

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

26. Jahrgang Heft 3/2014 August 2014

3/2014

Impressum

»Die $\text{\TeX}$ nische Komödie« ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Autoren wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nicht gestattet. Alle Rechte zur weiteren Verwendung außerhalb von DANTE e.V. liegen bei den jeweiligen Autoren.

Beiträge sollten in Standard- $\text{\LaTeX}$ -Quellcode unter Verwendung der Dokumentenklasse `dtk` oder `article` erstellt und per E-Mail oder Datenträger (CD/DVD) an untenstehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, $\text{\LaTeX}$ -Pakete oder Schriften notwendig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden. Weitere Informationen für Autoren findet man auf der Projektseite <http://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit Lua $\text{\TeX}$, Version beta-0.79.1 (TeX Live 2014) (rev 4971) erstellt. Als Standard-Schriften kamen Linux Libertine, Linux Biolinum, Luxi Mono und XITS Math zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2500

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung $\text{\TeX}$ e.V.
Postfach 10 18 40
69008 Heidelberg

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

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

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit:	Gert Ingold	Eberhard Lisse	Rolf Niepraschk
	Heiko Oberdiek	Christine Römer	Volker RW Schaa
	Gert Seidl		

Redaktionsschluss für Heft 4/2014: 15. Oktober 2014

ISSN 1434-5897

Editorial

Liebe Leserinnen und liebe Leser,

die dritte Ausgabe unserer Zeitschrift »Die T_EXnische Komödie« in diesem Jahr enthält traditionell die DVD »T_EX Collection«. Sie erscheint diesmal überpünktlich, um die Auslieferung der DVD nicht länger als nötig zu verzögern. Möglich war dies auch durch Artikel, die bereits länger vorliegen. Der Beitrag von Idris Samawi Hamid ist ein Bericht über ein von DANTE e.V. gefördertes Projekt. Derartige Projektberichte dienen vor allem auch zur Information über die Verwendung der Vereinsbeiträge. Da das Dokument selbst mit ConT_EXt gesetzt ist, musste es in die »Komödie« seitenweise per

```
\includegraphics[page=...,width=\textwidth,  
height=\textheight,keepaspectratio]{...}
```

eingebunden werden. Um den Unterschied zum aktuellen Layout so gering wie möglich zu halten, wurde beim ConT_EXt-Lauf zum einen dieselbe Schriftart (Libertine/Biolinum) und zum anderen dasselbe Seitenformat gewählt. Durch den Verzicht auf Kopf- und Fußzeilen und anschließender Anwendung von pdfcrop erhält man dann eine Ausgabe, die in den normalen Textbereich der »Komödienseiten« eingefügt werden konnte. Die Angaben für das Autoren- und Inhaltsverzeichnis erfolgten durch direkte Einträge in die entsprechenden externen Dateien.

Fußnoten sind ein traditionelles Mittel in (wissenschaftlichen) Arbeiten und spalten seit Ewigkeiten die typografisch interessierte Leserschaft. Für die Einen sind sie komplett überflüssig und für die Anderen ein willkommenes Stilmittel. Im T_EXbook findet sich am Ende des 15. Kapitels die mehrfach zitierte Anmerkung von Jill Knuth »Don't use footnotes in your books, Don.« (Jill Knuth – 1962). Nimmt man dagegen textkritische Editionen, beispielsweise die zu Immanuel Kant, so kommt man nicht um Fußnoten herum und landet sehr schnell beim Paket bigfoot. Jakob Wiersma beschreibt in seinem Beitrag die verschiedenen Pakete, die das Setzen von Fußnoten erleichtern.

Die von Jürgen Fenn seit Jahren regelmäßig erstellte Liste der neuen auf CTAN vorhandenen Pakete ist umfangreich wie eh und je. Steve Zakrzowsky und Christine Römer beschreiben jeweils eines der neueren Pakete.

Die Rubrik »Im Netz gefunden« enthält einen Beitrag, den ich seit zwei Jahren in meinem DTK-Ordner habe und der nun endlich den Weg in eine Ausgabe von »Die T_EXnische Komödie« gefunden hat. In dem von Marc Ensenbach

und Mark Trettin erstellten Dokument »l2tabu – Das L^AT_EX 2_ε-Sündenregister« (<http://www.ctan.org/pkg/l2tabu> oder Ausführen von »texdoc l2tabu«) wird empfohlen, den Befehl `\graphicspath` nicht zu verwenden, was bei der Erstellung der Komödie aber aus gegebenem Anlass regelmäßig geschieht. Dass diese Aussage bei den heutigen technischen Voraussetzungen der Hardware nicht mehr zweifelsfrei aufrecht erhalten werden kann, erläutert Markus Kohm in einer Stellungnahme.

Ich wünsche Ihnen wie immer viel Spaß beim Lesen und verbleibe
mit T_EXnischen Grüßen,

Ihr Herbert Voß

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

vor ein paar Tagen erst habe ich die druckfrische Ausgabe 2/2014 aus dem Briefkasten geholt. Nun sitze ich schon wieder an einem Grußwort und das noch vor dem üblichen Redaktionsschluss für die Ausgabe 3/2014. Der Grund ist sehr erfreulich: Die Ausgabe ist bereits fertig und auch die aktuelle $\text{\TeX}$ Collection liegt seit einigen Tagen zur Auslieferung bereit. Neben den aktuellen Versionen aller »modernen« $\text{\TeX}$ -Engines und Pakete, gibt es diesmal auch eine neue Version von $\text{\TeX}$ und METAFONT. Donald Knuth hat nach sechs Jahren die Liste von mehr als zwei Dutzend offener Punkte abgearbeitet. Eine ausführliche Erläuterung findet man unter <https://www.tug.org/TUGboat/tb35-1/tb109knut.pdf>.

Nachdem wir im letzten Jahr erstmals wieder selbst die Produktion organisiert und dabei andere europäische $\text{\TeX}$ -Nutzergruppen miteingebunden hatten, haben wir dieses erfolgreiche Modell in diesem Jahr erneut angewendet. Insgesamt elf Gruppen und damit nochmals zwei mehr als im Vorjahr haben sich diesmal an der Produktion beteiligt. Dadurch und durch das wieder einmal große Engagement vieler freiwilliger Helfer können wir so auch diesmal allen Mitgliedern sehr günstig »eine frische Scheibe $\text{\TeX}$ « anbieten. Zudem ist die DVD auch käuflich zu erwerben und trägt auf diese Weise zur Werbung für $\text{\TeX}$, aber auch für unseren Verein bei.

Ansonsten gibt es diesmal nicht allzuviel Neues zu berichten. Sehen Sie es mir also bitte nach, dass ich mich daher etwas kürzer fasse.

Meine Bitte um Anregungen für die Vorstandssitzung bleibt aktuell, ebenso wie die herzliche Einladung zur Herbsttagung am 20. September in Karlsruhe. Sowohl die Anmeldung zur Tagung als auch die Einreichung von Vorträgen ist über die Internetseite <http://www.dante.de/events/Herbst2014.html> möglich.

Abschließend wünsche ich Ihnen eine schöne Sommer- und Urlaubszeit sowie wie immer eine spannende Lektüre.

Herzlichst Ihr/Euer
Martin Sievers

Bretter, die die Welt bedeuten

Mehr Möglichkeiten mit Fußnoten

Jacob Wiersma

Es existieren schon längere Zeit verschiedene Pakete, mit denen sich einige Beschränkungen des Standardfußnotenmechanismus von L^AT_EX beheben lassen. In diesem Artikel werden die Pakete `bigfoot`, `manyfoot` und `footmisc` näher vorgestellt und einige Verbesserungsvorschläge erläutert.

Motivation

In einem meiner Bücher, an dem ich momentan arbeite, kommen manchmal auf einer Seite bis zu acht, meist kurze Fußnoten vor. Hier lässt der Standardfußnotenmechanismus das Dokument im Allgemeinen nicht gut aussehen. Abhilfe versprechen die drei oben erwähnten Pakete. Zumindest mehr oder weniger. Man braucht dennoch einige Tricks, um das gewünschte Ergebnis zu erreichen. Ich habe nicht vor, alle Möglichkeiten der Pakete zu besprechen. Dafür gibt es die (zugegeben manchmal nicht so üppigen) Dokumentationen der Pakete. Ich beschränke mich hier auf die Punkte, wo ich auf Probleme gestoßen bin – es sei dahingestellt, ob dies durch meine Unerfahrenheit kam.

Disclaimer

Ich bin L^AT_EX-Anwender und kein versierter L^AT_EX-Programmierer. Viele L^AT_EX-Befehle und fast alle T_EX-Befehle sind mir bisher relativ unbekannt geblieben. Die meisten Befehle, die ich im Laufe der Zeit selbst definiert habe, sind durch »trial & error« zustande gekommen. Ich habe noch nie L^AT_EX-Code veröffentlicht, und selbst wenn die *Die T_EXnische Komödie*-Redakteure die schlimmsten Fehler entdecken und korrigieren, nutzen Sie diese Beispiele auf eigene Gefahr.

Paket `bigfoot`

Das aktuelle T_EXLive bringt die Version 1.25 vom 15. 7. 2006 mit sich. Der Autor ist David Kastrup. [2] Die Dokumentation ist vielversprechend, denn sie sagt:

Purpose of this package is to provide a one-stop solution to almost all problems related to footnotes. You can use it as a drop-in replacement of the `manyfoot` package, but without many of its shortcomings, and quite a few features of its own. It uses the existing document class layouts for footnotes, so you can usually use it without having to worry about the looks.¹

Anhand der unten abgebildeten Datei testen wir, ob kurze Fußnoten in einer Zeile komprimiert werden können, was mit der Option `para` in Zeile 3 tatsächlich ohne Probleme gelingt.

```

1 \documentclass[a5paper,12pt]{article}
2 \usepackage{bigfoot}
3 \DeclareNewFootnote[para]{default}
4 \usepackage{perpage}
5 \MakePerPage{footnote}
6 \begin{document}
7 ~\vfill
8 \begin{minipage}{0.5\textwidth}
9 MPtest1\footnote{mp eins, S. 1}
10 \end{minipage}
11
12 test2\footnote{zwei}\footnote{auf S. 1}
13 \newpage
14 ~\vfill
15 test3\footnote{drei, S. 2}
16 test4\footnote{vier, S. 2}
17 \end{document}

```

MPtest1 ¹ test2 ²³	test3 ⁴ test4 ⁵
¹ mp eins, S. 1 ² zwei ³ auf S. 1	⁴ drei, S. 2 ⁵ vier, S. 2
1	2

Der obigen Ausgabe ist auch zu entnehmen, dass Fußnoten in einer `minipage`-Umgebung genauso wie andere Fußnoten unten auf der Seite erscheinen.

Die versprochene automatische Neunummerierung aber, die in Zeilen 4 und 5 der Datei angefordert wurde, fehlt dagegen. Die Ausgabe der zweiten Seite nummeriert die Fußnoten (4 und 5) einfach durch, statt wieder mit 1 anzufangen.

Der Grund dafür ist in Zeile 3 der Datei zu finden. Die Dokumentation verführt einen dazu, zu glauben, dass der Befehl

¹ Siehe [2, S. 1]

```
1 \DeclareNewFootnote[para]{default}
```

die Standardfußnote (die man mit `\footnote` initiiert) zu einer *para*-Fußnote umwandelt. Dies ist aber nicht der Fall. Um tatsächlich zu erreichen, dass die Standardfußnote zu einer *para*-Fußnote wird, ist folgender Code notwendig:

```
1 \documentclass[a5paper,12pt]{article}
2 \usepackage{bigfoot}
3 \DeclareNewFootnote[para]{default}
4 \usepackage{perpage}
5 \MakePerPage{footnotedefault}
6 \begin{document}
7 ~\vfill
8 \begin{minipage}{0.5\textwidth}
9 MPtest1\footnote{mp eins, S. 1}
10 \end{minipage}
11
12 test2\footnote{zwei}\footnote{auf S. 1}
13 \newpage
14 ~\vfill
15 test3\footnote{drei, Seite 2}
16 test4\footnote{vier, Seite 2}
17 \end{document}
```

Der Trick dabei ist, dass man die Standardeinstellungen des Pakets überschreibt. Dann ist die gewünschte Neunummerierung auf Seite 2 erreicht. Man beachte, wie bei diesem Paket die doppelte Fußnote der Zeile 12 im Text erscheint: Es sieht aus, als existiere nur eine Fußnote und zwar mit der Nummer 23.

MPtest1 ¹ test2 ²³	test3 ¹ test4 ²
¹ mp eins, S. 1 ² zwei ³ auf S. 1	¹ drei, Seite 2 ² vier, Seite 2
1	2

Weitere Funktionen

Das Paket `bigfoot` bietet viel mehr Möglichkeiten, als die bereits Erwähnten. Ich habe diese Möglichkeiten nicht ausprobiert, sie werden aber in der Paketdokumentation beschrieben. Kurz zusammengefasst betrifft es Folgendes:

- Man kann verschiedene Arten von Fußnoten definieren, deren Eigenschaften unabhängig voneinander sind.
- Man kann eine gewisse Sortierung der Fußnoten vornehmen.

- Man kann `\verb`-Konstrukte in Fußnoten aufnehmen.
- Es ist möglich, Farben in Fußnoten zu verwenden.
- Fußnoten können mathematische Ausdrücke enthalten.

Paket manyfoot

In das Testdokument laden wir jetzt das Paket `manyfoot`.

```

1 \documentclass[a5paper,12pt]{article}
2 \usepackage[para,perpage]{manyfoot}
3 \DeclareNewFootnote[para]{default}
4 \begin{document}
5 ~\vfill
6 \begin{minipage}{0.5\textwidth}
7 MPtest1\footnote{mp eins, S. 1}
8 \end{minipage}
9
10 test2\footnote{zwei}\footnote{auf S. 1}
11 \newpage
12 ~\vfill
13 test3\footnote{drei, S. 2}
14 test4\footnote{vier, S. 2}
15 \end{document}

```

Es ist in der Version 1.10 vom 11.9.2005 in $\text{\TeX}$ Live 2014 enthalten. Der Autor ist Alexander I. Rozhenko. [3] Für die *perpage*-Neunummerierung wird das `perpage`-Paket von David Kastrup benutzt, das auch beim `bigfoot`-Paket zum Einsatz kommt.

MPtest1 ^a ^a mp eins, S. 1 test2 ¹² ¹ zwei ² auf S. 1 1	test3 ³ test4 ⁴ ³ drei, S. 2 ⁴ vier, S. 2 2
--	--

In der Ausgabe oben fallen, außer der »alt aussehenden« doppelten Fußnote, noch ein paar Sachen auf:

- Fußnoten in einer `minipage`-Umgebung stehen direkt unter der Minipage mit *alpha*-Markierung, wie in $\text{\LaTeX}$ üblich.
- Die Fußnoten sind nicht in einer Zeile zusammengefügt.

- Die *perpage*-Neunummerierung zeigt keine Wirkung.

Dass augenscheinlich schon mal gar nichts funktioniert, erklärt sich dadurch, dass in der Zeile `\DeclareNewFootnote[para]{default}` im Gegensatz zum Paket `bigfoot`, mit *default* hier nicht die Standardfußnote des Befehls `\footnote` angegeben wird, sondern die neu kreierte Fußnote, die man mittels `\footnotedefault` anspricht. Um doch noch in den Genuss der versprochenen Eigenschaften zu kommen, kann man Folgendes machen:

```

1 \documentclass[a5paper,12pt]{article}
2 \usepackage[para,perpage]{manyfoot}
3 \DeclareNewFootnote[para]{X}
4 \renewcommand{\footnote}[1]{\footnoteX{#1}}
5 \begin{document}
6 ~\vfill
7 \begin{minipage}{\textwidth}
8 MPtest1\footnote{mp eins, S. 1}
9 \end{minipage}
10
11 test2\footnote{zwei}\footnote{auf S. 1}
12 \newpage
13 ~\vfill
14 test3\footnote{drei, S. 2}
15 test4\footnote{vier, S. 2}
16 \end{document}

```

Die große Ähnlichkeit mit dem Paket `bigfoot` ist offensichtlich. Lediglich das explizite Einbinden des Pakets `perpage` lässt sich hier über eine `manyfoot`-Option steuern. Von Vorteil könnte auch sein, dass die Neunummerierung durch *perpage* in diesem Fall für alle Fußnoten gilt. Nicht nur, dass der $\text{\LaTeX}$ -Code große Ähnlichkeit hat: Nach dem $\text{\LaTeX}$ -Durchlauf erhält man das gleiche Ergebnis wie bei Einbindung des Pakets `bigfoot`;

MPtest1 ¹ test2 ²³	test3 ¹ test4 ²
¹ mp eins, S. 1 ² zwei ³ auf S. 1	¹ drei, S. 2 ² vier, S. 2
1	2

Paket footmisc

Das letzte hier betrachtete Paket zur Fußnotensteuerung ist das Paket `footmisc` von Robin Fairbairns, das dem aktuellen $\text{\TeX}$ Live in der Version 5.5b vom 6. 6. 2011

beiliegt. [1] Es ist damit aktueller als die beiden vorher erwähnten Pakete und hat wesentlich mehr Funktionalität. Schauen wir uns das nächste Beispiel an:

```

1 \documentclass[a5paper,12pt]{article}
2 \usepackage[para,perpage]{footmisc}
3 \begin{document}
4 ~\vfill
5 \begin{minipage}{0.5\textwidth}
6 MPtest1\footnote{mp eins, S. 1}
7 \end{minipage}
8
9 test2\footnote{zwei}\footnote{auf S. 1}
10 \newpage
11 ~\vfill
12 test3\footnote{drei, S. 2}
13 test4\footnote{vier, S. 2}
14 \end{document}

```

Es fällt auf, dass die Anweisung `\DeclareNewFootnote[para]{default}` fehlt. Dieser Befehl ist tatsächlich in diesem Paket nicht mehr definiert. Die generierte Ausgabe ist unten zu sehen

MPtest1^a ^a mp eins, S. 1 test2¹² ¹ zwei ² auf S. 1 1	test3¹ test4² ¹ drei, S. 2 ² vier, S. 2 2
---	--

Es fällt positiv auf, dass sowohl die *para*- als auch die *perpage*-Option sofort ihre Wirkung zeigen. Minipages und doppelte Fußnoten bilden noch ein Problem, wobei man das letzte Übel mit einer Option namens *multiple* beseitigen kann, wie im nächsten Beispiel ersichtlich sein wird. Negativ aber – zumindest nach meinem Geschmack, worüber ich nicht streiten möchte – ist die gähnende Leere zwischen der Fußnotenmarkierung und dem Fußnotentext. Das Paket bietet jedoch keine Möglichkeit zur Steuerung dieser Lücke. Dieses »Übel« steckt im Befehl `\@makefn`text, wozu Robin Fairbairns bemerkt:

The standard classes set the footnote mark flush with the text of the footnote, but that's not appropriate for paragraph footnotes, we find.²

Die Lösung liegt in der Umdefinierung dieses Befehls, mittels:

² Siehe [1, S. 18]

```

1 \long\def\@makefnmark#1{\leavevmode\@makefnmark\nobreak
2 % \hskip.5em\relax
3 #1%
4 }

```

Wenn gewünscht, kann man natürlich auch andere Werte für den *hskip* benutzen, statt den Befehl zu unterdrücken. Das komplette Beispiel sieht jetzt so aus:

```

1 \documentclass[a5paper,12pt]{article}
2 \usepackage[multiple,para,perpage]{footmisc}
3 \makeatletter
4 \long\def\@makefnmark#1{\leavevmode\@makefnmark\nobreak #1}
5 \makeatother
6 \begin{document}
7 ~\vfill
8 \begin{minipage}{0.5\textwidth}
9 MPtest1\footnote{mp eins, S. 1}
10 \end{minipage}
11
12 test2\footnote{zwei}\footnote{auf S. 1}
13 \newpage
14 ~\vfill
15 test3\footnote{drei, S. 2}
16 test4\footnote{vier, S. 2}
17 \end{document}

```

Man beachte bitte die Option *multiple* in Zeile 2, die das Problem der doppelten Fußnoten löst. Es ergibt sich folgende Ausgabe:

MPtest1^a ^amp eins, S. 1 test2^{1,2} ¹zwei ²auf S. 1 1	test3¹ test4² ¹drei, S. 2 ²vier, S. 2 2
---	--

Weitere Funktionen

Das Paket *footmisc* hat viel mehr Optionen zu bieten als oben gezeigt. Einige davon sind:

- Option *side* setzt Fußnoten mittels `\marginpar`.
- Option *symbol* benutzt Symbole als Fußnoten-Markierung.

