

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

19. Jahrgang Heft 3/2007 August 2007

3/2007

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 Schreibenden wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nicht-kommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden.

Beiträge sollten in Standard-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und per E-Mail oder Datenträger (CD) an untenstehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, L^AT_EX-Pakete oder Schriften dafür nötig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden.

Diese Ausgabe wurde mit pdfT_EX 3.141592-1.40.4-2.2 (Web2C 7.5.6) erstellt. Als Standard-Schriften kamen die Type-1-Fonts Latin-Modern und LuxiMono zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2700

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

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

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

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit : Rudolf Herrmann Klaus Höppner Gert Ingold
Rolf Niepraschk Volker RW Schaa Martin Schröder

Redaktionsschluss für Heft 3/2007: 15. Juli 2007

ISSN 1434-5897

Die T_EXnische Komödie 3/2007

Editorial

Liebe Leserinnen und Leser,

mit diesem Heft geht es wieder einmal in die tieferen Gefilde von L^AT_EX: Werner Lemberg hat eine deutsche Übersetzung der Beschreibung zum sogenannten »encoding« erstellt. Sie nimmt von ihrem Umfang her fast das gesamte Heft ein. Dies geschah mit Absicht, denn eine Aufteilung auf zwei Komödien hätte zwar Platz für anderes gelassen, wäre aber dem Zweck eines »Nachschlagewerkes« nicht förderlich gewesen.

Der diesjährige Linuxtag fand erstmals in Berlin statt und scheint zumindest die Veranstalter so überzeugt zu haben, dass sie den nächsten wieder hier stattfinden lassen wollen. Wenigstens ich hatte dadurch eine erheblich verkürzte Anreise, wohingegen der Kern der Linuxtag-Mannschaft eine weite Anreise auf sich nehmen musste. Sie finden in diesem Heft einige Impressionen davon, was DANTE e. V. auf nur 8 m² Grundfläche alles bewegt hat ...

Ich wünsche Ihnen viel Spaß mit dieser Komödie, erinnere noch einmal an die Herbsttagung in Ulm und verbleibe wie gewohnt

mit T_EXnischen Grüßen,

Ihr Herbert Voß

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

diese Ausgabe von »Die T_EXnische Komödie« erhalten Sie kurz vor der diesjährigen Herbsttagung von DANTE e.V. an der Universität Ulm. Entsprechend des häufig geäußerten Wunsches nach einem Wochenendtermin finden dieses Mal Mitgliederversammlung und Tutorien an einem Samstag statt. Nähere Informationen finden Sie unter <http://www.dante.de/dante/events/mv37>.

Für die weitere Lektüre dieser Ausgabe möchten wir – neben dem bereits im Editorial erwähnten Übersichtsartikel zum Thema Fontencoding – auf den umfangreichen Bericht von der EuroBachOT_EX 2007 in Polen hinweisen, der vielleicht Lust auf den Besuch einer zukünftigen BachOT_EX- oder der TUG 2008-Konferenz in Cork, Irland macht. Frisch von der TUG 2007 in San Diego stammt die Ankündigung der Beta-Version von LuaT_EX.

Zum Abschluss möchten wir uns bei den Helfern bedanken, die wiederum einen Linuxtag-Stand von DANTE e.V. möglich gemacht haben. Nachdem die Veranstalter des Linuxtages eine langfristige Vereinbarung mit der Berliner Messe abgeschlossen haben, wird uns ein erneuter Umzug (der natürlich immer eine Herausforderung an die »Standlogistik« ist) hoffentlich möglichst lange erspart bleiben – auch wenn die Inspiration, die der Veranstalter am neuen Messeort gefühlt haben will, uns wie anderen Ausstellern eher unbemerkt blieb.

Mit freundlichem Gruß,

Klaus Höppner	Volker RW Schaa
Vorsitzender	Stellvertretender Vorsitzender

Von fremden Bühnen

Bericht über die EuroBachTeX 2007-Tagung¹

Michael Guravage, Hans Hagen, Taco Hoekwater,
Volker RW Schaa

Einführung

Das 17. Europäische Treffen der T_EX-User fand in diesem Jahr zeit- und ortsgleich zum jährlichen Treffen der polnischen »GUST« am Bachotek-See statt, so dass der Name zu EuroBachTeX 2007 verschmolzen wurde. Mehr als 50 Vorträge und zwei Workshops wurden in der Zeit vom 28. April–2. Mai 2007 gehalten, über 100 Teilnehmer aus der ganzen Welt wurden gezählt.

Samstag, 28. April 2007



Jerzy Ludwiczowski, Präsident der GUST und Vorsitzender des Organisationskomitees, eröffnete die Konferenz, die gemeinsam von der polnischen GUST und der tschechisch und slowakischen CSTUG ausgerichtet wurde. Er begrüßte die Teilnehmer und ermunterte sie, die Konferenz unter dem Motto »Paths to the Future« als Gelegenheit zu begreifen,

Diskussionen und Gespräche mit Vortragenden und Entwicklern zu führen, die richtungsweisend für zukünftige Entwicklungen sein können.

¹Mit Erlaubnis von Michael Guravage, Hans Hagen und Taco Hoekwater habe ich ihren Text, der in der MAPS#35 veröffentlicht wurde, ins Deutsche übertragen und um persönliche Anmerkungen ergänzt.

Das Vortragsprogramm eröffnete Jonathan Kew mit einem Überblick über die Historie und den aktuellen Status von XeTeX: Nach der ersten Version im Frühjahr 2004 für Mac OSX folgte in 2005 eine Version, die OpenType unterstützt. XeTeX für Linux wurde während der BachTeX-Tagung 2006 veröffentlicht, eine MS-Windows-Version folgte dann kurz darauf im Juni. Das Jahr 2007 markierte einen Meilenstein für das XeTeX-Projekt, da es nun als integraler Bestandteil in die TeXLive-Distribution aufgenommen wurde.

Die aktuelle XeTeX-Implementierung unterstützt jetzt Unicode, die Integration von existierenden Makropaketen wurde verbessert und es wurde eine elegante Einbindung von Trennmustern erreicht. Ein Ausblick in die Zukunft: vollständige Unterstützung betriebssystem-spezifischer Fonts (OpenType, TrueType und PostScript) ohne TeX-spezifische Installation. Ein weiteres geplantes Feature sind »inter-character tokens«, also Token, die in Abhängigkeit vom Zeichen (character class) zwischen benachbarten Zeichen eingefügt werden können. Schließlich soll es eine bessere Unterstützung für nicht lateinische Schriften für Minderheitensprachen geben und hier insbesondere für die, die (noch) nicht in Unicode aufgenommen sind. Die Unterstützung für Minderheitenschriften und -sprachen ist eine der Hauptprojekte der SIL-Organisation, dem Arbeitgeber von Jonathan Kew. XeTeX wird deshalb das SILs Graphite-Fontsystem unterstützen (mehr über SIL unter <http://www.silinternational.com/sil/>).

Taco Hoekwater begann seine Präsentation mit der Ankündigung, dass die METAPOST-Version 1.0 nun verfügbar sei. Neu sind zum Beispiel die Farbtypen `cmymk`, `grey-scale` und `marking-only`, Template-Benutzung für Dateinamen und ein verbessertes Benutzerhandbuch.

Ziel des nächsten Release 1.1 ist es, verschiedene Beschränkungen aufzuheben. So wird es dynamische Speicherverwaltung für Arrays geben und die Rechengenauigkeit wird erhöht. Um Kompatibilitätsprobleme mit existierenden Eingabedateien zu vermeiden, wird die Version 1.1 als separates Modul ausgeliefert. Abschließend äußerte Taco die Absicht, im nächsten Jahr die METAPOST-Funktionalität als Bibliothek zur Verfügung zu stellen.

Der Titel der Präsentation von Hans Hagen lautete »Beware of too much tokenspeak«: TeX konvertiert Zeichen zu »Token«, die in der Folge in »Nodes« umgewandelt werden. Hans gab einen kurzen Einblick, wie LuaTeX auf der Ebene der Node-Listen bisher komplexe Teile von TeX vereinfacht und rationalisiert. Dies ging mittlerweile schon so weit, dass er erhebliche Teile des existierenden ConTeXt-Codes ablösen konnte. Gemäß des Konferenz-Themas

beschrieb Hans, wie mit LuaTeX eine einmalige Möglichkeit besteht, einen weiten Schritt in Richtung Zukunft zu tun.



Joanna Ludmiła Ryćko stellte in ihrem Beitrag die T_EX-Klinik vor. Die Klinik öffnete letztes Jahr zur BachTeX-Tagung und behandelte jeden, der mit einem akuten T_EX-Problem Hilfe suchte. Sie stellte die T_EX-Klinik-Ärzte mit ihren Spezialgebieten vor und empfahl den anwesenden

»Leidtragenden« sie aufzusuchen.

Johannes Große präsentierte **MathPSfrag**, ein Werkzeug mit dem es möglich ist, in Encapsulated-PostScript-Grafiken Zeichen oder Zeichenketten durch L^AT_EX generierte auszutauschen. **MathPSfrag** erweitert das existierende **PSfrag**, und erlaubt dabei sowohl automatische als auch fein abgestimmte manuelle Kontrolle über den Inhalt und die Platzierung der Beschriftung.

Siep Kroonenberg präsentierte ihre **epspdf/epspdfTk**-Konversions-Tools. Sie sind in Ruby bzw. Ruby/Tk geschrieben und benutzen **Ghostscript** und **pdftops**. **epspdf** bietet sowohl einen Kommandozeilenmodus wie auch ein grafisches User-Interface (GUI) für die Konversion von PostScript nach PDF und zurück. Besondere Vorzüge der Tools sind die Unabhängigkeit vom Betriebssystem (Windows, Unix/Linux und Mac OSX werden unterstützt), Bounding-Box-Berechnung (Beschneidung von Rändern) für Grafiken und Dokumente und eine Einzelseitenauswahl bei der Konversion von Dokumenten.

Zofia Walczak demonstrierte einige grundlegende und fortgeschrittene Eigenschaften des Paketes »Portable Graphics Format« (**pgfname**). Es wurde von Till Tantau, dem Autor des bekannten **beamer**-Paketes, entwickelt. **pgfname** wird in drei Layer aufgeteilt: System, Basis und Front-End. **TikZ** ist das Front-End von **pgfname**. Es erlaubt den Zugang zu allen Features von **pgfname** und bietet eine einfach zu benutzende Schnittstelle. Beim näheren Hinsehen erkennt man, dass es Teile seiner Syntax von **METAFONT** und **pstricks** geerbt hat.

Norbert Preining berichtete in Vertretung von Jim Hefferon über eine neue »experimentelle« Schnittstelle für den Software-Upload auf CTAN, die das Hochladen von Paketen verständlicher und einfacher für den Nutzer machen soll. Der Arbeitsablauf umfasst die Schritte »Upload«, »Freigabe« und »Installation« und führt am Ende zu einem TDS-konformen Bündel. Hilfsmittel für die Aktualisierung von Metadaten sind ebenfalls vorhanden.

Jean-Michel Huffle gab eine Einführung in XSL-FO (Extensible Stylesheet Language – Formatting Objects) für L^AT_EX-Benutzer. Er verglich korrespondierende L^AT_EX- und XSL-FO-Strukturen und erläuterte die unterschiedlichen Vorgehensweisen dieser beiden Textformatierer.

Grzegorz Murzynowski gab eine Einführung in das von ihm entwickelte **gmdoc**-Paket, das für die Dokumentation von L^AT_EX-Klassen und -Paketen geeignet ist. Es unterscheidet sich von seinem Vorgänger **doc** durch die automatische Verknüpfung der Indexeinträge, des Inhaltsverzeichnisses und der Referenzen durch Hyperlinks und es erfordert ein minimales Markup ohne spezielle Umgebungen ausschließlich mit Hilfe von Kommentaren (%).

Trzy samoloty wywiozł
 [gręzacz tęci wzdłużny,
 Apelajda sęklina borowajkę
 |kuci...

In einem weiteren Vortrag beschrieb Grzegorz Murzynowski seine Pakete **gmverse** und **gmcontinuo**. Das Paket **gmverse** bietet für das Setzen von langen umbrochenen Verszeilen eine

Ausrichtung wahlweise mit einer eckigen Klammer (l) zur Abtrennung der umbrochenen Zeile oder einer optischen Zentrierung. Das Paket **gmcontinuo** erlaubt das Setzen von Paragraphen *in continuo*, ohne Zeilenumbruch oder Einrückung, also Fließtext, bei dem die Absatzenden durch ein ¶-Zeichen markiert sind.

Im letzten Vortrag am Samstag trug Marek Ryćko Argumente für eine Überarbeitung der vielfältigen Bausteine des T_EX-Systems vor. Seine Hoffnung ist es, durch eine bessere Komponentenarchitektur die gesamte T_EX-Funktionalität zu erweitern und zu verbessern. Wenn stärker berücksichtigt würde, dass es sich nicht um Stand-Alone-Komponenten handelt, sondern um Bausteine eines Gesamtsystems, könnte die dadurch mögliche Integration die (Weiter-)Entwicklung von T_EX maßgeblich beeinflussen.

Das Wetter am Samstagabend war klar und kühl. So genossen wir unter Sternen und einem zunehmenden Mond am späteren Abend das traditionelle Lagerfeuer mit Stockwurstbraten, bereichert durch Getränke, Musik und Gesänge und, nicht zu vergessen, dem Feuerspeien der pyroT_EXniker.



Abbildung 1: Konferenzfoto der EuroBachTeX2007

Sonntag, 29. April 2007

In diesem Jahr fanden wir die Hütten nicht in ihrem üblichen bewohnbaren Zustand vor. Eine Renovierungswelle hatte uns einen neu gestalteten Speisesaal ohne Heizung beschert, bei den Hütten war die Neugestaltungswelle aber noch nicht angekommen oder unterbrochen worden. So fehlten neben den Heizungskörpern beispielsweise alle Vorhänge und auch das Toilettenpapier war knapp.



Wir brauchten eine Weile, um herauszufinden, dass die Textilien und das Papier wohl für den Workshop »Papier selbstgemacht: zwischen Handwerk, Kunst und Fun« gebraucht wurden, der von Grażyna Jackowska parallel zum Tagungsprogramm abgehalten wurde. Im Laufe der Konferenz war man dann auch angehalten, vorsichtiger durch den

Wald zu den Hütten zu laufen. Es hingen nämlich überall selbstgemachte Papierbögen zum Trocknen auf Gardinenleinen zwischen den Bäumen.

Andrzej Tomaszewski begann den zweiten Tag der Tagung mit einem Vortrag, der den Titel trug »Buchdesign: mit Vergnügen und ... Furcht«. Er berichtete über die obskuren Randbedingungen und unterschiedlichsten Beschränkungen, denen er bei der Produktion des Buches »The Master of Life Arteries of the Greater Warsaw«, einem Buch zum Jubiläum der Warschauer Wasserbehörde, unterworfen war. In Anbetracht der vielen Hürden, die er bei diesem Projekt nehmen musste, ist das Ergebnis wirklich sehenswert.

Dorota Cendrowska präsentierte in ihrem Vortrag die unterschiedlichen, meist außer Acht gelassenen oder sogar missachteten, Designkriterien für die Verwendung von Aufzählungen im Druck oder in Multimedia-Präsentationen. TeX und L^ATeX gelten auf Grund ihrer Satzeigenschaften als Referenz. Dennoch wurden in diesem Beitrag Methoden zur Diskussion gestellt, wann und wie das Erscheinungsbild von Aufzählungen verändert werden kann und/oder muss, um dem **Inhalt** die nötige Aufmerksamkeit zu geben.

Jerzy Ludwichowski beschrieb in seinem Vortrag die Arbeit, die Karl Berry und er mit der Überarbeitung der »GUST SOURCE«- und »NONSOURCE«-Fontlizenzen und die Überführung in eine einzige »GUST Font License« (GFL) hatten. Das Resultat ist eine Lizenz, die rechtlich identisch mit der »L^ATeX Project Public License« (LPPL) ist, die bereits von der FSF und Debian als rechtlich gültige freie Software-Lizenz anerkannt ist.

Jean-Michel Hufflen erläuterte in seinem Vortrag »MLBibTeX: reporting the experience«, wie sich MLBibTeX von den Anfängen im Jahr 2000 von einer C-Implementation zu einer im Lisp-Dialekt Scheme entwickelte. Jean-Michel plant das erste öffentliche Release der Neuentwicklung im Mai diesen Jahres.

In seinem zweiten Vortrag an diesem Vormittag zeigte Jean-Michel Hufflen, wie MLBibTeX mit der Schwierigkeit, dass lexikographische Sortierungen sprachabhängig sind, umgeht. Bibliographie-Stile können sortiert oder unsortiert sein, die Sortierfunktion, die in der bst-Sprache Verwendung findet, eignet sich aber ausschließlich für Englisch. In MLBibTeX wird nbst benutzt. Zusammen mit Scheme ist man dann in der Lage, europäische Sprachen korrekt lexikographisch zu sortieren.

David Kastrup erklärte in seinem Vortrag »Writing L^ATeX documents with AU^CTeX in Emacs« wie man das Emacs-AU^CTeX-Package vom Server herunterlädt, installiert und benutzt. Die neueste Version von AU^CTeX ist bei <http://www.gnu.org/software/auctex> zu finden. Hervorzuheben sind der erweiterte Unicode-Support, die Unterstützung weiterer »Version-Control«-

Systeme und die neuen Portierungen für Windows, Mac OSX and GTK+ unter Benutzung plattformspezifischer Graphik-Bibliotheken. Für Wagemutige sei der Quellcode für ein Pre-Release der Emacs 22-Version unter <http://alpha.gnu.org/gnu/emacs/pretest> empfohlen.

