

Deutschsprachige
Anwendervereinigung $\text{T}_{\text{E}}\text{X}$ e.V.

Michael Niedermair: *Neues von $\varepsilon_{\mathcal{X}}\text{T}_{\text{E}}\text{X}$: TFM-Daten anschaulich darstellen*, Die $\text{T}_{\text{E}}\text{X}$ nische Komödie 3/2005, S. 27–31.

Reproduktion oder Nutzung dieses Beitrags durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nicht-kommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden. Für kommerzielle Nutzung ist die Zustimmung der Autoren einzuholen.

Die $\text{T}_{\text{E}}\text{X}$ nische Komödie ist die Mitgliedszeitschrift von DANTE, Deutschsprachige Anwendervereinigung $\text{T}_{\text{E}}\text{X}$ e.V. Einzelne Hefte können von Mitgliedern bei der Geschäftsstelle von DANTE, Deutschsprachige Anwendervereinigung $\text{T}_{\text{E}}\text{X}$ e.V. erworben werden. Mitglieder erhalten Die $\text{T}_{\text{E}}\text{X}$ nische Komödie im Rahmen ihrer Mitgliedschaft.

Neues von $\varepsilon_{\mathcal{X}}\text{T}_{\text{E}}\text{X}$: TFM-Daten anschaulich darstellen

Michael Niedermair

Dieser Artikel beschreibt, wie mit Hilfe von $\varepsilon_{\mathcal{X}}\text{T}_{\text{E}}\text{X}$ eine TFM-Datei durch eine Transformation in eine gestaltete HTML-Datei umgewandelt werden kann, um sie optisch »aufgepeppt« darzustellen.

Einleitung

Mit diesem Artikel möchte ich eine Serie einleiten, um in unregelmäßigen Abständen Neues von $\varepsilon_{\mathcal{X}}\text{T}_{\text{E}}\text{X}$ (siehe <http://www.extex.org>) zu berichten. Als Einstiegsthema möchte ich hier eine etwas andere Darstellung der TFM-Datei ($\text{T}_{\text{E}}\text{X}$ -Font-Metrik) zeigen.

TFM für die Metriken

Ein Ziel von $\varepsilon_{\mathcal{X}}\text{T}_{\text{E}}\text{X}$ ist es, die traditionellen Fontmetriken von $\text{T}_{\text{E}}\text{X}$ zu verwenden. Dazu müssen sie eingelesen und in ein eigenes Format umgewandelt

werden (mit AFM- oder TTF-Dateien wird genauso verfahren). Für die Darstellung der Metriken verwendet $\epsilon\chi\text{T}_{\text{E}}\text{X}$ ein XML-Format, welches die Dateiendung EFM ($\epsilon\chi\text{T}_{\text{E}}\text{X}$ -Font-Metrik) besitzt. Zum Testen, ob die Werte richtig ausgelesen werden, wurden sie in eine XML-Datei geschrieben und dann mit Hilfe eines JUnit-Tests verglichen. Ein JUnit-Test ist ein Test, der u. a. Java-Klassen testet und sehr einfach die Möglichkeit bietet, Werte zu vergleichen (z. B. `assertEquals("-1026142560", e1.getAttributeValue("checksum"));`). Das XML-Format hat in dieser Sicht mehrere Vorteile: Es ist direkt für den Menschen lesbar, es lässt sich beliebig erweitern und mit vielen Hilfsprogrammen verarbeiten.

Die XML-Datei wird dabei mit $\epsilon\chi\text{T}_{\text{E}}\text{X}$ wie folgt erzeugt (eine entsprechende Installation von $\epsilon\chi\text{T}_{\text{E}}\text{X}$ und Java vorausgesetzt):

```
java de.dante.util.font.TFM2XML cmr12.tfm /tmp/cmr12.xml
```

Als Ergebnis erhält man eine XML-Datei (siehe Abbildung 1). Dabei wird jede interne TFM-Tabelle (Informationen zu den Zeichen, Ligaturen, Maßen, zu Kerning usw.) entsprechend umgesetzt. Zusätzlich wird unter `/tfm/charinfo/char` zu jedem Zeichen eine informative Zusammenfassung aus den anderen Tabellen des Zeichens bereitgestellt. Im Gegensatz zur XML-Datei enthält die EFM-Datei nur die zusammengesetzten Informationen und nicht mehr alle Originaltabellen. Zum Testen werden aber alle Informationen benötigt. Tabelle 1 zeigt eine Übersicht über die Java-Klassen, die TFM-Dateien umwandeln.

