, bei denen wir davon ausgehen, dass die notwendigen Arbeiten gut verstanden sind und daher »nur« zu erledigen sind – das aber sehr gut! Zudem gibt es ein paar Projektaufgaben, für die ausführlichere Untersuchungen erforderlich sind. Ihre Beschreibungen in der Studie sind weit weniger

⁵ Die veröffentlichte, zitierte Version der Studie wurde im September 2020 durch ein paar kleinere Anpassungen, Überarbeitungen und Klarstellungen aktualisiert.

37-01 vollständig, weil die Entwicklungsdetails einfach noch nicht gut genug bekannt
37-02 oder verstanden sind. Diese Untersuchungsaufgaben werden sehr wahrscheinlich
37-03 zu weiteren Entwicklungsaufgaben führen.
37-04

37-05 Beachten Sie, dass die Aufgaben in diesem Teil nach Kategorien unterteilt und daher
37-06 nicht in ihrer chronologischen Ausführungsreihenfolge im Gesamtplan aufgeführt
37-07 sind: Die Nummer einer Aufgabe besagt daher nicht, wann die Aufgabe im Projekt
37-08 angegangen werden wird.
37-09
37-10

37-11

37-12

37-13 Der Projektplan

37-14

37-15 Im abschließenden Hauptteil der Studie (»Project Timeline«) entwickeln wir einen
37-16 vollständigen Projektplan, der aus sechs Phasen besteht. Die Aufgaben wurden so
37-17 auf diese Phasen verteilt, dass die Abhängigkeiten zwischen ihnen berücksichtigt
37-18 werden. Zudem (und wichtiger noch) haben wir sichergestellt, dass die Ergebnisse
37-19 jeder Phase direkte Vorteile für die LaTeX-Community bieten.
37-20
37-21

37-22 Diese letztere Anforderung ist wichtig, weil sie zu zeitnahe Feedback und einer
37-23 frühzeitigen Verwendung führen wird. Wie Sie weiter unten sehen können, erwarten
37-24 wir beispielsweise, dass es mit dem Abschluss von Phase II schon möglich sein
37-25 wird, automatisch Tagged PDFs für einen eingeschränkten Satz von Dokumenten
37-26 zu generieren. In späteren Phasen werden diese Automationsmöglichkeiten auf
37-27 mehr Dokumente ausgeweitet werden.
37-28
37-29
37-30

37-31

37-32

37-33 Phase I – Das Fundament legen

37-34

37-35 Diese Phase deckt Aufgaben ab, die wichtige Voraussetzungen für spätere Arbeiten
37-36 sind – diese sind schon in Arbeit. Ein wichtiges Ergebnis ist hier das neue allgemeine
37-37 Hook-Management-System für LaTeX, das auf der TUG-Konferenz 2020 präsentiert
37-38 wurde und nun seit dem Release im Oktober 2020 Teil des LaTeX-Kerns ist.
37-39
37-40

37-41 Eine weitere Aufgabe ist die initiale Arbeit an den Low-Level-Anforderungen für
37-42 das Erstellen von Tagged PDF. Dies ist aktuell im stark experimentellen Paket tagpdf
37-43 verfügbar, das Prototyp-Code aus pdfresources-bezogenen Dateien für das Erstellen
37-44 und Managen von PDF-Objekten nutzt. Dieses Paket unterstützt daher Aufgaben
37-45 wie das Erstellen des Strukturbaums im PDF und das Hinzufügen der notwendigen
37-46 Verbindungen zwischen den Knoten in diesem Baum und dem »Inhalts-Stream«
37-47 jeder Seite.
37-48
37-49
37-50

37-51 Es ermöglicht momentan das Erstellen von Tagged-PDF-Dokumenten (mit einiger
37-52 Unterstützung durch den Anwender oder die Anwenderin) – es gilt aber zu beachten,
37-53 dass diese Arbeiten *sehr* experimentell sind und sich sowohl der Code wie auch die
37-54 Schnittstellen sehr wahrscheinlich ändern oder auch wieder verschwinden können.
37-55
37-56

Diese Phase ist die einzige, die nicht dazu gedacht ist, Anwenderinnen und/oder Autoren signifikante neue Funktionalitäten zu bieten. Trotzdem werden die Standardisierung und die vom allgemeinen Hook-Management bereitgestellten Schnittstellen diesen Gruppen neben den eigentlich anvisierten Paket-Entwicklerinnen und -Entwicklern unbestreitbar Vorteile bringen.

Phase II – Einfache Dokumente taggen können

Das Hauptziel von Phase II ist eine grundlegende Unterstützung für das Automatisieren des Tagging-Prozesses. In dieser Phase werden durch das Automatisieren nur recht einfache Dokumente abgedeckt: Solche, die keine der komplizierteren Strukturen wie Formeln und Tabellen enthalten.

Das automatisierte Tagging wird in dieser Phase nicht im Kernel liegen, sondern durch das Laden eines extra Paketes bereitgestellt und es wird sich nur um eine prototypische Implementierung handeln.

Es werden in dieser Phase eine Reihe von »Workarounds« erforderlich sein, um die notwendigen, aber noch fehlenden Features bereitzustellen. So wird beispielsweise kein vollständiges Querverweissystem aufgesetzt (ähnlich dem Paket `zref` von Heiko Oberdiek, aber in den L^AT_EX-Kern eingebaut), sondern wir werden entweder dieses Paket verwenden oder zusätzlichen temporären Code einsetzen, der Probleme umgehen soll, die daraus resultieren, dass diese Funktionalität noch keine Kernelkomponente ist. All dieser »Workaround-Code« wird in einer späteren Phase entfernt oder angepasst werden müssen.

Diese Arbeitsreihenfolge wurde gewählt, um recht früh im Projekt greifbare Ergebnisse bieten zu können, statt in dieser Phase zu viel Zeit für rein interne Verbesserungen aufzuwenden, die keine direkt sichtbare Anwendbarkeit haben und damit für viele Benutzerinnen und Benutzern ohne Wert zu sein scheinen.

Phase III – Die Workarounds für das Taggen entfernen

Das Hauptziel von Phase III ist, die Abdeckung des automatischen Taggings auf eine größere Varietät von Dokumenten zu erweitern, indem mehr zentrale Dokumentelemente für das Tagging angepasst werden. Diese Erweiterungen ermöglichen es uns zudem, die in der vorigen Phase für das Bereitstellen eines funktionierenden Prototypen eingeführten »Workarounds« zu entfernen. Diese Phase wird außerdem einen Metadaten-Mechanismus enthalten, der in den L^AT_EX-Kern integriert ist.

Phase IV – Grundlegendes Tagging und Hyperlinks verfügbar machen

Hauptziel von Phase IV ist, den gesamten Code aus den Prototyp-Paketen in den Kernel aufzunehmen. Das muss sehr sorgfältig und vorsichtig geschehen, da es keine negativen Auswirkungen auf die Verarbeitung älterer Dokumente geben soll. Daher erwarten wir, mindestens einen vollständigen LaTeX-Releasezyklus zu benötigen,⁶ um diese Arbeit abzuschließen.

Phase V – Die Tagging-Fähigkeiten erweitern

Ist das grundlegende Tagging verfügbar, liegt der Fokus von Phase V darauf, weitere Unterstützung für das Tagging zu bieten, indem Tabellen und Formeln mit in die unterstützten Dokumentstrukturen aufgenommen werden.

In dieser Phase werden zudem Schnittstellen hinzugefügt, wenn es notwendig ist, weitere Aspekte von Tagged PDF zu unterstützen, wie zum Beispiel die Spezifikation von alternativem Text (für Formeln, Abbildungen und so weiter).

Phase VI – Standards unterstützen

Phase VI konzentriert sich schließlich auf das Bereitstellen von Unterstützung für die relevanten PDF-Standards (wie PDF/A [5], genauer beschrieben in [15], und PDF/UA [4], siehe auch [2]) – zumindest so weit, wie dies direkt mit LaTeX möglich ist, also ohne Post-Processing der PDF-Datei. Auch wird die Kernelunterstützung für einige weitere Funktionen Bestandteil dieser Phase sein, wie zum Beispiel das Erstellen von Outlines für ein Dokument und das Einbinden von »associated Files« (zugehörigen Dateien) in ein PDF.

Recherchearbeit

Neben diesen sechs Phasen, die nur Aufgaben enthalten, welche aus technischer Sicht größtenteils verstanden sind, gibt es eine Reihe weiterer Aufgaben, für die vorbereitende Untersuchungen notwendig sind, um zu verstehen, welche Entwicklungsaufgaben erforderlich sein werden. Diese Recherche wird parallel zu anderen Arbeiten durchgeführt und es ist – wie schon erwähnt – sehr wahrscheinlich, dass sie zur Spezifikation zusätzlicher Entwicklungsaufgaben führen wird. Daher werden die Rechercheergebnisse vermutlich zu Anpassungen und Erweiterungen in den Phasen IV bis VI führen.

⁶ Siehe »Zeitplan und andere Überlegungen« zu weiteren Informationen über die Phasen und diesen Releasezyklus.

Veränderungen, Anpassungen und Reporting

Insgesamt glauben wir, dass dieser Plan einen konsistenten und vernünftigen Ansatz bietet, um die gewünschten Ziele des Projekts zu erreichen. Aber es wird ohne Zweifel Änderungen geben, da beispielsweise die Forschungsaufgaben zu einer Erweiterung der geplanten Aufgaben und auch Änderungen an einigen von ihnen führen werden. Zudem haben wir schon festgestellt, dass wir ein paar der Deliverables (aus den größeren Aufgaben) in frühere Phasen vorziehen wollen. Und in manchen Fällen steht die Überlegung an, eine Aufgabe nicht vollständig innerhalb der geplanten Phase abzuschließen, weil es wichtiger erscheint, uns erst auf andere Arbeiten zu konzentrieren (natürlich werden solche Anpassungen nur in Fällen funktionieren, in denen diese Deliverables nicht für andere Aufgaben erforderlich sind). Schließlich mag die Arbeit an einer Aufgabe auch noch zusätzliche Arbeiten aufdecken, die erforderlich sind, aber bisher nicht berücksichtigt wurden, was zu neuen Unteraufgaben mit weiteren Deliverables führt.