Péter Szabó demonstrierte in seinem Vortrag »Automated DVD menu authoring with pdfL^AT_EX« das Paket `dvdmenuauthor`, eine Tool-Sammlung für die Erstellung von Video-DVDs auf Unix-Systemen.



Norbert Preining beschrieb die unter der Debian-Version »etch« parallel verfügbaren T_EX-Systeme T_EXLive 2005 und t_ET_EX und wie sowohl Systemadministratoren als auch normale Benutzer von diesem Nebeneinander zweier T_EX-Distributionen profitieren können. Norbert schloß seinen Vortrag mit einem Ausblick auf T_EXLive 2007 und der Aussage, dass

für Debian wegen des Rückzugs von Thomas Esser die Zukunft in der T_EXLive-Distribution läge.



Die Präsentation von Atif Gulzar (Kofautor: Shafiq-ur Rahman) aus Pakistan begann mit einer einführenden Erklärung über Urdu, einer Sprache, die weltweit von 60 Millionen Menschen in zwanzig Ländern als Muttersprache gesprochen wird. Urdu basiert auf der arabischen Schrift, benutzt aber Nastaliq, eine besonders schöne, aber auch manchmal schwer

zu entziffernde Variante der arabischen Schrift. Atif entwarf einen virtuellen Font für Omega mit 827 Glyphen und benutzte Omegas externe OTPs, um in zwei Durchläufen akzeptable Ligaturplatzierungen und Kerning zu erreichen². Aus den mehr als 20 000 gültigen Ligaturen in Urdu gelang es Atif, einen Satz von rund 7 000 Ligaturen korrekt wiederzugeben.

²Der Studienaufenthalt von Atif Gulzar in Brest an der ENST Bretagne bei Yanniss Haralambous wurde von DANTE e.V. gefördert

Hossam A.H. Fahmy präsentierte seine Arbeit mit Amir M.S. Hamdy zur Schaffung eines Fonts, der für den Satz des Korans geeignet ist. Anhand von Beispielen zeigte er die vielfältigen Probleme, die bei dem Versuch entstehen, eine kalligraphische Schrift wie das Arabische zu digitalisieren. Der zweite Teil des Vortrags beschäftigte sich dann mit einem Detail aus dieser Arbeit, nämlich der Frage, wie man in METAFONT die Spitze einer Schreibfeder simuliert.

Martin Schröder informierte uns über die neuen Funktionen der pdfTeX-Version 1.40. Besonders erwähnenswert sind die Eigenschaften, nun komprimierte Objekt-Streams zu erzeugen, die Unterstützung für JBIG2-kodierte Bilder und Colorstacks à la dvips. Die Colorstack-Eigenschaften werden bereits in dem neuen Release des hyperref-Paketes benutzt und lösen das L^AT_EX-Problem, dass sich die Textfarbe mit einem Seitenumbruch ändert.

Karel Horák führte uns durch die Geschichte des Háček (oder Neu-Adobisch: Caron) im tschechischen Schriftsatz. Er zeigte uns die historische Entwicklung der Symbolform und die Vielzahl unterschiedlicher Erscheinungsformen in aktuellen Zeichensätzen. Einige wenige bewertete er als gute Form, mehr als weniger gute, ein Großteil sind schlicht hässlich und völlig ohne Bezug zu der tschechischen Tradition. Als wahrscheinlicher Grund wird vermutet, dass die großen Schriftgießereien nie tschechische Typographen zu Rate zogen.



Auch Hàn Thê Thành sprach über Akzente, allerdings über die im Vietnamesischen gebräuchlichen. Die Schrift basiert auf dem lateinischen Alphabet, sie hat allerdings eine große Anzahl akzentuierter Zeichen (Vokale mit bis zu zwei Diakritika), um aussprache-abhängige Lautverschiebungen (Töne) eindeutig machen zu können. Mit seinem

VnTeX-Paket hat Hàn Thê Thành eine vollständige Lösung zum Setzen von Vietnamesisch zur Verfügung gestellt, die auch eine große Anzahl von Fonts unterstützt, von denen er einige selbst geschaffen hat.

Der Vortragstag endete mit zwei Präsentationen von Tomasz Łuczak. Die erste unter dem Thema »LyX: an editor not only for secretaries« beschäftigte sich mit weniger bekannten Fähigkeiten des Lyx-Editors (www.lyx.org). Der zweite Vortrag drehte sich um die Konvertierung von Wiki-Markup in T_EX-Quellen.

Beide Vorträge wurden in Polnisch gehalten, aber trotz der Simultanübersetzung ins Englische durch freundliche T_EXies aus dem Auditorium war es schwierig, ihnen zu folgen.

Montag, 30. April 2007

Für den Montag waren keine Vorträge angesetzt, stattdessen unternahmen wir eine Exkursion nach Toruń, wo wir die Kopernikus-Bibliothek besuchten und eine Führung durch die Altstadt auf dem Programm stand. Danach war der Besuch von Chełmno angesagt, wo auch das Bankett stattfinden sollte.

Toruń (dt.: Thorn), im Jahre 1231 gegründet und an der Weichsel gelegen, ist seit dem Mittelalter ein wichtiger Handelsknoten. Als Mitglied der Hanse besaß Toruń eine Flotte von 150 Schiffen, deren Handelsvolumen Toruń einen Wohlstand garantierte, der sich mit Brügge, Kopenhagen und London messen ließ. Die UNESCO erhob 1997 die Altstadt von Toruń zum Weltkulturerbe.



In der Kopernikus-Bibliothek wurden uns die Schmuckstücke der Sammlung präsentiert. Von den vielen kostbaren Büchern, die wir zu sehen bekamen, sei die Erstausgabe von Kopernikus' Werk »*Revolutionibus Orbium Coelestium*« (Von den Umdrehungen der Himmelskörper) aus dem Jahre 1543 zu nennen. Zum Abschluss bekamen wir eine »neue«

Ausgabe der Gutenberg-Bibel zu Gesicht. Es ist eines der 180 Exemplare, die als Original-Reproduktion produziert wurden, genau die Anzahl, die auch Gutenberg druckte.

Unsere Führung durch die Thorner Altstadt begann am historischen Rathaus unter der Statue von Nikolaus Kopernikus mit der Inschrift »Er bewegte die Erde und brachte die Sonne zum Stillstand«. Auf unserer Rundtour besuchten wir einige Kirchen und Sehenswürdigkeiten, darunter Kornspeicher aus dem 14.–17. Jh. und die Stadtmauer aus dem 13. Jh. mit Basteien und Stadttoren.

Am späten Nachmittag trafen wir in Chełmno (dt.: Kulm) ein. Auch Kulm ist eine Stadtgründung des Deutschen Ordens (1233) und gilt mit seinen sieben Kirchen auf neun Hügeln, umgeben von den fast unverändert erhaltenen Wehrmauern als Europas bestes Beispiel des mittelalterlichen Städtebaus.

Chełmno Kirchen stammen aus dem 13. und 14. Jahrhundert, in einer von ihnen werden seit mehreren hundert Jahre Reliquien des Heiligen Valentin aufbewahrt. Der Valentinstag am 14. Februar wird daher in Chełmno besonders gefeiert.

Nach einem kurzen Stadtrundgang begaben wir uns zum Restaurant, wo wir unser Dinnerbuffet genießen durften. Musikalische Unterhaltung wurde von einer Gruppe unter Leitung von Bogusław Jackowskis Tochter Kasia geboten.

Dienstag, 1. Mai 2007

Im ersten Vortrag am Dienstag präsentierte Hàn Thế Thành eine Zusammenstellung aller font-spezifischen Eigenschaften von pdfTeX. Einige wenige wie »Font-Expansion« und »Margin-Kerning« sind bereits im pdfTeX-Manual dokumentiert. Für den Rest existieren nur README- und Beispiel-Dateien in E-Mails der Mailinglisten. Zum ersten Mal wurden alle diese Eigenschaften zusammenhängend dargestellt. Diese beinhalteten »adjusting letter« und »inter-word spacing«, »adding additional kerning« vor oder nach bestimmten Zeichen eines Fonts, Unicode-Support für Aktionen wie »Ausschneiden«, »Einfügen« und »Suchen« und für Font-Subsetting, ein Mechanismus für die Unterstützung von CJK-Sprachen (Chinesisch, Japanisch und Koreanisch).



Hans Hagen begann seine Präsentation »ConTeXt and OpenType: what kind of font system do we need« mit der Beschreibung, wie in ConTeXt das Fontsystem aufgebaut ist, mit dem zwischen verschiedenen Fonts, Font-Stilen und -Größen einfach umgeschaltet werden kann und das auch eine sichere Behandlung im Mathematikmodus ermöglicht. In ConTeXt

wird dies über **typescripts** gelöst. Innerhalb eines **typescripts** kann man beispielsweise Palatino-Regular als Default-Serif-Font, Palatino-Sans als Sans-Serif-Font, Courier als Monospace-Font und Euler als Mathematik-Font definieren. Sobald das **typescript** aufgerufen wird, stehen die gewünschten Fonts über die Standardkommandos wie `\rm`, `\ss`, `\tt` und `$$` zur Verfügung. Es wurde diskutiert, welchen Einfluss der Trend zu OpenType-Fonts mit seinen erweiterten Eigenschaften, LuaTeX und XeTeX nun auf ein einheitliches Font-

User-Interface haben werden. Brauchen wir neue Mechanismen oder können alte angepasst werden? Und wie sieht diese Lösung in ConTeXt aus.

Taco Hoekwater erklärte im Anschluss, wie LuaTeX mit seiner eingebauten Unterstützung für OpenType-Fonts die Notwendigkeit für statische Fontmetriken (`tfm`) überflüssig macht. Zur Zeit sind in LuaTeX einige Dutzend Callbacks an strategischen Stellen implementiert. Mit diesen Callbacks können TeX-Codesequenzen durch eigene Behandlungsmethoden ersetzt werden. Taco führte vor, wie LuaTeX-Callbacks bei OpenType-Benutzung die Metrikinformationen direkt aus dem OpenType-Font extrahieren.

In seinem Vortrag »L^ATeX: A modification of the Logo« identifizierte Grzegorz Murzynowski zwei unterschiedliche Ansichten zu den in TeX & Co. definierten Logos. Die erste Gruppe verfißt die Betrachtungsweise, dass der Font ein integraler Bestandteil des Logos sei und daher die Kombination nicht geändert werden dürfe. Die zweite Gruppe vertritt die Meinung, dass ein TeX-Logo immer den Font des Kontextes benutzen sollte. Für die zweite Gruppe schlug Grzegorz einige leichte Veränderungen für das L^ATeX-Logo vor, damit es besser zu den benutzten Fonts passt und die Berührung der Buchstaben minimiert wird.



Sam Guravage (13), der jüngste Sprecher, der je auf einer europäischen TeX-Tagung einen Vortrag hielt, erklärte, warum er TeX für alle seine schulischen Aufgaben nutze: »Weil mein Vater mich dazu gezwungen hat«. Sam zählte dann auf, was er an TeX einfach findet: die Kapitel- und Abschnittsaufteilung und Listen, und die eher schwierigen Dinge wie

Einbindung von Bildern und die Deutung von Fehlermeldungen. Sams Schlussfolgerung war, dass die Benutzung von TeX seine Ausarbeitungen besser aussehen lässt und das bedeutet direkt eine bessere Note. Und er plant auch weiterhin TeX zu benutzen.



David Kastrup begann eine Serie von drei Beiträgen mit der Vorstellung von `qstest`, einem $\text{\LaTeX}$ -Makropaket das eine Kontrolle darüber erlaubt, dass Änderungen nicht zu Rückschritten in der Funktionalität führen. Der Benutzer kann in seinem `.dtx`-File neben den Makros auch die Dokumentation und die Unit-Tests unterbringen und damit

in einer Testumgebung sicherstellen, dass Makros und Register von $\text{\TeX}$ erwartete Werte aufweisen.

David fuhr fort mit Ausführungen zum `makematch`-Makropaket, einer Ausgliederung des `gctest`-Paketes. `makematch` ist in der Lage, eine Patternliste mit einer ungeordneten Schlüsselwortliste auf Übereinstimmung zu prüfen.

Der abschließende Vortrag von David Kastrup hatte das `bigfoot`-Paket zum Thema. Von diesem Makropaket, das ursprünglich für text-kritische Editionen entwickelt wurde, kann auch der normale $\text{\LaTeX}$ -Benutzer profitieren. So wird beispielsweise in $\text{\TeX}$ die globale Seitenumbruchsoptimierung umgangen, wenn eine Fußnote nicht komplett auf die Seite passt. In `bigfoot` hingegen wird für Fußnoten jeder mögliche Seitenumbruch überprüft. Das bedeutet, dass $\text{\TeX}$ bei Benutzung des Paketes eine optimale Kombination von Umbrüchen im Haupt- und Fußnotentext findet. Robustheit, Optimierung, Farbübergang und Absätze in Fußnotentexten sind weitere Gründe für die Benutzung von `bigfoot` statt des Standards in $\text{\TeX}$.

Klaus Höppner führte uns in den Workflow für die Erstellung eines PostScript-Type-1-Fonts aus einer METAPOST-Quelle bei Benutzung von `MetaType1` ein. Dieses Toolset aus der Werkstatt von Bogusław Jackowski, Janusz Nowacki und Piotr Strzelczyk umfasst METAPOST, `t1utils` und `AWK`. Zusammen werden sie benutzt, um PostScript-Type-1 AFM-, TFM- und PFB-Dateien zu erzeugen. Obwohl die Dokumentation spärlich ist, erwies sich `MetaType1` als das richtige Werkzeug für diesen Zweck.

Petr Sojka und Michal Růžicka zeigten in ihrem Vortrag Möglichkeiten für die Generierung von PDF, HTML und XHTML+MathML aus einer einzigen $\text{\LaTeX}$ -Quelle auf. Viele Einzelquellen-Projekte starten mit XML als Ausgangspunkt, in diesem Projekt machte der hohe Anteil von Mathematik jedoch $\text{\TeX}$ zum einzig möglichen Eingangsformat. Durch eine strikte Trennung von

Form und Inhalt, und durch Anpassungen in den T_EX4ht-Quellen, war es den Autoren möglich, getrennte Workflows für jedes Ausgabeformat zu realisieren.

Péter Szabó berichtet über seine Erfahrungen bei der Zusammenstellung von Konferenz-Proceedings, unter anderem auch der der letztjährigen EuroT_EX. Péter beschrieb, wie der Einsatz von verständlichen Prozeduren und Werkzeugen die Arbeit von Autoren, Editoren und Druckern vereinfachen kann. Versionskontroll-Software, Mailinglisten, Shellskripte und natürlich T_EX können in einer Weise verknüpft werden, die einen wirkungsvollen Workflow für Publikationen aufbaut.

David Kastrup hielt einen Vortrag über **DocScape Publisher**, ein XML-basiertes Datenbank-Publikationssystem der Firma QuinScape GmbH. Kernkomponenten von **DocScape** sind pdfL^AT_EX und David Carlisles **xmltex**. Aktuell wird die Applikation für Finanzreporte, eine Vielfalt von Produktkataloge und Online-Reports genutzt.

Karel Píška beschrieb die von ihm entwickelten Prozeduren und Programme für die Untersuchung und den Vergleich von Zeichensätzen. Seine Toolsammlung kann von dem Server <http://www-ep.fzu.cz/~piska/tfcpr.html> geladen werden. Aus dem Repertoire demonstrierte er die folgenden Programme für den Vergleich von zwei Bitmap-Repräsentationen eines Glyphenpaares in zwei verschiedenen Auflösungen: **cprpk**, **cprpkt1**, **cprpkt1c**, **cpcticpk** und **cprpkpk**.

Für den Vergleich von Kerning-Paaren in zwei oder drei T_EX-Fonts oder in zwei Versionen eines Fonts stehen die folgenden Tools zur Verfügung: **prfkrn**, **prfkrna**, **cpkrn** und **cpkrna**. **prprof** und **cprprof** sind für den Vergleich und Proof von Outline-Fonts vorgesehen.



In seiner zweiten Präsentation berichtete Karel Píška über den Einsatz seiner Font-Werkzeuge bei der Analyse der Latin-Modern-Fonts. Seine Ergebnisse enthielten Beispiele von defekten und inkonsistenten Glyphen. Interessanterweise fand er eine übermäßige Anzahl von Kerningpaaren, von denen der größte Teil in keiner Sprache Sinn macht. Durch seine an-

spruchsvolle Detailarbeit verbessert Karel die Qualität der Fonts, die wir jeden Tag nutzen.

Janusz M. Nowacki gab in seinem Vortrag die Veröffentlichung eines »neuen« Fonts bekannt: Cyklop. Das Design stammt von J. Idźkowski, die S-ka Giesserei in Warschau goss diesen Font in den 20er Jahren in Blei. Cyklop ist ein schwerer zweielementiger Sans-Serif-Zeichensatz, der im Original nur in einer Oblique-Form in Größen von 8 bis 48 pt produziert wurde. Er wird für Schlagzeilen, Poster, Formulare und Schilder benutzt. Janusz erweiterte nicht nur den Zeichensatz um alle lateinischen akzentuierten Glyphen, er ergänzte ihn auch um eine neue aufrechte Variante.