- Option *bottom* platziert die Fußnoten im Seitenfuß.
- Option *norule* unterdrückt die Fußnoten-Linie.
- Option *stable* ermöglicht Fußnoten z. B. im Kapitelkopf.

Fazit

Von den drei beschriebenen Paketen *bigfoot*, *manyfoot* und *footmisc* kommen meines Erachtens vor allem *bigfoot* und *footmisc* in Betracht. Abhängig von konkreten Wünschen wird man das eine oder andere Paket bevorzugen.

Ähnliche Pakete

Es gibt noch andere Pakete, die einem die Arbeit mit Fußnoten erleichtern. In einem kleinen Projekt, um *dynamische* Fußnoten zu realisieren, verwende ich z. B.

- *pagenotes* für Noten am Ende des Kapitels bzw. Dokuments.
- *tablefootnote* für Fußnoten in der Umgebung *table*.

Literatur

- [1] Robin Fairbairns: *footmisc – a portmanteau package customising footnotes in L^AT_EX*; 2011; CTAN:macros/latex/contrib/footmisc/; zuletzt besucht: 2014-04-17.
- [2] David Kastrup: *The bigfoot package*; 2006; CTAN:macros/latex/contrib/bigfoot/; zuletzt besucht: 2014-04-17.
- [3] Alexander I. Rozhenko: *The manyfoot package*; 2005; CTAN:macros/latex/contrib/ncctools/; zuletzt besucht: 2014-04-17.

Paket *skmath* für mathematische Formeln

Steve Zakrzowsky

Das $\text{\LaTeX}$ -Paket *skmath* wurde von Simon Sigurdhsson entwickelt, von dem auch einige spezielle Dokumentklassen stammen. Das Schreiben mathematischer Ausdrücke und Gleichungen kann schnell sehr unübersichtlich werden. Das Paket *skmath* stellt eine Erweiterung zur einfachen und intuitiven Schreibweise für mathematische Ausdrücke dar.

Einleitung

Das Paket *skmath* enthält einige neue und angepasste Kommandos für die Erzeugung von mathematischen Ausdrücken. Es soll dem Benutzer helfen, seine Formeln einfacher im $\text{\LaTeX}$ -Code zu definieren. Anfangs beabsichtigte der Autor Simon Sigurdhsson seine Erweiterung nicht für die Öffentlichkeit bereitzustellen, aber durch dessen Nützlichkeit entschied er sich dagegen. Ebenfalls kann der Autor keine Garantie geben, dass Fehler durch Inkompatibilitäten entstehen.

Einbindung

Zur Verwendung des Paketes wird es – wie üblich – in der Präambel eingebunden.

```
\usepackage[<Option>]{skmath}
```

skmath besitzt einen optionalen Parameter `commonsets`. Dieser ist standardmäßig deaktiviert (= `false`) und erlaubt bei Aktivierung die folgenden Kommandos für zusätzliche Zahlbereiche.

Verwendung

Neue Kommandos

Zahlbereiche

Das Paket *skmath* besitzt fünf einfache Befehle für die Anzeige von Zahlbereichen. Voraussetzung ist die erwähnte Aktivierung durch `commonsets`.

$\backslash\mathbb{N}$: $\mathbb{N}$ (natürliche Zahlen) $\backslash\mathbb{Z}$: $\mathbb{Z}$ (ganze Zahlen) $\backslash\mathbb{Q}$: $\mathbb{Q}$ (rationale Zahlen)
 $\backslash\mathbb{R}$: $\mathbb{R}$ (reelle Zahlen) $\backslash\mathbb{C}$: $\mathbb{C}$ (komplexe Zahlen)

Norm `\norm{<Ausdruck>}`

Im folgenden Abschnitt wird die Verwendung der mathematischen Norm von einem Ausdruck gezeigt. Das Kommando `\norm` hat den einzuschließenden Ausdruck als Argument. Zur Unterscheidung von verschiedenen Normen können sie mit einem Index mittels `_` versehen werden.

<code>\[\norm{x} \]</code>	$\ x\ $
-----------------------------	---------

Betragsfunktion `\abs{<Ausdruck>}`

Der Betrag eines Ausdrucks lässt sich mittels des Kommandos `\abs` visualisieren. Das Kommando fordert ebenfalls einen nicht-optionalen Parameter, nämlich den mathematischen Ausdruck, den es einzuschließen gilt.

<code>\[\abs{\vec{x} - \vec{y}} \]</code>	$ \vec{x} - \vec{y} $
--	-----------------------

Differential `\d{<Variable>}`

Die Angabe des Differentialteils eines Integrals wird mit dem Kommando `\d` realisiert. Als Argument erwartet das Kommando die einzuschließende Variable nach der differenziert wird.

<code>\[\int_{\mathbb{R}} \frac{\sin(x)}{x} \, \mathrm{d}x \]</code>	$\int_{\mathbb{R}} \frac{\sin(x)}{x} \, dx$
---	---

Partielle Ableitung `\pd*{<Funktion>}{<Variable>,<Variable>,...}`

Das Kommando `\pd` kann zur Beschreibung von partiellen Ableitungen genutzt werden. Die Variante mit dem Setzen des Sterns `*` ermöglicht das Schreiben von Ableitungen nach den gewünschten Variablen in der tiefer gestellten Schreibweise. Ohne den Stern im Kommando wird die partielle Ableitung in der Bruchschreibweise dargestellt. Dabei werden die beliebig vielen Variablen automatisch in der Ableitungssumme generiert.

<code>\[\pd*{f}{x,y} \]</code>	f_{xy}
<code>\[\pd{g}{x^i,y^j} \]</code>	$\frac{\partial^{i+j} g}{\partial x^i \partial y^j}$

Erwartungswert `\E{<Ausdruck>}`

Der Erwartungswert einer Variablen kann mit dem Makro `\E` geschrieben werden.

<code>\[\E{X} = \sum_{i=0}^I x_i p_i \]</code>	$E[X] = \sum_{i=0}^I x_i p_i$
---	-------------------------------

Wahrscheinlichkeit $\backslash P\{<Ausdruck>\given<Ausdruck>\}$

Wahrscheinlichkeiten können mit dem Kommando $\backslash P$ gesetzt werden. Das Angeben von bedingten Wahrscheinlichkeiten kann mittels des $\backslash given$ -Befehls im Argument des $\backslash P$ -Kommandos angegeben werden.

```
\[
  \P{A\given B} =
  \frac{\P{B\given A}\P{A}}{\P{B}}
\]
```

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

Varianz $\backslash var\{<Ausdruck>\}$ und Kovarianz $\backslash cov\{<Ausdruck>\}\{<Ausdruck>\}$

Die Varianz einer Variablen kann mittels des Befehls $\backslash var$ beschrieben werden, wobei der Ausdruck ein nicht-optionaler Parameter ist. Zur Angabe einer Kovarianz zweier Variablen kann das Makro $\backslash cov$ genutzt werden, das zwei Argumente erwartet.

```
\begin{align*}
  \var{X} &= \E{(X-\mu)^2}\\
  \cov{X}{Y} &= \E{XY} - \E{X}\E{Y}
\end{align*}
```

$$\begin{aligned} \text{Var}(X) &= E[(X - \mu)^2] \\ \text{Cov}(X, Y) &= E[XY] - E[X]E[Y] \end{aligned}$$

Erweiterte Kommandos

Das skmath-Paket bietet nicht nur neue Befehle für das Schreiben mathematischer Formeln, sondern auch eine Reihe neu definierter Kommandos.

Trigonometrische Funktionen

Tabelle 1: Zusammenstellung der trigonometrischen Funktionen.

<i>Trigonometrische Funktion</i>	<i>Syntax</i>
Sinus	$\backslash sin[(\text{Exponent})]\{<Ausdruck>\}$
Arkussinus	$\backslash arcsin\{<Ausdruck>\}$
Kosinus	$\backslash cos[(\text{Exponent})]\{<Ausdruck>\}$
Arkuskosinus	$\backslash arccos\{<Ausdruck>\}$
Tangens	$\backslash tan[(\text{Exponent})]\{<Ausdruck>\}$
Arkustangens	$\backslash arctan\{<Ausdruck>\}$
Kotangens	$\backslash cot[(\text{Exponent})]\{<Ausdruck>\}$
Sinus Hyperbolicus	$\backslash sinh[(\text{Exponent})]\{<Ausdruck>\}$
Kosinus Hyperbolicus	$\backslash cosh[(\text{Exponent})]\{<Ausdruck>\}$
Tangens Hyperbolicus	$\backslash tanh[(\text{Exponent})]\{<Ausdruck>\}$

Die neu definierten Befehle für die trigonometrischen Funktionen enthalten stets ein Argument für den einzuschließenden Ausdruck. Einige Makros können zusätzlich als optionalen Parameter einen Exponenten enthalten.

```
\begin{gather*}
  \sin[2]{\alpha}\\
  \arcsin{\beta}\\
  \cot[k]{\alpha + \beta}
\end{gather*}
```

$$\sin^2(\alpha)$$

$$\arcsin(\beta)$$

$$\cot^k(\alpha + \beta)$$

Natürlicher Logarithmus `\ln{<Ausdruck>}`

Zum Schreiben des natürlichen Logarithmus kann der neu definierte Befehl `\ln` genommen werden. Dieser erwartet einen beliebigen Ausdruck, der als Parameter für den natürlichen Logarithmus angegeben werden soll.

```
\[ \ln{e^x} = x \]
```

$$\ln(e^x) = x$$

Logarithmus `\log[<Basis>]{<Ausdruck>}`

Der Logarithmus kann mittels des Befehls `\log` geschrieben werden. Das Argument des Logarithmus muss durch die geschweiften Klammern angegeben werden. Dazu kann eine Basis im optionalen Parameter spezifiziert werden.

```
\[ \log[2]{x} \]
```

$$\log_2(x)$$

Exponentialfunktion `\exp{<Ausdruck>}`

Der Befehl `\exp` gibt die Exponentialfunktion in der kurzen Schreibweise `e` an. Das Argument wird dabei als Exponent ausgegeben. Die Sternvariante schreibt `exp` und klammert das Argument.

```
\begin{align*}
  \exp{2\pi i} &\&= 1\\
  \exp{x} &\&= \sum_{n=0}^{\infty} \frac{x^n}{n!}
\end{align*}
```

$$e^{2\pi i} = 1$$

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Minimum- und Maximum-Funktionen

Die Tabelle 2 zeigt alle möglichen Minimum- und Maximum-Funktionen im Paket `skmath`. Als Argument erwarten die Makros den Ausdruck, auf den die Funktion angewendet werden soll. Durch den optionalen Sternparameter lässt sich die Ausgabe der geschweiften Klammern unterdrücken. Zusätzlich kann durch einen weiteren optionalen Parameter ein Definitionsbereich angegeben werden, der zentriert unterhalb des Funktionsnamens angezeigt wird.

Tabelle 2: Zusammenstellung der Minimum- und Maximum-Funktionen.

Name	Syntax
Minimum	<code>\min*[\langle\text{Definitionsbereich}\rangle]\{\langle\text{Ausdruck}\rangle\}</code>
Argument des Minimums	<code>\argmin*[\langle\text{Definitionsbereich}\rangle]\{\langle\text{Ausdruck}\rangle\}</code>
Maximum	<code>\max*[\langle\text{Definitionsbereich}\rangle]\{\langle\text{Ausdruck}\rangle\}</code>
Argument des Maximums	<code>\argmax*[\langle\text{Definitionsbereich}\rangle]\{\langle\text{Ausdruck}\rangle\}</code>
Supremum	<code>\sup*[\langle\text{Definitionsbereich}\rangle]\{\langle\text{Ausdruck}\rangle\}</code>
Infimum	<code>\inf*[\langle\text{Definitionsbereich}\rangle]\{\langle\text{Ausdruck}\rangle\}</code>

```
\begin{gather*}
\min\{0,1,2,3\}\\
\min[x\in\mathbb{R}]{f(x)}\mid x<\pi\\
\argmax*[x\in\mathbb{N}]{g(x)}
\end{gather*}
```

$$\min\{0, 1, 2, 3\}$$

$$\min_{x \in \mathbb{R}} \{f(x) \mid x < \pi\}$$

$$\arg \max_{x \in \mathbb{N}} g(x)$$

Stil

Einige Kommandos wurden in ihrer Darstellung verändert, um mehr Möglichkeiten zur Darstellung mathematischer Formeln zu gewährleisten.

Bruchschreibweise `\frac{\langle\text{Zähler}\rangle}{\langle\text{Nenner}\rangle}`

Der Befehl `\frac` erlaubt nun in einigen Konstellationen das Einbinden von verschiedenen Darstellungsstilen für den Mathematikmodus in L^AT_EX.

`\bar`-Befehlsänderung `\bar{\langle\text{Ausdruck}\rangle}`

Mittels des `\bar`-Befehls lässt sich über dem als Parameter angegebenen Ausdruck eine waagrechte Linie zeichnen. Im Paket skmath wurde dieser Standardbefehl erweitert, indem sich die Linie nun an die Länge des Argumentes anpasst.

```
\[ \bar{XTR} \]
```

$$\overline{XTR}$$

Vektorschreibweise

Der bereits bestehende Befehl `\vec` wurde nach Standard ISO 80000-2 erweitert. Dieser gleicht nun dem Befehl `\vectorsym` aus dem Paket isomath.

DANTE Summary Report:

Introducing Arabic-Latin Modern Fixed

Idris Samawi Hamid

1 Background: The Oriental T_EX Project

The Oriental T_EX project was initiated in 2006 to facilitate the development of high quality typography and typesetting of academic and scholarly texts that require the Arabic script, such as critical editions and monographs. Although support for the Arabic script in modern typesetting software has been slowly improving over the past decade or so, the situation is still very far behind the Latin script in terms of features, available high-quality typefaces, and layout-processing software. For academic and scholarly work, it's still very much a wilderness out there. A full solution to the problems of advanced Arabic-script typography and typesetting, particularly one based on OpenType and Unicode standards, is still some ways off.

So far the Oriental T_EX project has worked closely with the LuaT_EX project, also initiated in 2006; an initial large-sum grant from Colorado State University in particular was an important boost to LuaT_EX development. In the last round of supplemental funding from DANTE (2009) we focused on the development, implementation, and testing of the OpenType layout engine in CONTEX_T's LuaT_EX-based MkIV (in close collaboration with Hans Hagen).¹ In this current round of funding through DANTE, we are focusing on a different aspect of the mission of Oriental T_EX, to be described in the next section.

2 The Problem

2.1 Editing Marks in an Abjad Writing System

Unlike Latin script, which is an *alphabetic* writing system, Arabic script is an *abjad* writing system. The key difference between the two is as follows:

An *alphabetic* writing system consists of *letters* that represent consonant sounds, as well as characters that represent vowel sounds. Thus in an alphabetic script each fundamental sound in a given language,² whether a consonant (e. g. ›b‹, ›ch‹) or a

¹ See »OpenType Engineering in T_EX«, *Die T_EXnische Komödie*, 4/2009.

² The technical term for such a fundamental sound is ›phoneme‹.

vowel sound (e. g., ›i‹, ›ai‹), is represented by one or more letters of the alphabet of that language.

On the other hand an *abjad*³ writing system is one where each letter represents a consonant sound, and most vowels are not represented by letters. Sometimes a consonant may function as a vowel in certain contexts in addition to its consonantal function (in Arabic the equivalent of the letters ›a‹, ›w‹, and ›y‹ do this), but most vowels remain implicit rather than explicit.