Daher planen wir, regelmäßig (öffentliche) fortlaufende Berichte (von sinnvollem Umfang) zum Status und zu den Ergebnissen des Projekts zu liefern, in denen auch Ergänzungen oder Richtungsänderungen enthalten sind. Die Machbarkeitsstudie mit ihrer ziemlich detaillierten Dokumentation und den Deliverables wird ein guter Ausgangspunkt für diese wichtige Aktivität sein. Die genaue Form dieser Berichte ist noch nicht festgelegt, aber wir werden sie zu gegebener Zeit auf der Website des L^AT_EX-Teams [18] und/oder im *TUGboat* Journal veröffentlichen.

Zeitplan und andere Überlegungen

Die von uns (in der Machbarkeitsstudie) erstellte Abschätzung für den zeitlichen Aufwand für die einzelnen Aufgaben basiert auf der Annahme, dass *mindestens* (das Äquivalent) eines Software-Entwicklers beziehungsweise einer Entwicklerin exklusiv und in Vollzeit an jeder Aufgabe in dem zugewiesenen Zeitabschnitt arbeitet und dabei nach Bedarf zusätzliche Unterstützung von anderen Entwicklern und Entwicklerinnen enthalten. Uns ist sehr wohl bewusst, dass Aktivitäten wie das Dokumentieren oder Testen substanzielle Beiträge von Spezialistinnen und Spezialisten enthält, für die ebenfalls Ressourcen geschaffen werden müssen.

Es hängt von einer Reihe von Faktoren ab, ob dieses Aufwandsniveau durchgehend möglich sein wird – ein wichtiger ist die Verfügbarkeit einer ausreichenden Finanzierung, die den verantwortlichen Entwicklerinnen und Entwicklern und anderen wichtigen Beteiligten die Sorge um ihren Lebensunterhalt nehmen kann. Die finanzielle Unterstützung durch Adobe hilft zwar sehr dabei, die Erfordernisse des Projekts abzudecken, aber sie wird nicht alleine ausreichen, um die Arbeiten vollständig umsetzen zu können. Aus diesen Gründen sollten die Angaben des

41-01 Zeitbedarfs in der Studie als Schätz-, und nicht als endgültige Werte betrachtet
41-02 werden.
41-03

41-04 Ein realistisches Szenario wäre, dass für jede Phase zwei bis drei Release-Zyklen
41-05 von LaTeX erforderlich sind, von denen es jeweils zwei pro Jahr gibt. Daraus ergibt
41-06 sich, dass dieses Projekt mindestens vier Jahre laufen wird – sehr wahrscheinlich
41-07 aber länger. Zusätzliche Finanzierung wird dabei helfen, jede Phase rechtzeitig
41-08 abschließen zu können, zudem erlaubt sie uns, den Leistungsumfang in manchen
41-09 Bereichen noch zu erweitern. Trotz alledem ist es angesichts der Komplexität des
41-10 Themas nicht realistisch, zu erwarten, dass Phasen vorzeitig abgeschlossen werden
41-11 können, auch wenn die Finanzierung vollständig gesichert ist.
41-12

41-13 Es sei auch darauf hingewiesen, dass alle notwendigen Updates an wichtigen ex-
41-14 ternen Paketen (die nicht vom LaTeX Project Team betreut werden) durch externe
41-15 Ressourcen vorgenommen werden müssen – also die Betreuerinnen und Betreuer
41-16 dieser Pakete. Diese Annahme ist vermutlich nicht in allen Fällen realistisch (siehe
41-17 beispielsweise die Diskussion in [9, Task 2.4.3]). In solchen Fällen wird zusätzli-
41-18 che Arbeit als Teil des Projekts anfallen, die ebenfalls die Zeitplanung beeinflusst
41-19 oder das Einstellen zusätzlicher Entwickler und Entwicklerinnen für diese Arbeit
41-20 erfordert.
41-21

41-22 Schließlich hängt der Erfolg des Projekts auch sehr stark von einer produktiven
41-23 Zusammenarbeit mit vielen Leuten in der LaTeX-Community ab: Zum Testen, Schrei-
41-24 ben von Paketen und möglicherweise auch mit denjenigen, die sich eher mit den
41-25 zugrundeliegenden TeX-Engines und den diversen Hilfsprogrammen befassen, die
41-26 zum Erzeugen von PDF-Ausgaben genutzt werden. Zudem benötigen wir Unterstüt-
41-27 zung von einer Reihe von Expertinnen und Experten für die Produktion qualitativ
41-28 hochwertiger PDFs, die uns mit ihrem professionellen Wissen erklären können, wie
41-29 die Consumer-Anwendungen diese Features in der Praxis verwenden.
41-30

41-31 Erste Ergebnisse 41-32 41-33 41-34 41-35 41-36 41-37 41-38 41-39 41-40

41-41 Wie schon erwähnt, haben wir das Projekt trotz COVID-19 auf den Weg gebracht.
41-42 Daher können wir schon ein paar Erfolgserlebnisse vermelden:
41-43

- 41-44 • Das neue LaTeX Hook Management System (Task 2.2.5 in [9]), präsentiert auf
41-45 der TUG 2020 [8] und in einem TUGboat-Artikel [7];
- 41-46 • PDF-Stringunterstützung (Task 2.2.1 in [9]), die vor allem intern wichtig für
41-47 viele weitere Aufgaben ist;
- 41-48 • Erste Experimente mit tagpdf-Prototyp-Code (Teil von Task 2.2.6 in [9]). In [3]
41-49 finden Sie mehr dazu.
41-50
41-51
41-52
41-53
41-54
41-55
41-56

Ausblick

Aktuell stecken wir mitten in Phase I und gehen davon aus, dass sie mit dem Frühlings-Release 2021 von L^AT_EX abgeschlossen sein wird.⁷ Da wir planen, die L^AT_EX-Releasetermine nächstes Jahr zu verschieben, um sie besser mit den jährlichen T_EX Live-Distributionen abzustimmen, erwarten wir, mit manchen der Aufgaben aus Phase II schon vor dem nächsten L^AT_EX-Release zu beginnen.

Neben der aktuellen Arbeit an Entwicklungsaufgaben werden auch administrative Aufgaben angegangen werden, zum Beispiel:

- Verbindungen zu und Zusammenarbeit mit externen Expertinnen und Experten ermöglichen, um Wissen in Bereichen zu erlangen, in denen das aktuelle Team noch Lücken hat, und um sicherzustellen, dass sich unsere Arbeit auf die relevanten Ziele konzentrieren kann;
- Ausschau halten nach zusätzlicher finanzieller Unterstützung, um mehr Expertise einholen und damit hoffentlich die späteren Phasen beschleunigen zu können;
- Ein professionelles System für die qualitativ und quantitativ hochwertige Produktion der Dokumentation einrichten.

Natürlich wird die Entwicklung und Wartung von L^AT_EX in den kommenden Jahren nicht nur auf dieses Projekt beschränkt sein. Es gibt viele andere Aktivitäten, die das L^AT_EX Project Team parallel dazu umsetzen will. Seine Ergebnisse werden – wie bisher – über die üblichen Kommunikationskanäle bekanntgegeben werden, zum Beispiel unsere Website, l^AT_EX-Newsletter, TUGboat-Artikel und/oder Internet-basierte Diskussionen im Chat von StackExchange, auf L^AT_EX-L und an anderen Orten.

Literatur

- [1] Adobe Document Cloud, https://en.wikipedia.org/wiki/Adobe_Document_Cloud.
- [2] Olaf Drümmer, Bettina Chang: PDF/UA in a Nutshell — Accessible documents with PDF, <https://pdfa.org/resource/pdfua-in-a-nutshell/>, PDF Association, 2013.
- [3] Ulrike Fischer: »Creating accessible pdfs with L^AT_EX«, TUGboat, 41.1 (2020) 26–28, ISSN: 0896-3207, <https://tug.org/TUGboat/tb41-1/tb127fischer-accessible.pdf>.

⁷ Zum Zeitpunkt der Übersetzung dieses Artikels (Sommer 2021) sind die Arbeiten an Phase I abgeschlossen und die an Phase II in Bearbeitung, d.h., bisher gibt es keine Verzögerungen im Zeitplan.

- 43-01 [4] ISO 14289-1:2014 Document management applications — Electronic do-
43-02 cument file format enhancement for accessibility – 1: Use of ISO 32000-1
43-03 (PDF/UA-1),
43-04 <https://iso.org/standard/64599.html>, Technical Committee ISO/TC
43-05 171/SC 2, 2014.
- 43-06 [5] ISO 19005-2:2011 Document management — Document file format for
43-07 long-term preservation — Part 2: Use of ISO 32000-1 (PDF/A-2),
43-08 <https://iso.org/standard/50655.html>, Technical Committee ISO/TC
43-09 171/SC 2, 2011.
- 43-10 [6] ISO 32000-1:2008 Document management — Portable document format
43-11 (PDF 1.7), <https://iso.org/standard/51502.html>. Freely available as [17],
43-12 Technical Committee ISO/TC 171/SC 2, 2008.
- 43-13 [7] Frank Mittelbach: »Quo vadis LaTeX(3) Team — A look back and at the
43-14 upcoming years«, *TUGboat*, 41.2 (128 2020), [https://latex-project.org/](https://latex-project.org/publications/indexbyyear/2020/)
43-15 [publications/indexbyyear/2020/](https://latex-project.org/publications/indexbyyear/2020/), 201–207, ISSN: 0896-3207.
- 43-16 [8] — »Video: Quo vadis LaTeX(3) Team — A look back and at the upcoming
43-17 years« (2020), Aufzeichnung des Vortrags auf der Online TUG 2020 Confe-
43-18 rence, <https://youtu.be/zNci41cb8Vo>.
- 43-19 [9] Frank Mittelbach, Ulrike Fischer, Chris Rowley: LaTeX Tagged PDF Feasibi-
43-20 lity Evaluation, [https://latex-project.org/publications/indexbyyear/](https://latex-project.org/publications/indexbyyear/2020/)
43-21 [2020/](https://latex-project.org/publications/indexbyyear/2020/), LaTeX Project, 2020.
- 43-22 [10] Frank Mittelbach, Chris Rowley: »LaTeX Tagged PDF – A blueprint for a
43-23 large project«, *TUGboat*, 41.3 (129 2020), [https://latex-project.org/](https://latex-project.org/publications/indexbyyear/2020/)
43-24 [publications/indexbyyear/2020/](https://latex-project.org/publications/indexbyyear/2020/), 292–298, ISSN: 0896-3207.
- 43-25 [11] Ross Moore: »Implementing PDF standards for mathematical publishing«,
43-26 *TUGboat*, 39.2 (122 2018), 131–135, ISSN: 0896-3207, [https://tug.org/](https://tug.org/TUGboat/tb39-2/tb122moore-pdf.pdf)
43-27 [TUGboat/tb39-2/tb122moore-pdf.pdf](https://tug.org/TUGboat/tb39-2/tb122moore-pdf.pdf).
- 43-28 [12] — Website: Tagged PDF Beispiele und Ressourcen, [http://maths.mq.edu.](http://maths.mq.edu.au/~ross/TaggedPDF)
43-29 [au/~ross/TaggedPDF](http://maths.mq.edu.au/~ross/TaggedPDF) und insbesondere /TUG2019-movies, 2020.
- 43-30 [13] Ross Moore u. a.: »Generation of PDF/X- and PDF/A-compliant PDFs with
43-31 pdfTeX — pdfx.sty«, <https://ctan.org/pkg/pdfx>, 2019.
- 43-32 [14] Heiko Oberdiek: »The bookmark package«, <https://ctan.org/pkg/bookmark>,
43-33 2019.
- 43-34 [15] Alexandra Oettler: PDF/A in a Nutshell — PDF for long-term archiving,
43-35 <https://pdfa.org/resource/pdfa-in-a-nutshell/>, PDF Association, 2013.
- 43-36
- 43-37
- 43-38
- 43-39
- 43-40
- 43-41
- 43-42
- 43-43
- 43-44
- 43-45
- 43-46
- 43-47
- 43-48
- 43-49
- 43-50
- 43-51
- 43-52
- 43-53
- 43-54
- 43-55
- 43-56