Zur Abrundung und zum Ausklang des Tages gab es einen informellen Empfang im Vortragssaal. Bei einem Glas Wein und netter Konversation konnte man sich Schwarz-Weiß-Fotos in Postergröße anschauen, die Janusz Nowacki in seiner Tätigkeit als Fotograf geschossen hat, bevor er mit den Arbeiten an Fonts begann.

Mittwoch, 2. Mai 2007



Paweł Jackowski präsentiert die diesjährige Ernte der T_EX-Schönheiten und T_EX-Obskuritäten, dieses Mal waren es sechzehn an der Zahl. Man muss diese »Perlen« gesehen haben, um zu glauben, dass sie existieren. Zu begutachten ist die Sammlung unter <http://www.gust.org.pl/pearls>.

Ross Moore sprach über seine Erfahrungen beim Setzen von Beiträgen für »The Journal of The Australian Mathematical Society«. Mit dem Einsatz von pdfT_EX ist es nun möglich, Metadaten in PDF-Dateien einzubetten und Unicode-Lesezeichen für alle Kapitel, Tabellen, Theoreme und Referenzen zu verwenden, die auch einfache mathematische Ausdrücke enthalten. Sogar die halbautomatische Generierung von Hyperlinks zu MathSciNet (einer AMS-Datenbank mit mehr als einer Million Abstracts mathematischer Veröffentlichungen) wurde durch den Einsatz von pdfT_EX möglich.

Um unseren Weg in die Zukunft zu beleuchten, rekapitulierte Arthur Reutenauer nochmals die letzten Jahren in der Weiterentwicklung von T_EX. Der Untertitel

seines Vortrags »Pax T_EXnica – The program on which the sun never sets« beleuchtete die Entwicklungsgeschichte von T_EX78 über Aleph, X_YT_EX und LuaT_EX, die rund um die Welt ging. Die verschiedenen T_EX-Erweiterungen und Makropakete haben uns mittlerweile in die Lage versetzt, (beinahe) alle Sprachen und Schriften der Welt zu setzen.

Als Erweiterung des historischen Aspekts präsentierte Ulrik Vieth einen Überblick über das T_EX-Archive, einem Archiv mit historischen T_EX-Distributionen und Paketen. Zu finden ist das historische Archiv auf dem TUG-Server <http://ftp.tug.org/historic/>. Die Geschichte von T_EX umfasst mittlerweile 30 Jahre, aber während die Entwicklungsgeschichte von T_EX von der Frühzeit an sehr gut dokumentiert ist, lässt sich die Geschichte verschiedener Makropakete und System wie METAFONT und METAPOST oft nur aus Anekdoten und Schnipseln rekonstruieren. Nach 30 Jahren bleibt die Geschichte von T_EX immer noch ein interessantes Forschungsfeld. Das Archiv enthält eine Fülle von Informationen, aber auch einige unübersehbare Lücken. Informationen sind jederzeit willkommen, insbesondere zu (pdf)T_EX und den Latin-Modern-Fonts.

Bogusław Jackowski, Jerzy Ludwowski und Janusz M. Nowacki gaben einen Bericht über den aktuellen Stand der beiden großen Fontentwicklungsprojekte: Latin Modern und T_EX-Gyre.

Das Latin-Modern-Font-Projekt wurde 2002 begonnen. Basierend auf Computer Modern, besteht die Latin-Modern-Fontfamilie derzeit aus 72 Text- und 20 Mathematikzeichensätzen, die sowohl in PostScript-Type-1- als auch im OpenType-Format zur Verfügung stehen.

Das T_EX-Gyre-Projekt startete 2006 und verfolgt das Ziel, die mit Ghostscript ausgelieferten 33 URW⁺⁺-Zeichensätze für alle latein-basierten Schriften auf die gleiche Zahl von Glyphen wie die LM-Zeichensätzen auszubauen. Darüberhinaus wurde bisher das Hinting verbessert und die Fonts im OpenType-Format angeboten. Für die nahe Zukunft ist die Erweiterung der Fonts mit mathematischen Zeichen geplant. Hier eine Liste der T_EX-Gyre-Fonts (mit den Namen der Originale in Klammern):

- Adventor (Avantgarde)
- Bonum (Bookman)
- Cursor (Courier)
- Heros (Helvetica)

- Pagella (Palatino)
- Schola (Century Schoolbook)
- Termes (Times)
- Chorus (Zaph Chancery)

Die Projektseiten für das Latin-Modern- und Gyre-Projekt findet man unter <http://www.gust.org.pl/projects/e-foundry/latin-modern> und [tex-gyre](http://www.gust.org.pl/projects/e-foundry/tex-gyre).

Marek Ryćko rief im letzten Vortrag der Tagung Niklaus Wirths Aussage »Algorithmen + Datenstrukturen = Programm« ins Gedächtnis. Er demonstrierte wie man LISP-ähnliche Strukturen in T_EX realisieren kann. Er argumentierte, dass ein sauberer und konsistenter Ansatz zur Behandlung von Elementlisten die Programmierung in T_EX einfacher und T_EX-Programme, sprich Makros, sicherer machen würde.



Jerzy Ludwichowski schloss die Konferenz mit einem Dank an die Organisatoren, Autoren und Teilnehmer. Den Spezialpreis des GUST-Vorstandes erhielt Sam Guravage für die beste Konferenz-Präsentation. Die Urkunde war auf selbstgemachtem Papiere gedruckt und wurde von einem Präsent begleitet, das dem Alter des Ausgezeichneten angemessen war.

LuaT_EX goes beta

Hans Hagen

At TUG 2007, in San Diego, the first official beta release of LuaT_EX was presented. Apart from the LuaT_EX team members (Taco Hoekwater, Hartmut Henkel and me) the author of Lua (Roberto Ierusalimsky) was present, which added a nice touch to the conference. Of course the usual DANTE delegation was there to keep an eye on us.

The qualification beta is bound to the objectives of the first stage of the project. As you may know, pascal/c code development (by Taco) is partly funded by the Oriental TeX project and therefore the ability to typeset Arabic was a condition for going beta. Fortunately we had a long flight to San Diego to sort out the last details. The ability to handle OpenType fonts (and to extend them at runtime) is one of the qualifications of LuaTeX, and all advanced font features are handled by Lua code. Handling the ZapfinoPro as well as ArabType fonts was our benchmark.

The development is driven by a more or less continuous chat model, where Taco and I use Skype as communication channel. Binaries flow in one direction, Lua code goes the other way. Ideas are explored and tested in intense chat sessions. We test code on real documents using ConTeXt, which we can adapt as we wish. The version of ConTeXt that will take full advantage of LuaTeX is tagged as MkIV, indicating that we have a major upgrade compared to MkII (the version that uses pdfTeX and XeTeX).

Although those involved in ConTeXt development have access to the experimental MkIV code, the first release of MkIV was a few weeks after the conference, at August 6, 2007. Of course the first days were rather hectic because we had to solve all kind of platform related issues, but eventually installation and updating will be as easy as with a traditional ConTeXt.

Next year at TUG 2008 the first official production version of LuaTeX will be presented. However, the order of TeX meetings will probably lead to sneak previews of the program (and therefore MkIV code) at the DANTE meeting (March 2008) and Bachotek (May 2008). Such meetings also provide us good deadlines. At that time we should also have a more clear picture of what components of MkIV are generic and usable in other environments.

More information: www.luatex.org.

Bretter, die die Welt bedeuten

Zeichensatzkodierungen von L^AT_EX

Frank Mittelbach, Robin Fairbairns, Werner Lemberg

Dieser Artikel beschreibt die Ideen, welche den Zeichensatzkodierungen von L^AT_EX zugrunde liegen, und die daraus resultierenden Beschränkungen für neue Kodierungen. Er enthält außerdem detaillierte Beschreibungen aller bisher registrierten Kodierungen.

Eine englischsprachige Version (`encguide.tex`) ist Bestandteil der L^AT_EX-Distribution.

Einleitung

Das Programm T_EX kennt drei Arten von Kodierungen, die (mehr oder weniger explizit) im T_EXbook [4] dokumentiert sind:

- Die Eingabekodierung, welche die Bedeutung aller Zeichen in den von T_EX gelesenen Dateien festlegt. KNUTH schreibt: »Your version of T_EX will recognise the characters you type on your keyboard.« Gemeint ist, dass das Programm T_EX auf das System angepasst wird, und zwar *vor* der Kompilation – die Eingabezeichen werden fest mit Hilfe einer statischen Tabelle auf eine interne Kodierung abgebildet. Weder L^AT_EX noch Plain-T_EX benutzen diesen (plattformabhängigen) Mechanismus, um verschiedene Eingabekodierungen zu verarbeiten.
- Der *Token*-Eingabestrom. Das T_EXbook beschreibt ausführlich die Konvertierung der Eingabezeichen zu Token und deren Weiterverarbeitung.
- Die Zeichensatzkodierung, also die Abbildung von Zeichencodes auf die Glyphen in den Zeichensätzen, mit denen das jeweilige Dokument gesetzt

wird. Auch hier beschreibt das $\text{\TeX}$ book einen Satz von Zeichensatzkodierungen, die sich aber als unzureichend für mehrsprachige Dokumente erwiesen haben.

Dieser Artikel behandelt nur den letzten Punkt. Es wird gezeigt, *warum* die originalen, von KNUTH entwickelten Kodierungen nicht den heutigen Anforderungen genügen; im Weiteren wird erläutert, wie neue Zeichensatzkodierungen konstruiert und definiert werden können.

Zeichensatzkodierungen sind nicht nur für den Textsatz wichtig, sondern auch für den Trennungsalgorithmus; weiter unten wird das genauer beschrieben.

Geschichtlicher Abriss der Zeichensatzkodierungen in $\text{\TeX}$

Bis zum Erscheinen von $\text{\TeX}$ Version 3 wurden Zeichensatzkodierungen kaum beachtet. Es wurden fast ausschließlich KNUTHs Zeichensätze benutzt (die *Computer Modern*-Schriftfamilie, deren Zeichensatzkodierungen heute als OT1 und OM bezeichnet werden), andernfalls war man auf sich allein gestellt.

Ein Problem, das sich bei der Verwendung von OT1 mit einer unveränderten $\text{\TeX}$ -Version ergibt, ist die fehlende Möglichkeit, Wörter mit diakritischen Zeichen automatisch umzubreaken. Obwohl OT1 Akzente aller Art für westeuropäische Sprachen enthält, verhindert diese Beschränkung optimalen Textsatz, falls nicht manuell eingegriffen wird. Seit $\text{\TeX}$ 3 besteht nun die Möglichkeit, Trennmuster zu wechseln. Daher musste ein Weg gefunden werden, um dieses Manko zu beseitigen. Das Ergebnis einer Arbeitsgruppe während der TUG-Konferenz 1989 in Cork (Irland) ist ein Vorläufer der Zeichensatzkodierung T1 mit 256 Glyphen, die Kompositglyphen (Basisglyphen mit diakritischem Zeichen) für die meisten westeuropäischen (und einige osteuropäischen) Sprachen zur Verfügung stellt. Jörg KNAPPEN hat diese Kodierung mit seiner *European Computer*-Familie realisiert; für die meisten gängigen POSTSCRIPT-Schriften gibt es virtuelle T1-Fonts, und mit Hilfe des *fontinst*-Paketes [12] kann man relativ leicht virtuelle Fonts für beliebige Zeichensätze automatisch erzeugen.

Inzwischen gibt es eine ganze Reihe von Kodierungen für $\text{\TeX}$ -Zeichensätze, wobei die Cork-Kodierung Pate stand.

Im Gegensatz dazu hat sich bei mathematischen Schriftkodierungen wenig geändert. Eine technische Arbeitsgruppe der TUG wurde während des Treffens in Cork eingerichtet, deren Ziel es war, eine oder mehrere Kodierungen zu

schaffen, die 256 Zeichen umfassen. Sie sollen die K $\text{\LaTeX}$ Thschen Originale regularisieren und (unter Verwendung anderer Zeichensätze) erweitern, wobei die Bedürfnisse von Mathematikern aller wissenschaftlichen Zweige berücksichtigt sind. Unabhängig von dieser Arbeitsgruppe gab es auch einen ersten Vorschlag von Justin ZIEGLER [16], das sogenannte *Aston-Proposal*, entstanden im Auftrag des $\text{\LaTeX}$ -Projekts; eine vorläufige Version wurde von Matthias CLASEN und Ulrik VIETH implementiert [18, 17]. Da jedoch kürzlich eine große Anzahl von mathematischen Zeichen in Unicode aufgenommen wurde [19], ist sicher noch mit Änderungen zu rechnen, sodass sich die Veröffentlichung von neuen mathematischen Kodierungen weiter verzögern wird.

Offizielle Zeichensatzkodierungen

In diesem Abschnitt werden alle bis jetzt registrierten Zeichensatzkodierungen dargestellt. Es werden der registrierte $\text{\LaTeX}$ -Name, der Verwendungszweck sowie der Autor angegeben, zusammen mit einer tabellarischen Auflistung der verwendeten Zeichencodes. Dazu kommt noch die Liste der *variablen Positionen* (siehe die Beschreibung von OT1 weiter unten) und eine Referenz für weitere Informationen.

Eigentlich sollten alle Zeichensätze, die einer bestimmten Zeichensatzkodierung folgen, auch tatsächlich die gleichen Glyphen haben, doch gibt es einige Kodierungen (im besonderen OT1 und deren Abkömmlinge), wo das nicht für alle Mitglieder einer Zeichensatzfamilie zutrifft.

Die Namen von Zeichensatzkodierungen

Folgende Namenskonvention wird für Zeichensatzkodierungen verwendet:

- Ein Kodierungsname besteht aus bis zu drei Zeichen (nur Großbuchstaben und Ziffern).
- Das $\text{\LaTeX}$ 3-Projekt hat folgende Anfangsbuchstaben von Kodierungsnamen reserviert: T (Standardkodierungen für Text mit 256 Zeichen), TS (Symbole, welche die zugehörige T-Kodierung erweitern), X (Textkodierungen, die den strengen Kriterien einer T-Kodierung nicht entsprechen), M (Standardkodierungen für Mathematik mit 256 Zeichen), S (andere Symbolkodierungen), A (Kodierungen für andere spezielle Anwendungen), OT (Standardkodierungen für Text mit 128 Zeichen) und OM (Standardkodierungen für Mathematik mit 128 Zeichen).

- Lokale Kodierungen sollten mit L starten, experimentelle Kodierungen für weite Verbreitung mit E und nichtklassifizierte (oder unbekannte) Kodierungen mit U.

Warnung: Neue Zeichensatzkodierungen sollten mit größter Vorsicht implementiert werden; auf jeden Fall sollte man vorher verschiedene L^AT_EX-Foren kontaktieren, ob es nicht eine andere Lösung gibt. Idealerweise sollten Kodierungen nicht nur für einen bestimmten Zeichensatz, sondern parallel für eine größere Anzahl von Zeichensätzen verwendet werden können; eine solche Vorgehensweise reduziert spätere Probleme bereits im Ansatz.

Ein Beispiel für eine schlechte Kodierung ist TS1, obwohl mit den besten Absichten definiert: Der Großteil der verfügbaren Schriften deckt nur Teile der Kodierung ab, sodass fehlende Glyphen im besseren Falle durch einen anderen Font ersetzt werden müssen, im schlechteren Falle leer bleiben oder ein kleines schwarzes Rechteck erzeugen.

Zeichensatzkodierungen für Text mit 128⁺ Glyphen

Die OT-Zeichensatzkodierungen sind KNUTHs ursprüngliche Textkodierungen, die bereits mit den frühesten Versionen von T_EX benützt wurden. Das ›O‹ steht für ›original‹ oder ›old‹ (engl. für ›alt‹).

Name in L ^A T _E X:	OT1
Kodierungsname:	T _E X text
Autor:	Donald Erwin KNUTH
Verwendete Glyphindizes:	0x00–0x7F
Variable Positionen:	0x0B–0x0F, 0x24, 0x3C, 0x3E, 0x5C, 0x7B–0x7D
Beispielzeichensatz:	cmr10
Referenz:	[4, S. 427]

Donald KNUTH entwickelte seine Zeichensatzkodierung (und seine Zeichensätze) in einer völlig anderen Umgebung im Vergleich zum heutigen Einsatz von T_EX: Sein Mainframe-Computer hatte sehr wenig Hauptspeicher, und es gab nur geringe Erfahrung (und Bedarf) für mehrsprachigen technischen Textsatz. Aus diesen Gründen wurde die Einheitlichkeit teilweise der Effizienz geopfert.

In Abhängigkeit vom Zeichensatzschnitt ergeben sich daher für manche Glyphen *variable Positionen*. Zum Beispiel sind <, >, \, { und } nur in

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl
1	ı	j	`	´	˘	˙	˚	˛	ß	æ	œ	ø	Æ	Œ	Ø	
2	ˆ	!	”	#	\$	%	&	’	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	i	=	ı	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	“	]	^	˙
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	—	—	”	˜	¨

Tabelle 1: Die OT1-Kodierung von `cmr10`.

Schreibmaschinenschrift verfügbar, und `$` teilt sich mit `£` den gleichen Index (in verschiedenen Zeichensätzen).