Thus a single letter string can be quite ambiguous and represent a number of different words. Furthermore, in the case of Arabic in particular the grammatical role of words in a sentence are determined by *inflection*, where the last letter of a word takes on a different sound depending on its context. In order to disambiguate words and inflection endings, early scholars of Arabic developed a system of diacritics and vowel markings to mark each pure consonant within a word. This was especially motivated by the case of the Qurʾān; within a generation of the passing of the Prophet of Islam the non-Arabic-speaking peoples who were converting to Islam in large numbers would frequently get especially the inflections incorrect, inadvertently affecting the meaning of Qurʾānic expressions in disastrous ways. For the purpose of precisely presenting a Classical Arabic text in a modern edition, getting the vowel markings correct is also crucial. However, entering the markings correctly for the purposes of typesetting has always been an arduous task. It is instructive to compare this with the situation in scholarly Greek.

Although its script constitutes an alphabetic system (as opposed to an abjad one), the precise representation of many aspirations (or »breathings«) and accents in Ancient and Hellenistic Greek came to be represented by a sophisticated system of diacritical markings. The entire Greek alphabetic script inclusive of the full system of diacritics is called *polytonic Greek*. In Unicode this system is represented by characters located within the Greek-Extended Block (U+1F00..U+1FFF). Most of these polytonic letters can be composed in Unicode from a combination of basic Greek consonants (located within the U+0370..U+03FF block) and other diacritics (mostly common to both Greek and Latin typography). For example, ›Ǻ‹ (U+1F08) can be decomposed into ›A‹ and ›'‹. This makes the digital entry of polytonic Greek text quite straightforward and easy to edit: One can easily proofread and edit a manuscript either

- a. by selecting and modifying each diacritic independent of the letter that it modifies (using decomposition); or
- b. by replacing one Unicode code point (e. g. ›Ǻ‹) with another (e. g. ›A‹).

³ The term ›abjad‹ comes from the first four letters of most Semitic languages, ›A‹, ›B‹, ›J‹, and ›D‹.

	ἑαυτὸν τούτων, καὶ τοιοῦτός ἐστιν οἷος ἀξιοῦν ἑαυτόν. οὗτος	
	ἐπαινετὸς καὶ μέσος τούτων.	4
15	ἐπεὶ οὖν περὶ τιμῆς αἵρεσιν καὶ	4
	χρήσιν καὶ τῶν ἄλλων ἀγαθῶν τῶν ἐντίμων ἀρίστη ἐστὶ δι-	5
	άθσεις ἢ μεγαλοψυχία, καὶ τοῦτ' ἀποδίδομεν, καὶ οὐ περὶ τὰ	
16	χρήσιμα, τῷ μεγαλοψύχῳ, ἅμα δὲ καὶ ἡ μεσότης αὕτη ἐπαι-	
	νετωτάτη· δῆλον ὅτι καὶ ἡ μεγαλοψυχία μεσότης ἂν εἴη, τῶν	
	δ' ἐναντίων, ὥσπερ διεγράψαμεν, ἡ μὲν ἐπὶ τὸ ἀξιοῦν ἑαυ-	
	τὸν ἀγαθῶν μεγάλων ἀνάξιον ὄντα χαυνότης (τοὺς τοιούτους	10
	γὰρ χαύνους λέγομεν, ὅσοι μεγάλων οἶονται ἄξιοι εἶναι οὐκ	
	ὄντες), ἡ δὲ περὶ τὸ ἄξιον ὄντα μὴ ἀξιοῦν ἑαυτὸν μεγάλων	
	μικροψυχία (μικροψύχου γὰρ εἶναι δοκεῖ, ὅστις ὑπαρχόντων	
	δι' ἃ δικαίως ἂν ἡξιούτο, μὴ ἀξιοῖ μηθενὸς μεγάλου ἑαυτόν),	
	ὥστ' ἀνάγκη καὶ τὴν μεγαλοψυχίαν εἶναι μεσότητα χαυ-	15
17	νότητος καὶ μικροψυχίας. ὁ δὲ τέταρτος τῶν διορισθέντων	
	οὔτε πάμπαν ψεκτὸς οὔτε μεγαλόψυχος, περὶ οὐδὲν ἔχον· ὦν	
	μέγεθος· οὔτε γὰρ ἄξιος οὔτε ἀξιοῖ μεγάλων, διὸ οὐκ ἐναν-	
18	τίος. καίτοι δόξειεν ἂν ἐναντίον εἶναι τῷ μεγάλων ἀξίῳ ὄντι	
	μεγάλων τὸ μικρῶν ὄντα ἄξιον μικρῶν ἀξιοῦν ἑαυτόν. οὐκ	20
	ἐστὶ δ' ἐναντίος οὔτε τῷ μὴ	
	1233 ^a 9–30 = EN 1125 ^a 16–34; cf. 1122 ^b 30–1123 ^a 18	
	3 οἷον ἀξιοῖ ἑαυτόν· Richards: οἷος ἀξιοῦν ἑαυτόν, codd.	6 <περὶ
	suppl. Walzer 7 τὰ post οὐ suppl. Fritzsche, τὸ Ross τὸν μεγαλόψυχον	
	codd.: τῷ μεγαλοψύχῳ Fritzsche 8 αὕτη secl. Rackham 10 ἐπὶ	
	codd. (cf. 22 ^a 23): ἐπεὶ Ald.: περὶ Ross (cl. 33 ^a 12) τῷ τῷ L	

Figure 1 Polytonic Greek in action (image provided by Thomas Schmitz).

Classical editions take the effort to properly represent polytonic Greek in all its glory (see Figure 1). Monotonic Greek (i. e., Greek script mostly stripped of the diacritics) is thus hardly a credible option in Classical scholarship, philology, or textual studies (except perhaps for certain narrow, specialized cases).

Unfortunately, in the Arabic script and language the situation is much more complicated. Not only is there an array of intra-word vowels plus inflections that have to be represented on most letters (by *marks*), but each letter can take on a multiple of shapes, with almost limitless possibilities. Representing each and every one of these possible letter-mark combinations by a single Unicode code point is unwieldy and impractical. A quick look at the (in my strong opinion) very ill-conceived Arabic Presentation Forms A (U+FB50..U+FDFF block) will let one view a non-exhaustive list of some possible letter-shape possibilities that is virtually unworkable for digital typesetting purposes. Now if one were to add all of the possible letter-mark combinations for each character of the two Presentation-Forms blocks (A and B,

for a total of 891 characters) the number of resultant code points would expand exponentially.

Even if we consider only the basic four Arabic-script shapes-per-letter needed for normal text entry into an editor (isolated, initial, medial, and final), managing separate code points for each possibility still becomes quite impractical. Hence the OpenType solution of entering marks separate from base letters, then combining them using OpenType rules which are then displayed by the layout engine. That layout engine may be within a text editor or a text processor such as T_EX.

But this is where the problem begins. OpenType rules specify *general positioning* (GPOS) rules for the *vertical* placement of marks over base letters. This is entirely correct and indispensable (see Figure 2). On the other hand, in a text editor this makes it very impractical and tiresome to proofread and correct vowels in an Arabic text of any substantial length. This is because in an editor one selects text, not vertically, but *horizontally*. Anyone who has tried to select and edit Arabic-script marks in an editor separately from the consonants knows what a pain it is. It is a two-dimensional task boxed into a one-dimensional standard of digital input with a cursor. No text editor or processor of which I am aware has ever been designed to handle two-dimensional input.



Figure 2 Mark on consonant in Arabic.

The above problem is not unique to digital typography. Lead-press typography as well had a hard time dealing with marked Arabic script. Ironically, the best examples of Arabic-script typesetting with marks over multiple shapes is found in early 20th century Arabic typography. But over the following decades the use of marks decreases, along with less and less shape variations of the letters themselves. This is due, no doubt, to the overall unwieldiness and growing expense of the process. So Arabic critical editions are rarely, if ever, done to the same degree of orthographical precision as a Classical-Greek edition (see Figure 3 for a standard example of an Arabic critical edition).

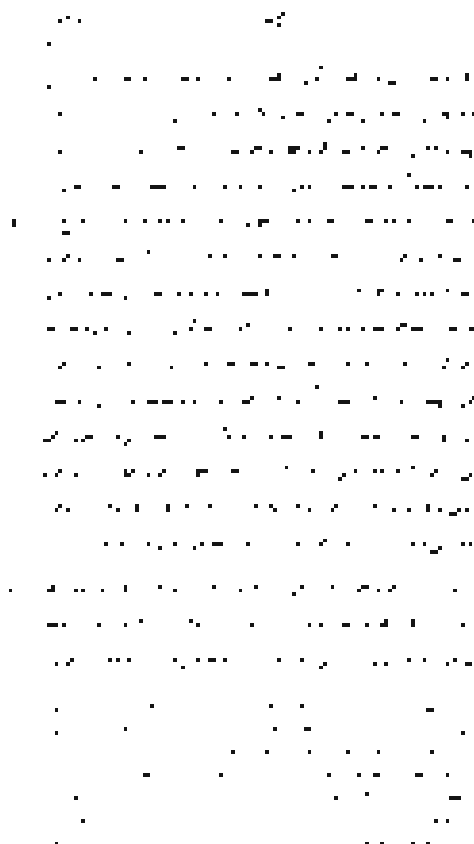


Figure 3 From George Anawati's critical edition of Ibn Sīnā's *Metaphysics* (1950's).

As a part of my doctoral dissertation I did a critical edition of the Arabic text of *Wisdom Observations* by Shaykh Aḥmad al-Aḥsā'ī (d. 1826). In this case full marks were applied (see Figure 4), a task made immensely easier by the use of Klaus Lagally's ArabT_EX transcription method. This method involves an entirely romanized Latin transcription of the Arabic script, using combinations of one or more Latin vowel letters to represent Arabic marks. If this had to be done in a text editor with a digital Arabic font the task of managing the marks would have been much more difficult.

With the growing popularity of Unicode and its implementation within T_EX (particularly X_YT_EX and CONT_EXT) Lagally's method has become mostly obsolete. Thankfully, today no one should have to enter text via Lagally's transcription method (as complete and as reliable as it was), particularly for extensive texts. Yet the problem

of entering and editing marks within digital Arabic text remains a thorn in the side of scholars who have to work with the Arabic script on a continual basis.

427

إِلَيْهِ شَيْءٌ فَيُخْبِرُ عَمَّا هُنَاكَ وَ يَصِفُ مَا فِيهِ. وَإِذَا كَانَ كَذَلِكَ، لَا يَعْرِفُهُ أَحَدٌ إِلَّا بِمَا
وَصَفَ بِهِ نَفْسَهُ.

وَهُوَ كَمَا يَقُولُ: لَا يُدْرِكُهُ غَيْرُهُ. فَلَا يَعْرِفُ كُنْهَهُ إِلَّا هُوَ لِأَنَّ عِلْمَهُ بِنَفْسِهِ غَيْرُ
نَفْسِهِ. فَإِذَا وَصَفَ نَفْسَهُ كَانَ وَصْفُ الْحَقِّ لِلْحَقِّ حَقًّا. وَ يَقَعُ عَلَيْنَا وَصْفُهُ خَلْقًا. وَ
نَحْنُ ذَلِكَ الْوَصْفُ الْوَاقِعُ عَلَيْنَا بِنَا، فَقَدْ تَعَرَّفَ لَنَا بِنَا. فَكَانَ وَصْفُهُ
55 [الْأَلَاكُ] إِلَى نَظَائِرِهَا. فَلَا يُدْرِكُ شَيْءٌ إِلَّا مَا كَانَ مِنْ جَنِينِهِ.

وَمَعْنَى أَنَّهُ لَا يَتَعَرَّفُ لِأَحَدٍ بِنَحْوِ مَا عَرَفَهُ مِنْ غَيْرِهِ أَنَّهُ مُبْحَاثُهُ عَرَفَ آخِلَقَ
لِلْخَلْقِ بِمَا هُمْ عَلَيْهِ أَتَاهُمْ خَلْقٌ. وَ هُوَ عَرَفَ نَفْسَهُ أَنَّهُ لَيْسَ بِخَلْقٍ؛ وَ لَا يُشَبِّهُهُ شَيْئًا
مِنَ الْخَلْقِ. فَلَا يُدْرِكُ مَا تَعَرَّفَ لَهُمْ بِهِ شَيْءٌ مِنْ بَصَائِرِهِمْ وَ لَا أَبْصَارِهِمْ. وَ إِنَّمَا
يُعْرِفُ بِبَصِيرٍ مِنْهُ. قَالَ (ع): «إِغْرِفُوا اللَّهَ بِاللَّهِ» وَ قَالَ الشَّاعِرُ:

إِذَا زَامَ عَاشِقُهَا نَظْرَةً وَ لَمْ يَسْتَطِعْهَا فَمِنْ لُطْفِهَا [175]

أَعَارَظُهُ طَرَفًا رَأَاهَا بِهِ فَكَانَ الْبَصِيرُ بِهَا طَرَفَهَا
وَمَعْنَى «فَهُوَ الْمَغْلُومُ وَ الْمَجْهُولُ» إِنْخِ أَنَّهُ الْمَغْلُومُ بِصُنْعِهِ، الْمَجْهُولُ بِكُنْهِهِ؛
الْمُوجُودُ بِأَيَاتِهِ، الْمُنْقُودُ بِدَائِهِ. فَظَهَرَ فَلَا شَيْءَ أَظْهَرَ مِنْهُ؛ وَ إِنَّمَا ظَهَرَ كُلُّ شَيْءٍ بِأَثَرِ
ظُهُورِهِ. وَ يَظُنُّ فَلَا شَيْءَ أَبْظَنَ مِنْهُ لِأَنَّهُ لَا شَيْءَ أَظْهَرَ مِنْهُ. وَ إِنَّمَا خَفِيَ لِشِدَّةِ
65 ظُهُورِهِ؛ وَ أَشْتَرَّ لِعَظَمِ نُورِهِ.

Figure 4 From Idris Samawi Hamid's draft critical edition of Shaykh Aḥmad's *Wisdom Observations* (doctoral dissertation; note the extensive use of marks).

52-53 ذَاتِهِ [نَفْسِهِ] T., p. 26.

54-55 [الْحَقُّ لِلْخَلْقِ خَلْقًا] T., p. 26.

55 وَ تُشِيرُ الْأَلَاكُ [و تُشِيرُ] T., p. 26.

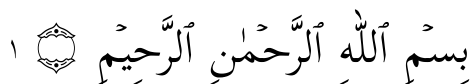
59 وَ لَا مِنْ [و لَا] T., p. 27.

63 وَ الْمَجْهُولُ [الْمَجْهُولُ] T., p. 28.

2.2 Enter SC Unipad

For some time this author struggled over the choice between the use of Unicode-based Arabic script and that of Lagally's romanized transcription. Finally, I discovered a Unicode editor called SC Unipad, a \$200.00 utility developed by some Persian-American developers. In addition to having perhaps the best implementation of the Unicode Bidirectional Algorithm in any text editor, the developers implemented a simple yet ingenious solution to the problem of entering text with marks. If the mark belongs to a given letter that is initial or medial, extend the

letter via *taṭwīl* and place the mark *horizontally* to the left of that letter. Consider the following Unicode Arabic text rendered with a simple and basic OpenType font such as SIL's Scheherazade:



Note the vertical marks above and below the base letters. Now let's compare the above to precisely the same string rendered by Unipad (see Figure 5). Note the positioning of the marks now makes them easily editable in one dimension (horizontal).

As exciting as this development was (and Unipad has lots of useful features for editing Unicode text) it soon became clear that this was a solution with many limitations. For example:

- The interline spacing is too tight, so marks come very close to clashing.
- The Unicode font is bitmap and hardcoded into the application. This means it cannot be used in any other editor.
- Although designed for its size (making it crystal clear at that size) it cannot be enlarged or shrunk.
- A $\text{\TeX}$ -code editor needs lots of important features if one is to have an efficient workflow; Unipad is hardly any good at all as a $\text{\TeX}$ -code editor.
- Perhaps worst of all, Unipad hasn't been updated since 2006 and is only up to Unicode version 4.1.0 in compatibility (Unicode is currently at 7.0). Numerous characters (including from Arabic script) are thus missing.

Despite its severe limitations, SC Unipad had indeed found a simple and ingenious solution to the problem of entering and editing Arabic-script marked text. The more general solution to the problem, then, would be to develop an OpenType font with capabilities similar to (and beyond) those of Unipad's bitmap font.

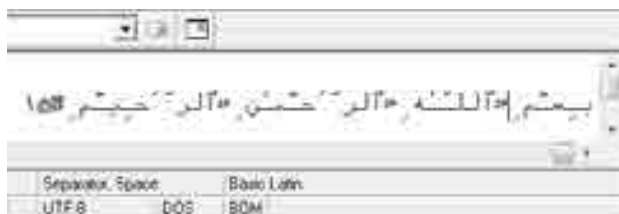


Figure 5 SC Unipad and Arabic-script marks.

2.3 Verbatim

In the process of consulting with Hans Hagen in the ongoing development of `CONTEXT`'s bidirectional capabilities, lots of verbatim issues arise. Verbatim is important because it helps to illustrate how a string of text is supposed to behave at different levels of the bi-directional analysis. So a common procedure would be to place a string of text into a buffer, then process that buffer to show both verbatim as well as fully processed output in the resultant pdf. No current font really meets the needs for verbatim Arabic text. So far we have been using DejaVu Sans Mono, whose limitations and lacunae as far as Arabic script are concerned are considerable.

A verbatim font is usually a fixed-width text font, the same kind of font generally used in a text editor. Thus a properly designed OpenType fixed-width font would address both issues: that of text editing as well as of verbatim representation in $\text{\TeX}$ processing.

2.4 Completeness

Finally, there is no generally available fixed-width (or even variable-width) font that is *complete* with respect to Arabic script. Nearly every available text-editing font is missing important characters. Any solution to the above problem would need to be as complete as possible as well as support the Arabic-script ranges of Unicode 7.0, the latest version as of this writing.