- [16] Scott Pakin: »The hyperxmp package«, <https://ctan.org/pkg/hyperxmp>, 2020.
- [17] PDF Reference 1.7, https://www.adobe.com/devnet/pdf/pdf_reference.html. Freely available version of [6], Adobe Systems Inc., 2006.
- [18] L^AT_EX Project Team: Website des L^AT_EX Project, <https://latex-project.org/>.

L^AT_EX News – Issue 32, Oktober 2020¹

Frank Mittelbach, L^AT_EX Project

Einleitung

Das L^AT_EX-Release 2020-10-01 zeigt, dass die Arbeit an der Verbesserung von L^AT_EX erneut intensiviert wurde. Die zwei wichtigsten neuen Features sind die Kernel-Unterstützung für xparse und die Einführung des neuen Hook-Management-Systems für L^AT_EX, aber es gibt auch viele kleinere Verbesserungen und Bugfixes im Kernel und in diversen Paketen.

xparse im Format bereitstellen

Die offizielle Schnittstelle im L^AT_EX-Kernel zum Erstellen von Befehlen auf Dokumentenebene war immer schon `\newcommand`. Das war im Vergleich zu L^AT_EX 2.09 ein großer Schritt nach vorne. Aber die Art von Befehlen, die so erzeugt werden konnten, war immer noch sehr beschränkt: Höchstens ein optionales Argument in eckigen Klammern, gefolgt von null oder mehr verpflichtenden Argumenten. Für eine ausgefeiltere Syntax musste die T_EX-Primitive `\def` zusammen mit entsprechender Makroprogrammierung auf einer tieferen Ebene eingesetzt werden.

Das L^AT_EX-Team begann mit der Arbeit an xparse, einem umfassenden Parser für Befehle auf Dokumentenebene, in den späten 1990er Jahren. Im letzten Jahrzehnt wurden die experimentellen Ideen, die darin einfließen, sorgfältig aufgearbeitet und auf ein stabiles Fundament gesetzt. Seitdem wird xparse genutzt, um eine große Zahl von Dokumenten- und Paketbefehlen zu definieren. Das funktioniert, weil das Paket eine umfangreiche und in sich konsistente Syntax zum Beschreiben vieler verschiedener Schnittstellen bietet, die in L^AT_EX-Paketen zu finden sind.

Die Ideen, die in xparse entwickelt wurden, sind nun ausreichend getestet, so dass der größte Teil in den L^AT_EX-Kernel übernommen werden kann. Daher wurden folgende Befehle hinzugefügt:

¹ Das Newsletter wurde zuvor im TUGboat 41:3 [2] veröffentlicht und von Thomas Demmig übersetzt.

- \NewDocumentCommand, \RenewDocumentCommand,
 \ProvideDocumentCommand, \DeclareDocumentCommand
- \NewExpandableDocumentCommand,
 \RenewExpandableDocumentCommand,
 \ProvideExpandableDocumentCommand,
 \DeclareExpandableDocumentCommand
- \NewDocumentEnvironment, \RenewDocumentEnvironment,
 \ProvideDocumentEnvironment, \DeclareDocumentEnvironment
- \BooleanTrue \BooleanFalse
- \IfBooleanTF, \IfBooleanT, \IfBooleanF
- \IfNoValueTF, \IfNoValueT, \IfNoValueF
- \IfValueTF, \IfValueT, \IfValueF
- \SplitArgument, \SplitList, \TrimSpaces, \ProcessList,
 \ReverseBoolean
- \GetDocumentCommandArgSpec \GetDocumentEnvironmentArgSpec

Die meisten – allerdings nicht alle – Argumenttypen, die durch xparse definiert werden, werden nun auf Kernel-Ebene unterstützt. Speziell die Typen g/G, l und u werden *nicht* vom Kernel-Code angeboten – diese sind veraltet, stehen aber durch das explizite Laden von xparse immer noch zur Verfügung. Alle anderen Argumenttypen *sind* nun direkt im LaTeX 2_ε-Kernel vorhanden. Die Dokumentation dieser Kommandos findet man in `usrguide3.pdf` als Teil der LaTeX-Distribution.

Ein Hook-Management-System für LaTeX

Mit dem LaTeX-Release aus dem Herbst 2020 stellen wir ein allgemeines Hook-Management-System für den Kernel und für Pakete zur Verfügung. Damit können Pakete Code sicher in eine Reihe von Kernel- und Paket-Hooks einhängen und bei Bedarf Regeln definieren, um den Code in den Hooks neu zu ordnen, falls es zu den typischen Reihenfolge-Problemen beim Laden von Paketen kommt. Dieses Hook-System ist in der Programmierschicht L3 geschrieben und bildet damit die erste größere Anwendung innerhalb des Kernels, die die nun verfügbare LaTeX3-Funktionalität nutzt (wenn wir xparse außer Acht lassen, das es schon lange als eigenes Paket gibt).

Die Datei `lthooks.dtx` enthält den zentralen Management-Code für Hooks und hier werden auch die grundlegenden Hooks für Umgebungen definiert (wie zuvor durch `etoolbox` angeboten), `ltshipout.dtx` liefert Kernel-Hooks in den Shipout-Prozess (wodurch Pakete wie `atbegshi` und so weiter überflüssig werden) und die Datei `ltfilehook.dtx` enthält Redefinitionen für Befehle wie `\input` oder `\usepackage`, so dass diese Hooks in einer Art und Weise anbieten, die der vom Paket `filehook` ähnelt.

Momentan ist die Integration schlank, sie überschreibt Definitionen, die vorher während des Generierens des Formats erstellt wurden (das wird sich allerdings später ändern, wenn umfangreicher getestet wurde). Aus diesem Grund liegt die Dokumentation auch noch nicht in ihrer endgültigen Form vor und man muss sich drei verschiedene Dokumentationen durchlesen:

lthooks-doc.pdf Zentrale Management-Schnittstelle und grundlegende Hooks für Umgebungen, die durch den Kernel bereitgestellt werden.

ltskipout-doc.pdf Hooks, die beim Ausgeben einer Seite verfügbar sind.

lfilehook-doc.pdf Hooks, die beim Lesen einer Datei angeboten werden.

Wer sich auch den Code anschauen möchte, ersetzt -doc durch -code, also beispielsweise lthooks-code.pdf. Alle Dokumente sollten über texdoc zur Verfügung stehen, zum Beispiel sollte

```
texdoc lthooks-doc
```

die zentrale Dokumentation liefern.

Andere Änderungen am L^AT_EX-Kernel

`\symbol` im Mathematik-Modus für große Unicode-Werte

Der L^AT_EX 2_ε-Kernel definiert den Befehl `\symbol`, mit dem man Zeichen setzen kann, indem man ihre ›Slotnummer‹ angibt. Bei den LuaT_EX- und XeL^AT_EX-Engines lassen sich diese Slotnummern auf sehr große Werte ausdehnen, um Unicode-Zeichen in den höheren Unicode-Planes zu erreichen (zum Beispiel hat Bold Mathematical Capital A die Slotnummer "1D400 in Hex oder 119808 dezimal). In XeL^AT_EX war es nicht möglich, für `\symbol` Werte über 2¹⁶ im Mathematik-Modus einzusetzen – diese Einschränkung wurde nun aufgehoben.

(Github Issue 124)

Korrekter Unicode-Wert von `\=y` ($\bar{y}$)

Der Unicode-Slot für $\bar{y}$ zeigte fälschlicherweise auf den Slot für $\bar{Y}$. Das wurde korrigiert.

(Github Issue 326)

Unterstützung für Unicode Soft Hyphen ergänzt

Seit langer Zeit sorgt die UTF-8-Option für inputenc dafür, dass das Unicode-Zeichen Soft Hyphen (U+00AD) ein Alias für den weichen Bindestrich `\-` bei L^AT_EX ist. Die Unicode-Engines XeL^AT_EX und LuaT_EX haben sich aber anders verhalten: Entweder ignorierten sie U+00AD oder sie interpretierten das Zeichen als echten Bindestrich.