Für den Anwender bedeutet das konkret, dass die direkte Auswahl dieser Glyphen unvorhergesehene Resultate produzieren kann. So kann etwa `<` oder `\symbol{"3E}` das Zeichen `ı` erzeugen.

Name in $\text{\LaTeX}$:	OT2
Kodierungsname:	UW Cyrillic encoding
Autor:	University of Washington
Verwendete Glyphindizes:	0x00–0x7F
Variable Positionen:	—
Beispielzeichensatz:	wmr10
Referenz:	[5]

Der benützte Zeichensatz stammt aus der LH-Fontfamilie [13]. Für praktisch alle Anwendungen ist es besser, stattdessen eine der T2-Kodierungen zu verwenden.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Ъ	Ь	Ц	Э	І	Є	Ђ	Ћ	Њ	Љ	Џ	Э	і	є	ђ	ћ
1	Ю	Ж	Й	Ё	Ѵ	Ѳ	Ѕ	Я	ю	ж	й	ё	ѵ	ѳ	ѕ	я
2	“	!	”	Ђ	ѵ	%	’	’	(	)	*	Ђ	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	«	ı	»	?
4	˘	А	Б	Ц	Д	Е	Ф	Г	Х	И	Ј	К	Л	М	Н	О
5	П	Ч	Р	С	Т	У	В	Щ	Ш	Ы	З	[	“	]	Ь	Ъ
6	‘	а	б	ц	д	е	ф	г	х	и	ј	к	л	м	н	о
7	п	ч	р	с	т	у	в	щ	ш	ы	з	—	—	№	ь	ъ

Tabelle 2: Die OT2-Kodierung von `wnr10`.

Name in $\LaTeX$: OT3
Kodierungsname: UW IPA encoding
Autor: University of Washington
Verwendete Glyphindizes: 0x00–0x7F
Variable Positionen: —
Beispielzeichensatz: `wsuipa10`
Referenz: [6, S. 149]

Durch die Einführung des `tipa`-Paketes, welches viel bessere Unterstützung für das Alphabet der Internationalen Phonetischen Gesellschaft (IPA) liefert, ist OT3 heute obsolet. Beachtenswert ist im Besonderen, dass es bis dato keine Kodierungs-Definitionsdatei `ot3enc.def` gibt.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	ę	ą	ǫ	ɔ	ł	ś	ś	ł	ł	β	ø	ę	Ć	đ	đ	đ
1	đ	đ	đ	đ	D	ə	ə	ə	ε	z	z	z	g	g	G	γ
2	ŷ	ŷ	lv	h	fi	fj	q	i	z	l	I	z	J	ł	ł	ł
3	ł	ł	λ	λ	uj	uj	j	uj	uj	N	⊙	θ	o	ω	ω	ω
4	∞	þ	þ	φ	r	r	l	l	l	R	B	ş	ş	ş	σ	σ
5	ł	ł	ł	θ	u	u	u	U	U	u	u	χ	Ł	Y	z	z
6	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z
7	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z

Tabelle 3: Die OT3-Kodierung von wsuipa10.

Name in L ^A T _E X:	OT4
Kodierungsname:	Polish text encoding
Autoren:	B. JACKOWSKI und M. RYĆKO
Verwendete Glyphindizes:	0x00–0x7F, 0x81, 0x82, 0x86, 0x8A, 0x8B, 0x91, 0x99, 0x9B, 0xA1, 0xA2, 0xA6, 0xAA, 0xAB, 0xAE, 0xAF, 0xB1, 0xB9, 0xBB, 0xD3, 0xF3, 0xFF
Variable Positionen:	0x0B–0x0F, 0x24, 0x3C, 0x3E, 0x5C, 0x7B–0x7D
Beispielzeichensatz:	plr10
Referenz:	—

KNUTH hat zwar Unterstützung für den polnischen Buchstaben Ł in OT1 integriert, doch fehlt das Ogonek (.), ein diakritisches Zeichen, das für Polnisch gebraucht wird. Aus diesem Grunde wurde OT4 definiert – eine Kodierung, die es schon lange vor T1 gab.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8		Ա	Շ				Է				Լ	Ռ				
9		Տ								Ճ		Յ				
A		ա	ճ				ե				լ	ռ			«	»
B		Տ								Ճ		Յ				
C																
D				Օ												
E																
F				Օ												”

Tabelle 4: Die obere Hälfte der OT4-Kodierung von plr10. Die untere Hälfte ist identisch mit OT1.

Name in $\text{\LaTeX}$:	OT6
Kodierungsname:	Armenian text encoding
Autor:	Serguei DACHIAN
Verwendete Glyphindizes:	0x03–0x0F, 0x13–0xFF
Variable Positionen:	—
Beispielzeichensatz:	artmr10
Referenz:	—

Diese Kodierung wurde definiert, um armenische Zeichensätze des ArmTeX -Paketes in einer Standard- $\text{\LaTeX}$ -Umgebung verwenden zu können.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				2	ŀ	đ	Ł	Ł	Œ	Š	Ž	Q	Ŧ	Ÿ	Φ	Ō
1				ă	ŋ	đ	Ł	Ł	Œ	Š	Ž	Q	Ŧ	Ÿ	Φ	Ō
2	u	‘	”	#	\$	%	&	’	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	«	=	»	’
4	@	U	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ
5	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ	Ŧ	Ÿ
6	`	w	p	g	ŋ	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł
7	u	p	p	u	u	u	u	u	u	u	u	u	u	u	u	u

Tabelle 5: Die OT6-Kodierung von artmr10.

Zeichensatzkodierungen für Text mit 256 Glyphen

Name in L ^A T _E X:	T1
Kodierungsname:	Cork encoding
Autoren:	EuroT _E X-Konferenz in Cork, Irland
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	—
Beispielzeichensatz:	ecrm1000
Referenz:	[7, S. 514], [9, S. 99]

Die Cork-Kodierung wurde entwickelt, um alle Vorteile der damals neuen Version 3 von T_EX auszunützen. Durch sie wird automatische Worttrennung für die meisten westeuropäischen (und einige osteuropäische) Sprachen ermöglicht, wobei eine speziell modifizierte Version von T_EX nicht notwendig ist.

T1 wurde konstruiert, ohne auf vorhandene Zeichensätze zurückzugreifen, und erst später gab es tatsächlich verfügbare METAFONT-Schriften (die sogenannten EC-Fonts) und Implementationen mittels virtueller Zeichensätze (allerdings nicht alle Glyphen). Letztere bilden die Glyphen entweder auf K_NU_TH_S originale, OT1-kodierte Schriften oder

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	`	´	^	~	¨	˜	ˆ	ˇ	˘	˙	˚	¸	ˆ	˜	˘	˙
1	“	”	„	«	»	—	—		°	¹	ª	«	¬	®	¯	°
2	¡	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	-
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ä	Å	Ć	Č	Ď	Ě	Ė	Ğ	Ĭ	Ĺ	Ł	Ń	Ň	Đ	Ŕ	Ŗ
9	Ř	Ś	Ŝ	Ş	Ť	Ț	Ũ	Û	Ÿ	Ž	Ż	Ž	ı	ı	đ	š
A	ă	ą	ć	č	ď	ě	ė	ğ	ĭ	ĺ	ł	ń	ň	đ	ŕ	ŗ
B	ř	ś	ŝ	ş	ť	ț	ũ	û	ÿ	ž	ż	ž	ı	ı	ı	ı
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	ø	ù	ú	û	ü	ý	þ	þ	þ

Tabelle 6: Die T1-Kodierung von ecrm1000.

auf kommerzielle Zeichensätze, welche die 224 Glyphen des *Standard-Glyphsets* von Adobe enthalten, ab.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Г	Ғ	Ђ	Ѓ	Һ	Җ	Ӷ	Љ	Ӣ	Қ	Ҡ	К	Æ	Ң	Н	Ѕ
9	Ө	Ҫ	Ӣ	Ү	Ү	Ҳ	Ц	Ч	Ч	€	Ә	Ӣ	Ӣ	№	Ѡ	§
A	г	ғ	ђ	ѓ	һ	җ	ӷ	љ	ӥ	қ	ҡ	к	æ	ң	н	ѕ
B	ө	ҫ	ӥ	ү	ү	ҳ	ц	ч	ч	€	ә	ӥ	ӥ	„	«	»
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Tabelle 7: Die obere Hälfte der T2A-Kodierung von larm1000.

Name in L ^A T _E X:	T2A, T2B, T2C
Kodierungsname:	Cyrillic encodings
Autoren:	Das CyrTUG Font-Team
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	—
Beispielzeichensatz:	larm1000, lbrm1000, lcrm1000
Referenz:	[20]

Die T2-Kodierungen benutzen Glyphen aus dem X2-Container (s. u.) – es gibt zu viele kyrillische Zeichen, als dass sie mit einer einzigen L^AT_EX-konformen T-Kodierung abgedeckt werden können, daher die Aufteilung in drei Kodierungen. Die ausgewählten Glyphsets wurden nach linguistischen Gesichtspunkten erstellt, um verschiedene Sprachen vollständig abdecken zu können. Aus diesem Grund ist z.B. das kyrillische Standard-Alphabet (für Russisch) in allen drei Kodierungen enthalten.

Die unteren Hälften der T2-Kodierungen sind mit der unteren Hälfte von T1 identisch.

In einem späteren Abschnitt wird genauer beschrieben, welche Eigenschaften eine L^AT_EX-konforme Kodierung erfüllen muss.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ғ	Ғ	Ғ	Ғ	h	Ж	δ	З	Љ	Қ	Д	Қ	Ј	Н	Н	Н
9	Ө	С	Ў	У	Х	Х	Х	Ч	Ч	Н	Ә	Е	Ё	№	¤	§
A	ғ	ғ	ғ	ғ	h	ж	δ	з	љ	қ	д	қ	ј	н	н	н
B	ө	с	ў	у	х	х	х	ч	ч	н	ә	е	ё	„	«	»
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	a	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Tabelle 8: Die obere Hälfte der T2B-Kodierung von lbrm1000.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	П	Ц	Т	Б	h	Р	Р	З	М	Қ	Д	К	Ј	Н	М	Н
9	Ө	е	е	Б	Й	Х	Ц	Q	Ч	Н	Ә	Р	Ё	№	¤	§
A	п	ц	т	б	h	р	р	з	м	қ	д	к	ј	н	м	н
B	ө	е	е	б	й	х	ц	q	ч	н	ә	р	ё	„	«	»
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	a	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Tabelle 9: Die obere Hälfte der T2C-Kodierung von lcrm1000.

Name in $\text{\LaTeX}$:	T3
Kodierungsname:	IPA encoding
Autor:	FUKUI Rei
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	—
Beispielzeichensatz:	tipa10
Referenz:	[10, S. 102]

Die **T3**-Kodierung (und die dazugehörigen Makros) stellt Glyphen für phonetische Umschrift entsprechend den Empfehlungen der IPA (International Phonetic Association) zur Verfügung.

T3 ist *keine* $\text{\LaTeX}$ -konforme Kodierung der T-Serie! Der Name hat historische Gründe – die richtige Bezeichnung wäre **X3** – und wird aus Kompatibilitätsgründen, bedingt durch die weite Verbreitung dieser Schriftfamilie, nicht mehr geändert.

Name in $\text{\LaTeX}$:	T4
Kodierungsname:	African Latin (fc)
Autor:	Jörg KNAPPEN
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	0x24
Beispielzeichensatz:	fcr10
Referenz:	[8, S. 104]

Die untere Hälfte von **T4** ist identisch mit der unteren Hälfte von **T1**; die obere Hälfte enthält Buchstaben und Symbole für afrikanische Sprachen, welche das lateinische Alphabet benützen.

Aus Platzgründen teilen sich **\$** und **£** eine Glyphposition; ein Anwender sollte also stets `\textdollar` und `\textsterling` benützen.

Anstelle einer größeren Anzahl von Makros für spezielle Buchstaben gibt es drei virtuelle, akzentähnliche Kontrollsequenzen: `\m` (Modifizierer 1), `\M` (Modifizierer 2) und `\B` (Balken). Details dazu finden sich in Anhang C der Datei `fc.rme`. Alle Standard- $\text{\LaTeX}$ -Kontrollsequenzen wie `\L` (**Ł**) oder `\AE` (**Æ**) funktionieren, nur die speziellen Buchstaben für Isländisch, nämlich `\TH` (**Þ**), `\th` (**þ**) und `\dh` (**ð**) sind nicht verfügbar und werden durch »Annäherungen« aus **T4** (**T**, **t** und **d** mit Balken) ersetzt.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	`	´	^	~	¨	”	°	˘	˙	–	·	ˆ	˚	”	˘	˙
1	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	J	+	+	+	+	+
2	/	!	!	!	!	!	!	!	()	*	+	,	-	.	/	
3	⌘	ı	Λ	3	ı	ı	ı	ı	ı	ı	ı	ı	ı	=	ı	?
4	ə	ɑ	β	ε	ð	ε	ϕ	γ	h	ı	j	ı	ı	ı	ı	ı
5	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	[	ı	ı	ı	ı
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z			ı	ı	ı
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
9	/	/			↓	↑	↗	↘	˘	˙	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
A	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
B	ω	Ω	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
C	A	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
D	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
E	B	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
F	N	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı

Tabelle 10: Die T3-Kodierung von tipa10.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ɓ	Ɗ	Ɛ	Ɛ	Ƒ	Ǝ	Ƨ	ƨ	Ƙ	ƞ	Ɵ	Ɲ	ƿ	Ƭ	Ʈ	Ƴ
9	Ɔ	Ƣ	Š	Ň	Ņ	Ş	Ʒ	Ʀ	È	Ě	Ƨ	Ƨ	Ƨ	Ƨ	Ƨ	Ƨ
A	Ɓ	Ɗ	Ɛ	Ɛ	Ƒ	Ǝ	Ƨ	ƨ	Ƙ	ƞ	Ɵ	Ɲ	ƿ	Ƭ	Ʈ	Ƴ
B	Ɔ	Ƣ	Š	Ň	Ņ	Ş	Ʒ	Ʀ	È	Ě	Ƨ	Ƨ	Ƨ	Ƨ	Ƨ	Ƨ
C	Ɓ	Ɗ	Ɛ	Ɛ	Ƒ	Ǝ	Ƨ	ƨ	Ƙ	ƞ	Ɵ	Ɲ	ƿ	Ƭ	Ʈ	Ƴ
D	Ɔ	Ƣ	Š	Ň	Ņ	Ş	Ʒ	Ʀ	È	Ě	Ƨ	Ƨ	Ƨ	Ƨ	Ƨ	Ƨ
E	Ɓ	Ɗ	Ɛ	Ɛ	Ƒ	Ǝ	Ƨ	ƨ	Ƙ	ƞ	Ɵ	Ɲ	ƿ	Ƭ	Ʈ	Ƴ
F	Ɔ	Ƣ	Š	Ň	Ņ	Ş	Ʒ	Ʀ	È	Ě	Ƨ	Ƨ	Ƨ	Ƨ	Ƨ	Ƨ

Tabelle 11: Die obere Hälfte der T4-Kodierung von fcr10.

Name in $\text{\LaTeX}$:	T5
Kodierungsname:	Vietnamese text encoding
Autoren:	Werner LEMBERG und Vladimir VOLOVICH
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	—
Beispielzeichensatz:	vnr10
Referenz:	[2]

T5 deckt Vietnamesisch ab. Dass diese Kodierung trotz der großen Zahl von Buchstaben außerhalb des erlaubten Bereichs für Glyphpositionen normaler T-Kodierungen nicht X5 heißt, verdankt sie einer Besonderheit der vietnamesischen Sprache: Es gibt keine Worttrennungen. Dadurch kann auf Kompatibilität mit den durch T1 vorgegebenen Werten von `\lccode` und `\uccode` (s. u.) verzichtet werden. Intern werden alle Buchstaben mit diakritischen Zeichen und Akzenten auf $\text{\LaTeX}$ -Makros abgebildet, sodass `\MakeUppercase` und `\MakeLowercase` korrekt funktionieren.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	`	´	^	~	¨	.	°	˘	˙	–	·	„	”	,	<	>
1	“	”	„	«	»	–	—		°	ı	Ÿ	ŷ	Ỳ	ỳ	Đ	đ
2	˘	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	-
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	À	Á	Ã	Ả	Ạ	Â	À	Á	Ã	Ả	Ạ	Ă	À	Á	Ã	Ả
9	Ằ	È	É	Ẽ	Ẻ	Ẹ	Ê	È	É	Ẽ	Ẻ	Ệ	Ì	Í	Ĩ	Ỉ
A	à	á	ã	ả	ạ	â	à	á	ã	ả	ạ	ă	à	á	ã	ả
B	ă	è	é	ẽ	ẻ	ẹ	ê	è	é	ẽ	ẻ	ệ	ì	í	ĩ	ỉ
c	Ị	Ò	Ó	Õ	Ỏ	Ô	Ồ	Ố	Ỗ	Ổ	Ỗ	Ợ	Ơ	Ờ	Ớ	Ỡ
D	Ỡ	Ợ	Ừ	Ú	Ủ	Ự	Ư	Ừ	Ú	Ủ	Ự	Ự	Ỡ	Ỡ	Ỡ	Ỡ
E	ị	ò	ó	õ	ỏ	ô	ồ	ố	ỗ	ổ	ỗ	ộ	ơ	ờ	ớ	ỡ
F	ỡ	ợ	ù	ú	ủ	ự	ư	ừ	ứ	ữ	ự	ự	ỳ	ý	ỹ	ỹ

Tabelle 12: Die T5-Kodierung von vnr10.