3 From Latin Modern Mono to Arabic-Latin Modern Fixed

3.1 $\text{\TeX}$ Gyre Latin Modern Mono

For our project we decided to extend Latin Modern Mono Regular 10 to Arabic. We chose this particular font for a number of reasons, including

- Most $\text{\TeX}$ verbatim work is done in Latin Modern Mono Regular.
- The $\text{\TeX}$ Gyre fonts are OpenType, and currently support everything in Latin needed for the *transliteration* of Arabic script into Latin. This is crucial for the editing of academic texts that work with transliteration.

$\text{\TeX}$ Gyre Cursor (close in design style to Courier New, a very nice editing fixed-width font) was also considered as a starting point. On the other hand, the metric width of Cursor is 600 font units, whereas that of Latin Modern Mono is 525 font units. Since the horizontal moving of marks is already going to stretch out the length of word strings, and the vertical metrics are going to have to be increased as well, using a smaller metric width allows a more economical use of space for marked Arabic-script expressions.

Another reason for going with Latin Modern is in honor of Donald Knuth's original verbatim design (Computer Modern Mono). My own motivations for getting into digital typography in the first place are much the same as those of Knuth, so an extension of Knuth in this instance seemed to be in the spirit of things.

3.2 Design

3.2.1 Knuthian Inspiration

The Arabic-script portion of Arabic-Latin Modern Fixed is a completely original font design, not based on any other Arabic-script font or typeface. It is inspired in part by the SC Unipad bitmap font. Its developers did an excellent job of designing it for readability at its intended resolution (which is where bitmaps are supposed to excel). But the actual letters of our font were developed from a careful study of Knuth's letters in Latin Modern Mono, such as the characters ›l‹, ›r‹, ›c‹, and ›7‹. The aim was to use elements in the original Knuthian design to develop a culturally authentic and aesthetically pleasing Arabic-script fixed-width font. In other words, Arabic-Latin Modern Fixed is a natural extension of and complement to the Knuthian design and inspiration.

In the following example, the Arabic-script letters are on the right:

l	ا	
c	ع	ف
r	ج	ب
m	س	
7	ص	

Thus one notices that the »ear« of the ›r‹ (top-right »terminal« or extremity) harmonizes well with many of the extremities of the Arabic letters. Now traditional Arabic-script does not contain these »ears«: We integrated the ears into the Arabic design in a way that we hope expands the possibilities of culturally authentic Arabic in a natural way. For example: In traditional Arabic terminal endings in characters are usually tapered from thick to thin. In some of the characters above one will notice we did exactly the opposite: Tapering goes from thin to thick (at the »ears«). Yet I believe we have managed to maintain a powerful, authentic Arabic feel, that is in harmony with the Knuthian design.

3.2.2 Vertical Metrics and Latin

Note that the width and length of the stem of the Arabic letter Alif (ا, first letter of the Arabic alphabet) is exactly the same as that of the Latin lower case Ell (l). Using this as a benchmark, it becomes easy to see why Arabic script nearly always needs more vertical-metric space than Latin in mixed script contexts. The maximum ascenders and descenders for Arabic-Latin Modern Fixed are considerably greater than those of Latin Modern Mono. This is to be expected: As we design the rest of the Arabic letters with respect to Alif, the ascender-descender space naturally has to expand to accommodate things. Note also that the baseline for Arabic letters is higher than it is for Latin: This is also natural and to be expected.

Virtually every aspect of the base Latin Modern Mono design has been incorporated into Arabic-Latin Modern Fixed. I did notice defects in a number of glyphs but except in a few minor cases have not changed anything in Latin Modern. The OpenType tables have been ported over. The only thing that prevents Arabic-Latin Modern Fixed from serving as a drop-in replacement for its Latin Modern Mono base is that the vertical metrics are different. Indeed, this is one reason for choosing to name the font ›Arabic-Latin Modern Fixed‹ instead of ›Latin-Arabic Modern‹. Of course, in T_EX one can set the interline spacing of Arabic-Latin Modern Fixed to precisely match that of its Latin Modern counterpart.

3.2.3 On-screen Applications

Due to its intended use for editors, a considerable effort has been made to make Arabic-Latin Modern Fixed comfortable for on-screen viewing. A number of Arabic-script characters have gone through numerous iterations in pursuit of this goal. Each shape iteration often involves having to then adjust dozens of other characters that depend on that shape, making this a very time-consuming process (the letter ط has been a real challenge, with its very close teeth). I did not change the Latin Modern counterparts (except in some very minor places as mentioned above). On the other hand, the hinting of Arabic-Latin Modern Fixed is, I hold, much better than that of Latin Modern Mono.

One reason I did not previously use Latin Modern Mono for on-screen applications such as text editing is because of its very limited hinting. For example: An on-screen string of, e. g., all capital letters at normal typing sizes (9-14 points) shows sometimes wild inconsistencies in the letter heights. Now, in Arabic-Latin Modern Fixed, the Latin component of Arabic-Latin Modern Fixed looks much better on-screen than in the former; it is to my eye virtually as comfortable as, say, Courier New on Windows.

Arabic-script fonts generally require greater point sizes for readability than Latin. Arabic-Latin Modern Fixed is just readable on-screen with marks even as low as

seven points. Comfortable point sizes for editing purposes range from 9 to 14 points (I particularly like the hinting effects in sizes 11 and 13; your own mileage may vary of course).

3.2.4 Some Notes:

- In a future maintenance release I may fix some of the minor errors in a few of the Latin Modern Mono glyphs.
- Although it has plenty of room for improvement (as almost any font does) the shapes have been more successful than I expected. I can envision Arabic-script applications where this could be used apart from editing or verbatim (such as web-page design).
- Related to this, I may eventually add a proportional version of this design to complement T_EX Gyre Latin Modern Proportional 10. Much of the current design elements can be reused. Although not as useful for verbatim and editing purposes, the aesthetics appear to me as robust enough to make a future proportional version a worthwhile pursuit.

3.3 Character Coverage

3.3.1 Arabic Blocks

The original Latin Modern Mono contains 785 glyphs, including about 617 standard Unicode characters. Oriental-T_EX Arabic-Latin Modern Mono contains 2630 glyphs, for an increase of 1845 glyphs. Of these, 1113 are Unicode code points. Arabic-Latin Modern Fixed covers every single character in all Unicode-Arabic blocks: Arabic, Arabic Supplement, Arabic Extended, Arabic Presentation-Forms A, and Arabic Presentation-Forms B.

One of the main features of Arabic-Latin Modern Fixed is its unique treatment of Arabic-script marks. Our font contains a total of 66 Unicode marks, more than any other publicly available font of which I am aware. It is almost certainly the most complete fixed-width font as far as the Arabic-script blocks are concerned.⁴

3.3.2 Presentation Forms

The Arabic Presentation Forms A includes what are commonly called »ligatures«, although the ligature concept doesn't really make sense for the Arabic script (for reasons discussed elsewhere). These characters have been decomposed into their base components: For two-component forms the metric width has been doubled; for three-component forms the metric width has been tripled. In part for this reason we call this font »fixed-width« rather than »mono«. That is, in a fixed-width

⁴ Nor have I seen any commercial font that covers these ranges so completely. On the other hand, this is not to deny that others may exist or be under development.

font, occasional or special characters may have a width that is a natural-number multiple of the fixed width; in a monospaced font every glyph must have precisely the same width. The expressions ›fixed-width‹ and ›monospaced‹ are normally used synonymously; we have introduced a subtle distinction in our adaptation of this nomenclature. Some examples:

ن ط ح سم تجم يمم

Column U+FDFF of Arabic Presentation-Forms A has been maintained without decomposition: A special character has been designed for each, within the standard fixed-width:

صله قل الله اكبر محمد صلعم رسول
 عليه صلى الله عليه وسلم جلاله جلاله
 الرحمن الرحيم

The famous Lām-Alif »ligature« (لا) has been maintained *only* for U+FEFB and U+FEFC. In every other case (U+FEF5..U+FEFA) they have been decomposed. In addition, U+0644 plus U+0621 do *not* automatically combine into the Lām-Alif combination. This is because Arabic-Latin Modern Fixed's purpose is primarily to serve as an input, editing, and verbatim font. Ligatures and the like get in the way of that task. Once input is done the text can be processed via a layout processor such as T_EX to another font (or one can make a simple font switch in a word processor). U+FEFB and U+FEFC (initial and final forms) have been kept because these two have an ancient pedigree in the Arabic script and deserve an independent representation.⁵

With a few exceptions (such as U+FD3E–U+FD3F ﴿ ﴾ ornate parenthesis), the bulk of Presentation-Forms A are useless for digital text entry and their use should be severely discouraged. They are there purely for esoteric legacy purposes, and in my view even those legacy purposes are suspect. Very few applications ever used the bulk of these characters; it is better to convert what few (if any) texts encoded using them into what even the Unicode Consortium calls the »preferred« Arabic Block (0600-06FF). On the other hand, it is the aim of Arabic-Latin Modern Fixed to be complete, so we have included the entire range of presentation characters.

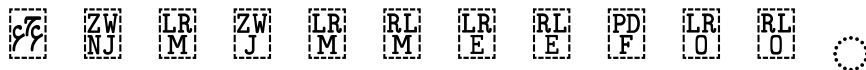
Note that Arabic-Latin Modern Fixed does *not* support initial, medial, or final processing of Presentation Forms; this is intentional. The most these forms should

⁵ Indeed, in some ancient texts (لا or U+FEFB) is considered a distinct letter of the Arabic alphabet. On the other hand, U+FEFB proper should only be used in special cases, otherwise U+0644 plus U+0621 should always be input explicitly.

be used for is visual presentation for purposes of conversion to the main Unicode Arabic block.⁶

3.3.3 Format and Spacing Glyph Representations

We added twelve additional Unicode format and spacing characters (with explicit glyph representations) and also designed a glyph representation for Latin Modern U+00A0 (no-break space), as well as U+25CC (dotted circle):



The glyph representations of these characters are important for at least a couple of reasons:

1. For verbatim representation of multi-directional and multilingual text.
2. For visual control and confirmation in text editors.⁷

The Arabic Language Mark —  or U+061C — is a new format character, placed in the main Arabic block. We will discuss its role in bi-directional typesetting in future research. Given the decision by the Unicode Consortium to place it in the Arabic Block, I decided to give it an Arabic-script representation. The abbreviation is short for **علامة الحرف العربي**, which means »Arabic Language Mark«.

The dotted circle is needed for Uniscribe to implement its »invalid marks« algorithm as well as the individual representation of marks in running text. We will say more about Uniscribe later.

4 OpenType Features

4.1 T_EX Gyre Latin Modern Mono and OpenType

As mentioned earlier, Arabic-Latin Modern Fixed has imported all T_EX Gyre Latin Modern Mono OpenType features. A few redundancies have been removed, and

⁶ Indeed, I strongly considered a purely *implicit* representation of the Presentation Forms in terms of OpenType lookups and the `ccmp` feature. Uniscribe, Microsoft's ubiquitous OpenType and language-layout processor, would not apply the feature, although the lookups were technically correct (we will discuss some of Uniscribe's other peculiarities as well later in this report). After many experiments and careful analysis it was decided to go ahead with an *explicit* representation of each and every Presentation-Forms character.

⁷ Oddly enough, Microsoft Notepad is one of the only text editors that actually let's you see the glyph representations of format (or »control«) and spacing characters. The glyph representation of format characters can also be turned on and off, while those of spacing characters seem to remain on permanently if the font has them.

one important bug that caused Uniscribe-based applications to switch to another fallback font under some conditions has been fixed.

4.2 Scripts and Language-Specific Discretionaries

There are now two supported scripts: Arabic (OpenType tag `<arab>`) and Latin (tag `<latn>`). In `CONTEXT` one activates them using the `<script>` key (e. g., `script=arab`).

Given an OpenType script one can implement lookup and feature rules for each language supported by that script. For most purposes languages follow the same script rules so a Default language (OpenType tag `<dflt>`)⁸ is usually enough. In some cases, however, certain modifications are needed. In the case of $\text{\TeX}$ Gyre Latin Modern Mono, a few conventions related to punctuation and numerals are implemented for a small handful of Latin-script languages.

In a similar spirit we have implemented some lookups for punctuation and numeric conventions for the three primary and most used Arabic-script languages: Arabic, Persian, and Urdu. For each supported language these are implemented under the Discretionary Ligatures feature (`<dlig>`). Most other Arabic-script languages can use either one of these three conventions or the default language option. When and where necessary, more language-specific discretions (appropriate to the purpose of this font) can be added as needed by users in the future.

I'll give just one example. Persian-Indo numerals (column `U+06FX`) and Arabic numerals (column `U+066X`) mostly look the same, but they follow different bidirectional rules.⁹ Urdu numerals also look mostly the same. But two or three numerals use a language-specific shape. For example, the numeral `ۛ4ۛ` takes on a distinctly different shape in Arabic, Persian, and Urdu respectively:

ع ۴ ۴

Users of each language (or a regionally related one) can thus choose which language to activate via the `<locl>` feature.

⁸ Internally this is converted to `DefaultLangSys` internally within the OpenType font file, but higher-level syntax usually uses `<dflt>`.

⁹ Loosely, the Arabic numerals have stronger bidirectional characteristics than the Persian-Indo numerals, which basically follow the same rules as Latin-script digits.

Here is a map of currently supported scripts and languages:

- Arabic script <arab>
 - Default <dflt>
 - Arabic <ARA>
 - Persian <FAR>
 - Urdu <URD>
- Latin script <latn>
 - Default <dflt>
 - Azeri <AZE>
 - Crimean Tatar <CRT>
 - Moldavian <MOL>
 - Romanian <ROM>
 - Turkish <TRK>

Latin-script languages support the official features Discretionary Ligatures <dlig>, Fractions <frac>, Lining Figures <lnum>, Oldstyle Figures <onum>, and Slashed Zero <zero>.

In each Arabic-script language (*except* Default) the language-specific features are found under the feature Localized Forms <locl>. For each Arabic-script language (*including* Default) punctuation alternates are found under Discretionary Ligatures <dlig>. Thus all three languages and default (<dflt>) support switching punctuation from Latin to a more authentic Arabic-script look. For example, the period on the left is the usual U+002E, the one on the right is more suited to Arabic-script but does not involve a different Unicode symbol:

• •

Notice that the Arabic-script period is both higher (the Arabic-script baseline is also higher) and shaped more squarish. This feature is turned on under <dlig>

Under Localized Forms locl we have situated figure substitutions (discussed above), and even some analogous language-specific letter substitutions such as

س → ذ (Persian)
 د → ن (Urdu)

Although these localized forms are also encoded in Unicode and can be input directly, <locl> allows users of one language system to, when desired, switch these shapes without changing the encoding of the source text.

4.3 Mark Editing

One of the chief motivations for the development of Arabic-Latin Modern Fixed was improved marks handling. Unfortunately Uniscribe, Microsoft's language-layout processor and the most used one in the world by far, has some built-in,

hardcoded rules that will override OpenType instructions as it sees fit. In other words, Uniscribe is not only an OpenType layout processor but in some cases also functions as a syntax enforcer. These restrictions make it difficult for font developers to be creative and develop features useful for, e. g., Arabic script.

For example, one could begin to approach the problem of moving marks to the left of the modified letter as follows: Since the font is already fixed-width, keep the marks at the same width (in most fonts a mark glyph is always zero width); and do not apply anchor attachment. Doing this will show up correctly when proofing and previewing the OpenType tables, but Uniscribe will override it, forcing the mark glyphs to occupy the same width space as the modified letter.¹⁰

Overcoming this limitation of Uniscribe (as well as a couple of others) required some OpenType trickery. In our case, we first converted the marks to base characters (in OpenType each character has to be labelled as either base, mark, ligature, or component). Once converted to a base character, the mark can then be treated mostly as a base character by Uniscribe. I say »mostly« because Uniscribe also has other, more esoteric, restrictions as well (we'll mention a couple of cases later).

After converting the marks to base characters, we can then make a substitution: For each initial or medial character, we substitute the mark (now converted to base) for itself plus the Arabic-script letter extender character (*taʔwīl* or U+0640). For example:

/ + — + J < — / + J

Then we position the mark over the *taʔwīl* using pair-adjustment (GPOS). There are some other subtleties in programming the tables but this is the main idea. So we now have

/ J

In a text editor, what had originally been a tedious and impractical task (selecting and editing marks) is just as straightforward as editing any other base character.

¹⁰ Some fonts that come with Windows operating systems (such as Courier New) also have mark glyphs that are not zero-width but fixed width. In either case Uniscribe ignores the width of mark glyphs.

4.4 Esoteric Subtleties

In Uniscribe it is not allowed to form a string of marks in succession, e. g.,

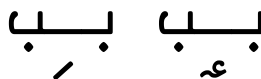


When an »invalid« string is encountered Uniscribe will often place the invalid mark over a dotted circle (◌ or U+25CC). We have added this symbol to Arabic-Latin Modern Fixed. Now, according to Microsoft,¹¹ »It should also be noted that the dotted circle is not inserted into the application’s backing store. This is a run-time insertion into the glyph array...«. This makes it very difficult to override via OpenType trickery, at least without unwanted side effects. However, we did find a workaround: Using the Zero-Width Joiner (◌ or U+200D) and contextual substitution of the dotted circle with the *taṭwīl* the following string in a Uniscribe-based application will give the same effect as above – with format-character glyph representation turned *off*, as is default. If you turn the format-character glyph representations *on* you will see something like this in Uniscribe-based applications:



This particular case is somewhat esoteric, but will help make Arabic-Latin Modern Fixed more useful and complete. In `CONTEXT` this trick is unnecessary.