47-01 Diese Inkonsistenz ist nun behoben und LaTeX behandelt U+00AD immer als \-.
 47-02 (Github Issue 323)
 47-03
 47-04
 47-05

47-06 Akzente für Großbuchstaben in Unicode-Engines korrigieren
 47-07

47-08 In Unicode-Engines wurden Akzente für Großbuchstaben (wie zum Beispiel
 47-09 \capitalcedilla) als triviale Abkürzungen für die normalen Akzente implementiert
 47-10 (weil sie abgesehen von Computer Modern von so gut wie keinem Font unterstützt
 47-11 werden), aber das ging schief, wenn hyperref geladen wurde. Dies wurde nun
 47-12 korrigiert.
 47-13 (Github Issue 332)
 47-14
 47-15
 47-16

47-17 calc in einer Reihe von Kernel-Befehlen unterstützen
 47-18

47-19 Die Befehle \hspace, \vspace, \addvspace, \ und weitere übergeben ihr Argument
 47-20 einfach an eine TeX-Primitive, um den erforderlichen Platz zu erzeugen. Dadurch
 47-21 war es unmöglich, in solchen Argumenten irgendetwas anderes als einen einfachen
 47-22 Dimensionswert anzugeben. Das wurde geändert, so dass bei diesen Befehlen nun
 47-23 auch calc-Syntax unterstützt wird.
 47-24 (Github Issue 152)
 47-25
 47-26
 47-27

47-28 ε -TeX-Längenausdrücke in picture-Koordinaten unterstützen
 47-29

47-30 Picture-Modus-Koordinaten, die per $\langle x \rangle, \langle y \rangle$ angegeben werden, erlaubten bisher
 47-31 Mehrfache von \unitlength. Nun unterstützen sie auch ε -TeX-Längenausdrücke
 47-32 (wie zum Beispiel von der Primitive \glueexpr genutzt, auch wenn alle Verwendun-
 47-33 gen im picture-Modus nicht dehnbar sind).
 47-34
 47-35
 47-36

47-37 Also sind wie zuvor Angaben wie \put(2,2) erlaubt, nun aber auch so etwas wie
 47-38 \put(\textwidth-5cm,0.4\textheight).
 47-39

47-40 Beachten Sie, dass Sie nur Ausdrücke mit Längen verwenden können – Eingaben
 47-41 wie \put(1+2,0) werden nicht unterstützt.
 47-42
 47-43

47-44 Leerzeichen in Namen von eingebundenen Dateien
 47-45

47-46 Dateinamen mit Leerzeichen führen zu unerwarteten Ergebnissen, wenn man sie in
 47-47 den Befehlen \include und \includeonly einsetzt. Das wurde nun behoben und das
 47-48 Argument von \include kann einen Dateinamen mit Leerzeichen enthalten. Führen-
 47-49 de oder abschließende Leerzeichen werden entfernt, aber Leerzeichen im Dateina-
 47-50 men bleiben erhalten. Das Argument von \includeonly, eine kommaseparierte Liste
 47-51
 47-52
 47-53
 47-54
 47-55
 47-56

mit zu verarbeitenden Dateien, kann ebenfalls Leerzeichen enthalten, wobei alle führenden und abschließenden Leerzeichen der einzelnen Dateinamen entfernt werden, während Leerzeichen *in* Dateinamen erhalten bleiben. (*Github Issues 217 und 218*)

Zusätzliche Zeilen in `\centering`, `\raggedleft` oder `\raggedright` vermeiden

Richten wir Absätze nicht aus, kann ein sehr langes Wort (länger als eine Zeile) zu einer unnötigen Zusatzzeile führen, um einen Trennstrich in der zweitlängsten Zeile des Absatzes zu verhindern. Das wird nun vermieden, indem `\finalhyphendemerits` im Flattersatz, bzw. zentriertem Satz auf null gesetzt wird. (*Github Issue 274*)

`\baselineskip` in Textskripten auf einen Wert ungleich null setzen

Da `\textsuperscript` und `\textsubscript` normalerweise nur ein paar Zeichen in einer Zeile enthalten, wurde `\baselineskip` auf null gesetzt. Allerdings nutzt `hyperref` diesen Wert, um die Höhe einer Linkbox zu bestimmen, die daher deutlich zu klein erstellt wurde. Das ist nun angepasst worden. (*Github Issue 249*)

Abstandsprobleme beim Einsatz von `\linethickness`

In manchen Situationen hat der Einsatz von `\linethickness` zu überflüssigem Abstand geführt, durch den Objekte in einer `picture`-Umgebung nach rechts verschoben wurden. Dies ist nun korrigiert worden. (*Github Issue 274*)

Bessere Unterstützung für die Standard-Schnittstelle der Schriftstärke bei Altdokumenten

In der ersten Implementierung von L^AT_EXs Font-Auswahlschema (NFSS) wurden Änderungen an einem Standardwert umgesetzt, indem bestimmte Befehle neu definiert wurden, zum Beispiel `\seriesdefault`. 2019 haben wir eine Reihe von Erweiterungen eingeführt und damit neue Methoden, bestimmte Teile von NFSS anzupassen. Zum Beispiel ist der empfohlene Weg, Schriftstärken-Standard zu ändern, nun `\DeclareFontSeriesDefault`, siehe [5]. In diesem Release haben wir die Unterstützung für Altdokumente verbessert, die die alte Methode verwenden, um zusätzliche Randfälle abzudecken. (*Github Issues 306 and 315*)

Unterstützung für ungewöhnliche Schriftstärken als Standardwerte

Wurde eine Font-Familie mit recht ungewöhnlichen Standardwerten für die Schriftstärke eingerichtet, zum Beispiel

```
\renewcommand\ttdefault{lmvt}
\DeclareFontSeriesDefault[t]{md}{lm}
\DeclareFontSeriesDefault[t]{bf}{bm}
```

wurde der Text nach einem Wechsel der Haupt-Dokumentfamilien, zum Beispiel `\ttfamily... \rmfamily`, nicht immer korrekt weiter in Medium oder Fett gesetzt, wenn dazu die Werte für `\mdseries` oder `\bfseries` anzupassen waren. Das wurde nun korrigiert. *(Github Issue 291)*

Den Kontext der aktuellen Schriftstärke prüfen

Manchmal ist es notwendig, Befehle zu definieren, die sich anders verhalten, wenn sie in einem fetten Umfeld genutzt werden (zum Beispiel innerhalb von `\textbf`). Nachdem es in LaTeX nun möglich ist, unterschiedliche »bf«-Standards für die drei Metafamilien (rm, sf und tt) über `\DeclareFontSeriesDefault` zu definieren, ist es nicht mehr länger einfach, die Frage zu beantworten, ob man sich in einem fetten Kontext befindet. Um bei diesem Problem zu helfen, wurde ein neuer Befehl bereitgestellt:

```
\IfFontSeriesContextTF{<context>}{<true code>}{<false code>}
```

Der `<context>` kann entweder `bf` (fett) oder `md` (medium) sein und abhängig davon, ob der aktuelle Font als über `\bfseries` oder `\mdseries` ausgewählt erkannt wurde, wird `<true code>` oder `<false code>` ausgeführt. Ein Beispiel:²

```
\usepackage{bm} % (bold math)
\newcommand\vbeta{\IfFontSeriesContextTF{bf}%
{\ensuremath{\bm{\beta}}}%
{\ensuremath{\beta}}}
```

So können Sie etwa `\vbeta`-Isotope schreiben, welches in einer Überschrift dann fett als **β -Isotope** gesetzt würde. *(Github Issue 336)*

Überflüssige Warnungen bei Paketoptionen vermeiden

Wird ein Paket mit einer Reihe von Optionen geladen, zum Beispiel X, Y und Z, und wird dann später erneut versucht, das Paket zu laden – dieses Mal aber mit einer Untermenge der Optionen oder gar keinen Optionen –, war es möglich,

² In lualatex würde man `\sympfit` an Stelle von `\bm` verwenden.

eine Fehlermeldung zu erhalten, dass Option X dem Paket nicht bekannt ist. Diese offensichtlich falsche Meldung entstand durch ein Timing-Problem, bei dem die Liste der verfügbaren Optionen zu früh verworfen wurde. Das ist nun korrigiert worden. (Github Issue 22)

fleqn anpassen

In amsmath ist der Parameter `\mathindent` im fleqn-Design ein dehnbarer Längenparameter, der es erlaubt, ihn auf einen Wert wie `1em minus 1em` zu setzen, so dass die normale Einrücktiefe bei sehr breiten Formeln verringert werden kann. Das wird nun auch von den L^AT_EX-Standardklassen unterstützt.

Zudem wurde ein verringerbarer Abstand zwischen Formel und Formelnummer in der equation-Umgebung ergänzt, wenn die fleqn-Option genutzt wird, so dass eine sehr breite Formel nicht in die Formelnummer läuft. (Github Issue 252)

\clap bereitstellen

L^AT_EX hat `\llap` und `\rlap` von Plain-T_EX übernommen (Boxen der Breite null, deren Inhalt links beziehungsweise rechts daran klebt), aber es gibt keinen entsprechenden `\clap`-Befehl, der das Material zentriert. Dieser fehlende Befehl wurde durch diverse Pakete ergänzt, zum Beispiel durch `mathtools`, und ist nun im Kernel hinzugefügt worden.

Fix für die alte Mathematik-Alphabet-Schnittstelle

Beim Einsatz der alten Mathematik-Alphabet-Schnittstelle aus L^AT_EX 2.09, zum Beispiel `\sf -1$` statt `\mathsf{-1}$`, wurde ein zusätzliches Mathematik-Ord-Atom zur Formel hinzugefügt, falls das Mathematik-Alphabet erstmalig zum Einsatz kam. In manchen Fällen hat dieses Mathematik-Atom die Abstände verändert, zum Beispiel das unäre Minuszeichen im obigen Beispiel in ein binäres Minuszeichen umgewandelt. Das wurde endlich behoben. (Gnats Issue latex/3357)

Zusätzliche Prüfungen für Format-, Paket- und Klassen-Datumswerte

Um Kompatibilitäts-Code zu implementieren oder sicherzustellen, dass bestimmte Features verfügbar sind, ist es hilfreich und oft notwendig, das Datum des Formats eines Pakets oder einer Klasse zu prüfen und abhängig vom Ergebnis unterschiedlichen Code auszuführen. Dafür gab es bei L^AT_EX bisher nur interne Befehle (`\ifpackagelater` und `\ifclasslater`), um Paket- oder Klassennamen zu prüfen,

51-01 aber nichts sinnvolles für das Testen des Format-Datums. Dafür musste man auf
 51-02 einen obskuren Befehl namens `\@ifl@t@r` zurückgreifen, der angesichts seines kryp-
 51-03 tischen Namens niemals dazu gedacht war, im Code von Paketen oder Klassen
 51-04 verwendet zu werden. Zudem waren auch die bestehenden Schnittstellen-Befehle
 51-05 nicht ganz korrekt, da sie auf »gleich oder neuer« und nicht auf »neuer« geprüft
 51-06 haben, wie man eigentlich bei ihrem Namen erwarten würde.
 51-07

51-08 Wir haben daher drei neue CamelCase-Befehle als offizielle Schnittstelle für solche
 51-09 Prüfungen eingeführt:
 51-10

51-11 `\IfFormatAtLeastTF{<date>}{<true code>}{<false code>}`
 51-12

51-13 und für Pakete- und Klassentests:
 51-14

51-15 `\IfClassAtLeastTF{<class name>}{<date>}`
 51-16 `{<true code>}{<false code>}`
 51-17