Name in $\text{\LaTeX}$:	T6
Kodierungsname:	Reserviert für Armenisch
Autor:	—
Verwendete Glyphindizes:	—
Variable Positionen:	—
Beispielzeichensatz:	—
Referenz:	—

T6 ist für eine zukünftige Erweiterung von armenischen Zeichensätzen reserviert, die derzeit **OT6** benutzen.

Name in $\text{\LaTeX}$:	T7
Kodierungsname:	Reserviert für Griechisch
Autor:	—
Verwendete Glyphindizes:	—
Variable Positionen:	—
Beispielzeichensatz:	—
Referenz:	—

T7 ist für eine offizielle griechische Kodierung reserviert (die meisten Fonts benutzen derzeit **LGR**).

Mathematische Zeichensatzkodierungen mit 128⁺ Glyphen

Name in $\text{\LaTeX}$:	OML
Kodierungsname:	$\text{\TeX}$ math italic
Autor:	Donald Erwin KNUTH
Verwendete Glyphindizes:	0x00-0x7F
Variable Positionen:	—
Beispielzeichensatz:	cmmi10
Referenz:	[4, S. 430]

OML enthält kursive lateinische und griechische Buchstaben mit mathematischer Funktion (in erster Linie für Variablennamen) sowie einige weitere Symbole.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	Φ	Ψ	Ω	α	β	γ	δ	ϵ
1	ζ	η	θ	ι	κ	λ	μ	ν	ξ	π	ρ	σ	τ	υ	ϕ	χ
2	ψ	ω	ε	ϑ	ϖ	ϱ	ς	φ	$\leftarrow$	$\rightharpoonup$	$\rightarrow$	$\mapsto$	$\circ$	$\circ$	$\triangleright$	$\triangleleft$
3	o	1	2	3	4	5	6	7	8	9	.	,	<	/	>	*
4	∂	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	b	h	#	~	^
6	ℓ	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	i	j	$\wp$	$\rightarrow$	$\sim$

Tabelle 13: Die OML-Kodierung von `cmmi10`.

Name in $\text{\LaTeX}$: OMS
Kodierungsname: $\text{\TeX}$ math symbol
Autor: Donald Erwin KNUTH
Verwendete Glyphindizes: 0x00-0x7F
Variable Positionen: —
Beispielzeichensatz: `cmsy10`
Referenz: [4, S.431]

Mathematische Symbole in unveränderlicher Größe sowie kalligraphische lateinische Großbuchstaben (für Mengen und ähnliches) sind als OMS kodiert.

Name in $\text{\LaTeX}$: OMX
Kodierungsname: $\text{\TeX}$ math extension
Autor: Donald Erwin KNUTH
Verwendete Glyphindizes: 0x00-0x7F
Variable Positionen: —
Beispielzeichensatz: `cmex10`
Referenz: [4, S.432]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	−	⋅	×	*	÷	◊	±	∓	⊕	⊖	⊗	⊘	⊙	◯	◦	●
1	≠	≡	⊆	⊇	≤	≥	≲	≳	∼	≈	⊂	⊃	⋈	⋉	⋊	⋌
2	←	→	↑	↓	↔	↗	↘	≈	⇐	⇒	↑↑	↓↓	↔	↖	↗	∞
3	/	∞	∈	∋	△	▽	/		∀	∃	¬	∅	ℜ	ℑ	⊤	⊥
4	ℵ	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>	<i>K</i>	<i>L</i>	<i>M</i>	<i>N</i>	<i>O</i>
5	<i>P</i>	<i>Q</i>	<i>R</i>	<i>S</i>	<i>T</i>	<i>U</i>	<i>V</i>	<i>W</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	∪	∩	⊕	∧	∨
6	⊢	⊣	⊤	⊥	⌈	⌋	{	}	⟨	⟩			↕	↕	\	?
7	√	∏	∇	∫	⊔	⊓	⊆	⊇	§	†	‡	¶	♣	♦	♥	♠

Tabelle 14: Die OMS-Kodierung von cmsy10.

OMS kodiert mathematische Symbole mit veränderlicher Größe. Hierhin gehören Symbole wie das $\sum$ -Zeichen sowie die Bausteine für große runde und eckige Klammern u. ä.

Mathematische Zeichensatzkodierungen mit 256 Glyphen

Wie in einem früheren Abschnitt bereits besprochen, gibt es derzeit noch keine mathematischen Zeichensatzkodierungen mit 256 Glyphen.

Allgemeine Zeichensatzkodierungen für Text

Hierher gehören Kodierungen für Text, welche nicht die Beschränkungen der T-Serie erfüllen. Derzeit ist nur X2 definiert; wie schon weiter oben erwähnt, gehört eigentlich auch T3 in diese Kategorie.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	(	)	[	]	[	]	[	]	{	}	<	>			/	\
1	(	)	(	)	[	]	[	]	[	]	{	}	<	>	/	\
2	(	)	[	]	[	]	[	]	{	}	<	>	/	\	/	\
3	(	)	[	]	[	]			(	)	(	)	{	}	'	'
4	(	)	'	'	<	>	□	□	§	§	⊙	⊙	⊕	⊕	⊗	⊗
5	Σ	Π	∫	∪	∩	⊕	∧	∨	Σ	Π	∫	∪	∩	⊕	∧	∨
6	Π	Π	ˆ	ˆ	˜	˜	˜	˜	[	]	[	]	[	]	{	}
7	√	√	√	√	√		Γ		↑	↓	˘	˘	˘	˘	↗	↘

Tabelle 15: Die OMX-Kodierung von cmex10.

Name in $\text{\LaTeX}$:	X2
Kodierungsname:	Cyrillic glyph container
Autoren:	Das CyrTUG Font-Team
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	—
Beispielzeichensatz:	rxrm1000
Referenz:	[20]

Der Glyphcontainer für kyrillische Zeichen.

Zeichensatzkodierungen für Symbole in Text

Name in $\text{\LaTeX}$:	TS1
Kodierungsname:	Text Symbols
Autor:	Jörg KNAPPEN
Verwendete Glyphindizes:	0x00–0x0D, 0x12, 0x15, 0x16, 0x18–0x1D, 0x20, 0x24, 0x27, 0x2A, 0x2C–0x3A, 0x3C– 0x3E, 0x4D, 0x4F, 0x57, 0x5B, 0x5D–0x60, 0x62–0x64, 0x6C–0x6E, 0x7E–0xBF, 0xD6, 0xF6
Variable Positionen:	—
Beispielzeichensatz:	tcrml1000
Referenz:	[9]

TS1 stellt Symbole bereit, die oft in normalem Text vorkommen und deren Schnitt sich mit dem umgebenden Text gemeinsam ändert.

Wie bereits in einem früheren Abschnitt angesprochen, wurde **TS1** ohne Rücksicht auf bestehende kommerzielle Zeichensätze definiert, sodass nur speziell für $\text{\TeX}$ entwickelte Schriften alle Glyphen enthalten. In der Regel fehlen Nicht- $\text{\TeX}$ -Fonts rund die Hälfte der Zeichen, wie die Kodierungstabelle des Times-Roman PostScript-Fonts **ptmr8c** demonstriert. Das **textcomp**-Paket hilft hier weiter, indem es fünf Subkodierungen von **TS1** definiert und vielen der gängigen, populären Schriften eine davon bereits zuordnet: Wird nun ein Zeichen verwendet, das nicht in der passenden Subkodierung enthalten ist, stellt **textcomp** es zur Verfügung, indem es stillschweigend das gesuchte Zeichen aus einer anderen Schrift nimmt.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	`	´	^	~	¨	˘	˚	ˇ	˘	ˉ	˙	˚	˚	I	<	>
1	“	”	ˆ	˜	˘	—	—		˚	˘	˘	˘	δ	δ	Ж	ж
2	˘	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	Æ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ
5	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ
6	‘	æ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ
7	p	q	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г
9	Θ	Ç	Ÿ	Υ	Υ	Χ	Χ	Ч	Ч	€	Θ	€	€	№	⊠	§
A	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г
B	Θ	Ç	Ÿ	Υ	Υ	Χ	Χ	Ч	Ч	€	Θ	€	€	„	«	»
C	A	B	B	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	P	C	T	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	a	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	p	c	t	y	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Tabelle 16: Die X2-Kodierung von rxrm1000.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	˘	˙	ˆ	˜	¨	˘	˚	ˇ	˘	–	˙	˘	˙	˙		
1			”			—	—		←	→	ˆ	ˆ	ˆ	ˆ		
2	ˆ				\$			’			*		,	=	.	/
3	o	1	2	3	4	5	6	7	8	9			<	—	>	
4														U		○
5								Ω				⌈		⌋	↑	↓
6	˘		★	o o	†								☛	∞	♪	
7															~	=
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	˘	˘	”	”	†	†	⌈	% ₀₀	●	°C	\$	¢	f	©	W	N
9	G	P	£	R	†	‡	‡	™	% ₀₀₀	¶	B	Nº	%	e	o	SM
A	{	}	¢	£	¤	¥		§	¨	©	ª	©	¬	®	®	—
B	°	±	²	³	´	µ	¶	·	※	¹	º	√	¼	½	¾	€
C																
D							×									
E																
F							÷									

Tabelle 17: Die TS1-Kodierung von tcrm1000.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	`	´	^	~	¨	ˆ	°	ˇ	˘	–	·	‚	˙	‚		
1			„			—	—		■	■	■	■	■	■		
2	■				\$			'			*		,	■	.	/
3	■	■	■	■	■	■	■	■	■				■	—	■	
4														■		■
5								■				[		]	■	■
6	`		■	■	■								■	■	■	
7															~	■
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	˘	˘	ˆ	ˆ	†	‡		‰	•	°C	■	■	f	■	■	■
9	■	■	■	■	■	■	■	TM	■	■	■	■	■	■	■	■
A	■	■	¢	£	¤	¥	¦	§	¨	©	ª	■	¬	■	®	¯
B	°	±	²	³	´	µ	¶	·	■	¹	º	■	¼	½	¾	€
C																
D							×									
E																
F							÷									

Tabelle 18: Die T51-Kodierung von ptmr8c. Die schwarzen Rechtecke repräsentieren nichtvorhandene Glyphen.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	˘	˙	˚	˛	◻	←	↵	↓	↑	→	↔					
1																
2	æ	ɑ	ə	ɹ	ɕ	ɓ	ɔ	ɹ̥	ɹ̥	ɔ̥	ɛ̥	ɛ̥	ɣ̥	ɣ̥	ɣ̥	ɦ̥
3	ħ	ɹ̥	ɹ̥	ɹ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥	ɦ̥
4	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥
5	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥	ɹ̥
6																
7	˘	˙	˚	˛	◻	←	↵	↓	↑	→	↔					

Tabelle 19: Die TS3-Kodierung von tipx10.

Name in L ^A T _E X:	TS3
Kodierungsname:	IPA symbol encoding
Autor:	FUKUI Rei
Verwendete Glyphindizes:	0x00–0x0A, 0x20–0x49, 0x50–0x56, 0x70–0x7B
Variable Positionen:	—
Beispielzeichensatz:	tipx10
Referenz:	[10]

Die TS3-Kodierung (und die dazugehörigen Makros) stellt Glyphen für phonetische Umschrift entsprechend den Empfehlungen der IPA (International Phonetic Association) zur Verfügung. Sie wird vom tipa-Paket verwendet.

Andere Zeichensatzkodierungen

Name in L ^A T _E X:	PD1
Kodierungsname:	PDF DocEncoding
Autor:	Adobe
Verwendete Glyphindizes:	0x08-0x0A, 0x0C, 0x0D, 0x18-0x7E, 0x80-0x9E, 0xA0-0xAE, 0xB0-0xFF
Variable Positionen:	—
Beispielzeichensatz:	—
Referenz:	[11], [14]

PD1 ist eine virtuelle Kodierung mit 256 Zeichen, um Bookmarks u.ä. in PDF-Dokumenten mit pdfL^AT_EX zu erzeugen. Virtuuell deshalb, weil es (bewusst) keine Zeichensätze gibt, die PD1 abdecken. Details dazu finden sich im Anhang D.1 der PDF-Referenz.

Name in L ^A T _E X:	PU
Kodierungsname:	PDF Unicode Encoding
Autor:	Adobe
Verwendete Glyphindizes:	—
Variable Positionen:	—
Beispielzeichensatz:	—
Referenz:	[11], [14]

Eine weitere virtuelle Kodierung (mit mehr als 600 Zeichen) für unicode-kodierte Bookmarks in PDF-Dokumenten.

Name in L ^A T _E X:	C. .
Kodierungsname:	CJK Encoding
Autor:	Werner Lemberg
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	—
Beispielzeichensatz:	—
Referenz:	[3]

Das CJK-Paket definiert eine Reihe von Kodierungen, welche auf chinesische, japanische und koreanische Zeichensätze zugreifen.

Name in $\text{\LaTeX}$:	U
Kodierungsname:	Unknown Encoding
Autor:	—
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	alle
Beispielzeichensatz:	—
Referenz:	[15, S. 416]

Diese Kodierung sollte für Zeichensätze verwendet werden, die sich einer anderen Klassifizierung entziehen – im Besonderen für die Fälle, wo wohl nur ein einziger Font diese Kodierung besitzt.

Name in $\text{\LaTeX}$:	E . .
Kodierungsname:	Experimental Encoding
Autor:	—
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	alle
Beispielzeichensatz:	—
Referenz:	[15, S. 416]

Name in $\text{\LaTeX}$:	L . .
Kodierungsname:	Local Encoding (site dependent)
Autor:	—
Verwendete Glyphindizes:	0x00–0xFF
Variable Positionen:	alle
Beispielzeichensatz:	—
Referenz:	[15, S. 416]

Ursprünglich war vom $\text{\LaTeX}$ -Team beabsichtigt, dass L-Kodierungen nur lokal beschränkt eingesetzt werden. Jedoch haben sich einige davon als De-facto-Standards durchgesetzt und finden breite Verwendung, wie weiter unten beschrieben.

Name in $\LaTeX$: LY1
 Kodierungsname: Y&Y 256 glyph encoding
 Autor: Berthold Horn
 Verwendete Glyphindizes: 0x00–0x08, 0x0C, 0x10, 0x12–0xFF
 Variable Positionen: *wahrscheinlich keine*
 Beispielzeichensatz: **ptmr8y**
 Referenz: [15, S. 416]

Name in $\LaTeX$: LV1
 Kodierungsname: MicroPress encoding
 Autor: Michael Vulis
 Verwendete Glyphindizes: *nicht bekannt*
 Variable Positionen: *nicht bekannt*
 Beispielzeichensatz: —
 Referenz: [15, S. 416]

Name in $\LaTeX$: LGR
 Kodierungsname: Greek 256 glyph encoding
 Autor: *nicht bekannt*
 Verwendete Glyphindizes: 0x00–0xFF
 Variable Positionen: *wahrscheinlich keine*
 Beispielzeichensatz: **grmn1000**
 Referenz: [15, S. 575]

Diese Kodierung für Griechisch erfüllt übrigens nicht die Bedingungen für eine T-Kodierung, da die ASCII-Zeichen fehlen.

T-Kodierungen

In den folgenden Abschnitten werden die notwendigen Bedingungen von T-Kodierungen beschrieben, sodass ein sie benützender Zeichensatz als *allgemeiner Textfont* verwendet werden kann.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0					/	·	˘		fi				fi			
1	ı		`	´	˘	˙	-	˚	,	ß	æ	œ	ø	Æ	Œ	Ø
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	..
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ł	'	,	f	„	...	†	‡	^	‰	Š	<	Œ	Ž	^	-
9	ł	‘	’	“	”	•	-	—	~	™	š	>	œ	ž	~	ÿ
A		ı	¢	£	¤	¥	ı	§	¨	©	ª	«	¬		®	-
B	°	±	²	³	´	µ	¶	·	,	ı	°	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Tabelle 20: Die LY1-Kodierung von ptmr8y.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	—	˘	◻	◼	◽	◾	⸍	⸎	ı	A _I	H _I	Ω _I	A	Ÿ	α	ü
1	,	˘	ı	ı	˘	ı	Γ	Λ	€	% ₀₀	ə	Ḃ	‘	’	˘	—
2	˘	!	˘	˘	˘	%	˘	’	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	.	˘	=	˘	;
4	˘	A	B	˘	Δ	E	Φ	Γ	H	I	Θ	K	Λ	M	N	O
5	Π	X	P	Σ	T	Υ	˘	Ω	Ξ	Ψ	Z	[	˘	]	˘	˘
6	˘	α	β	ς	δ	ε	φ	γ	η	ι	θ	κ	λ	μ	ν	ο
7	π	χ	ρ	σ	τ	υ		ω	ξ	ψ	ζ	«	,	»	˘	—

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	ά	ά	ά	ά	ά	ά	ά	ά	ά	ά	ά	ά	ά	ά	ά	ά
9	ά	ά	ά	ƒ	ά	ά	ά	˘	ή	ή	ή		ή	ή	ή	
A	ή	ή	ή	ή	ή	ή	ή	ή	ή	ή	ή	ή	ή	ή	ή	ή
B	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ	ώ
C	ώ	ώ	ώ	F	ώ	ώ	ώ		ι	ι	ι	ι	υ	υ	υ	υ
D	ί	ί	ί	ι	ύ	ύ	ύ	υ	ι	ι	ι	Ι	υ	υ	υ	Υ
E	è	é	é	è	ò	ó	ó	ò	é	é	é	è	ó	ó	ó	ò
F	ı	ı	ı	ı	ü	ü	ü	ü	α	η	ω	ρ	ρ		’	,

Tabelle 21: Die LGR-Kodierung von grmn1000.