Mit XSL-T transformieren

Eine Möglichkeit, diese XML-Datei weiterzuverarbeiten, stellt XSLT dar. (XSL-Transformation ist eine Möglichkeit, mit Regeln eine XML-Datei in verschiedene andere Formate, wie z. B. XML, HTML, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ usw., umzuwandeln. Dabei können entweder die ganze XML-Datei oder nur Teilbereiche davon für die Ausgabe verwendet werden.) Für unser Beispiel werden nur einige Informationen aus der XML-Datei transformiert und als Ausgabeformat HTML festgelegt. Der Aufruf von XSLT lässt sich auch direkt mit $\epsilon\chi\text{T}_{\text{E}}\text{X}$ durchführen, da Java einen XSLT-Prozessor integriert hat. Ein Beispiel einer solchen XSLT-Datei ist in $\epsilon\chi\text{T}_{\text{E}}\text{X}$ enthalten.

```
java de.dante.util.xslt.Transform /tmp/cmr12.xml
    src/xslt/tfm2html.xsl /tmp/cmr12.html
```

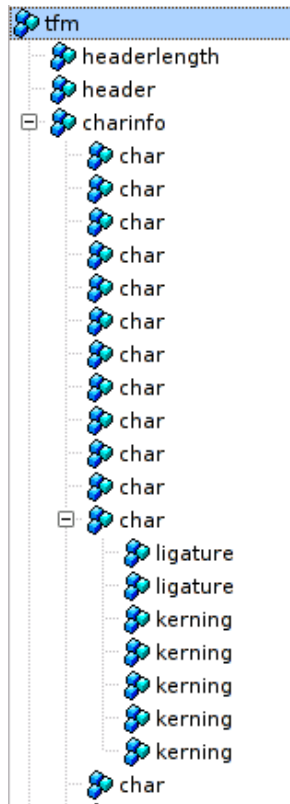


Abbildung 1: Grafische Struktur einer EFM-Datei

| Font | | | | | | | | | | | | | | | |
|---|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| 1. Fontfamily CMR 2. Designsize 12.0 3. Checksum 1487622411 4. Codingscheme TEX TEXT 5. Fonttype VANILLA 6. Sevenbitsafe false 7. Xeroxfacecode 230 | | | | | | | | | | | | | | | |
| Übersicht | | | | | | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |

Abbildung 2: Ansicht mit dem Browser

| 102 | f | | | | |
|-----|---|------------------|---------------|---------------|---------------|
| | | width | height | depth | italic |
| | | 299.186707 | 694.443703 | 0 | 69.73362 |
| | | letter-id | letter | lig-id | lig |
| | | 105 | i | 12 | |
| | | 102 | f | 11 | |
| | | 108 | l | 13 | |
| | | id | char | size | |
| | | 39 | ' | 69.73362 | |
| | | 63 | ? | 69.73362 | |
| | | 33 | ! | 69.73362 | |
| | | 41 |) | 69.73362 | |
| | | 93 |] | 69.73362 | |

Abbildung 3: Ansicht des Zeichens »f«

| | |
|-------------|---|
| Java-Klasse | Beschreibung |
| TFM2XML | Wandelt eine TFM-Datei in eine XML-Datei um (alle Informationen der TFM-Datei werden komplett übernommen). |
| TFM2EFM | Wandelt eine TFM-Datei in eine EFM-Datei um. Es werden nur die zusammengesetzten Informationen verwendet, die der Typesetter benötigt. |
| TFM2PL | Wandelt eine TFM-Datei in das PL-Format (wie <i>tftopl</i>) um. (Auch hier vergleicht ein JUnit-Test die beiden Ergebnisse und meldet entsprechende Fehler). |

Tabelle 1: Java-Klassen, die TFM-Dateien umwandeln

Verwendet man zusammen mit der HTML-Datei eine CSS-Datei – ein Beispiel ist bei $\epsilon\chi\text{T}_{\text{E}}\text{X}$ enthalten –, so kann man das Aussehen deutlich aufpeppen (siehe Abbildung 2). Dabei werden zuerst allgemeine Informationen des Fonts und eine Übersicht (mit Link) aller vorhandenen Zeichen angezeigt. Durch einen Klick kommt man dann direkt zu den Informationen des entsprechenden Zeichens (siehe Abbildung 3). Dabei werden die Breite, Höhe, Tiefe und der Wert für italic-Korrektur angezeigt. Ist für den Buchstaben eine Ligatur definiert, wird diese direkt darunter angezeigt, ebenso wird mit dem Kerning verfahren.

Ansicht anpassen

Wem die Ansicht oder die Darstellung der Informationen nicht gefällt, kann sehr leicht eine entsprechende Anpassung durchführen. $\epsilon\chi\text{T}_{\text{E}}\text{X}$ mit allen seinen Dateien findet man unter <http://www.extex.org>.

Zusammenfassung

Mit $\epsilon\chi\text{T}_{\text{E}}\text{X}$ lassen sich TFM-Dateien in XML, EFM oder in andere Formate wie PL einfach umwandeln. Durch das flexible XML-Format lässt sich aus diesen Informationen eine beliebige andere Darstellung erzeugen. $\epsilon\chi\text{T}_{\text{E}}\text{X}$ verwendet übrigens XML-Dateien nicht nur für die Metriken, sondern für vieles mehr. Es gibt sogar schon ein XML-Backend (Testphase), der alle Informationen des Typesetters in eine XML-Datei schreibt, so dass man mit einer entsprechenden Transformation beliebige andere Formate erzeugen kann.