51-18 `\IfPackageAtLeastTF{<package name>}{<date>}`
 51-19 `{<true code>}{<false code>}`
 51-20

51-21 Aus Kompatibilitätsgründen bleiben die alten Befehle weiter verfügbar, aber wir
 51-22 schlagen vor, sie im Laufe der Zeit zu ersetzen und in neuem Code auf die neuen
 51-23 Schnittstellen zurückzugreifen. (Github Issue 186)
 51-24
 51-25

51-26 Problematische Leerzeichen nach `\verb` vermeiden
 51-27
 51-28

51-29 Hat ein Anwender oder eine Anwenderin unabsichtlich `\verb!~!_foo` statt
 51-30 `\verb!~!_foo` eingetippt, war das Ergebnis überraschenderweise »!~!foo« – ohne
 51-31 Warnung oder Fehlermeldung. Das `_` wurde hier aufgrund der ziemlich komplexen
 51-32 Verarbeitung durch `\verb` für ein möglichst unverändertes Ausgeben zu einem
 51-33 Argumentbegrenzer. Das wurde behoben und Leerzeichen direkt nach den Befehlen
 51-34 `\verb` oder `\verb*` werden nun wie sonst auch ignoriert. (Github Issue 327)
 51-35
 51-36
 51-37
 51-38

51-39 Eine Möglichkeit, robuste Befehle zu kopieren ...
 51-40
 51-41

51-42 Im letzten LaTeX 2_ε-Release wurden eine Reihe von Befehlen auf User-Level robust
 51-43 gemacht, womit die Notwendigkeit für eine Möglichkeit, Kopien dieser Befehle zu
 51-44 erstellen (häufig, um sie neu zu definieren), gestiegen ist, und der LaTeX 2_ε-Kernel
 51-45 hatte solch eine Möglichkeit nicht. Zuvor wurde diese Funktionalität teilweise durch
 51-46 das Paket `letltxmacro` von Heiko Oberdiek bereitgestellt, mit dem ein robuster
 51-47 Befehl `\foo` mit Hilfe von `\LetLtxMacro\bar\foo` nach `\bar` kopiert werden konnte.
 51-48
 51-49

51-50 Seit diesem Release stellt der LaTeX 2_ε-Kernel `\NewCommandCopy` (und die Varianten
 51-51 `\Renew...` und `\Declare...`) zur Verfügung, die sehr ähnlich wie `\LetLtxMacro`
 51-52 funktionieren. Für den Endanwender oder die Endanwenderin sollten beide gleich
 51-53
 51-54
 51-55
 51-56

agieren und man sollte sich nicht um die Definition des Befehls kümmern müssen: `\NewCommandCopy` kann die harte Arbeit erledigen.

`\NewCommandCopy` kennt die unterschiedlichen Definitionsarten aus dem L^AT_EX 2_ε-Kernel und auch aus anderen Paketen, wie zum Beispiel die Befehlsdeklarationen wie `\NewDocumentCommand` in `xparse` oder `\newrobustcmd` in `etoolbox`, und kann erweitert werden, um zusätzliche Pakete abzudecken. (Github Issue 239)

... und eine Möglichkeit, sie mit `\show` zu zeigen

Manchmal ist es notwendig, die Definition eines Befehls nachzuschlagen, und häufig weiß man nicht nur nicht, wo dieser Befehl definiert ist, sondern auch, ob er von einem Paket neu definiert wurde, daher hilft es auch nicht, sich den Quellcode anzuschauen. Üblicherweise nutzt man dann T_EXs `\show-Primitive`, um die Definition eines Befehls auszugeben – was funktioniert, wenn der Befehl nicht robust ist. Bei `\show\frac` erhält man

```
>> \frac=macro:
->>\protect \frac .
```

was nicht sehr hilfreich ist. Um den eigentlichen Befehl anzuzeigen, muss man bemerken, dass sich die eigentliche Definition von `\frac` im Makro `\frac_` befindet, und `\expandafter\show\csname frac\space\endcsname` ausführen.

Aber mit dem Coding zum Kopieren robuster Befehle ist es schon möglich, einen Befehl zu analysieren und zu erkennen (soweit es eine Makroerweiterungs-Sprache erlaubt), wie er definiert wurde. `\ShowCommand` weiß das und mit `\ShowCommand\frac` wird am Terminal folgendes ausgegeben:

```
>> \frac=robust macro:
->>\protect \frac .

>> \frac =\long macro:
#1#2->>{\begingroup #1\endgroup \over #2}.
```

(Github Issue 373)

`l3docstrip` nach `docstrip` mergen

Die Datei `l3docstrip.tex` bot eine kleine Erweiterung gegenüber der Originaldatei `docstrip.tex`, indem die `%<@=<module>`-Syntax von `expl3` unterstützt wurde. Das ist nun in `docstrip` eingebunden, so dass dies sowohl für klassische `.dtx`-Dateien wie auch für Dateien mit Code genutzt werden kann, die in der L₃-Programmiersprache `expl3` geschrieben sind. (Github Issue 337)

53-01 Vertikales Setzen mit doc unterstützen

53-02

53-03 Die Umgebung macrocode nutzt intern eine trivlist und richtet als Teil davon
53-04 die \@labels-Box so ein, dass sie einen horizontalen Abstand erzeugt, aber diese
53-05 Box wird nie verwendet. Dadurch entsteht aber in manchen Situationen ein Pro-
53-06 blem, wenn die Satzrichtung vertikal wie etwa im Japanischen ist. Das wurde nun
53-07 korrigiert, um auch solche Anwendungsfälle zu unterstützen. (Github Issue 344)
53-08
53-09
53-10

53-11

53-12 Den Zählernamen erfassen, der durch \refstepcounter hochgezählt wird
53-13

53-14 \refstepcounter speichert nun den Zählernamen in \@currentcounter. Damit kön-
53-15 nen Pakete sich wie zref und hyperref den Namen merken, ohne \refstepcounter
53-16 patchen zu müssen. (Github Issue 300)
53-17
53-18

53-19

53-20

53-21 Natives LuaTeX-Verhalten für \-

53-22

53-23 LaTeX verändert \-, um auch dann einen optionalen Trennstrich hinzuzufügen, wenn
53-24 \hyphenchar auf -1 gesetzt ist. Diese Änderung ist unter LuaTeX nicht erforderlich,
53-25 weil \- dort nicht durch \hyphenchar beeinflusst wird. Daher wurde dieses Ver-
53-26 halten für LuaTeX geändert, um sicherzustellen, dass LuaTeXs (sprachspezifische)
53-27 Trennzeichen korrekt von \- berücksichtigt werden.
53-28
53-29

53-30

53-31

53-32 \par-Befehle in \typeout zulassen
53-33

53-34

53-35 \typeout hatte Probleme, wenn es auf eine Leerzeile oder einen \par-Befehl in
53-36 seinem Argument traf. Aber manchmal wird dieser Befehl verwendet, um beliebige
53-37 Eingaben oder Code von Benutzern oder Benutzerinnen auszugeben (zum Beispiel
53-38 verpackt in \unexpanded), die explizite \par-Befehle enthalten können. Das wurde
53-39 nun erlaubt. (Github Issue 335)
53-40

53-41

53-42

53-43 Abstandsbefehle aus amsmath in den Kernel übernommen
53-44

53-45

53-46 LaTeX hat ursprünglich nur sehr wenige Abstandsbefehle für den Einsatz in Text-
53-47 und Mathematik-Umgebungen angeboten – manche davon, wie zum Beispiel \; ,
53-48 gab es nur im Mathematikmodus. Das amsmath-Paket hat dies normalisiert und alle
53-49 Befehle für Text und Formeln bereitgestellt. Dieser Code wurde nun in den Kernel
53-50 verschoben, so dass er immer zur Verfügung steht.
53-51
53-52

53-53

53-54

53-55

53-56

Befehlsname(n)	Mathe	Text
<code>\, \thinspace</code>	$x\,x$	$x\,x$
<code>!\ \negthinspace</code>	xx	xx
<code>\: \> \medspace</code>	$x\,x$	$x\,x$
<code>\negmedspace</code>	xx	xx
<code>\; \thickspace</code>	$x\,x$	$x\,x$
<code>\negthickspace</code>	xx	xx

(Github Issue 303)

In Lua_T_EX auf alle Glyphen zugreifen, ohne Fonts neu laden zu müssen

Die L^AT_EX-Definitionen für `\textquotesingle`, `\textasciigrave` und `\textquotedbl` für die TU-Kodierung in Lua_T_EX erfordern eine spezielle Verarbeitung, um den Shaper davon abzuhalten, diese Zeichen durch Anführungszeichen zu ersetzen. Das geschah durch das erneute Laden des aktuellen Fonts ohne das `tlig`-Feature, was aber eine Reihe von Nachteilen mit sich brachte: Der Code verhält sich anders als der entsprechende X_YL^AT_EX-Code und er ist nicht sehr effizient. Dieser Code wurde nun durch eine Implementierung ersetzt, die einen geschützten Glyphen-Knoten injiziert, welcher durch das Font Shaping nicht beeinflusst wird. (Github Issue 165)

`\contentsline` um ein viertes leeres Argument erweitert

Bisher schrieb der L^AT_EX-Befehl `\addcontentsline` einen `\contentsline`-Befehl mit drei Argumenten in die `.toc`-Datei und ähnliche Dateien. Das `hyperref`-Paket definierte dann `\addcontentsline` neu, um ein viertes Argument zu schreiben. Die Änderung vereinheitlicht nun die Anzahl der Argumente, indem wenn nötig eine zusätzliche leere Klammergruppe weggeschrieben wird. (Github Issue 370)

Lua_T_EX-Callback `new_graf` exklusive gemacht

Es wurde ein falscher Callback-Typ korrigiert, der dafür gesorgt hat, dass Rückgabewerte aus dem Callback `new_graf` ignoriert und die Absatzrückung unterdrückt wurden. In der neuen Version kann nun ein `new_graf`-Callback-Handler gleichzeitig aktiv sein, wodurch dieser Handler die vollständige Kontrolle über die Absatzrückung erhält. (Github Issue 188)

Änderungen an Paketen der Kategorie graphics

Eine Warnung erzeugen, wenn eine bestehende Farbdefinition verändert wird

Wird eine Farbe mit `\DefineNamedColor` zweimal definiert, wurde bisher aufgrund eines Tippfehlers in der Prüfung kein Infotext `Redefining color ... in named color model ...` in die Log-Datei geschrieben. Das ist nun korrigiert worden.