Benötigte Zeichen

Damit ein Zeichensatz als ein allgemeiner Textfont gelten kann, müssen bestimmte Zeichen an ganz bestimmten Positionen stehen. Ein solcher kann für beliebigen Text verwendet werden, ohne Einschränkung auf spezielle Umgebungen (wie z. B. das phonetische Alphabet) oder auf bestimmte Zeichen (wie z. B. die TS1-Kodierung).

Zeichen	Position	Zeichen	Position	Zeichen ¹	Position
!	33	:	58	<	60
,	39	;	59	>	62
(	40	=	61		124
)	41	?	63		
*	42	@	64		
+	43	A – Z	65 – 90		
,	44	[	91		
-	45	]	93		
.	46	‘	96		
/	47	a – z	97 – 122		
0 – 9	48 – 57				

Zusätzlich müssen die folgenden Zeichen irgendwo in der Kodierung zusammen mit den Ligaturprogrammen in den zugehörigen TFM-Dateien vorkommen.²

Zeichen	Ligaturprogramm
“	“ “
”	” ”
—	--
---	---

Insgesamt ergeben sich $33 + 2 \times 26 = 85$ benötigte Zeichen; 171 Positionen sind daher verfügbar.

Wenn in der Kodierung noch Platz frei sein sollte, ist es am besten, die übrig gebliebenen Positionen mit allen oder einigen Akzenten aufzufüllen. Ist dagegen zu wenig Platz, könnte man einige Nichtbuchstaben in eine

¹Diese drei Zeichen sind in OT-Kodierungen für lateinische Alphabete nicht an den erforderlichen Positionen.

²Da die Glyphen nicht direkt, sondern nur durch Ligaturen angesprochen werden, ist die exakte Position im Zeichensatz unwesentlich.

TS-Kodierung schieben, da ja nur Buchstaben (im $\text{\TeX}$ schen Sinne) durch den Trennalgorithmus beeinflusst werden.

Probleme mit `\lccode` und `\uccode`

Aufgrund technischer Beschränkungen in $\text{\TeX}$ s Trennalgorithmus ist es in $\text{\LaTeX}$ nicht möglich, jeweils mehr als eine Tabelle für `\lccode` und `\uccode` zu verwenden. Daher müssen alle T-Kodierungen die beiden Tabellen benützen, die für T1 definiert sind.

Unglücklicherweise hat T1 einige Besonderheiten, die bestimmte Positionen für viele andere Kodierungen praktisch wertlos machen, wenn man sich an diese Beschränkung hält. Da aber T1 so weit verbreitet und die Grundlage einer großen Anzahl von Sprachen ist, erschien es dem $\text{\LaTeX}$ -Team besser, sich mit dieser Situation abzufinden, als dass man T1 durch eine besser genormte Kodierung ersetzt, die dann auf lange Zeit Probleme mit vorhandenen $\text{\LaTeX}$ -Installationen macht, wo diese neue Kodierung und dazu passende Schriften nicht verfügbar sind.

Folgende Positionen sind problematisch:

- 25 (i) gleicher `\uccode`-Wert wie 105 (i)
- 26 (j) gleicher `\uccode`-Wert wie 106 (j)
- 27 (ff) `\lccode`-Wert ist ebenfalls 27, sodass die Position durch den Trennalgorithmus beeinflusst werden kann³
- 157 (İ) gleicher `\lccode`-Wert wie 73 (I)
- 158 (đ) gleicher `\uccode`-Wert wie 240 (đ)

Eine Möglichkeit zur Benutzung dieser Positionen besteht darin, sie mit Ligaturglyphen aufzufüllen, da $\text{\TeX}$ die `\uccode`- und `\lccode`-Tabellen nur für einzelne Zeichen benützt, nicht aber für die durch Ligaturprogramme erzeugten Zeichen.

Eine vollständige Liste aller `\uccode`/`\lccode`-Paare findet sich am Ende des Artikels.

³Dieser Wert ist zur Unterstützung der OT1-Kodierung definiert.

Kodierungsspezifische Befehle

Ein kodierungsspezifischer Befehl erzeugt ein Zeichen, das je nach Kodierung unterschiedlich implementiert sein kann (ohne dass sich der Benutzer darum kümmern muss). Details dazu finden sich in [15, S. 442].

Die folgenden beiden Tabellen zeigen die kodierungsspezifischen Befehle der T1- und OT1-Kodierung. In anderen Kodierungen können noch weitere Befehle dieser Art definiert sein (so z. B. `\h` für T5).

Hier die kodierungsspezifischen Befehle für diakritische Zeichen der T1- und OT1-Kodierung (alle Befehle benötigen ein Argument; so ist z. B. `\v{c}` die L^AT_EX-interne Darstellung von ›č‹):

<code>\‘</code>	OT1, T1	˘	Gravis
<code>\’</code>	OT1, T1	˙	Akut
<code>\^</code>	OT1, T1	ˆ	Zirkumflex
<code>\~</code>	OT1, T1	˜	Tilde
<code>\"</code>	OT1, T1	¨	Umlaut, Trema
<code>\H</code>	OT1, T1	˝	Doppelakut
<code>\r</code>	OT1, T1	°	Ring, Kroužek
<code>\v</code>	OT1, T1	ˇ	Háček, Caron
<code>\u</code>	OT1, T1	˘	Brevis
<code>\t</code>	OT1, T1	ˆ	Ligaturbogen (zwischen zwei Zeichen)
<code>\=</code>	OT1, T1	ˉ	Makron
<code>\.</code>	OT1, T1	˙	Punkt
<code>\b</code>	OT1, T1	ˉ	Unterstrich
<code>\c</code>	OT1, T1	¸	Cedilla
<code>\d</code>	OT1, T1	˙	Punkt darunter
<code>\k</code>	T1	˛	Ogonek

Und nun die übrigen kodierungsspezifischen Befehle der T1- und OT1-Kodierung:

<code>\AE</code>	OT1, T1	Æ
<code>\DH</code>	T1	Ð
<code>\DJ</code>	T1	Đ
<code>\L</code>	OT1, T1	Ł
<code>\NG</code>	T1	Ŋ
<code>\OE</code>	OT1, T1	Œ

<code>\O</code>	OT1, T1	Ø
<code>\SS</code>	OT1, T1	Š
<code>\TH</code>	T1	Þ
<code>\ae</code>	OT1, T1	æ
<code>\dh</code>	T1	ð
<code>\dj</code>	T1	đ
<code>\guillemotleft</code>	T1	« (guillemet)
<code>\guillemotright</code>	T1	» (guillemet)
<code>\guilsinglleft</code>	T1	‹ (guillemet)
<code>\guilsinglright</code>	T1	› (guillemet)
<code>\i</code>	OT1, T1	ı
<code>\j</code>	OT1, T1	ĵ
<code>\l</code>	OT1, T1	ł
<code>\ng</code>	T1	ŋ
<code>\oe</code>	OT1, T1	œ
<code>\o</code>	OT1, T1	ø
<code>\quotedblbase</code>	T1	„
<code>\quotesinglbase</code>	T1	,
<code>\ss</code>	OT1, T1	ß
<code>\textasciicircum</code>	OT1, T1	^
<code>\textasciitilde</code>	OT1, T1	~
<code>\textbackslash</code>	OT1, T1	\
<code>\textbar</code>	OT1, T1	
<code>\textbraceleft</code>	OT1, T1	{
<code>\textbraceright</code>	OT1, T1	}
<code>\textcompwordmark</code>	OT1, T1	(nicht sichtbar)
<code>\textdollar</code>	OT1, T1	\$
<code>\textemdash</code>	OT1, T1	—
<code>\textendash</code>	OT1, T1	–
<code>\textexclamdown</code>	OT1, T1	ı
<code>\textgreater</code>	OT1, T1	>
<code>\textless</code>	OT1, T1	<
<code>\textquestiondown</code>	OT1, T1	¿
<code>\textquotedbl</code>	T1	"
<code>\textquotedblleft</code>	OT1, T1	“
<code>\textquotedblright</code>	OT1, T1	”
<code>\textquoteleft</code>	OT1, T1	‘
<code>\textquoteright</code>	OT1, T1	’
<code>\textregistered</code>	OT1, T1	®

<code>\textsection</code>	OT1, T1	§
<code>\textsterling</code>	OT1, T1	£
<code>\texttrademark</code>	OT1, T1	™
<code>\textunderscore</code>	OT1, T1	—
<code>\textvisiblespace</code>	OT1, T1	□
<code>\th</code>	T1	þ

Literatur

- [1] *The Unicode Standard*. <http://unicode.org>.
- [2] *Die vnr-Schriftfamilie*, entwickelt von Hàn Thê Thành, dem Autor von pdf $\TeX$. <http://vntex.org/download/vntex>.
- [3] *Das CJK-Paket*: <http://cjk.ffii.org>.
- [4] Donald E. KNUTH: *The $\TeX$ book*. Band A von *Computers & Typesetting*, Mai 1989.
- [5] Barbara BEETON: *Mathematical symbols and cyrillic fonts ready for distribution*. In: TUGBoat, Jahrgang 6 (Heft 3), 1985. <http://tug.org/TUGboat/Articles/tb06-3/tb13beetcyr.pdf>.
- [6] Dean GUENTHER und Janene WINTER: *An international phonetic alphabet*. In: TUGBoat, Jahrgang 12 (Heft 1), 1991.
- [7] Michael J. FERGUSON: *Report on multilingual activities*. In: TUGBoat, Jahrgang 11 (Heft 4), 1990.
- [8] Jörg KNAPPEN: *Fonts for Africa: The fc Fonts*. In: TUGboat, Jahrgang 14 (Heft 2), 1993.
- [9] Jörg KNAPPEN: *The dc fonts 1.3: Move towards stability and completeness*. In: TUGBoat, Jahrgang 17 (Heft 2), 1996. <http://www.tug.org/TUGboat/Articles/tb17-2/tb51knap.pdf>.
- [10] FUKUI Rei: *TIPA: A system for processing phonetic symbols in $\LaTeX$* . In: TUGBoat, Jahrgang 17 (Heft 2), 1996. <http://www.tug.org/TUGboat/Articles/tb17-2/tb51rei.pdf>.
- [11] *PDF reference*: Adobe portable document format version 1.6. Adobe Systems Incorporated, 2005. <http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf>.

- [12] *Das fontinst-Paket*: CTAN:/fonts/utilities/fontinst.
- [13] *Die lh-Fonts für Cyrillich*: CTAN:/fonts/cyrillic/lh.
- [14] *Das hyperref-Paket*: <http://www.tug.org/applications/hyperref>.
- [15] Frank MITTELBACH und Michel GOOSSENS: *The $\LaTeX$ Companion second edition*. Mit Johannes BRAAMS, David CARLISLE, and Chris ROWLEY. Addison-Wesley, Reading, Massachusetts, 2004.
- [16] Justin ZIEGLER, *Technical Report on Math Font Encodings*, June 1994, <http://www.tug.org/twg/mfg/papers/ltx3pub/l3d007.ps.gz>.
- [17] Matthias CLASEN and Ulrik VIETH: *Towards a new Math Font Encoding for (La) $\TeX$* . March 1998, <http://www.tug.org/twg/mfg/papers/current/mfg-euro-all.ps.gz>.
- [18] Matthias CLASEN: *A new implementation of $\LaTeX$ math*, 1997-98. <http://www.tug.org/twg/mfg/papers/current/newmath.ps.gz>.
- [19] Barbara BEETON: *Unicode and math, a combination whose time has come – Finally!*. In: TUGBoat, Jahrgang 21 (Heft 3), 2000. <http://www.tug.org/TUGboat/Articles/tb21-3/tb68beet.pdf>.
- [20] A. BERDNIKOV, O. LAPKO, M. KOLODIN, A. JANISHEVSKY und A. BURYKIN: *The Encoding Paradigm in $\LaTeX$ 2 ϵ and the Projected X $\LaTeX$ 2 Encoding for Cyrillic Texts*. Euro $\TeX$ 98. <http://www.gutenberg.eu.org/pub/GUTenberg/publicationsPDF/28-29-berdnikova.pdf>.

$\backslash$ lccode- und $\backslash$ uccode-Tabelle

Die folgende Tabelle zeigt die Werte für $\backslash$ lowercase und $\backslash$ uppercase aller Positionen der T-Kodierungen:

Pos	Die Position in der Kodierung (0–255)
lc	Der Wert der <code>\lowercase</code> -Tabelle an dieser Position (falls der Wert 0 ist, hat <code>\lowercase</code> keine Funktion für dieses Zeichen, und der Trennalgorithmus wird nicht angewendet)
uc	Der Wert der <code>\uppercase</code> -Tabelle an dieser Position (falls der Wert 0 ist, hat <code>\uppercase</code> keine Funktion für dieses Zeichen)
Zeichen	Die Zeichen der T1-Kodierung an dieser Posi- tion, dargestellt als $\langle \text{Zeichen} \rangle (\langle \text{Kleinbuchstabe} \rangle /$ $\langle \text{Großbuchstabe} \rangle)$

Pos	lc	uc	Zeichen	Pos	lc	uc	Zeichen	Pos	lc	uc	Zeichen	Pos	lc	uc	Zeichen
0	0	0	` (-/-)	32	0	0	□ (-/-)	64	0	0	@ (-/-)	96	0	0	' (-/-)
1	0	0	´ (-/-)	33	0	0	! (-/-)	65	97	65	A (a/A)	97	97	65	a (a/A)
2	0	0	^ (-/-)	34	0	0	" (-/-)	66	98	66	B (b/B)	98	98	66	b (b/B)
3	0	0	˘ (-/-)	35	0	0	# (-/-)	67	99	67	C (c/C)	99	99	67	c (c/C)
4	0	0	¨ (-/-)	36	0	0	\$ (-/-)	68	100	68	D (d/D)	100	100	68	d (d/D)
5	0	0	” (-/-)	37	0	0	% (-/-)	69	101	69	E (e/E)	101	101	69	e (e/E)
6	0	0	° (-/-)	38	0	0	& (-/-)	70	102	70	F (f/F)	102	102	70	f (f/F)
7	0	0	˘ (-/-)	39	0	0	' (-/-)	71	103	71	G (g/G)	103	103	71	g (g/G)
8	0	0	˘ (-/-)	40	0	0	((-/-)	72	104	72	H (h/H)	104	104	72	h (h/H)
9	0	0	- (-/-)	41	0	0	) (-/-)	73	105	73	I (i/I)	105	105	73	i (i/I)
10	0	0	· (-/-)	42	0	0	* (-/-)	74	106	74	J (j/J)	106	106	74	j (j/J)
11	0	0	˘ (-/-)	43	0	0	+ (-/-)	75	107	75	K (k/K)	107	107	75	k (k/K)
12	0	0	˘ (-/-)	44	0	0	, (-/-)	76	108	76	L (l/L)	108	108	76	l (l/L)
13	0	0	, (-/-)	45	45	0	- (-/-)	77	109	77	M (m/M)	109	109	77	m (m/M)
14	0	0	˘ (-/-)	46	0	0	. (-/-)	78	110	78	N (n/N)	110	110	78	n (n/N)
15	0	0	˘ (-/-)	47	47	0	/ (-/-)	79	111	79	O (o/O)	111	111	79	o (o/O)
16	0	0	“ (-/-)	48	0	0	0 (-/-)	80	112	80	P (p/P)	112	112	80	p (p/P)
17	0	0	” (-/-)	49	0	0	1 (-/-)	81	113	81	Q (q/Q)	113	113	81	q (q/Q)
18	0	0	„ (-/-)	50	0	0	2 (-/-)	82	114	82	R (r/R)	114	114	82	r (r/R)
19	0	0	« (-/-)	51	0	0	3 (-/-)	83	115	83	S (s/S)	115	115	83	s (s/S)
20	0	0	» (-/-)	52	0	0	4 (-/-)	84	116	84	T (t/T)	116	116	84	t (t/T)
21	0	0	- (-/-)	53	0	0	5 (-/-)	85	117	85	U (u/U)	117	117	85	u (u/U)
22	0	0	— (-/-)	54	0	0	6 (-/-)	86	118	86	V (v/V)	118	118	86	v (v/V)
23	23	0	(/-)	55	0	0	7 (-/-)	87	119	87	W (w/W)	119	119	87	w (w/W)
24	0	0	o (-/-)	56	0	0	8 (-/-)	88	120	88	X (x/X)	120	120	88	x (x/X)
25	25	73	ı (ı/I)	57	0	0	9 (-/-)	89	121	89	Y (y/Y)	121	121	89	y (y/Y)
26	26	74	ı (ı/J)	58	0	0	: (-/-)	90	122	90	Z (z/Z)	122	122	90	z (z/Z)
27	0	0	ff (-/-)	59	0	0	; (-/-)	91	0	0	[(-/-)	123	0	0	{ (-/-)
28	0	0	fi (-/-)	60	0	0	< (-/-)	92	0	0	\ (-/-)	124	0	0	(-/-)
29	0	0	fl (-/-)	61	0	0	= (-/-)	93	0	0	] (-/-)	125	0	0	} (-/-)
30	0	0	ffi (-/-)	62	0	0	> (-/-)	94	0	0	^ (-/-)	126	122	0	~ (z/-)
31	0	0	ffl (-/-)	63	0	0	? (-/-)	95	0	0	_ (-/-)	127	127	0	- (-/-)