Related: In Uniscribe, newly added marks to the Unicode standard will not be supported unless their status is explicitly hard-coded into Uniscribe. This can be a nightmare for some Arabic-script users of Windows text-processing applications. Consider the following two Arabic-script contextual strings:



Both strings are rendered correctly in `CONTEXT`. The one on the right uses a mark whose introduction to the Unicode standard is rather recent. Since Uniscribe does not recognize it, even though the OpenType tables characterize it as a mark it doesn’t get properly displayed by Uniscribe no matter what. Instead Uniscribe gives



¹¹ <http://www.microsoft.com/typography/OpenTypeDev/arabic/intro.htm>

Our OpenType lookups allow a user to partially get around this by adding a *taṭwīl* before the unsupported mark, but that will add an extra *taṭwīl*. For Uniscribe applications this results in



Until there are more readily accessible alternatives to Uniscribe (which is by far the most used and depended-upon multi-lingual OpenType language-layout processor out there) these kinds of issues are things one simply has to live with. Arabic-Latin Modern Fixed has worked around some issues and/or ameliorated others; but given the run-time nature of Uniscribe's limitations OpenType trickery cannot by itself eliminate all peculiarities of Uniscribe. Fortunately, in `CONTEXT` these issues do not arise, and at least the `TEX` community is not locked in to the interests or time-table of a commercial corporation to get support as needed.

5 Conclusion

Arabic-Latin Modern Mono will be a boon to scholars, academics, coder editors, and anyone who wants to manage the entry of the wide array of Unicode Arabic text for editing, verbatim, or other purposes. In this regard I believe it is unique (I have not found a single other fixed-width font that comes close to matching it). Arabic-script critical editions can now use the full array of available marks at an expense no greater than that of other characters. I will periodically issue maintenance releases as needed to fix bugs or, on a selective case-by-case basis, add features needed by users (such as other language-dependent discretionaries that may arise).

6 Appendix

6.1 Sample of Arabic-Latin Modern Fixed in `CONTEXT`

Normally one gives displays a text in verbatim, then gives the result after processing. In this case the main text is the same in both cases (since Arabic-Latin Modern Fixed is also the verbatim font). In addition, getting verbatim bidi correct is for a future project (one that the current font project helps to pave the way for). Here we present the preamble and postscript (without the body explicit, only an indication of the body buffer in its place), then we show the output.

Preamble and postscript:

```
\setupbodyfont[modern,tt]
\definefont[ALM][file:almmo10-regular*arabic at 14pt]
\setuplayout[width=6in]
\setupwhitespace[big]
\starttext
\setupinterlinespace[line=4.9ex] % One can play with this.
\setupalign[r2l]
\ALM

\getbuffer[almmo-sample] % Sample goes in this buffer.

\stoptext
```

The result can be seen in Figure 6.

6.2 License

Upon release, Arabic-Latin Modern Fixed will be released under the GUST Font License or something very close in spirit to it. See

<http://www.gust.org.pl/projects/e-foundry/licenses>

6.3 Unicode Arabic-Script Blocks

Notes:

- The grayed blocks are *unassigned* Unicode code points (as of version 6.3; see the last bullet); every *assigned* code point is supported by Arabic-Latin Modern Fixed as displayed here.
- The last page of Arabic Presentation Forms-A (FD20-FDFF) uses a smaller font size to fit the larger strings.
- The characters marked in black in the last page of Arabic Presentation Forms-A (FD20-FDFF) are »noncharacters«, i. e., these »codes are intended for process-internal uses« and will never be given character assignments by Unicode.
- Only the primary Unicode code points of Arabic-Latin Modern Fixed are displayed. The various variant and alternate forms are in the remainder of the font.
- As this report was going to press, Unicode 7.0 was released (the previous release was version 6.3). I have gone back and updated Arabic-Latin Modern Fixed to the latest Unicode. The new characters are in the 0600 block (one character) and the 08A0 block (eight characters). You'll notice that each new character is

in a gray box (meaning it was unassigned until the latest version). Again, any variant and alternate forms of these characters will be found in the remainder of the font.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ ١٥
 الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ ٢٥
 الرَّحْمَنِ الرَّحِيمِ ٣٥
 مَلِكِ يَوْمِ الدِّينِ ٤٥
 إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ ٥٥
 اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ ٦٥
 صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ ٧٥
 غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ ٧٥

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
 الْم ١٥
 ذَٰلِكَ الْكِتَابُ لَا رَيْبَ ٤٥ فِيهِ ٤٥
 هُدًى لِّلْمُتَّقِينَ ٢٥
 الَّذِينَ يُؤْمِنُونَ بِالْغَيْبِ ١٥ وَيُقِيمُونَ الصَّلَاةَ ١٥
 وَمِمَّا رَزَقْنَاهُمْ يُنْفِقُونَ ٢٥
 وَالَّذِينَ يُؤْمِنُونَ بِمَا أُنزِلَ إِلَيْكَ ١٥ وَمِمَّا
 أُنزِلَ مِنْ قَبْلِكَ ٤٥
 وَبِالْآخِرَةِ هُمْ يُوقِنُونَ ٤٥

Figure 6 Sample of Arabic-Latin Modern Fixed in action.

0600

Arabic

06FF

	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0	ع	ص	ي	ز	ـ	ح	ـ	ا	ب	د	غ	ك	ه	ي	و	ـ
	0600	0610	0620	0630	0640	0650	0660	0670	0680	0690	06A0	06B0	06C0	06D0	06E0	06F0
1	م	ف	ع	ر	ف	س	ا	ا	ع	ط	ف	ك	ه	ي	ح	ا
	0601	0611	0621	0631	0641	0651	0661	0671	0681	0691	06A1	06B1	06C1	06D1	06E1	06F1
2	و	ن	آ	ز	ق	ه	ي	ا	خ	ز	ف	ك	ه	ع	و	ي
	0602	0612	0622	0632	0642	0652	0662	0672	0682	0692	06A2	06B2	06C2	06D2	06E2	06F2
3	م	ن	أ	س	ك	ـ	ي	ا	ج	ب	ف	ك	ت	ه	س	ي
	0603	0613	0623	0633	0643	0653	0663	0673	0683	0693	06A3	06B3	06C3	06D3	06E3	06F3
4	م	ـ	و	ش	ل	ع	ـ	ـ	ج	ب	ق	ك	و	ـ	ـ	ـ
	0604	0614	0624	0634	0644	0654	0664	0674	0684	0694	06A4	06B4	06C4	06D4	06E4	06F4
5	ـ	ط	ا	ص	م	ـ	ـ	ا	خ	ب	ي	ل	و	ه	و	ـ
	0605	0615	0625	0635	0645	0655	0665	0675	0685	0695	06A5	06B5	06C5	06D5	06E5	06F5
6	ـ	ـ	ي	ض	ن	ـ	ـ	و	ج	ب	ق	ل	و	ـ	ـ	ـ
	0606	0616	0626	0636	0646	0656	0666	0676	0686	0696	06A6	06B6	06C6	06D6	06E6	06F6
7	ـ	ـ	ا	ط	ه	ـ	ـ	و	ج	ب	ق	ل	و	ـ	ـ	ـ
	0607	0617	0627	0637	0647	0657	0667	0677	0687	0697	06A7	06B7	06C7	06D7	06E7	06F7
8	ـ	ـ	ب	ظ	و	ـ	ـ	ا	ي	ب	ق	ل	و	ـ	ـ	ـ
	0608	0618	0628	0638	0648	0658	0668	0678	0688	0698	06A8	06B8	06C8	06D8	06E8	06F8
9	ـ	ـ	ة	ع	ي	ـ	ـ	ا	ب	ق	ك	ب	و	ـ	ـ	ـ
	0609	0619	0629	0639	0649	0659	0669	0679	0689	0699	06A9	06B9	06C9	06D9	06E9	06F9
A	ـ	ـ	ت	غ	ي	ـ	ـ	ا	ب	ي	ك	ب	ق	ـ	ـ	ـ
	060A	061A	062A	063A	064A	065A	066A	067A	068A	069A	06AA	06BA	06CA	06DA	06EA	06FA
B	ـ	ـ	ث	ك	ـ	ـ	ـ	ا	ب	ي	ك	ب	ق	ـ	ـ	ـ
	060B	061B	062B	063B	064B	065B	066B	067B	068B	069B	06AB	06BB	06CB	06DB	06EB	06FB
C	ـ	ـ	ج	ك	ـ	ـ	ـ	ا	ب	ي	ك	ب	ق	ـ	ـ	ـ
	060C	061C	062C	063C	064C	065C	066C	067C	068C	069C	06AC	06BC	06CC	06DC	06EC	06FC
D	ـ	ـ	ح	ي	ـ	ـ	ـ	ا	ب	ي	ك	ب	ق	ـ	ـ	ـ
	060D	061D	062D	063D	064D	065D	066D	067D	068D	069D	06AD	06BD	06CD	06DD	06ED	06FD
E	ـ	ـ	خ	ي	ـ	ـ	ـ	ا	ب	ي	ك	ب	ق	ـ	ـ	ـ
	060E	061E	062E	063E	064E	065E	066E	067E	068E	069E	06AE	06BE	06CE	06DE	06EE	06FE
F	ـ	ـ	د	ي	ـ	ـ	ـ	ا	ب	ي	ك	ب	ق	ـ	ـ	ـ
	060F	061F	062F	063F	064F	065F	066F	067F	068F	069F	06AF	06BF	06CF	06DF	06EF	06FF

0750

Arabic Supplement

077F

	075	076	077
0	يٰٓ 0750	يٰٓ 0760	طٓ 0770
1	ثٓ 0751	يٰٓ 0761	نٓ 0771
2	پٓ 0752	كٓ 0762	طٓ 0772
3	تٓ 0753	كٓ 0763	ٲٓ 0773
4	نٓ 0754	كٓ 0764	ٲٓ 0774
5	بٓ 0755	مٓ 0765	ئٓ 0775
6	بٓ 0756	مٓ 0766	ئٓ 0776
7	چٓ 0757	نٓ 0767	يٓ 0777
8	خٓ 0758	نٓ 0768	وٓ 0778
9	طٓ 0759	نٓ 0769	وٓ 0779
A	طٓ 075A	لٓ 076A	هٓ 077A
B	وٓ 075B	زٓ 076B	هٓ 077B
C	سٓ 075C	رٓ 076C	جٓ 077C
D	عٓ 075D	سٓ 076D	سٓ 077D
E	عٓ 075E	چٓ 076E	سٓ 077E
F	غٓ 075F	چٓ 076F	كٓ 077F

08A0

Arabic Extended-A

08FF

	08A	08B	08C	08D	08E	08F
0	ﻁ 08A0	ﻁ 08B0				ﻁ 08F0
1	ﻁ 08A1	ﻁ 08B1				ﻁ 08F1
2	ﻁ 08A2	ﻁ 08B2				ﻁ 08F2
3	ﻁ 08A3					ﻁ 08F3
4	ﻁ 08A4				ﻁ 08E4	ﻁ 08F4
5	ﻁ 08A5				ﻁ 08E5	ﻁ 08F5
6	ﻁ 08A6				ﻁ 08E6	ﻁ 08F6
7	ﻁ 08A7				ﻁ 08E7	ﻁ 08F7
8	ﻁ 08A8				ﻁ 08E8	ﻁ 08F8
9	ﻁ 08A9				ﻁ 08E9	ﻁ 08F9
A	ﻁ 08AA				ﻁ 08EA	ﻁ 08FA
B	ﻁ 08AB				ﻁ 08EB	ﻁ 08FB
C	ﻁ 08AC				ﻁ 08EC	ﻁ 08FC
D	ﻁ 08AD				ﻁ 08ED	ﻁ 08FD
E	ﻁ 08AE				ﻁ 08EE	ﻁ 08FE
F	ﻁ 08AF				ﻁ 08EF	ﻁ 08FF

FB50

Arabic Presentation Forms-A

FC3F

	FB5	FB6	FB7	FB8	FB9	FBA	FBB	FBC	FBD	FBE	FBF	FC0	FC1	FC2	FC3
0	ٱ	ڢ	ڤ	ڤ	ڪ	ڻ	ڻ	ط		و	ئو	ئج	تي	صح	فم
	FB50	FB60	FB70	FB80	FB90	FBA0	FBB0	FBC0		FBE0	FBF0	FC00	FC10	FC20	FC30
1	ٱ	ڢ	ڤ	ڤ	ڪ	ڻ	ڻ	ط		و	ئو	ئج	ئج	صم	في
	FB51	FB61	FB71	FB81	FB91	FBA1	FBB1	FBC1		FBE1	FBF1	FC01	FC11	FC21	FC31
2	ڀ	ڙ	ڙ	ڙ	ڳ	ڳ	ڳ			و	ئو	ئم	ئم	ضج	في
	FB52	FB62	FB72	FB82	FB92	FBA2	FBB2			FBE2	FBF2	FC02	FC12	FC22	FC32
3	ڀ	ڙ	ڙ	ڙ	ڳ	ڳ	ڳ			و	ئو	ئي	ئي	ضح	قح
	FB53	FB63	FB73	FB83	FB93	FBA3	FBB3		FBD3	FBE3	FBF3	FC03	FC13	FC23	FC33
4	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	ئي	ئي	ضخ	قم
	FB54	FB64	FB74	FB84	FB94	FBA4	FBB4		FBD4	FBE4	FBF4	FC04	FC14	FC24	FC34
5	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	بج	جج	ضم	قي
	FB55	FB65	FB75	FB85	FB95	FBA5	FBB5		FBD5	FBE5	FBF5	FC05	FC15	FC25	FC35
6	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	بج	جم	طح	قي
	FB56	FB66	FB76	FB86	FB96	FBA6	FBB6		FBD6	FBE6	FBF6	FC06	FC16	FC26	FC36
7	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	بخ	حج	طم	كا
	FB57	FB67	FB77	FB87	FB97	FBA7	FBB7		FBD7	FBE7	FBF7	FC07	FC17	FC27	FC37
8	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	بم	حم	ظم	كج
	FB58	FB68	FB78	FB88	FB98	FBA8	FBB8		FBD8	FBE8	FBF8	FC08	FC18	FC28	FC38
9	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	بي	خج	عج	كج
	FB59	FB69	FB79	FB89	FB99	FBA9	FBB9		FBD9	FBE9	FBF9	FC09	FC19	FC29	FC39
A	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	بي	خج	عم	كنج
	FB5A	FB6A	FB7A	FB8A	FB9A	FBAA	FBBA		FBDa	FBEa	FBFa	FC0A	FC1A	FC2A	FC3A
B	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	تج	خم	غج	كل
	FB5B	FB6B	FB7B	FB8B	FB9B	FBAB	FBBB		FBDB	FBEb	FBFb	FC0B	FC1B	FC2B	FC3B
C	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	تج	لج	غم	كم
	FB5C	FB6C	FB7C	FB8C	FB9C	FBAc	FBBc		FBDc	FBEc	FBFc	FC0C	FC1C	FC2C	FC3C
D	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	تخ	لج	فج	كي
	FB5D	FB6D	FB7D	FB8D	FB9D	FBAc	FBBd		FBDd	FBEc	FBFd	FC0D	FC1D	FC2D	FC3D
E	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	تم	لج	فح	كي
	FB5E	FB6E	FB7E	FB8E	FB9E	FBAE	FBBE		FBDc	FBEc	FBFE	FC0E	FC1E	FC2E	FC3E
F	ڀ	ڙ	ڙ	ڙ	ڳ	ه	ه			و	ئي	سم	فخ	لج	لج
	FB5F	FB6F	FB7F	FB8F	FB9F	FBAF	FBBF		FBDf	FBEf	FBFF	FC0F	FC1F	FC2F	FC3F

FC40

Arabic Presentation Forms-A

FD1F

	FC4	FC5	FC6	FC7	FC8	FC9	FCA	FCB	FCC	FCD	FCE	FCF	FD0	FD1
0	لح FC40	ني FC50	ني FC60	تر FC70	كا FC80	ي FC90	به FCA0	سم FCB0	فخذ FCC0	مذ FCD0	مئ FCE0	يم FCF0	حي FD00	نصر FD10
1	لخ FC41	هج FC51	ني FC61	تز FC71	كل FC81	ير FC91	تج FCA1	صد FCB1	فم FCC1	مم FCD1	بم FCE1	يه FCF1	جي FD01	طي FD11
2	لم FC42	هم FC52	ني FC62	تم FC72	كم FC82	يز FC92	تد FCA2	صذ FCB2	قد FCC2	نجد FCD2	به FCE2	ـ FCF2	جي FD02	طي FD12
3	لي FC43	هي FC53	ني FC63	تن FC73	كي FC83	يم FC93	تذ FCA3	صم FCB3	قم FCC3	ند FCD3	تم FCE3	ـ FCF3	خي FD03	عي FD13
4	لي FC44	هي FC54	ني FC64	تري FC74	كي FC84	ين FC94	تم FCA4	ضج FCB4	نجد FCC4	نذ FCD4	ته FCE4	ـ FCF4	خي FD04	عي FD14
5	مج FC45	يج FC55	ئز FC65	تي FC75	لم FC85	يي FC95	ته FCA5	ضد FCB5	كد FCC5	نم FCD5	ثم FCE5	طي FCF5	صي FD05	غى FD15
6	مح FC46	يخ FC56	ئم FC66	ثري FC76	لي FC86	يي FC96	ثم FCA6	ضذ FCB6	كد FCC6	نخ FCD6	ثم FCE6	طي FCF6	صي FD06	غى FD16
7	مخ FC47	يخ FC57	ئن FC67	ثري FC77	لي FC87	ئج FC97	جد FCA7	ضم FCB7	كل FCC7	هجد FCD7	سلم FCE7	عى FCF7	ضى FD07	للى FD17
8	مم FC48	يم FC58	ئى FC68	ثم FC78	ما FC88	ئد FC98	جم FCA8	طد FCB8	كم FCC8	هم FCD8	له FCE8	عى FCF8	ضى FD08	للى FD18
9	مى FC49	يى FC59	ئى FC69	ثمن FC79	مم FC89	ئذ FC99	جد FCA9	ظم FCB9	لج FCC9	ه' FCD9	شم FCE9	غى FCF9	شج FD09	شى FD19
A	مي FC4A	يى FC5A	برى FC6A	ثرى FC7A	نرى FC8A	ئم FC9A	حم FCAA	عج FCBA	لد FCCA	يد FCDA	شه FCEA	غى FCFA	شج FD0A	شى FD1A
B	نچ FC4B	زا FC5B	بز FC6B	ثري FC7B	نري FC8B	ئھ FC9B	خج FCAB	عم FCBB	لذ FCCB	يد FCDB	كل FCEB	سى FCFB	شخ FD0B	حى FD1B
C	نح FC4C	را FC5C	بم FC6C	فى FC7C	نم FC8C	بج FC9C	خم FCA0	عج FCBC	لم FCCC	يد FCD0	كم FCE0	سى FCFC	شم FD0C	حى FD1C
D	نخ FC4D	ي FC5D	بن FC6D	فى FC7D	نن FC8D	بد FC9D	سج FCA0	غم FCBD	له FCCD	يم FCD0	لم FCE0	شى FCFD	شر FD0D	جى FD1D
E	نم FC4E	ني FC5E	بى FC6E	قى FC7E	نى FC8E	بذ FC9E	سد FCAE	فج FCBE	مج FCEE	يه FCE0	نم FCE0	شى FCFE	سر FD0E	جى FD1E
F	نى FC4F	ني FC5F	بى FC6F	قى FC7F	نى FC8F	بم FC9F	سذ FCAF	فد FCBF	مد FCCF	ئم FCD0	نھ FCE0	حى FCFF	صر FD0F	خى FD1F