(Gnats Issue graphics/3635)

Den Viewport im Paket graphics angeben

Die Angabe einer `BoundingBox` hat keine echte Bedeutung, wenn man Nicht-EPS-Grafiken in pdfTeX und LuaTeX einbindet. Seit ein paar Jahren wurde der Schlüssel `bb` des `graphicx`-Pakets (mit einer Warnung) als `viewport`-Schlüssel interpretiert. Dieses Feature wurde nun auch der Zwei-Argumente-Form von `\includegraphics` hinzugefügt, welche im Paket `graphics` am häufigsten zum Einsatz kommt. `\includegraphics[1,2][3,4]{file}` wird in pdfTeX und LuaTeX nun genauso interpretiert wie der Befehl `\includegraphics[viewport=1 2 3 4]{file}` von `graphicx`.

`\endlinechar` normalisieren

Wird `\endlinechar` auf `-1` gesetzt, so dass Zeilenenden in bestimmten Kontexten ignoriert werden, wurde durch Code, der `BoundingBox`-Kommentare parst, ein Low-Level-Fehler in TeX erzeugt. Das Paket setzt nun lokal `\endlinechar` auf seinen Standardwert, während Grafikdateien gelesen werden.

(Github Issue 286)

Dateien mit mehreren Erweiterungen

Manchmal hat man eine Grafikdatei – zum Beispiel `file.svg` –, konvertiert diese in ein anderes Format, um sie in LaTeX aufzunehmen, und erhält dann eine Datei namens `file.svg.png`. Hat man in früheren Releases `\includegraphics{file.svg}` genutzt, erzeugte das einen Fehler und das Einbinden der Grafik hätte aufgrund der unbekannten Erweiterung `.svg` nicht funktioniert. Das Paket `graphics` prüft nun, ob die angegebene Erweiterung bekannt ist und wenn das nicht der Fall ist, versucht es, die bekannten Erweiterungen anzuhängen, bis es eine Grafikdatei mit einer gültigen Erweiterung findet. Nur wenn es auf diesem Weg keine Datei findet, erzeugt es eine Fehlermeldung.

(Github Issue 355)

Änderungen an Paketen der Kategorie tools

array: Dehnbaren Leerraum in w-Spalten unterstützen

Wenn dehnbare Glue, zum Beispiel `\dotfill`, in tabular-Spalten genutzt wird, die mit Hilfe des Pakets array erstellt werden, dehnt er sich wie in normalem Absatztext. Die eine Ausnahme waren w-Spalten (nicht aber W-Spalten), bei denen er in seine nominale Breite gezwungen wurde (welche bei `\hfill` oder `\dotfill` den Wert `0pt` besitzt). Das wurde korrigiert und nun verhalten sich w-Spalten diesbezüglich wie alle anderen Spaltentypen.

(Github Issue 270)

array: Mathematik-Modus für w- und W-Felder in array nutzen

Die w- und W-Spalten sind LR-Spalten, die l, c und r stark ähneln. Daher ist es ganz natürlich, zu erwarten, dass ihre Feldinhalte im Mathematik- statt im Textmodus gesetzt werden, wenn sie in einer array-Umgebung zum Einsatz kommen. Dies wurde nun angepasst. Beachten Sie, dass es sich dabei um eine Änderung in Version v2.5 handelt, die alte Dokumente verändern kann! Haben Sie w- oder W-Spalten in älteren Dokumenten verwendet, fügen Sie für solche Spalten entweder `>\$`...`<\$` hinzu oder entfernen Sie die `\$`-Zeichen in den Feldern. Alternativ können Sie zur alten Version zurückkehren, indem Sie array in solchen Dokumenten mit

```
\usepackage{array}[v2.4]
```

laden.

(Github Issue 297)

array: Fix für `\firstline` und `\lastline`

Ein Ersetzen von `\hline` durch `\firstline` oder `\lastline` konnte in manchen Fällen zu einer Vergrößerung der Tabellenbreite führen. Das wurde nun korrigiert.

(Github Issue 322)

varioref: Japanisch als Sprachoption unterstützen

Das Paket erkennt nun `japanese` als Sprachoption. Dies war besonders kompliziert, weil aus grammatikalischen Gründen `\vref`, `\Vref`, `\vrefrange` und `\fullref` eine Struktur benötigen, die sich von der aller anderer bisher unterstützten Sprachen unterscheidet. Um das zu erreichen, wurden `\vrefformat`, `\Vrefformat`, `\vrefrangeformat` und `\fullrefformat` auch allen anderen Sprachen hinzugefügt.

(Github Issue 352)

57-01 xr: Unterstützung für Leerzeichen in Dateinamen

57-02

57-03 Der Befehl `\externaldocument`, bereitgestellt durch `xr`, unterstützt nun auch Dateina-
57-04 men mit Leerzeichen – genauso wie `\include` und `\includeonly`. (*Github Issue 223*)
57-05

57-06

57-07 Änderungen an Paketen der Kategorie `amsmath`
57-08

57-09

57-10 Positionierungskorrekturen für zwei Akzentbefehle
57-11

57-12

57-13 Die Akzentbefehle `\dddot` und `\ddddot` (die drei oder vier Punkte als Akzente erzeugen) haben in bestimmten Situationen das Basiszeichen vertikal verschoben, wenn es
57-14 sich um eine einzelne Glyphen handelte – beispielsweise befanden sich $\$Q \dddot{Q}\$$
57-15 nicht auf der gleichen Grundlinie. Das wurde korrigiert. (*Github Issue 126*)
57-16
57-17

57-18

57-19

57-20 Korrekturen an `aligned` und `gathered`
57-21

57-22

57-23 Die Umgebungen `aligned` und `gathered` haben ein abschließendes optionales Ar-
57-24 gument, um die vertikale Position der Umgebung in Bezug zum Rest der Zeile
57-25 anzugeben. Erlaubte Werte sind `t`, `b` und `c`, aber der Code hatte nur auf `b` und `t`
57-26 geprüft und ist davon ausgegangen, dass alles andere `c` sein muss. Dadurch wurde
57-27 eine Formel, die mit einer Gruppe aus eckigen Klammern begann, ohne Warnung
57-28 falsch dargestellt – die Gruppe wurde verworfen und als Wunsch nach Zentrierung
57-29 interpretiert. Nach mehr als 25 Jahren wurde das nun korrigiert. Wird solch eine
57-30 Gruppe gefunden, wird eine Warnung ausgegeben und die Daten werden als Teil
57-31 der Formel verarbeitet. (*Github Issue 5*)
57-32
57-33

57-34

57-35

57-36

57-37 Unicode-Engines erkennen, wenn `\std@minus` und `\std@equal` gesetzt werden
57-38

57-39

57-40 Das `amsmath` Paket erkennt nun die Unicode-Engines und nutzt ihre erweiterten
57-41 Befehle zum Definieren von `\std@minus` und `\std@equal`. Damit wird vermieden,
57-42 dass ein Paket wie `unicode-math` den Code im `Begin-Document-Hook` patchen muss,
57-43 um die Befehle an die Engine anzupassen.
57-44

57-45

57-46

57-47 Wenn möglich `LuaTeX`-Primitive verwenden
57-48

57-49

57-50 Seit einigen Jahren hat `lualatex-math` die Befehle `\frac` und `\genfrac` sowie die
57-51 Umgebung `subarray` gepatcht, um sicherzustellen, dass neue `LuaTeX`-Primitive
57-52 genutzt werden. Dieser Code wurde nun nach `amsmath` migriert.
57-53

57-54

57-55

57-56

Änderungen am Paket babel

In den letzten Jahren hat sich mehrsprachiges Setzen deutlich weiterentwickelt und babel ist wie L^AT_EX selbst Unicode und dem W3C-Konsortium gefolgt, um eine saubere Ausgabe in vielen Sprachen zu erzeugen.

Zudem wurde das klassische Modell zum Definieren und Auswählen von Sprachen (auch als »vertikal« bezeichnet), das auf in sich abgeschlossenen Dateien basiert und das in einsprachigen Dokumenten immer noch bevorzugt ist, durch ein neues Modell erweitert (als »horizontal« bezeichnet), das auf von babel bereitgestellten *Services* aufbaut und das Definieren und Redefinieren sprachspezifischer Anpassungen mit Hilfe einfacher ini-Dateien ermöglicht, die auf Schlüssel/Wert-Paaren basieren. Das Paket babel stellt ungefähr 250 dieser Dateien bereit, die mit Hilfe des Unicode Common Language Data Repository erzeugt wurden.

Dank der kürzlich erfolgten Fortschritte in lualatex und luaotfload unterstützt babel nun *Services* für bidirektionalen Textsatz, Zeilenumbrüche für südostasiatische und CJK-Schriftsysteme, Nicht-Standard-Worttrennungen (wie ff nach ff-f), alphabetische und additive Zähler, die automatische Wahl von Fonts und Sprachen abhängig vom Schriftsystem und so weiter. Das bedeutet, dass babel zum Setzen sehr vieler verschiedener Sprachen genutzt werden kann, wie zum Beispiel für Russisch, Arabisch, Hindi, Thai, Japanisch, Bengalisch, Amharisch, Griechisch und weitere.

Und da sich diese ini-Dateien einfach parsen lassen, können sie auch als Quelle für andere Pakete dienen.

Weitere Details finden Sie in der Dokumentation des babel-Pakets [1].

Literatur

- [1] Javier Bezos, Johannes Braams: Babel—Localization and internationalization. <https://www.ctan.org/pkg/babel>.
- [2] L^AT_EX Project Team: »L^AT_EX news, issue 32, October 2020«, 41.3 (129 2020), 286–291, ISSN: 0896-3207, <https://tug.org/TUGboat/tb41-3/tb1291tnews32.pdf>.
- [3] — L^AT_EX-Dokumentation auf der Website des L^AT_EX Project, <https://latex-project.org/help/documentation/>.
- [4] — L^AT_EX Issue Tracker, <https://github.com/latex3/latex2e/issues/>.
- [5] — L^AT_EX News 31, <https://latex-project.org/news/latex2e-news/1tnews31.pdf>.

59-01
59-02
59-03
59-04
59-05
59-06
59-07
59-08
59-09
59-10
59-11
59-12
59-13
59-14
59-15
59-16
59-17
59-18
59-19
59-20
59-21
59-22
59-23
59-24
59-25
59-26
59-27
59-28
59-29
59-30
59-31
59-32
59-33
59-34
59-35
59-36
59-37
59-38
59-39
59-40
59-41
59-42
59-43
59-44
59-45
59-46
59-47
59-48
59-49
59-50
59-51
59-52
59-53
59-54
59-55
59-56

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.

to-be-determined von *Yegor Bugayenko* erlaubt es, Textteile auszuzeichnen, die der Überarbeitung bedürfen. Sie können farbig hinterlegt, durch einen Platzhalter ersetzt oder ganz ausgeblendet werden.