Pos	lc	uc	Zeichen	Pos	lc	uc	Zeichen	Pos	lc	uc	Zeichen	Pos	lc	uc	Zeichen
128	160	128	Ä (ä/Ä)	160	160	128	ä (ä/Ä)	192	224	192	À (à/À)	224	224	192	à (à/À)
129	161	129	Å (å/Å)	161	161	129	å (å/Å)	193	225	193	Á (á/Á)	225	225	193	á (á/Á)
130	162	130	Ć (ć/Ć)	162	162	130	ć (ć/Ć)	194	226	194	Â (â/Â)	226	226	194	â (â/Â)
131	163	131	Č (č/Č)	163	163	131	č (č/Č)	195	227	195	Ã (ã/Ã)	227	227	195	ã (ã/Ã)
132	164	132	Ď (ď/Ď)	164	164	132	ď (ď/Ď)	196	228	196	Ä (ä/Ä)	228	228	196	ä (ä/Ä)
133	165	133	Ě (ě/Ě)	165	165	133	ě (ě/Ě)	197	229	197	Å (å/Å)	229	229	197	å (å/Å)
134	166	134	Ě (ě/Ě)	166	166	134	ě (ě/Ě)	198	230	198	Æ (æ/Æ)	230	230	198	æ (æ/Æ)
135	167	135	Ĝ (ĝ/Ĝ)	167	167	135	ĝ (ĝ/Ĝ)	199	231	199	Ç (ç/Ç)	231	231	199	ç (ç/Ç)
136	168	136	Ĺ (ĺ/Ĺ)	168	168	136	ĺ (ĺ/Ĺ)	200	232	200	È (è/È)	232	232	200	è (è/È)
137	169	137	Ł (ł/Ł)	169	169	137	ł (ł/Ł)	201	233	201	É (é/É)	233	233	201	é (é/É)
138	170	138	Ł (ł/Ł)	170	170	138	ł (ł/Ł)	202	234	202	Ê (ê/Ê)	234	234	202	ê (ê/Ê)
139	171	139	Ń (ń/Ń)	171	171	139	ń (ń/Ń)	203	235	203	Ë (ë/Ë)	235	235	203	ë (ë/Ë)
140	172	140	Ň (ň/Ň)	172	172	140	ň (ň/Ň)	204	236	204	Ì (ì/Ì)	236	236	204	ì (ì/Ì)
141	173	141	Đ (đ/Đ)	173	173	141	đ (đ/Đ)	205	237	205	Í (í/Í)	237	237	205	í (í/Í)
142	174	142	Ö (ö/Ö)	174	174	142	ö (ö/Ö)	206	238	206	Î (î/Î)	238	238	206	î (î/Î)
143	175	143	Ř (ř/Ř)	175	175	143	ř (ř/Ř)	207	239	207	Ï (ï/Ï)	239	239	207	ï (ï/Ï)
144	176	144	Ř (ř/Ř)	176	176	144	ř (ř/Ř)	208	240	208	Ð (ð/Ð)	240	240	208	ð (ð/Ð)
145	177	145	Š (š/Š)	177	177	145	š (š/Š)	209	241	209	Ñ (ñ/Ñ)	241	241	209	ñ (ñ/Ñ)
146	178	146	Š (š/Š)	178	178	146	š (š/Š)	210	242	210	Ò (ò/Ò)	242	242	210	ò (ò/Ò)
147	179	147	Ŝ (ŝ/Ŝ)	179	179	147	ŝ (ŝ/Ŝ)	211	243	211	Ó (ó/Ó)	243	243	211	ó (ó/Ó)
148	180	148	Ť (ť/Ť)	180	180	148	ť (ť/Ť)	212	244	212	Ô (ô/Ô)	244	244	212	ô (ô/Ô)
149	181	149	Ť (ť/Ť)	181	181	149	ť (ť/Ť)	213	245	213	Õ (õ/Õ)	245	245	213	õ (õ/Õ)
150	182	150	Ů (ů/Ů)	182	182	150	ů (ů/Ů)	214	246	214	Ö (ö/Ö)	246	246	214	ö (ö/Ö)
151	183	151	Ů (ů/Ů)	183	183	151	ů (ů/Ů)	215	247	215	Œ (œ/Œ)	247	247	215	œ (œ/Œ)
152	184	152	Ÿ (ÿ/Ÿ)	184	184	152	ÿ (ÿ/Ÿ)	216	248	216	Ø (ø/Ø)	248	248	216	ø (ø/Ø)
153	185	153	Ž (ž/Ž)	185	185	153	ž (ž/Ž)	217	249	217	Ù (ù/Ù)	249	249	217	ù (ù/Ù)
154	186	154	Ž (ž/Ž)	186	186	154	ž (ž/Ž)	218	250	218	Ú (ú/Ú)	250	250	218	ú (ú/Ú)
155	187	155	Ž (ž/Ž)	187	187	155	ž (ž/Ž)	219	251	219	Û (û/Û)	251	251	219	û (û/Û)
156	188	156	IJ (ij/IJ)	188	188	156	ij (ij/IJ)	220	252	220	Ü (ü/Ü)	252	252	220	ü (ü/Ü)
157	105	157	İ (i/İ)	189	0	0	ı (ı/ı)	221	253	221	Ý (ý/Ý)	253	253	221	ý (ý/Ý)
158	158	208	đ (đ/Đ)	190	0	0	ĵ (ĵ/ĵ)	222	254	222	Þ (þ/Þ)	254	254	222	þ (þ/Þ)
159	0	0	§ (—/—)	191	0	0	£ (—/—)	223	255	223	Š (ß/Š)	255	255	223	ß (ß/Š)

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit Februar 2007 bis zum Redaktionsschluss vor. Die Liste folgt der umgekehrten chronologischen Reihenfolge. Bloße Updates werden nicht aufgeführt. Sie können auf der moderierten *tex-announce*-Mailingliste verfolgt werden, die auch unter <http://blog.gmane.org/gmane.comp.tex.ctan.announce> online verfügbar ist.

fge ist ein Font zum Setzen von *Freges »Grundgesetze der Arithmetik«* und zugehöriger Sekundärliteratur. Autor ist *J. J. Green*.

CTAN:fonts/fge

numname wurde aus der *memoir*-Klasse von *Peter Wilson* per *copy and paste* ausgegliedert und dient dazu, Zähler in die entsprechenden englischen Numeralia umzusetzen (z. B.: $1 \rightarrow one$). Auch Ordinalzahlen werden unterstützt (z. B.: $1 \rightarrow first$). Es kann mit jeder beliebigen L^AT_EX-Klasse verwendet werden. Autor ist *S. Joshua Swamidass*.

CTAN:macros/latex/contrib/numname

automata von *Gabriele Puppis* ist ein METAPOST-Paket zum Zeichnen von Graphen zur Automatentheorie.

CTAN:graphics/metapost/contrib/macros/automata

boldtensors stellt fette lateinische und griechische Zeichen innerhalb von `\mathversion{normal}` bereit, die mit vorangestellten `~` und `"` angesprochen werden. Autor ist *Werner Fink*.

CTAN:macros/latex/contrib/boldtensors

xfor von *Nicola Talbot* definiert `\""@for` neu, so dass die Schleife auch vorzeitig beendet werden kann.

CTAN:macros/latex/contrib/xfor

japanese dient dazu, japanische Texte mit L^AT_EX zu setzen. Das Paket stellt Befehle zur Verfügung, die denen des Pakets *babel* ähneln, das bisher keine Option für Japanisch enthält. Autor ist *»ING«*.

CTAN:language/japanese/japanese

`templates-sommer` enthält neue L^AT_EX-Vorlagen für Hausarbeiten von Völkerkundlern und für psychologische Diplomarbeiten, die für die Universität Jena entwickelt wurden. Autor ist *Jörg Sommer*. `templates-fenn` wurde nach CTAN: info/templates/fenn verlegt.

CTAN:info/templates/sommer

`oberdiek` enthält folgende neuen Pakete: `stringenc` kann einen String zwischen verschiedenen Kodierungen konvertieren. – `catchfile` extrahiert den Inhalt einer Datei und schreibt sie in ein Makro. – `selinput` wählt das Inputencoding aufgrund der Zeichen in der Eingabedatei und den Namen der zugehörigen Glyphen. Autor ist *Heiko Oberdiek*.

CTAN:macros/latex/contrib/oberdiek

`jj_game` von *D. P. Story* ist eine L^AT_EX-Klasse zum Setzen von Jeopardy-Spielstellungen.

CTAN:macros/latex/contrib/jj_game

`unroman` von *Joseph Wright* dient dazu, römische in arabische Zahlen umzuwandeln.

CTAN:macros/latex/contrib/unroman

`trivfloat` hilft dabei, neue Float-Typen zu definieren. Autor ist *Joseph Wright*.

CTAN:macros/latex/contrib/trivfloat

`pdf-forms-tutorial` ist ein Tutorium auf Englisch und auf Deutsch zum Erstellen von PDF-Formularen mit `pdflatex`, `hyperref` und `insdljs`. Autor ist *Dirk Krause*.

CTAN:info/pdf-forms-tutorial

`synproof` ist ein `pstricks`-Paket zum Zeichnen von *syntactic proofs* zur formalisierenden Darstellung der Syntax von logisch-semantischen Schlussregeln. Autor ist *Paul Isambert*.

CTAN:macros/latex/contrib/synproof

`gcard` erzeugt Grußkarten und plazierte den Text so, dass er auf Vorder- und Rückseite eines zweimal gefalteten DIN A4-Blattes richtig erscheint. Autor ist *George McBane*.

CTAN:macros/latex/contrib/gcard

`latexpaper` von *S. Brown* berechnet die L^AT_EX-Einstellungen und -Größen für beliebige Schrift- und Papierformate. Dies ist vor allem hilfreich, um Texte für Sehbehinderte zu setzen, es kann aber auch zum Setzen von Postern verwendet werden.

CTAN:support/latexpaper

`cleveref` formatiert Referenzen abhängig von der Art des Labels, auf das sie sich beziehen (Gleichung, Abschnitt usw.). Außerdem können mehrfache Bezüge als Liste ausgegeben werden. Autor ist *Toby Cubitt*.

CTAN:macros/latex/contrib/cleveref

songs hilft beim Gestalten von Gesangbüchern und Handouts zum Mitsingen. Autor ist *Kevin W. Hamlen*.

CTAN:macros/latex/contrib/songs

mlist dient dazu, Form und Inhalt bei Vektoren, Matrizen und Funktionen im Mathematiksatz zu trennen, so dass man auf einzelne Elemente oder Argumente von Ausdrücken Bezug nehmen kann. Autor ist *Will Robertson*.

CTAN:macros/latex/contrib/mlist

dlfltxb sind Makros und Vorlagen aus dem dänischen Buch »*Introduktion til L^AT_EX*« von *Lars Madsen*.

CTAN:macros/latex/contrib/dlfltxb

GFS Solomos ist ein neuer griechischer Font von *Antonis Tzolomitis*.

CTAN:fonts/greek/GFS/GFSSolomos

glossaries dient zur Erstellung von Glossaren und ersetzt das Paket **glossary**, das nunmehr obsolet geworden ist. Man kann damit eine Liste der Akronyme, die in einem Text vorkommen, sowie eines oder mehrere Glossare in einem Dokument erstellen. Das Glossar wird über ein beigefügtes Perl-Skript erzeugt. Autorin ist *Nicola Talbot*.

CTAN:macros/latex/contrib/glossaries

philosophersimprint ist eine Klasse von *Boris Veytsman* für Aufsätze in der gleichnamigen Zeitschrift <http://www.philosophersimprint.org/>.

CTAN:macros/latex/contrib/philosophersimprint

sdrt dient zum Zeichnen von Baumstrukturen nach der »[*Structured*] *Discourse Representation Theory*« ([S]DRT) sowie zum vereinfachten Zugriff auf mathematische Symbole, die darin ebenfalls benötigt werden. Autor ist *Paul Isambert*.

CTAN:macros/latex/contrib/sdr

verbatimcopy von *Lars Madsen* liest eine Textdatei und speichert deren Inhalt in einer anderen Datei (ohne Schreibschutz).

CTAN:macros/latex/contrib/verbatimcopy

aeb_tileb von *D. P. Story* dient dazu, den Seitenhintergrund mit einer rechteckigen Grafik in kachelartiger Weise auszufüllen.

CTAN:macros/latex/contrib/aeb_tilebg

aebenvelope baut auf dem jeweils neuesten **aeb_pro** (s. u.) und **acrotex** auf und dient dazu, PDF-Dateien zu erzeugen, die L^AT_EX-Dokumente als *eEnvelopes*-Anhänge enthalten. Dazu müssen Acrobat Pro ab Version 7 und Distiller verwendet werden. Autor ist *D. P. Story*.

CTAN:macros/latex/aeb_pro/aebenvelopepy

`aeb_pro` ergänzt das *AcroTeX eDucation Bundle* für Anwender von Acrobat Pro ab Version 7 und dient zum Aktivieren diverser PDF-Features. Autor ist *D. P. Story*.
 CTAN:macros/latex/contrib/aeb_pro/aeb_pro

`linux-libertine` enthält die Schriften aus dem gleichnamigen Projekt mit lateinischen, kyrillischen, griechischen und hebräischen Fonts sowie den dazugehörigen L^AT_EX-Support. Autor ist *Michael Niedermair*.
 CTAN:fonts/linuxlibertine

`etexcmds` von *Heiko Oberdiek* erweitert die Namen der neuen ε -T_EX-Primitiven, um Verwechslungen mit bestehenden Makros zu vermeiden. Beispielsweise wird `\inexpanded` in ε -T_EX hierdurch zu `\etex@unexpanded`.
 CTAN:macros/latex/contrib/oberdiek

`elatem` von *Philipp Lehman* ist eine Sammlung von L^AT_EX-Frontends zu den neuen ε -T_EX-Primitiven, die sich an Paketautoren richtet.
 CTAN:macros/latex/contrib/elatem

`leading` von *Will Robertson* dient zum Festlegen des Durchschusses mithilfe des Befehls `\linespread`. Der Zeilenabstand wird als absolute Länge angegeben, z. B. `\leading{5mm}`.
 CTAN:macros/latex/contrib/leading

`Math_into_LaTeX-4` sind die Beispiel-Dateien zur vierten Auflage des gleichnamigen Buchs von *George Grätzer*. Das Paket enthält auch eine kurze Einführung zu L^AT_EX – `Short_Course.pdf` – und eine Tabelle `SymbolTables.pdf` zu den Symbolen unter L^AT_EX.
 CTAN:info/examples/Math_into_LaTeX-4
 CTAN:info/Math_into_LaTeX-4/Short_Course.pdf
 CTAN:info/examples/Math_into_LaTeX-4/SymbolTables.pdf

`infwarerr` ist ein neues Paket von *Heiko Oberdiek*, das Makros für die Fehlermeldungen und Warnungen von plain-T_EX bereitstellt.
 CTAN:macros/latex/contrib/oberdiek

`gmeometric` ermöglicht es, den Befehl `\geometry` aus dem gleichnamigen Paket innerhalb eines Dokuments zu verwenden. Autor ist *Grzegorz Murzynowski*.
 CTAN:macros/latex/contrib/gmeometric

`inversepath` von *Will Robertson* berechnet inverse Dateipfade, z. B. von `eins/zwei/` nach `../..`.
 CTAN:macros/latex/contrib/inversepath

`simplecv` von *Jean-Marc Lasgouttes* ist eine einfache L^AT_EX-Klasse, mit der man seinen Lebenslauf schreiben kann.
 CTAN:macros/latex/contrib/simplecv

kpfonts ist eine neue Schriftart für den Text- und Mathematiksatz, einschließlich echter Kapitälchen und der $\mathcal{AMS}$ -Symbole. Autor ist *Christophe Caignaert*.

CTAN:fonts/kpfonts

xnewcommand erlaubt es, neue Befehle mit einem optionalen Argument zu definieren, die `\protected` und `\global` sind. Autor ist *Josselin Noirel*.

CTAN:macros/latex/contrib/xnewcommand

xoptarg dient dazu, Befehle, die ein optionales Argument annehmen, expandierbar zu machen. Autor ist *Josselin Noirel*.

CTAN:macros/latex/contrib/xoptarg

cmdstring stellt den Befehl `\cmdstring` bereit, der den Namen eines Befehls liefert. Autor ist *Josselin Noirel*.

CTAN:macros/latex/contrib/cmdstring

fenixpar von *Javier A. Múgica* enthält zwei Pakete: mit **fenixtok** kann man einem *token-register* Daten links oder rechts hinzufügen, die beim Ausführen des Registers automatisch wieder entfernt werden. Das Paket **fenixpar** stellt ein Interface zur Manipulation des `\everypar-token registers` bereit.

CTAN:macros/generic/fenixpar

Rezensionen

Joachim Schlosser: Wissenschaftliche Arbeiten mit $\text{\LaTeX}$ schreiben

Jürgen Fenn

Der Beitrag stellt die Neuerscheinung *Wissenschaftliche Arbeiten mit $\text{\LaTeX}$ schreiben. Leitfaden für Anfänger* von Joachim Schlosser (mitp Verlag, Readline: Heidelberg 2007) vor.