FD20

Arabic Presentation Forms-A

FDFE

	FD2	FD3	FD4	FD5	FD6	FD7	FD8	FD9	FDA	FDB	FDC	FDD	FDE	FDF
0	خي FD20	شم FD30		تجم FD50	سمد FD60	ضخم FD70	لحم FD80		تجي FDA0	يمي FDB0	مجي FDC0	FDD0	FDE0	عط FDF0
1	صي FD21	سه FD31		تجج FD51	سمج FD61	طمج FD71	لحي FD81		تخي FDA1	ممي FDB1	غمي FDC1	FDD1	FDE1	قأ FDF1
2	صي FD22	شه FD32		تحد FD52	لسم FD62	طمد FD72	لحي FD82	مجد FD92	تخي FDA2	قمي FDB2	بحي FDC2	FDD2	FDE2	الله FDF2
3	ضي FD23	طم FD33		تحم FD53	لسم FD63	طمم FD73	لجد FD83	همج FD93	تمي FDA3	نحي FDB3	كمم FDC3	FDD3	FDE3	لهر FDF3
4	ضي FD24	سج FD34		تخم FD54	صصح FD64	ططي FD74	لجج FD84	همم FD94	تمي FDA4	قمم FDB4	عجم FDC4	FDD4	FDE4	قأ FDF4
5	شج FD25	لسد FD35		تمج FD55	صحد FD65	عجم FD75	لخم FD85	نحم FD95	جمي FDA5	لحم FDB5	صمم FDC5	FDD5	FDE5	لمم FDF5
6	شج FD26	لسد FD36		تمد FD56	صمم FD66	عمم FD76	لخم FD86	نحي FD96	جحي FDA6	عمي FDB6	لسخي FDC6	FDD6	FDE6	سبا FDF6
7	شخ FD27	شج FD37		تمذ FD57	شحد FD67	عمم FD77	لمح FD87	نجم FD97	جمي FDA7	كمي FDB7	نجي FDC7	FDD7	FDE7	علي FDF7
8	شم FD28	شد FD38		حمج FD58	شحم FD68	عمي FD78	لمد FD88	نجم FD98	سخي FDA8	نجد FDB8		FDD8	FDE8	وأم FDF8
9	شر FD29	شد FD39		جمد FD59	شجي FD69	غمم FD79	مجد FD89	نجي FD99	صحي FDA9	مخي FDB9		FDD9	FDE9	ع FDF9
A	سر FD2A	طم FD3A		حمي FD5A	ششم FD6A	غمي FD7A	محم FD8A	نمي FD9A	شحي FDAA	لجم FDBA		FDDA	FDEA	شحم FDFa
B	صر FD2B	ظم FD3B		حمي FD5B	شمد FD6B	غمي FD7B	محي FD8B	نمي FD9B	ضحي FDAB	كمم FDBB		FDOB	FDEB	مأ FDFB
C	فصر FD2C	ل FD3C		سحي FD5C	ششم FD6C	فخم FD7C	مجد FD8C	ييم FD9C	لجي FDAc	لجم FDBc		FDDC	FDEc	وال FDFC
D	شج FD2D	ل FD3D		سجي FD5D	شمم FD6D	فخم FD7D	مجم FD8D	ييم FD9D	لمي FDA0	نجم FDBD		FDDD	FDED	شحم FDFD
E	شد FD2E	ف FD3E		سجي FD5E	فحي FD6E	قمج FD7E	مجد FD8E	بجي FD9E	يجي FDAE	جحي FDBE		FDEE	FDEE	
F	شد FD2F	ف FD3F		للمح FD5F	فضم FD6F	قمم FD7F	مخم FD8F	تجي FD9F	يجي FDAF	جحي FDBF		FDDF	FDEF	

FE70

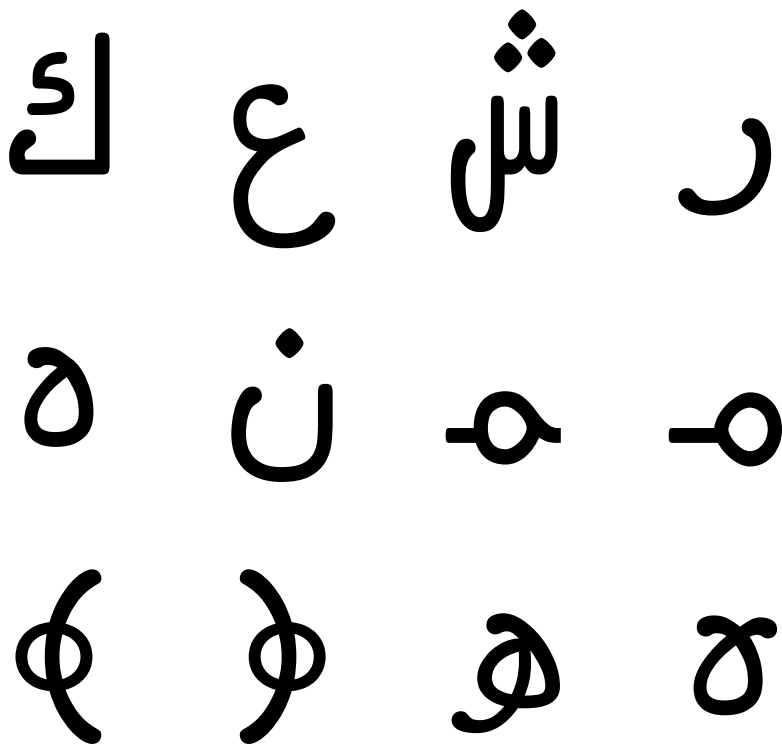
Arabic Presentation Forms-B

FEFF

	FE7	FE8	FE9	FEA	FEB	FEC	FED	FEE	FEF
0		ع	ب	ج	ز	ض	غ	ل	ى
1		آ	ـ	ح	س	ط	ف	م	ي
2		آ	ـ	ح	س	ط	ف	م	ي
3	،	أ	ة	د	س	ط	ف	م	ي
4		أ	ة	د	س	ط	ف	م	ي
5		ؤ	ت	خ	ش	ظ	ق	ن	ـ
6		ؤ	ت	خ	ش	ظ	ق	ن	ـ
7	ـ	إ	ت	ذ	ش	ظ	ق	ن	ـ
8		إ	ت	ذ	ش	ظ	ق	ن	ـ
9	ـ	ئ	ث	د	ص	ع	ك	ه	ـ
A		ئ	ث	د	ص	ع	ك	ه	ـ
B	ـ	ك	ث	ز	ص	ع	ك	ه	لا
C		ك	ث	ز	ص	ع	ك	ه	لا
D	ـ	ا	ج	ر	ض	غ	ل	و	
E		ل	ج	ر	ض	غ	ل	و	
F	ـ	ب	ج	ز	ض	غ	ل	ى	

6.4 A Large Glyph Sample

It's useful to look at a few letters up close to examine some of the subtleties in the design. In particular, the tapering towards extremities is the opposite of traditional Arabic design, in keeping with the Knuthian spirit, but it still has a culturally authentic feel.



Mit etoc Inhaltsverzeichnisse anpassen

Christine Römer

Das neue Paket `etoc` erweitert die Möglichkeiten, Inhaltsverzeichnisse individuell zu gestalten. Es ist besonders nützlich für das Einfügen von lokalen Inhaltsverzeichnissen.

Einführung

Das Paket `etoc` (*enhanced table of contents*) ist eine Erweiterung des standardmäßigen $\text{\LaTeX}$ -Befehls `\tableofcontents`, um benutzerdefinierte Anpassungen von globalen und/oder lokalen Inhaltsverzeichnissen (*Table of contents* – TOC) zu ermöglichen. Es wurde von Jean-François Burnol anfangs ohne Kenntnis und Bezug auf die etablierten Pakete zur eigenen Gestaltung von Inhaltsverzeichnissen (wie `tocloft`, `titledoc`, `minitoc`) geschaffen. [1, S. 1] Er hat dann später die Synchronisation zu den vorhandenen Paketen angestrebt und eingefügt.

$\text{\LaTeX}$ unterstützt automatisch das Erzeugen von Inhaltsverzeichnissen. Die Datei `jobname.toc` enthält die Verzeichniseinträge und wird von dem Befehl `\tableofcontents` eingelesen, wobei dieser jedoch nur einmal anwendbar ist. In `etoc` kann dagegen `\tableofcontents` beliebig oft benutzt werden. Somit sind verschiedene Inhaltsverzeichnisse (TOCs) und lokale Versionen davon möglich, wenn gewünscht mit veränderten Layouts.

Die Gliederungstiefe wird von vielen Klassen standardmäßig auf `\subsubsection` gesetzt (Wert = 3), kann aber mit `\setcounter{<Zähler>}{<Wert>}` verändert werden. `etoc` stellt für Modifizierungen den Befehl `\etocsettocdepth{<Wert>}` bereit.

Die Zuweisung von Ebenen zu den unterteilenden Einheiten kann zu jeder Zeit geändert werden und `etoc` kann dadurch in einer ziemlich generellen Weise genutzt werden, um selbsterstellte »Listen von« zu erzeugen. [1, S. 1]

Zur Kompatibilität mit den Dokumentklassen führt der Entwickler aus:

`etoc` scheint (mindestens teilweise) mit den Dokumentenklassen `article`, `book`, `report`, `scrartcl`, `scrbook`, `scrreprt` und `memoir` kompatibel zu sein. [1, S. 2]

Für das Zurückkehren zu den Standardeinstellungen der gewählten Textklasse gibt es den Kompatibilitätsmodus (siehe weiter dazu [1, Kap. 3.3]).

```
\etocstandardlines      %'Zeileneinträge' wie ohne \usepackage{etoc}
\etocstandarddisplaystyle %'Inhaltsstil' wie ohne \usepackage{etoc}
```

Die 43 Hauptbefehle des Pakets sind in [1, Kap. 6] verlinkt zu den Erklärungen aufgeführt, hier kann nicht auf alle eingegangen werden.

Zu Beginn des Überblickteils der Dokumentation wird die Frage gestellt »Muss ich Computerfreak sein, um etoc benutzen zu können?« Auch wenn der Autor wahrscheinlich berechtigt die Antwort »Nicht wirklich.« gibt, ist die Dokumentation doch ziemlich schwierig zu verstehen, da sehr T_EX-nisch abgefasst.

Lokale Inhaltsverzeichnisse

```
\localtableofcontents
```

Das Erzeugen von vielen Inhaltsübersichten möglich zu machen, ist ein wichtiges Anliegen des Pakets, wobei alle Verzeichnisse in nur einer toc-Datei enthalten sind. Die lokalen Verzeichnisse können beliebig ins Dokument eingefügt werden, normalerweise an den Anfang eines Kapitels. etoc erkennt automatisch, in welchem Abschnitt sich das Verzeichnis befindet und lässt alle anderen irrelevanten Abschnitte weg.

Zeilenstile (*line styles*)

```
\etocsetstyle{<Ebene>}{<1>}{<2>}{<3>}{<4>}
```

line styles meint die Gestaltung der Kapitelnummern, Seitenzahlen und der Benennung der Abschnitte. `\etocsetstyle` hat fünf Pflichtargumente und wird in der Regel unmittelbar vor der Anforderung des Inhaltsverzeichnisses für jede Gliederungsebene (Abschnittstyp) extra gesetzt und gilt so lange, bis es überschrieben wird.

Beispielsweise können die Gliederungsebenen `section` und `subsection` folgendermaßen umdefiniert werden:

```
\etocsetlevel{section}{1}
\etocsetlevel{subsection}{2}
\etocsettocdepth{2}
\etocsetstyle {section}
{
  {\leavevmode\leftskip 0cm\relax}
  {\bfseries\normalsize\color{blue}\makebox[0.5cm][l]{%
    \expandafter\ifx\expandafter\relax\etocnumber\relax\else\etocnumber.\fi}%
  \etocnumber.}%
  \etocname\nobreak\hfill\nobreak
  \rlap{\makebox[1cm]{\mdseries\etocpage}}\par}
{}
```

```

\etocsetstyle {subsection}
{
  {\leavevmode\leftskip .5cm\relax }
  {\mdseries\normalsize\color{red}\makebox[1cm][l]{\etocnumber}%
    \etocname\nobreak\hfill\nobreak
    \rlap{\makebox[1cm]{\etocpage}}\par}
}
\tableofcontents

```

Christine Römer und Herbert Voß

Deutsche Silbentrennmuster – aus linguistischer und T_EXnischer Sicht

Inhaltsverzeichnis

1	Einführung	2
2	Beginn der Normierung von Worttrennungen	2
3	Relevante typologische Besonderheiten des Deutschen	3
3.1	Doppelte Gliederung der Sprache	3
3.2	Fonologisch-typologischer Wandel: von einer Silben- zu einer Wortsprache	4
4	Aktuelle Silbentrennung am Zeilenende	5
4.1	Hierarchie der deutschen Silbentrennung	5
4.2	Änderungen der letzten Rechtschreibreform	6
5	T _E Xnische Grundlagen	7
5.1	Was ist T _E X?	7
5.2	Wie funktionieren Silbentrennalgorithmien?	8
6	Absatzumbruch und Silbentrennung bei T _E X	8
6.1	Absatzumbruch	8
6.2	Silbentrennung	9
7	Ästhetische Gesichtspunkte	12
8	Ausgewählte Silbentrennprobleme in der deutschen Sprache	13
8.1	Linguistische Probleme	13
8.2	T _E Xnische Probleme	16
9	Perspektiven für die elektronische Silbentrennung:	
	Vorschläge aus linguistischer Sicht	17
9.1	Gegenwärtige Trennmuster in T _E X	17
9.2	Perspektiven	18
9.3	Ausblick	20
	Literaturverzeichnis	20

Abbildung 1: Zeilenstil angepasst

TOC-Stile (*toc display styles*)

Durch `\etocsettocstyle[<Optionen>]{<Inhaltsangabe>}` wird der TOC-Stil festgelegt: Welche Ebenen, welches Format, ein- oder mehrspaltig, etc. (siehe [1, S. 40]) Es werden vier anpassbare TOC-Stile angeboten, die standardmäßig zweispaltig sind, der Einspaltenmodus kann durch das optionale Argument eingestellt werden. Die TOC-Stil-Festlegung muss auch vor der Anforderung des Inhaltsverzeichnisses erfolgen:

`\etocframedstyle` : mit Rahmen
`\etocruledstyle` : mit Linien
`\etocmulticolstyle` : Überschriften hintereinander
`\etoc tocstyle` : Struktur der Abschnitte modifizieren

Stil `\etocframedstyle`

```

\etocstandardlines
\renewcommand\etocbkgcolorcmd{\color{green!5}}
\renewcommand\etocbelowtocskip{0pt}
\etocframedstyle[1]{\fboxsep=1ex\fbox{\makebox[.5\linewidth]{%
  \etocfontminusone\hyperref[toc:a]{Inhaltsverzeichnis}}}}
\tableofcontents
  
```

Christine Römer und Herbert Voß

Deutsche Silbentrennmuster – aus linguistischer und T_EXnischer Sicht

Inhaltsverzeichnis		
1	Einführung	2
2	Beginn der Normierung von Worttrennungen	2
3	Relevante typologische Besonderheiten des Deutschen	3
3.1	Doppelte Gliederung der Sprache	3
3.2	Fonologisch-typologischer Wandel: von einer Silben- zu einer Wortsprache	4
4	Aktuelle Silbentrennung am Zeilenende	5
4.1	Hierarchie der deutschen Silbentrennung	5
4.2	Änderungen der letzten Rechtschreibreform	6
5	T_EXnische Grundlagen	7
5.1	Was ist T _E X?	7
5.2	Wie funktionieren Silbentrennalgorithmen?	8
6	Absatzumbruch und Silbentrennung bei T_EX	8
6.1	Absatzumbruch	8
6.2	Silbentrennung	9
7	Ästhetische Gesichtspunkte	12
8	Ausgewählte Silbentrennprobleme in der deutschen Sprache	13
8.1	Linguistische Probleme	13
8.1.1	Beziehung Fonem-Grafem	13
8.1.2	Problemfall "Binde-Trenn-Strich"	14
8.1.3	Problemfall rechter Wortteil	16
8.2	T _E Xnische Probleme	16
9	Perspektiven für die elektronische Silbentrennung: Vorschläge aus linguistischer Sicht	17
9.1	Gegenwärtige Trennmuster in T _E X	17
9.2	Perspektiven	18
9.3	Ausblick	20
	Literaturverzeichnis	20

Abbildung 2: TOC mit Rahmen

Stil \etocruledstyle

```
\etocruledstyle[1]{\etocfontminusone\color{blue}%
\fbxrule1pt\fbxseplex
\framebox[\linewidth]{\normalcolor\hss Inhaltsverzeichnis\hss}}
\tableofcontents
```