CTAN:macros/latex/contrib/to-be-determined

cartonaugh von *Jamal Bouajjaj* ist ein Fork des Pakets *karnaugh-map* von *Mattias Jacobsson* zum Zeichnen von Karnaugh-Veitch-Diagrammen mit *pgf/TikZ* und *Lua $\LaTeX$* .

CTAN:graphics/pgf/contrib/cartonaugh

macrolist von *Dennis Chen* stellt neue Listenumgebungen bereit.

CTAN:macros/latex/contrib/macrolist

pyluatex von *Tobias Enderle* erlaubt es, Python-Code in einem *$\LaTeX$* -Dokument zur Laufzeit auszuführen. Die Kompilierung muss mit *Lua $\TeX$* erfolgen.

CTAN:macros/luatex/latex/pyluatex

biblatex-spbasic von *Herbert Voß* ist eine Anpassung des *Bib $\TeX$* -Stils *spbasic.bst* für *Biblatex* und *Biber*. Damit kann man Bibliografien für Zeitschriften des Springer-Verlags setzen.

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

newcastle-bst von *Luke Briggs* stellt einen *Bib $\TeX$* -Stil bereit, um Fundstellen und Bibliografien im »Harvard-at-Newcastle-Stil« zu formatieren, der von der Universität Newcastle empfohlen wird.

CTAN:biblio/bibtex/contrib/newcastle-bst

tikz-swigs von *Thomas S. Richardson* dient zur Darstellung eines neuen Typs von Graphen, den der Autor gemeinsam mit J. M. Robins im Jahr 2013 vorgeschlagen hatte: *Single World Intervention Graphs (SWIGs)*.

CTAN:graphics/pgf/contrib/tikz-swigs

texsurgery von *Pablo Angulo* ist ein Paket, das das gleichnamige Python-Projekt <https://framagit.org/pang/texsurgery> ergänzt.

CTAN:macros/latex/contrib/texsurgery

<i>iexec</i> von <i>Yegor Bugayenko</i> ermöglicht es, Shell-Befehle aus dem Dokument heraus auszuführen und deren Ausgabe in das Dokument zu übernehmen.	60-01
CTAN:macros/latex/contrib/iexec	60-02
<i>thubeamer</i> von <i>Jingxuan Yang</i> ist ein Beamer-Theme für Präsentationen an der Peking University.	60-03
CTAN:macros/latex/contrib/beamer-contrib/themes/thubeamer	60-04
<i>basicalatex-ru</i> von <i>Alexey Kuznetsov</i> ist eine neue Einführung zu $\LaTeX$ in russischer Sprache.	60-05
CTAN:info/russian/basicalatex-ru	60-06
<i>inputnormalization</i> von <i>Marcel Krüger</i> normalisiert die Eingabedatei, bevor sie von $\XeTeX$ oder $\LuaTeX$ gelesen wird.	60-07
CTAN:macros/unicodetex/generic/inputnormalization	60-08
<i>strands</i> von <i>Diego Arcis</i> zeichnet Partitionen und weitere ähnliche Diagramme aus der Mengenlehre.	60-09
CTAN:graphics/pgf/contrib/strands	60-10
<i>ifallfalse</i> von <i>Dennis Chen</i> vergleicht einen String mit mehreren anderen Strings und führt ein Kommando aus, wenn keiner der anderen Strings dem ersten gleicht.	60-11
CTAN:macros/latex/contrib/ifallfalse	60-12
<i>texdimens</i> von <i>Jean-François Burnol</i> konvertiert $\TeX$ -Längen in Dezimalzahlen, um Rechenfehler zu vermeiden.	60-13
CTAN:macros/generic/texdimens	60-14
<i>spbmark</i> von <i>Qu Yi</i> dient zum Formatieren von hoch- und tiefgestelltem Text, sowohl im Absatz- als auch im Mathematikmodus.	60-15
CTAN:macros/latex/contrib/spbmark	60-16
<i>book-of-common-prayer</i> von <i>Arlie Coles</i> ist ein Paket, mit dem man Texte im Layout des <i>Book of Common Prayer</i> , der Gottesdienstordnung der Anglikanischen Kirche, setzen kann.	60-17
CTAN:macros/xetex/latex/book-of-common-prayer	60-18
<i>nndraw</i> von <i>Carlos Cardoso Dias</i> ist ein weiteres Paket zum Zeichnen künstlicher neuronaler Netze.	60-19
CTAN:graphics/pgf/contrib/nndraw	60-20
<i>optexcount</i> von <i>Richard Hartmann</i> ist ein Python-Skript, das Wörter in $\text{Op}\TeX$ -Dokumenten zählen kann.	60-21
CTAN:support/optexcount	60-22
<i>bilingualpages</i> von <i>Arlie Coles</i> setzt auf dem Paket <i>paracol</i> von <i>Hiroshi Nakashi-</i> <i>ma</i> auf und dient dazu, das Original und die Übersetzung eines Texts in zwei	60-23
	60-24
	60-25
	60-26
	60-27
	60-28
	60-29
	60-30
	60-31
	60-32
	60-33
	60-34
	60-35
	60-36
	60-37
	60-38
	60-39
	60-40
	60-41
	60-42
	60-43
	60-44
	60-45
	60-46
	60-47
	60-48
	60-49
	60-50
	60-51
	60-52
	60-53
	60-54
	60-55
	60-56

- 61-01 Kolumnen nebeneinander zu setzen.
 61-02 CTAN:macros/latex/contrib/bilingualpages
 61-03 *docassembly* von *Donald P. Story* stellt eine gleichnamige Umgebung bereit, in der
 61-04 JavaScript-Code *einmalig* in einer PDF-Datei ausgeführt werden kann.
 61-05 CTAN:macros/latex/contrib/docassembly
 61-06 *acrotex-js* von *Donald P. Story* enthält zwei JavaScript-Dateien für *acrotex*, die
 61-07 früher zusammen mit den Paketen *insdljs*, *aeb_pro* und *thorshammer* sowie mit
 61-08 dem Paket verteilt wurden. Sie müssen nun separat installiert werden. Auch das
 61-09 neue Paket *docassembly* benötigt *acrotex-js*.
 61-10 CTAN:macros/latex/contrib/acrotex-js
 61-11 *apalike-ejor* von *Adam Rumpf* ist ein BibTeX-Stil nach den Vorgaben des *European*
 61-12 *Journal of Operational Research*.
 61-13 CTAN:biblio/bibtex/contrib/apalike-ejor
 61-14 *arabic-book* von *Mohammed Obaid Alziyadi* ist eine Buchklasse, die die Sprachen
 61-15 Arabisch und Englisch mit der Engine XeTeX unterstützt.
 61-16 CTAN:macros/xetex/latex/arabic-book
 61-17 *href-ul* von *Yegor Bugayenko* setzt auf den Paketen *hyperref* und *ulem* auf und
 61-18 ermöglicht es, Hyperlinks zu unterstreichen.
 61-19 CTAN:macros/latex/contrib/href-ul
 61-20 *ffcode* von *Yegor Bugayenko* verwendet die Pakete *minted* und *tcolorbox* zum
 61-21 Setzen von Quelltexten in einer wissenschaftlichen Arbeit.
 61-22 CTAN:macros/latex/contrib/ffcode
 61-23 *minim* von *Esker Renkema* bietet ein modernes Plain-Format, das viele LuaTeX- und
 61-24 PDF-Erweiterungen unterstützt. *minim* ist modular aufgebaut, deshalb gibt es
 61-25 *minim-mp* und *minim-hatching* für die Arbeit mit METAPOST, *minim-math* für den
 61-26 Unicode-Mathmetiksat, *minim-pdf* für Tagged PDFs und *minim-xmp* für XMP-
 61-27 Metadaten in PDFs.
 61-28 CTAN:macros/luatex/generic/minim
 61-29 CTAN:macros/luatex/generic/minim-mp
 61-30 CTAN:graphics/minim-hatching
 61-31 CTAN:macros/luatex/generic/minim-math
 61-32 CTAN:macros/luatex/generic/minim-pdf
 61-33 CTAN:macros/luatex/generic/minim-xmp
 61-34 *wichura-table* von *Michael Wichura* enthält die TaBLE-Makros in Plain TeX, die
 61-35 nach 30 Jahren erstmals auf CTAN veröffentlicht werden.
 61-36 CTAN:macros/generic/wichura-table
 61-37 *bangla* von *Nahid Hossain* ist eine umfassende Unterstützung für die Bengalische
 61-38 Sprache, die in Bangladesch und in Indien gesprochen wird.
 61-39 CTAN:language/bengali/bangla
 61-40
 61-41
 61-42
 61-43
 61-44
 61-45
 61-46
 61-47
 61-48
 61-49
 61-50
 61-51
 61-52
 61-53
 61-54
 61-55
 61-56