... Ich bin Student ... und muss innerhalb der nächsten 3 Monate meine Bachelorarbeit und kommendes Jahr meine Masterarbeit schreiben. Ein Kommilitone machte mich auf $\text{\LaTeX}$ aufmerksam. Nun meine Frage: ... reichen 2–3 Monate für die Einarbeitung in $\text{\LaTeX}$?...

E-Mail vom 6. März 2007 an d. Verf.

Die zunehmende zeitliche Verdichtung an den (Hoch-)Schulen fordert ihren Tribut. Auch der $\text{\LaTeX}$ -Anfänger ist infolgedessen ungeduldiger geworden. Wer eine wissenschaftliche Abschlussarbeit zu schreiben hat, dem ist eine Einführung von mehreren hundert [1, 5] oder gar über tausend Seiten [4] – sei sie noch so empfehlenswert – oftmals keine wirkliche Hilfestellung. Auch die Vielzahl von Anleitungen, die frei im Netz verfügbar sind – vgl. etwa die Übersicht bei [6] – hilft oftmals wenig weiter, weil auch deren Umfang beträchtlich ist: Allein die (noch) aktuelle PDF-Version der FAQ von DANTE e.V. zählt immerhin 189 Seiten.



Joachim Schlosser, der bisher vor allem durch seine hilfreiche Website [8] hervorgetreten ist, auf der er die Installation eines $\text{\TeX}$ -Systems einschließlich der ergänzenden Programme unter Microsoft Windows gut nachvollziehbar beschreibt, gelingt die insoweit geforderte Konzentration auf das Wesentliche recht gut. Trotz der Eingrenzung des Stoffs und der Zielgruppe, scheinen 300 Seiten bei dieser Aufgabenstellung aber immer noch das umfängliche Minimum zu sein. Das darf jedoch nicht abschrecken, denn der Text ist didaktisch gut gegliedert und leicht verständlich geschrieben, auch absolute Anfänger sollten keine Verständnisschwierigkeiten haben, solange sie elementare Kenntnisse im Umgang mit einem Computer haben und einen Editor bedienen können.

Schlosser spricht den absoluten Anfänger an, der – realistischerweise – mit einem MS Windows-System versehen, versuchen wird, seine ersten Schritte mit $\text{\LaTeX}$ beim Schreiben einer wissenschaftlichen Arbeit zu wagen.

Alle Elemente der Sprache $\text{\LaTeX}$ werden auf dem aktuellen Stand dargestellt, ohne weitschweifig oder allzu knapp zu werden. Man findet alles, was man üblicherweise für eine wissenschaftliche Arbeit benötigen dürfte. Angenehm fällt auf, dass der Autor einen modernen Wissenschaftsbegriff vertritt, der auch die Geisteswissenschaften einbezieht, so dass in dem Kapitel über $\text{\BibTeX}$ neben `natbib` auch das Ergänzungspaket `jurabib` vorgestellt wird (nicht aber das hochgelobte `biblatex`, das sich ja noch in Entwicklung befindet). Sehr hilfreich ist es, dass auch die Technik der logischen Auszeichnung von Texten ausführlich erklärt wird, einschließlich der Nutzung eigener Erweiterungen mit `\newcommand` und `\newenvironment`. Auch viele Features von `KOMA-Script` werden vorgestellt. Verwirren dürften den Anfänger lediglich kleine Inkonsistenzen, beispielsweise dass manchmal die `babel`-Option `german`, manchmal aber `ngerman` vorgeschlagen wird (vgl. S. 37 vs. S. 50).

Die beigelegte CD-ROM enthält die in dem Buch behandelten Beispiele in Form von lauffähigen Dateien zum Testen und Lernen, einschließlich einer ausführlich kommentierten Vorlage für eine wissenschaftliche Arbeit, die man an die eigenen Bedürfnisse anpassen kann.

Ärgerlich ist, dass dem Verlag bei der Produktion der CD-ROM ein Fehler unterlaufen ist, so dass der Installer nicht funktioniert. Vor allem für Anfänger dürfte das eine unüberwindbare Hürde für die Installation des beigelegten `proTeXt`-Systems darstellen. Hierfür steht zwar unter <http://schlosser.info/latexbuch.html> bzw. <http://www.mitp.de./vmi/mitp/detail/pWert/1633> ein Patch zum Download bereit. Unter dieser Verlags-URL wird auch eine E-Mail-

Adresse genannt, unter der man ersatzweise eine funktionierende CD-ROM kostenlos bestellen kann. Dem unbedarften Käufer bleiben dieser Mangel und seine Ursache aber zunächst verborgen, er wird lediglich feststellen, dass seine Installationsversuche fehlschlagen und womöglich schlussfolgern, dass $\text{\LaTeX}$ auf seinem System gar nicht funktioniert. Zwar weisen Verlag und Autor auf ihren Webseiten deutlich auf dieses Problem hin, wünschenswert wäre aber vor allem der Austausch der CDs in den Exemplaren, die sich im Handel befinden, oder zumindest, so ist es Brauch, Corrigenda als Beilage zum Buch. Überhaupt hätte man dem Band einen Korrektor gewünscht, gibt es doch bekanntlich nichts Schwierigeres, als einen selbstgeschriebenen Text Korrektur zu lesen.

Doch damit nicht genug: Auch der Index des Buchs ist bei der Produktion nicht richtig erstellt worden. Er zeigt einen Offset von zwei Seiten. Auch ein korrigierter Index ist deshalb – neben der Einleitung und dem Kapitel 2 zur Installation des $\text{\TeX}$ -Systems – auf den oben genannten Seiten herunterzuladen.

Diese technischen Mängel sollten indes zumindest die DANTE-Mitglieder vom Kauf des Buchs nicht abhalten, hat man doch ohnehin die jährliche $\text{\TeX}$ -Collection zur Hand, die aktueller als jede Buch-Beilage sein wird. Das Buch ist daher mit dieser Maßgabe für den im Titel genannten Anwenderkreis zu empfehlen, wobei der Leser evtl. auch die konkurrierenden Titel von *Detig* [2, leider derzeit im Buchhandel vergriffen], *Knappen* [3] und *Willms* [9] vergleichsweise heranziehen wird, die von Umfang und Preis her gesehen etwa in der gleichen Liga spielen und die schon bei früherer Gelegenheit besprochen worden sind.

Ob das, was *Schlosser* lehrt, freilich ausreicht, um eine bestimmte Arbeit zu schreiben, hängt von vielem ab – nicht zuletzt von dem Vermögen des Lesers, den Stoff, der ihm vorgetragen wird, einzuüben und darüber hinauszugehen, indem er $\text{\LaTeX}$ -Ressourcen heranzieht und selbständig weiterlernt; aber auch von den Anforderungen, die die eigene Arbeit stellt, und den Ansprüchen, die der Autor an die technische und typographische Gestaltung seines Manuskripts hat. Manches ist – umgekehrt gesehen – vielleicht auch etwas knapp ausgefallen, so dass man sich fragt, ob es nicht besser ganz entfallen wäre, beispielsweise die Seiten über $\text{\MetaPost}$ und $\text{\pgf/TikZ}$, die man in einem einführenden Buch dieses Umfangs ja gar nicht erwartet hätte, oder der Abschnitt zur Versionsverwaltung mit $\text{\cvs}$ und $\text{\svn}$. Ausführlicheres zu letzterem hat der Autor vor kurzem an dieser Stelle veröffentlicht. [7]

Wer *Schlossers* Pflichtstoff beherrscht, sollte jedenfalls für die allermeisten Aufgaben, wenn nicht für die eigene Kür, allemal gut gewappnet sein.

Joachim Schlosser. Wissenschaftliche Arbeiten schreiben mit $\LaTeX$. Leitfaden für Anfänger. mitp-Verlag, Redline: Heidelberg 2007, 287 Seiten, ISBN-13: 978-3-8266-1633-4, mit einer CD-ROM. Preis: 16,95 Euro (in Deutschland).

Literatur

- [1] Klaus Braune, Joachim Lammarsch und Marion Lammarsch: *$\LaTeX$. Basissystem, Layout, Formelsatz.*; Juli 2006; Besprechung in DTK 4/06, S. 58–60 und in c't 9/07, S. 222.
- [2] Christine Detig: *Der $\LaTeX$ -Wegweiser*; Mitp-Verlag; 2. Aufl.; Juli 2004; Besprechung in DTK 1/06, S. 57 ff.
- [3] Jörg Knappen: *Schnell ans Ziel mit $\LaTeX$ 2 ϵ* ; Oldenbourg; 2. Aufl.; Mai 2004; Besprechung in DTK 1/06, S. 48 ff. und in DTK 4/05, S. 66 ff.
- [4] Frank Mittelbach, Michel Goossens et al.: *Der $\LaTeX$ -Begleiter*; Pearson Studium; München; 2. Aufl.; Okt. 2005; überarbeitete und erweiterte Auflage; Besprechung in DTK 3/06, S. 59–68 und c't 2/06, S. 212.
- [5] Elke Niedermair und Michael Niedermair: *$\LaTeX$. Das Praxisbuch. Studienausgabe*; Franzis' Professional Series; Franzis; 3. Aufl.; 2006; Besprechung der ersten gebundenen Ausgabe in DTK 4/04, S. 45–51; zur zweiten Auflage vgl. DTK 2/05, S. 67 ff.
- [6] Matthias Pospiech: *$\LaTeX$ – ein Dokumentensatzsystem. Dokumentationen*; <http://www.matthiaspospiech.de/latex/dokumentation/>.
- [7] Joachim Schlosser: *Große Dokumente mit $\LaTeX$ – Versionierung, Varianten, Automatisierung und Änderungsverfolgung*; DTK; (2/07), S. 45–67; 2007.
- [8] Joachim Schlosser: *$\TeX/\LaTeX$ unter Windows*; 2007; <http://schlosser.info/latexsystem.html>.
- [9] Roland Willms: *$\LaTeX$ echt einfach, m. CD-ROM*; Franzis; 2005; Besprechung in DTK 3/04, S. 57 ff.

Spielplan

Termine

2007

15. 9. **Herbsttagung**
und 37. Mitgliederversammlung von DANTE e.V.
Universität Ulm
<http://www.dante.de/dante/events/mv37/>
13. 10. **GuiT 2007 meeting**
Pisa, Italien
<http://www.guit.sssup.it/GuITmeeting/2007/2007.en.php/>

2008

5. 3. – 7. 3. **DANTE 2008**
Frühjahrstagung und 38. Mitgliederversammlung
Friedrich-Schiller-Universität Jena
<http://www.dante.de/dante2008/>
21. 7. – 24. 7. **TUG 2008**
TeX's 30th birthday
Cork, Irland
<http://tug2008.ucc.ie/>

Linuxtag 2007 in Berlin





(Fotos: Karheinz Geyer)

TeX Users Group 2008 Conference

University College Cork
Cork, Ireland
21–24 July 2008
<http://tug2008.ucc.ie/>

TeX's 30th birthday
Interfaces to TeX
Workshops
Presentations



Hosted by the Human Factors Research Group (<http://hfrg.ucc.ie>)

Stammtische

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

Aachen

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

Berlin

Rolf Niepraschk
Tel.: 030/3 48 13 16
rolf.niepraschk@ptb.de
Freie Universität Berlin,
Silberlaube, Raum J 25/10 (Blizzard)
Habelschwerdter Allee 45
Zweiter Donnerstag im Monat, 19.00 Uhr

Bremen

Martin Schröder
Tel.: 04 21/2 23 94 25
martin@oneiros.de
Wechselnder Ort
Erster Donnerstag im Monat, 18.30 Uhr

Darmstadt

Karlheinz Geyer
geyerk.fv.tu@nds.tu-darmstadt.de
Restaurant Poseidon, Rheinstraße 41
64283 Darmstadt
Erster Freitag im Monat, ab 19.30 Uhr

Dresden

Carsten Vogel
lego@wh10.tu-dresden.de
Studentenwohnheim, Borsbergstraße 34,
Dresden, Ortsteil Striesen
ca. alle 8 Wochen, Donnerstag, 19.00 Uhr

Düsseldorf

Georg Verweyen
Georg.Verweyen@web.de
Bistro/Café Zicke
Böckerstr. 5 a (Ecke Bergerallee)
40213 Düsseldorf
Zweiter Mittwoch in ungeraden Monaten,
20.00 Uhr

Erlangen

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

Freiburg

Heiko Oberdiek
Tel.: 07 61/4 34 05
oberdiek@uni-freiburg.de
Wechselnder Ort
Dritter Donnerstag im Monat, 19.30 Uhr

Hamburg

Lothar Fröhling
lothar@thefroehlings.de
Zum Schwarzenberg
Schwarzenbergstr. 80 – 21073 HH
4. Dienstag im Monat, 19.30 Uhr

Hannover

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

Heidelberg

Luzia Dietsche
Tel.: 0 62 21/54 45 27
luzia.dietsche@urz.uni-heidelberg.de
»Pizza Garden«
Alte Glockengießerei 9
Letzter Mittwoch im Monat, 19.30 Uhr

Karlsruhe

Klaus Braune
Tel.: 07 21/6 08 40 31
braune@rz.uni-karlsruhe.de
Universität Karlsruhe, Rechenzentrum
Zirkel 2, 3. OG, Raum 316
Erster Donnerstag im Monat, 19.30 Uhr

Kiel

Karsten Heymann

karsten.heyman@gmx.de

Letzter Donnerstag im Monat, 19.00 Uhr

Ort siehe <http://zauberer.net/latexwiki>

Köln

Helmut Siegart

Institut für Kristallographie

Zölpicher Straße 49b

Letzter Mittwoch im Monat, 19.30 Uhr

München

Michael Niedermair

[http://www.uwe-siart.de/typografie/](http://www.uwe-siart.de/typografie/stammtisch.html)

[stammtisch.html](http://www.uwe-siart.de/typografie/stammtisch.html)

Gaststätte »Scheidegger«

Bauerstraße 16

Erster Mittwoch im Monat, 19.00 Uhr

Münster

Johannes Reese

reese@linguist.de

Gaststätte »Sabroso«

Mauritzstraße 19

nach Vereinbarung

Stuttgart

Bernd Raichle

bernd.raichle@gmx.de

Bar e Ristorante »Valle«

Geschwister-Scholl-Str. 3

Zweiter Dienstag im Monat, 19.30 Uhr

Trier

Martin Sievers

stammtisch-trier@texberatung.de

Fetzenkneipe (Haus Fetzenreich)

Sichelstraße 36 (beim Sieh-Um-Dich)

54290 Trier

Dritter Montag des Monats, 20.15 Uhr

Ulm

Adelheid Grob

adelan@heidi.in-ulm.de

<http://latex.in-ulm.de>

Gaststätte »Peppers Ulm«

Deinselsgasse 8

Erster Donnerstag im Monat, 19.30 Uhr

Wuppertal

Andreas Schrell

Tel.: 02193/53 10 93

as@schrell.de

Restaurant Croatia »Haus Johannisberg«

Südstraße 10

an der Schwimmoper Wuppertal-Elberfeld

Zweiter Donnerstag im Monat, 19.30 Uhr

Würzburg

Bastian Hepp

dante@sning.de

nach Vereinbarung

Zürich

Johannes Reese

reese@spw.unizh.ch

nach Vereinbarung



(Linuxtag 2007 – Fotos: Karheinz Geyer)

Adressen

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

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

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

Präsidium

Präsident:	Klaus Höppner	president@dante.de
Vizepräsident:	Volker RW Schaa	vice-president@dante.de
Schatzmeister:	Tobias Sterzl	treasurer@dante.de
Schriftführer:	Manfred Lotz	secretary@dante.de
Beisitzer:	Günter Partosch	
	Bernd Raichle	
	Herbert Voß	advisor@dante.de

Server

ftp: [ftp.dante.de](ftp://ftp.dante.de)
WWW: <http://www.dante.de/>

Autoren/Organisatoren

Robin Fairbairns Robin.Fairbairns@cl.cam.ac.uk	[22]	Klaus Höppner siehe Seite 74	[4]
Jürgen Fenn Friedensallee 174/20 63263 Neu-Isenburg juergen.fenn@gmx.de	[60, 65]	Werner Lemberg Kl. Beurhausstr. 1 44137 Dortmund wl@gnu.org	[22]
Michael Guravage NL-3632 NT Loenen aan de Vecht, The Netherlands	[5]	Frank Mittelbach Zedernweg 62 55128 Mainz Frank.Mittelbach@latex-project.org	[22]
Hans Hagen PRAGMA ADE, 8061 GH Hasselt, The Netherlands	[5, 20]	Volker RW Schaa siehe Seite 74	[4, 5]
Taco Hoekwater Elvenkind BV, NL-3311 GP Dordrecht, The Netherlands	[5]	Herbert Voß siehe Seite 74	[3]

Die T_EXnische Komödie

19. Jahrgang Heft 3/2007 August 2007

Impressum

Editorial

Hinter der Bühne

4 Grußwort

Von fremden Bühnen

5 Bericht über die EuroBach_TE_X 2007-Tagung

20 Lua_TE_X goes beta

Bretter, die die Welt bedeuten

22 Zeichensatzkodierungen von L^AT_EX

60 Neue Pakete auf CTAN

Rezensionen

65 Joachim Schlosser: Wissenschaftliche Arbeiten mit L^AT_EX schreiben

Spielplan

69 Termine

72 Stammtische

Adressen

75 Autoren/Organisatoren