Christine Römer und Herbert Voß

Deutsche Silbentrennmuster – aus linguistischer und T_EXnischer Sicht

Inhaltsverzeichnis		
1	Einführung	2
2	Beginn der Normierung von Worttrennungen	2
3	Relevante typologische Besonderheiten des Deutschen	3
3.1	Doppelte Gliederung der Sprache	3
3.2	Fonologisch-typologischer Wandel: von einer Silben- zu einer Wortsprache	4
4	Aktuelle Silbentrennung am Zeilenende	5
4.1	Hierarchie der deutschen Silbentrennung	5
4.2	Änderungen der letzten Rechtschreibreform	6
5	T _E Xnische Grundlagen	7
5.1	Was ist T _E X?	7
5.2	Wie funktionieren Silbentrennalgorithmien?	8
6	Absatzumbruch und Silbentrennung bei T _E X	8
6.1	Absatzumbruch	8
6.2	Silbentrennung	9
7	Ästhetische Gesichtspunkte	12
8	Ausgewählte Silbentrennprobleme in der deutschen Sprache	13
8.1	Linguistische Probleme	13
8.1.1	Beziehung Fonem-Grafem	13
8.1.2	Problemfall "Binde-Trenn-Strich"	14
8.1.3	Problemfall rechter Wortteil	16
8.2	T _E Xnische Probleme	16
9	Perspektiven für die elektronische Silbentrennung: Vorschläge aus linguistischer Sicht	17
9.1	Gegenwärtige Trennmuster in T _E X	17
9.2	Perspektiven	18
9.3	Ausblick	20
	Literaturverzeichnis	20

Abbildung 3: TOC mit ruledstyle

Stil \etocmulticolstyle

```
\etocmulticolstyle{\noindent\bfseries\Large
\leaders\hrule height1pt\hfill
\MakeUppercase{Inhaltsverzeichnis}}
\tableofcontents
```

Christine Römer und Herbert Voß

Deutsche Silbentrennmuster – aus linguistischer und T_EXnischer Sicht

INHALTSVERZEICHNIS			
1	Einführung	2	6.1 Absatzumbruch 8
2	Beginn der Normierung von Worttrennungen	2	6.2 Silbentrennung 9
3	Relevante typologische Besonderheiten des Deutschen	3	7 Ästhetische Gesichtspunkte 12
3.1	Doppelte Gliederung der Sprache	3	8 Ausgewählte Silbentrennprobleme in der deutschen Sprache 13
3.2	Fonologisch-typologischer Wandel: von einer Silben- zu einer Wortsprache	4	8.1 Linguistische Probleme 13
4	Aktuelle Silbentrennung am Zeilenende	5	8.1.1 Beziehung Fonem-Grafem 13
4.1	Hierarchie der deutschen Silbentrennung	5	8.1.2 Problemfall "Binde-Trenn-Strich" 14
4.2	Änderungen der letzten Rechtschreibreform	6	8.1.3 Problemfall rechter Wortteil 16
5	T _E Xnische Grundlagen	7	8.2 T _E Xnische Probleme 16
5.1	Was ist T _E X?	7	9 Perspektiven für die elektronische Silbentrennung: Vorschläge aus linguistischer Sicht 17
5.2	Wie funktionieren Silbentrennalgorithmen?	8	9.1 Gegenwärtige Trennmuster in T _E X 17
6	Absatzumbruch und Silbentrennung bei T _E X	8	9.2 Perspektiven 18
			9.3 Ausblick 20
			Literaturverzeichnis 20

Abbildung 4: Mehrspaltiges TOC

Stil \etocstyle

```

\begingroup
\etocsettocdepth{2}
\parindent 0pt \leftskip 0cm \rightskip .75cm \parfillskip -\rightskip
\renewcommand*{\etoclineleaders}
  {\hbox{\normalfont\normalsize\hbox to .75ex {\hss.\hss}}}
\newcommand*{\EndParWithPagenoInMarginAndLeaders}
  {\nobreak\leaders\etoclineleaders\hfill
   \nobreak\makebox[0.75cm][r]{\mdseries\normalsize\etocpage}%
   \par }
\parindent 0pt \leftskip 0cm \rightskip .75cm \parfillskip -\rightskip
\etocsetstyle {section}
  {}
  {\leavevmode\leftskip 1.75cm\relax}
  {\bfseries\normalsize\llap{\makebox[.75cm][l]{\etocnumber}}}%
  \etocname\EndParWithPagenoInMarginAndLeaders}
  {}
\etocsetstyle {subsection}
  {}
  {\leavevmode\leftskip 2.75cm\relax }
  {\mdseries\normalsize\llap{\makebox[1cm][l]{\etocnumber}}}%
  \etocname\EndParWithPagenoInMarginAndLeaders}
  {}
\renewcommand\etoctrule {\hrule height 3pt\relax }
\renewcommand\etoctrulecolorcmd {\color{blue}}
\renewcommand\etocaftercontentshook{\medskip\color{blue}\hrule height 3pt}}
\etocruledstyle [1]{\Large\bfseries\fbbox{\makebox[8cm]{Inhaltsverzeichnis}}}
\tableofcontents
\endgroup

```

Christine Römer und Herbert Voß

Deutsche Silbentrennmuster – aus linguistischer und T_EXnischer Sicht

Inhaltsverzeichnis		
1	Einführung	2
2	Beginn der Normierung von Worttrennungen	2
3	Relevante typologische Besonderheiten des Deutschen	3
3.1	Doppelte Gliederung der Sprache	3
3.2	Fonologisch-typologischer Wandel: von einer Silben- zu einer Wortsprache	4
4	Aktuelle Silbentrennung am Zeilenende	5
4.1	Hierarchie der deutschen Silbentrennung	5
4.2	Änderungen der letzten Rechtschreibreform	6
5	T _E Xnische Grundlagen	7
5.1	Was ist T _E X?	7
5.2	Wie funktionieren Silbentrennalgorithmien?	8
6	Absatzumbruch und Silbentrennung bei T _E X	8
6.1	Absatzumbruch	8
6.2	Silbentrennung	9
7	Ästhetische Gesichtspunkte	12
8	Ausgewählte Silbentrennprobleme in der deutschen Sprache	13
8.1	Linguistische Probleme	13
8.2	T _E Xnische Probleme	16
9	Perspektiven für die elektronische Silbentrennung:	
	Vorschläge aus linguistischer Sicht	17
9.1	Gegenwärtige Trennmuster in T _E X	17
9.2	Perspektiven	18
9.3	Ausblick	20
	Literaturverzeichnis	20

Abbildung 5: Mit etocstyle die Gliederung modifizieren

Freie Gestaltungsmöglichkeiten

Weitere Gestaltungsmöglichkeiten werden in der Dokumentation aufgezeigt, beispielsweise die Gestaltung des TOC als Tabelle (siehe [1, Kap. 31.6]), als Baum mit dem pgf/tikz-basierten Paket forest (siehe [1, Kap. 18]) oder nach der TikZ-Syntax für Bäume als Molekül (siehe [1, Kap. 19]).

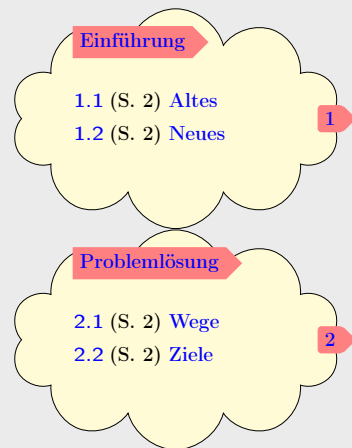
Nachfolgend kommt abschließend ein kleines Beispiel mit tikz. Der Quellcode zeigt, dass es für Anfänger/innen nicht so einfach ist, da Hilfsmakros zu definieren sind.

```
\usepackage{tikz} \newcounter{sectioncount}% TOC-Design von \section ändern.
\newcommand\DOSECTIONPICTURE{%
\tikzstyle{fancytitle} = [fill=red!50, text=white, signal]
\begin{tikzpicture}
\node [cloud, draw, fill=yellow!20, aspect=2] (box){%
\begin{minipage}{.5\textwidth} \subsectionlist \end{minipage}};
\node[fancytitle, right=12pt] at (box.north west){\SectionName};
\node[fancytitle, rounded corners] at (box.east) {\SectionNumber};
\end{tikzpicture}\par}
\newcommand\xpandaddtomacro[2]
{\expandafter\expandafter\expandafter\def
\expandafter\expandafter\expandafter #1\expandafter\expandafter\expandafter
{\expandafter#1#2}}%
```

```

\newcommand\addtomacro[2]{\expandafter\def\expandafter #1\expandafter {#1#2}}
\etocsetstyle{section}
{
\setcounter{sectioncount}{0}}
{\stepcounter{sectioncount}}
{\etocfontone\normalfont\huge\bfseries}
{}
\etocsetstyle{subsection}
{\def\subsectionlist{\begin{description}}}%
\let\SectionName\etocthelinkedname
\let\SectionNumber\etocthelinkednumber }
{}
{\addtomacro \subsectionlist{\item []}%
\xpandaddtomacro \subsectionlist{\etocthelinkednumber} (S. )}%
\xpandaddtomacro \subsectionlist{\etocthelinkedpage) }}%
\xpandaddtomacro \subsectionlist{\etocthelinkedname }}%
}%
{\addtomacro\subsectionlist {\end{description}}\DOSECTIONPICTURE}
\etocsettocstyle {}{}
\etocsettocdepth {subsection}
\begin{document}
\tableofcontents

```



1	Einführung
1.1	Altes
1.2	Neues
2	Problemlösung
2.1	Wege
2.2	Ziele

Abbildung 6: Mit tikz Kapiteluntergliederungen als Wolken.

Literatur

[1] Jean-François Burnol: *Das etoc-Paket*; <http://www.ctan.org/pkg/etoc>.

Von fremden Bühnen

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss vor. Bloße Updates können auf der moderierten *CTAN-ann*-Mailingliste verfolgt werden, die auch auf Twitter als @ctanannounce verfügbar ist.

babelx-multiple-dm von Maïeul Rouquette ist ein Paket, mit dem man mehrere neue »Datenmodelle« aus mehreren Quellen in *babelx* laden kann, ohne gleich einen vollständigen *babelx*-Stil neu schreiben zu müssen.

CTAN:macros/latex/contrib/babelx-contrib/babelx-multiple-dm

babelx-reauthor von Maïeul Rouquette fügt *babelx* ein Feld *reauthor* hinzu, um den bürgerlichen Namen des Autors von seinem Pseudonym unterscheiden zu können.

CTAN:macros/latex/contrib/babelx-contrib/babelx-reauthor

babel-latvian von Javier Bezos ist die *LaTeX*-Unterstützung für lettische Texte. Das Paket greift auf die Vorarbeit von Andris Lasis zurück.

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

sphdthesis von Saurabh Garg ist eine Klasse, die zum Schreiben einer Doktorarbeit an der *School of Computing* der *National University of Singapore* entwickelt wurde.

CTAN:macros/latex/contrib/sphdthesis

dithesis von Yannis Mantzouratos ist eine Dokumentenklasse zum Schreiben von Hausarbeiten am Fachbereich Informatik und Telekommunikation an der Universität Athen.

CTAN:macros/latex/contrib/dithesis

l3build vom *LaTeX3 Project* ist eine Test- und Building-Umgebung für *LaTeX*, die in Lua geschrieben wurde und die auf allen gängigen Plattformen lauffähig ist.

CTAN:macros/latex/contrib/l3build

longfigure von Tim Arnold stellt eine *figure*-ähnliche Umgebung bereit, die sich über mehrere Seiten erstrecken kann. Der Name lehnt sich deshalb an *longtable* an.

CTAN:macros/latex/contrib/longfigure

bagpipe von Walt Innis vereinfacht das Setzen von Stücken für Dudelsack mit *MusiXTeX*.

CTAN:macros/generic/bagpipe

getmap von *Josef Kleber* startet über den (zu aktivierenden) Befehl `\write18` ein externes Lua-Skript, um eine Karte von OpenStreetMap herunterzuladen.

CTAN:macros/latex/contrib/getmap

gitinfo2 von *Brent Longborough* dient zur Verwendung von Git-Metadaten in L^AT_EX. Nicht rückwärtskompatibler Nachfolger von *gitinfo*.

CTAN:macros/latex/contrib/gitinfo2

docbytex von *Petr Olsak* dient dazu, die Softwaredokumentation mit T_EX zu erstellen, vergleiche <http://www.olsak.net/docbytex.html>.

CTAN:macros/generic/docbytex

newtxtt von *Michael Sharpe* enthält verbesserte Schreibmaschinenfonts von *newtx*.

CTAN:fonts/newtxtt

tudscr von *Falk Hanisch* dient zum Setzen von Dokumenten an der *Technischen Universität Dresden* mit KOMA-Script.

CTAN:macros/latex/contrib/tudscr

universalis von *Bob Tennent* enthält eine Schriftfamilie, die als Ersatz für die Schriftarten *Univers* und *Frutiger* von *Adrian Frutiger* dienen kann.

CTAN:fonts/universalis

matlab-prettyfier von *Julien Cretel* ist ein Pretty-Printer für Matlab-Quelltext.

CTAN:macros/latex/contrib/matlab-prettyfier

unamth-template von *Jesús Velázquez* ist eine Vorlage für Arbeiten an der *Facultad de Ingeniería* an der *Universidad Nacional Autónoma de México*.

CTAN:macros/latex/contrib/unamth-template

heuristica von *Michael Sharpe* enthält die Schriftart *Heuristica* von *Andrej Panov* nebst L^AT_EX-Unterstützung, um den Font *Utopia* unter anderem um kyrillische Glyphen und um Kapitälchen zu ergänzen.

CTAN:fonts/heuristica

asyptureB von *Charles Staats III* ist ein Ersatz für das offizielle Paket *asymptote* zum Einbinden von *Asymptote*-Bildern in L^AT_EX, das vor allem Fehlermeldungen von *Asymptote* an L^AT_EX weiterreicht. Auch zum unveränderten Verarbeiten von Abbildungen, die nicht noch einmal kompiliert werden müssen, gibt es einen robusteren Mechanismus.

CTAN:graphics/asyptureb

placeat von *Arno L. Trautmann* erlaubt es, Inhalte mit LuaL^AT_EX absolut auf einer Seite zu platzieren. Zum Testen kann ein Raster unterlegt werden. Einfache grafische Elemente unterstützen beispielsweise Präsentationen.

CTAN:macros/luatex/latex/placeat

ucbthesis von *Paul Vojta* ist eine Klasse für Masterarbeiten und Dissertationen an der *University of California, Berkeley*, die elektronisch publiziert werden sollen.

CTAN:macros/latex/contrib/ucbthesis

repltext von *Scott Pakin* ermöglicht es, einen alternativen Ersatztext zu definieren, der aus einem PDF beim Kopieren anstelle des dort gesetzten Texts in die Zwischenablage übernommen wird.

CTAN:macros/latex/contrib/repltext

labyrinth von *Francesco Zigliotto* ist ein Paket, mit dem man Labyrinth sowie Lösungen zeichnen kann, um aus Labyrinth wieder herauszufinden.

CTAN:macros/latex/contrib/labyrinth

obnov von *Basil Solomykov* enthält die Schrift Obyknovennaya Novaya, die in der Sowjetunion häufig für wissenschaftliche Veröffentlichungen verwendet worden war (lateinische und kyrillische Buchstaben sowie die METAFONT-Quellen).

CTAN:fonts/obnov

baskervaldx von *Michael Sharpe* bietet einen Ersatz für die Schriftart Baskerville in den Formaten OpenType und PostScript mit L^AT_EX-Unterstützung.

CTAN:fonts/baskervaldx

cinzel von *Bob Tennent* enthält die L^AT_EX-Unterstützung für die Schriftarten *Cinzel* und *Cinzel Decorative* von *Natanael Gama*, die sich an alte römische Inschriften anlehnen.

CTAN:fonts/cinzel

biblatex-true-citepages-omit von *Maïeul Rouquette* korrigiert einige Beschränkungen der Option `citepages=omit` in dem Paket *biblatex*.

CTAN:macros/latex/contrib/biblatex-contrib/biblatex-true-citepages-omit

grundgesetze von *Marcus Rossberg* definiert Befehle für den Mathematikmodus, die dem Layout von *Gottlob Freges* »Grundgesetze der Arithmetik« nachempfunden wurden.

CTAN:macros/latex/contrib/grundgesetze

fifo-stack von *Stephen Gaito* ist eine Erweiterung des Pakets *stack* von *Benjamin Bayart* zum Führen eines FIFO-Stacks in L^AT_EX.

CTAN:macros/latex/contrib/fifo-stack

logicproof von *Alan Davidson* erlaubt das Setzen von Beweisen in Boxen in der Aussagen- und der Prädikatenlogik.

CTAN:macros/latex/contrib/logicproof

handout von *Maïeul Rouquette* dient zum Setzen eines Handouts, das einen wissenschaftlichen Vortrag begleitet.

CTAN:macros/latex/contrib/handout

efbox von *José Romildo Malaquias* erzeugt eine Inline-Box, die gerade so groß ist, dass sie den ihr mitgegebenen Text aufnehmen kann. Ein Rahmen und die Hintergrundfarbe der Box können optional gesetzt werden.

CTAN:macros/latex/contrib/efbox

clearsans von *Bob Tennent* enthält die Schriftart Clear Sans von *Daniel Ratighan* (Monotype) mit der entsprechenden L^AT_EX-Unterstützung. Eine Schriftart, die sich besonders für Bedienoberflächen eignet.

CTAN:fonts/clearsans

fira von *Bob Tennent* enthält die Fira-Fonts, die *Erik Spiekermann* und *Ralph du Carrois* für FirefoxOS gestaltet hatten, mit der L^AT_EX-Unterstützung.

CTAN:fonts/fira

playfair von *Bob Tennent* ist eine Schriftart, die sich besonders für Titel und Überschriften eignet. Das Paket enthält den Font und die L^AT_EX-Unterstützung.