<i>kdpcover</i> von <i>Yegor Bugayenko</i> erzeugt passende Umschläge für Bücher, die bei Kindle Direct Publishing verlegt werden.	62-01
CTAN:macros/latex/contrib/kdpbook	62-02
<i>biblatex-cv</i> von <i>Daniel E. Shub</i> ist ein Paket, das mithilfe von biblatex und Biber einen tabellarischen Lebenslauf aus einer BibTeX-Datei erstellt.	62-03
CTAN:macros/latex/contrib/biblatex-contrib/biblatex-cv	62-04
<i>bjfuthesis</i> von <i>Changxin Liu</i> ist eine Klasse für Abschlussarbeiten an der Beijing Forestry University.	62-05
CTAN:macros/latex/contrib/bjfuthesis	62-06
<i>bmstu-iu8</i> von <i>CatInCosmicSpace</i> ist eine Klasse für Arbeiten an der Staatlichen Technischen Universität Moskau.	62-07
CTAN:macros/latex/contrib/bmstu-iu8	62-08
<i>pgfmth-xfp</i> von <i>Jonathan P. Spratte</i> ist nach Auffassung des Autors nur eine behelfsweise Lösung, um Fließkommaberechnungen mit pgf und xfp durchführen zu können.	62-09
CTAN:macros/latex/contrib/pgfmth-xfp	62-10
<i>tiscreen</i> von <i>Mustafa Ibrahim</i> ahmt das Display älterer Taschenrechner von Texas Instruments nach, insbesondere den TI-82 STATS.	62-11
CTAN:macros/latex/contrib/tiscreen	62-12
<i>pdfextra</i> von <i>Michal Vlasák</i> erweitert optex um Features für den Umgang mit PDF-Dateien (Hyperlinks, Multimedia, Anhänge etc.).	62-13
CTAN:macros/luatex/optex/pdfextra	62-14
<i>tabularray</i> von <i>Jianrui Lyu</i> stellt Makros bereit, um Tabellen und Matrizen mit L ^A T _E X ₃ zu setzen. Die Syntax ist kompatibel mit L ^A T _E X _{2_ε} .	62-15
CTAN:macros/latex/contrib/tabularray	62-16
<i>projlib</i> von <i>Jinwen Xu</i> ist ein Bundle, das eine eklektische Sammlung von Erweiterungen bereithält, die das Schreiben von L ^A T _E X-Dokumenten (auch im Entwurfsstadium und für mehrsprachige Dokumente) erleichtern soll.	62-17
CTAN:macros/latex/contrib/projlib	62-18
<i>codehigh</i> von <i>Jianrui Lyu</i> ist ein weiteres Paket zum Setzen von Quelltexten, das auf l3regex aufsetzt. Mit LuaTeX kommt die Bibliothek LPeg zum Einsatz.	62-19
CTAN:macros/latex/contrib/codehigh	62-20
<i>coffeestains</i> von <i>Hanno Rein</i> und <i>Patrick Bideault</i> sorgt für die gewohnten Kaffeeflecken auf den Ausdrucken sorgsam gesetzter Texte. Dank pgf/TikZ stehen vier Varianten in verschiedenen Ausführungen bereit.	62-21
CTAN:graphics/pgf/contrib/coffeestains	62-22
	62-23
	62-24
	62-25
	62-26
	62-27
	62-28
	62-29
	62-30
	62-31
	62-32
	62-33
	62-34
	62-35
	62-36
	62-37
	62-38
	62-39
	62-40
	62-41
	62-42
	62-43
	62-44
	62-45
	62-46
	62-47
	62-48
	62-49
	62-50
	62-51
	62-52
	62-53
	62-54
	62-55
	62-56

63-01
63-02
63-03
63-04
63-05
63-06
63-07
63-08
63-09
63-10
63-11
63-12
63-13
63-14
63-15
63-16
63-17
63-18
63-19
63-20
63-21
63-22
63-23
63-24
63-25
63-26
63-27
63-28
63-29
63-30
63-31
63-32
63-33
63-34
63-35
63-36
63-37
63-38
63-39
63-40
63-41
63-42
63-43
63-44
63-45
63-46
63-47
63-48
63-49
63-50
63-51
63-52
63-53
63-54
63-55
63-56

Spielplan

2021

5. 8. – 8. 8. 2021 **TUG 2021 – Presentations covering the T_EX world**
The 42nd Annual Conference of the T_EX Users Group
Die Tagung findet dieses Jahr coronabedingt online statt.
<https://tug.org/tug2021>



18. 9. 2021 **Herbsttagung**
und 63. Mitgliederversammlung von DANTE e.V.
Landesinstitut für Pädagogik und Medien
Beethovenstraße 26
66125 Saarbrücken
<https://www.dante.de/veranstaltungen/herbst2021/>



20. 9. – 25. 9. 2021 **15th International ConT_EXt Meeting**
Bassenge, Château Boirs (Belgien)
<https://meeting.contextgarden.net/>



2022

12. 9. – 18. 9. 2022 **ConT_EXt Meeting 2022**
Evang. Freizeitstätte in Dreifelden/Westerwald
<https://www.rittersmorgen.de>



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
Restaurant Galilei, Marienbongard 24, 52062 Aachen
Erster Donnerstag im Monat, 20.00 Uhr
pausiert z. Zt. wegen Corona

Bad Doberan

Carsten Vogel,
texnicer@web.de
pausiert z. Zt. wegen Corona

Berlin

Michael-E. Voges, Tel.: (03362) 50 18 35,
mevoges@t-online.de
Mantee – Café Restaurant, Chausseestraße 131, 10115 Berlin
Zweiter Donnerstag im Monat, 19.00 Uhr
z. Zt. online wegen Corona. Link siehe Mailingliste dante-ev oder auf Anfrage

Darmstadt

Karlheinz Geyer,
geyerk@posteo.de
pausiert z. Zt. wegen Corona, weitere Treffen wieder ab 2022 geplant

Erlangen

Walter Schmidt, Peter Seitz,
w.a.schmidt@gmx.net
Gaststätte »Deutsches Haus«, Luitpoldstraße 25, 91052 Erlangen
Dritter Dienstag im Monat, 19.00 Uhr
z. Zt. online wegen Corona. Link auf Nachfrage bei obiger Email-Adresse

Frankfurt a. Main

Harald Vajkonny,
vajkonny@t-online.de
<https://wiki.lug-frankfurt.de/TeXStammtisch>
zur Zeit inaktiv, Interessenten bitte per Mail melden



Göttingen

Holger Nobach,
holger.nobach@nambis.de, <http://goetex.nambis.de/>
Restaurant Mazzoni Cucina Italiana,



64-01
64-02
64-03
64-04
64-05
64-06
64-07
64-08
64-09
64-10
64-11
64-12
64-13
64-14
64-15
64-16
64-17
64-18
64-19
64-20
64-21
64-22
64-23
64-24
64-25
64-26
64-27
64-28
64-29
64-30
64-31
64-32
64-33
64-34
64-35
64-36
64-37
64-38
64-39
64-40
64-41
64-42
64-43
64-44
64-45
64-46
64-47
64-48
64-49
64-50
64-51
64-52
64-53
64-54
64-55
64-56

65-01 Hermann-Rein-Straße 2, 37075 Göttingen

65-02 Dritter Donnerstag im Monat, 18.00 Uhr

65-03 pausiert z. Zt. wegen Corona

65-04

65-05 **Hamburg**

65-06 Günther Zander,

65-07 guenther.zander@lug-balista.de

65-08 Pausiert z. Zt. wegen Corona

65-09

65-10 **Hannover**

65-11

65-12 Reiko Kaps,

65-13 kaps@luis.uni-hannover.de

65-14 <https://tex-hannover.de> (im Entstehen)

65-15 Jeden zweiten Donnerstag im Monat. Zur Zeit online (Link über Mailingliste)

65-16

65-17 **Heidelberg**

65-18

65-19 Martin Wilhelm Leidig, Tel.: 0170 418 33 29,

65-20 moss@moss.in-berlin.de

65-21 Anmeldeseite zur Mailingliste: <https://tinyurl.com/stammtisch-HD>

65-22 pausiert z. Zt. wegen Corona

65-23

65-24 **Köln**

65-25

65-26 Uwe Ziegenhagen

65-27 uwe@dante.de

65-28 zur Zeit inaktiv, Interessenten bitte per Mail melden

65-29

65-30 **Leipzig**

65-31

65-32 Erhard Pross

65-33 Erhard.Pross@gmx.de

65-34 ab Herbst 2021 geplant. Interessenten bitte per Mail melden

65-35

65-36 **München**

65-37

65-38 Uwe Siart,

65-39 uwe.siart@tum.de, <http://www.siart.de/typografie/stammtisch.xhtml>

65-40 pausiert z. Zt. wegen Corona

65-41

65-42 **Stralsund**

65-43

65-44 Heiner Richter,

65-45 Heiner.Richter@hochschule-stralsund.de

65-46 z. Zt. inaktiv, Nachfolge in Planung

65-47

65-48 **Stuttgart**

65-49

65-50 Bernd Raichle,

65-51 bernd.raichle@gmx.de

65-52 z. Zt. online. Link auf Anfrage

65-53

65-54

65-55

65-56



Adressen

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

Tel.: (0 62 21) 2 97 66 (Mo., Mi., Do. von 9.00–11.30 Uhr)
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:	Dr. Uwe Ziegenhagen	vice-president@dante.de
Schatzmeisterin:	Dr. Doris Behrendt	treasurer@dante.de
Schriftführer:	Volker RW Schaa	secretary@dante.de
Beisitzer:	Dr. Klaus Höppner	
	Harald König	
	Stephan Lukasczyk	
	Herbert Voß	

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

67-01
67-02
67-03
67-04
67-05
67-06
67-07
67-08
67-09
67-10
67-11
67-12
67-13
67-14
67-15
67-16
67-17
67-18
67-19
67-20
67-21
67-22
67-23
67-24
67-25
67-26
67-27
67-28
67-29
67-30
67-31
67-32
67-33
67-34
67-35
67-36
67-37
67-38
67-39
67-40
67-41
67-42
67-43
67-44
67-45
67-46
67-47
67-48
67-49
67-50
67-51
67-52
67-53
67-54
67-55
67-56

Autoren/Organisatoren

Adelheid Bonnetsmüller
bonnetsmueller@icloud.com

Luzia Dietsche
71394 Kernen
dtkred@dante.de

Jürgen Fenn
Neu-Isenburg
juergen.fenn@gmx.de

Rainer-M. Fritsch
Speerweg 67
13465 Berlin
mail@rmf.berlin

Harald Lichtenstein
Gerhart-Hauptmann-Str. 29/1
70734 Fellbach
harald@linese.de

[11] **Thomas Hilarius Meyer** [5]
thomas.hilarius.meyer@gmail.com

[3,7] **Frank Mittelbach, L^AT_EX Project** [44]
frank.mittelbach@latex-project.org

[59] **Frank Mittelbach** [30]
frank.mittelbach@latex-project.org

Chris Rowley [30]
chris.rowley@latex-project.org

[23] **Martin Sievers** [4,5]
siehe Seite 66

Die T_EXnische Komödie

33. Jahrgang Heft 3/2021 August 2021

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 5 Einladung zur Herbsttagung 2021
- 7 Vorstellung des Vorstands

Bretter, die die Welt bedeuten

- 11 Having Fun with $\text{\TeX}$: Jetzt werden Boxen bunt!
- 18 Mit *arara* externe $\text{\TeX}$ -Dokumente erzeugen und einfügen
- 23 Wochenkalender

Von fremden Bühnen

- 30 $\text{\TeX}$ Tagged PDF – Ein Entwurf für ein großes Projekt
- 44 $\text{\TeX}$ News – Issue 32, Oktober 2020
- 59 Neue Pakete auf CTAN

Spielplan

- 63 Termine
- 64 Stammtische

Adressen

- 67 Autoren/Organisatoren