CTAN:fonts/playfair

texfot von *Karl Berry* ist ein kleines Perlskript, das die Ausgabe eines $\TeX$ -Laufs »filtert« und nur diejenigen Meldungen übrig lässt, die wahrscheinlich eine Änderung am Quelltext erforderlich machen.

CTAN:support/texfot

Practical₂LaTeX von *George Grätzer* enthält Dateien mit Beispielen aus seinem Buch *Practical LaTeX* aus dem Springer Verlag, 2014.

CTAN:info/examples/Practical₂LaTeX

dccpaper von *Alex Ball* ist ein Bundle aus zwei Klassen, die zum Setzen von Artikeln für das *International Journal of Digital Curation* dienen.

CTAN:macros/latex/contrib/dccpaper

zhmakeindex von *Leo Liu* ist ein Ersatz für *makeindex* für Chinesisch. Der Indexprozessor unterstützt Unicode und beherrschaft das Sortieren von chinesischen Schriftzeichen.

CTAN:indexing/zhmakeindex

pgf-pie von *Yuan Xu* ist eine Sammlung von Makros zum Erstellen von Tortengrafiken mithilfe des Pakets PGF/TikZ.

CTAN:graphics/pgf/contrib/pgf-pie

pgf-umlcd von *Yuan Xu* ist ein TikZ-Paket zum Zeichnen von UML-Klassen-Diagrammen.

CTAN:graphics/pgf/contrib/pgf-umlcd

pst-intersect von *Christoph Bersch* berechnet die Schnittpunkte von beliebigen Kurven.

CTAN:graphics/pstricks/contrib/pst-intersect

zlmmt von *Michael Sharpe* erlaubt die Aktivierung von Latin-Modern-Typewriter-Fonts einschließlich Skalierung und allen weiteren verfügbaren Features.

CTAN:fonts/zlmmt

tabstackengine von *Steven B. Segletes* ist ein Frontend zur Verwendung von Tabulatoren mit dem Paket *stackengine*.

CTAN:macros/latex/contrib/tabstackengine

yandy von *Clerk Ma* enthält die Y&Y- $\TeX$ -Distribution, die unter Windows 7 und 8 getestet worden ist. Näheres unter <http://tug.org/yandy/>.

CTAN:systems/win32/yandy

rubik von *Apostolos Syropoulos* und *RWD Nickalls* ist ein Bundle, mit dem man Rubiks Zauberwürfel (und Zugfolgen beim Ordnen) mithilfe von TikZ zeichnen kann.

CTAN:macros/latex/contrib/rubik

pkgloader von *Michiel Helvensteijn* ist ein Paket, mit dem man die Reihenfolge festlegen kann, in der Paketoptionen und andere Pakete beim $\LaTeX$ -Lauf geladen werden sollen.

CTAN:macros/latex/contrib/pkgloader

perfectcut von *Guillaume Munch-Maccagnoni* stellt mathematische Klammern bereit, deren Größe sich bei Schachtelungen unabhängig von deren Inhalt an die jeweils benötigte Größe anpasst.

CTAN:macros/latex/contrib/perfectcut

tikz-opm von *Torben Hoffmann* ist ein Paket zum Zeichnen von *Object-Process-Methodology*-Diagrammen zur Beschreibung von Informationssystemen.

CTAN:graphics/pgf/contrib/tikz-opm

biblatex-manuscripts-philology von Maïeul Rouquette ergänzt in biblatex einen neuen Eintragsstyp @manuscript zur Verwaltung von Handschriften in der klassischen Philologie.

CTAN:macros/latex/contrib/biblatex-contrib/biblatex-manuscripts-philology
cv4tw von Geoffrey Gouez ist eine Klasse zum Setzen von Lebensläufen mit L^AT_EX.

CTAN:macros/latex/contrib/cv4tw

aobs-tikz von Claudio Fiandrino stellt TikZ-Stile bereit, die hilfreich sein können, um Overlays (Überblendungen) in Präsentationen mit dem Paket beamer zu erstellen.

CTAN:graphics/pgf/contrib/aobs-tikz

graphicxbox von D. P. Story stellt Befehle bereit, um eine Grafik als Hintergrund für Boxen zu verwenden.

CTAN:macros/latex/contrib/graphicxbox

lobster2 von Bob Tennent enthält den Handschriftenfont Lobster Two von Pablo Impallari.

CTAN:fonts/lobster2

copac-clean von Guido Milanese ist ein Tool für Linux, mit dem man BibT_EX-Einträge aus dem britischen Bibliothekskatalog <http://copac.ac.uk/> nachbearbeiten kann.

CTAN:support/copac-clean

scanpages von Michael Sharpe ist ein Paket aus T_EX-Makros und Skripten, das es leichter macht, eingescannte Inhalte in T_EX-Dokumente mit PDFT_EX selektiv innerhalb eines Rasters zu platzieren und zu übernehmen.

CTAN:macros/latex/contrib/scanpages

cnbwp von Zdenek Wagner ist ein Paket zum Erstellen von Arbeitspapieren der Tschechischen Nationalbank.

CTAN:macros/latex/contrib/cnbwp

Im Netz gefunden

Herbert Voß

In den verschiedenen Mailinglisten, Webforen, Newsgroups u. a. findet man immer wieder hilfreiche Angaben zur Arbeit mit und um das Thema Textsatz mit $\TeX$, $\LaTeX$, Con $\TeX$ t usw.

Die Makros `\@bsphack` und `\@esphack`³

Ein Befehl wie `\label{xyz}` führt dazu, dass die Eingabe zwei Leerzeichen enthalten kann: Eines davor und eines danach. Das Makro `\label` setzt keinen Text und so würde ein unnatürlich großer Leerraum zwischen zwei Worten erscheinen, wenn die Definition von `\label` nicht mit `\@bsphack` anfangen und mit `\@esphack` enden würde. `\@bsphack` ermittelt den vorangegangenen horizontalen Leerraum. Dieser wird von `\@esphack` getestet. Ist der Wert größer als 0 pt, dann ignoriert `\@esphack` das folgende Leerzeichen. Das Ziel ist es, dass $\LaTeX$ im Fall von zwei Leerzeichen nur eines setzt. Die meisten Befehle, die keinen Text an der aktuellen Stelle erzeugen, benutzen diesen »Hack«. Beispiele sind `\glossary`, `\index`, `\marginpar`, `\nopagebreak`, `\vspace` und Gleitumgebungen.

Das Dokument `l2tabu`: How to use `\graphicspath`⁴

Kann man das »Tabu« getrost ignorieren?

Die Empfehlung in `l2tabu`, dass man `\graphicspath` nicht benutzen sollte, basiert auf einer Antwort des $\LaTeX$ -Teams auf einen Vorschlag zur Erweiterung des Mechanismus von $\LaTeX$ zum Auffinden von Dateien, sowie einer Antwort auf einen Bug-Report zu `\graphicspath` selbst. In beiden bezeichnet David Carlisle die Bereitstellung von `\graphicspath` als einen Fehler. Siehe dazu <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/2618> und <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=graphics/2612>.

Ich habe bereits damals darauf hingewiesen, dass das Geschwindigkeitsproblem immer weniger relevant wird. Heutzutage spielt es vor allem dann eine Rolle, wenn man tiefe Verzeichnisstrukturen nicht per `ls-R`, sondern direkt durchsuchen

³ Dan Luecking in `comp.text.tex`

⁴ Markus Kohm in `dante-ev`

lässt und eine Datei ungünstig liegt oder gar nicht vorhanden ist. Dann nützt das Ausweichen auf `TEXINPUTS` allerdings auch nichts.

Das Speicherproblem wurde irgendwann etwas entschärft. Während früher Rechner mit ein- bis zweistelligen MB an RAM auskommen mussten, ist dies in $\TeX$ -Größenordnungen heute kein Problem mehr. Also wurde irgendwann u. a. `pool_size` und `max_strings` drastisch erhöht. Wenn ich mich recht erinnere, wurde auch im Web2C-Code von $\TeX$ etwas geändert, damit Dateinamen nicht gefundener Dateien nicht mehr sowohl `pool` als auch `string memory` belegen. Da bin ich mir aber nicht ganz sicher. Jedenfalls gelangt man heute selbst in Dokumenten mit vielen Grafiken und bei Verwendung von `\graphicspath` mit mehreren Einträgen bei weitem nicht mehr so leicht an die Grenze wie noch 1997. Gleichzeitig verstehen heute alle modernen $\TeX$ -Implementierungen den Slash als Pfadtrenner, so dass das damals in Report 2618 vorgeschlagene `\dir@sep` bei nüchterner Betrachtung heute ebenfalls nicht mehr notwendig ist. So gesehen könnte man sagen, dass sowohl der Vorschlag als auch die Antwort von der Entwicklung überholt wurden.

Lohnt sich das »Tabu« bei 10, 100 oder erst 10000 Bildern zu beachten?

Das Problem tritt sicher nicht bei einigen 10 Bildern in Kombinationen mit einigen wenigen Einträgen in `\graphicspath` auf. Bei einigen 10000 Bildern, bei denen außerdem die Dateiendung nicht angegeben ist, könnte es jedoch schnell gehen. `max_strings` liegt üblicherweise bei um die 500000. Hat man nun 32 Pfade in `\graphicspath` und die Dateiendungen `.png`, `.jpg`, `.mps`, `.pdf` und `.jpeg`, sowohl in Groß- als auch in Kleinbuchstaben – also 10 mögliche Endungen – dann könnte im ungünstigsten Fall der Strings-Speicher bereits nach weit weniger als 1500 Bildern ausgenutzt sein (er wird ja nicht nur für Dateinamen benötigt und intern vom Parser für die Dateinamen ggf. mehr als einmal). Bei langen Datei- oder Verzeichnisnamen könnte auch `pool_size` bereits früher ausgehen. Getestet habe ich das nicht. Welcher der beiden zuerst zu Ende ist und ob das überhaupt passiert, hängt auch davon ab, wie die Änderung an $\TeX$ selbst, die ich oben angedeutet habe, ausgefallen ist und ob diese tatsächlich generell vorgenommen wurde oder auf einzelne $\TeX$ -Implementierungen beschränkt war.

Wenn ich mich recht erinnere, spielt es auch eine Rolle, ob man den Weg `pdf-latex`→`pdf` verwendet, oder `latex`→`dvi`→`pdf`, weil `pdftex.def` die Suche nach der Datei und deren Größe mit »`pdftex-only`-Makros« durchführt, die keine Strings und auch keinen `string-space` im `pool` belegen. Ich bin da bezüglich der Implementierungsdetails nicht ganz auf dem Laufenden.

Vor ein paar Jahren hatte ich bei einem Produktkatalog das Problem mit `max_strings` trotz Verwendung von `TEXINPUTS` an Stelle von `\graphicspath`.

Die Suche nach verschiedenen Endungen war damals schlicht zu teuer. Beheben ließ sich das einfach durch Angabe der Dateiendung bei allen Dateien.

Meiner Meinung nach wäre zu überlegen, den Passus in `l2tabu` zu überarbeiten und dort lediglich darauf hinzuweisen, dass in `\graphicspath` nicht exorbitant viele Verzeichnisse enthalten sein sollten oder man bei einem Überlauf von `pool_size` oder `max_strings` auf `TEXINPUTS` ausweichen kann.



(Fotos: Martin Wilhelm Leidig – Chemitzer Linuxtag)

Spielplan

2014

8. 9. – 13. 9. **8th International ConT_EXt Meeting**
’t Sjetootje, Bassenge (Belgien).
<http://meeting.contextgarden.net/2014/>
20. 9. **Herbsttagung**
und 51. Mitgliederversammlung von DANTE e.V.
KIT, Karlsruhe
<http://www.dante.de/events/Herbst2014.html>

2015

- April **DANTE 2015**
und 52. Mitgliederversammlung von DANTE e.V.
Fachhochschule Stralsund



(Fotos: Martin Wilhelm Leidig – Chemitzer Linuxtag)

Stammtische

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

Aachen

Torsten Bronger,
bronger@physik.rwth-aachen.de
Gaststätte Knossos, Templergraben 28, 52062 Aachen
Zweiter Donnerstag im Monat, 19.00 Uhr

Berlin

Michael-E. Voges, Tel.: (03362) 50 18 35,
mevoges@t-online.de
Mantee – Café Restaurant, Chausseestraße 131, 10115 Berlin
Zweiter Mittwoch im Monat, 19.00 Uhr

Bielefeld

Jürgen Schwarze, Tel.: (0521) 5 57 39 06,
juergen.schwarze@bitel.net
Ferdis Pizza Pinte, Schmiedestraße 9, 33613 Bielefeld,
Zweiter Montag im Monat, 19.30 Uhr

Bremen

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

Erlangen

Walter Schmidt, Peter Seitz,
w.a.schmidt@gmx.net
Gaststätte »Deutsches Haus«, Luitpoldstraße 25, 91052 Erlangen
Dritter Dienstag im Monat, 19.00 Uhr

Frankfurt

Harald Vajkonny,
<http://wiki.lug-frankfurt.de/TeXStammtisch>
Restaurant »Zum Jordan«, Westerbachstr. 7, 60489 Frankfurt
Zweimonatlich, Vierter Donnerstag im Monat, 19.30 Uhr

Hamburg

Lothar Fröhling,
lothar@thefroehlings.de
Letzter Dienstag im Monat an wechselnden Orten, 19.00 Uhr

Hannover

Mark Heisterkamp,
heisterkamp@rrzn.uni-hannover.de
Seminarraum RRZN, Schloßwender Straße 5, 30159 Hannover
Zweiter Donnerstag im Monat, 18.30 Uhr

Heidelberg

Martin Wilhelm Leidig, Tel.: (06203) 40 22 03,

moss@moss.in-berlin.de

Anmeldeseite zur Mailingliste: <http://mailman.moss.in-berlin.de/mailman/listinfo/stammtisch-hd-moss.in-berlin.de>

Wechselnder Ort

Letzter Freitag im Monat, ab 19.30 Uhr

Karlsruhe

Klaus Braune, Tel.: (0721) 608-4 40 31,

klaus.braune@kit.edu,

SCC (Steinbuch Centre for Computing) des KIT (vormals Universität Karlsruhe, Rechenzentrum),

Zirkel 2, 2. OG, Raum 203, 76131 Karlsruhe

Erster Donnerstag im Monat, 19.30 Uhr

Köln

Uwe Ziegenhagen

Ersatzbau Geowissenschaften, Greinstr. 6, 50674 Köln

Zweiter Dienstag im Monat, 19.00 Uhr

München

Uwe Siart,

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

Erste Woche des Monats an wechselnden Tagen, 19.00 Uhr

Stuttgart

Bernd Raichle,

bernd.raichle@gmx.de

Bar e Ristorante »Valle«, Geschwister-Scholl-Straße 3, 70197 Stuttgart

Zweiter Dienstag im Monat, 19.30 Uhr

Trier

Martin Sievers,

ttt@schoenerpublizieren.de

Anmeldeseite zur Mailingliste: <http://lists.schoenerpublizieren.de/cgi-bin/mailman/listinfo/ttt>

Universität Trier

nach Absprache

Wuppertal

Andreas Schrell, Tel.: (02193) 53 10 93,

as@schrell.de

Restaurant Croatia »Haus Johannisberg«, Südstraße 10, 42103 Wuppertal

Zweiter Donnerstag im Monat, 19.30 Uhr

Würzburg

Bastian Hepp,

LaTeX@sning.de

nach Vereinbarung

Adressen

DANTE, Deutschsprachige Anwendervereinigung 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: VR Bank Rhein-Neckar eG
BLZ 670 900 00 IBAN DE67 6709 0000 0002 3100 07
Kontonummer 2 310 007 SWIFT-BIC GEN0DE61MA2

Vorstand

Vorsitzender:	Martin Sievers	president@dante.de
Stellvertr. Vorsitzender:	Herbert Voß	vice-president@dante.de
Schatzmeister:	Klaus Höppner	treasurer@dante.de
Schriftführer:	Manfred Lotz	secretary@dante.de
Beisitzer:	Volker RW Schaa Dominik Wagenführ Uwe Ziegenhagen	

Ehrenmitglieder

Peter Sandner	22.03.1990	Klaus Thull †	22.03.1990
Yannis Haralambous	05.09.1991	Barbara Beeton	27.02.1997
Luzia Dietsche	27.02.1997	Donald E. Knuth	27.02.1997
Eberhard Mattes	27.02.1997	Hermann Zapf	19.02.1999
Joachim Lammarsch	12.04.2014	Rainer Schöpf	12.04.2014

Server

DANTE: <http://www.dante.de/> (Rainer Schöpf, Joachim Schrodtt)
CTAN: <http://mirror.ctan.org/>

FAQ

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

Autoren/Organisatoren

Jürgen Fenn Friedensallee 174/20 63263 Neu-Isenburg juergen.fenn@gmx.de	[55]	Martin Sievers siehe Seite 66	[5]
Idris Samawi Hamid Colorado State University ishamid@colostate.edu	[19]	Herbert Voß Wasgenstraße 21 14129 Berlin herbert@dante.de	[3, 60]
Christine Römer Institut für germanistische Sprachwissenschaft FSU Jena Christine.Roemer@uni-jena.de	[47]	Jacob Wiersma jacob@wiersma-online.de	[6]
		Steve Zakrzowsky steve.zakrzowsky@gmail.com	[14]

Die T_EXnische Komödie

26. Jahrgang Heft 3/2014 August 2014

Impressum

Editorial

Hinter der Bühne

5 Grußwort

Bretter, die die Welt bedeuten

6 Mehr Möglichkeiten mit Fußnoten

14 Paket skmath für mathematische Formeln

19 DANTE Summary Report: Introducing Arabic-Latin Modern Fixed

47 Mit etoc Inhaltsverzeichnisse anpassen

Von fremden Bühnen

55 Neue Pakete auf CTAN

60 Im Netz gefunden

Spielplan

63 Termine

64 Stammtische

Adressen

67 Autoren/Organisatoren