

# Zpravodaj Československého sdružení uživatelů TeXu

---

Tomáš Mráz

Postscriptová písma v TeXu

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 4 (1994), No. 2, 70–76

Persistent URL: <http://dml.cz/dmlcz/149705>

## Terms of use:

© Československé sdružení uživatelů TeXu, 1994

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:  
*The Czech Digital Mathematics Library* <http://dml.cz>

Tento článek by měl pojednávat o mých zkušenostech s použitím postscriptových písem v T<sub>E</sub>Xu. Zmíním se zejména o postupech jejich převodu do podoby použitelné s T<sub>E</sub>Xem a problémech, se kterými jsem se při něm setkal. Vůbec se nebudu zabývat problémy použití postscriptových písem v matematické sazbě, neboť jsem zatím neměl příležitost setkat se s postscriptovými fonty vhodnými pro tento druh sazby, zejména co se týče matematických symbolů.

Při konverzi postscriptových písem existuje několik možných cest, po kterých je možno postupovat. Zmíním se zde o některých z nich.

## Konverze metrických informací

### Cesta první — česká písma

Podaří-li se nám sehnat někde kvalitní česká písma, která obsahují české znaky, pak máme za předpokladu, že jsou skutečně kvalitní, již zčásti vyhráno. Problémem však zůstává, kde a za jakou cenu lze tato písma sehnat. Z mých zkušeností s písmi vyrobenými či lépe řečeno počestěnými českými firmami vyplývá, že jsou většinou horší kvality a že způsob jejich počestění je značně nekompatibilní se standardními postscriptovými fonty. Například názvy jednotlivých českých písmen neodpovídají tomu, co se pod nimi ve skutečnosti skrývá. Třeba pod názvem *ydieresis* se schovává písmeno ř. Další vadou těchto fontů bývá vypuštění ligatur *fi* a *fl*, malý počet kerningových párů a nedostatečné nebo přímo chybné hinty. Naopak je jasné, že tyto fonty mají obrovskou výhodu v ceně oproti písmům firem Adobe a URW, což jsou jediná zahraniční písma, o kterých vím, že se u nás prodávají a obsahují české znaky.

Přesto pokud se nám podaří tato písma sehnat, přichází další problém. Zejména u méně kvalitních fontů nenajdeme soubory s příponou *.afm*, což jsou soubory Adobe Font Metrics, obsahující metrické informace k fontům. Vzhledem k tomu, že tato písma jsou určena k použití pod MS Windows, jsou k nim dodávány soubory *.pfm*. Získat soubory

.afm k počestěným fontům je možné, pokud je mi známo, snadno pouze pomocí komerčních programů, jako je například Fontographer.<sup>1</sup>

Existuje ještě jedna cesta, kterou však získáme pouze informace o velikosti jednotlivých písmen. Pomocí chytrě napsaného postscriptového makra můžeme přinutit tiskárnu, nebo lépe nějaký interpret PostScriptu — typicky Ghostscript, aby si jednotlivá písmena vykreslil a pak nám sdělil, jak jsou velká. Bohužel tak nemůžeme dostat informace o kerningových párech. Tyto bychom naopak mohli vyextrahovat ze souborů .pfm, ale o žádném programu, který by to uměl, nevím. Na druhou stranu, pokud fonty zakoupíme přímo u firmy, která je vyrobila, můžeme požádat o vygenerování souborů .afm přímo ji.

Další úkol, který před námi stojí, je zjistit kódování fontu s názvy jednotlivých písmen, což je opět nejjednodušší pomocí programu Fontographer. Pokud známe anglické názvy akcentů, které používá firma Adobe, a odpovídají-li názvy písmen jejich tvarům, pak se lze jednoduše podívat do souboru .afm a názvy jednotlivých písmen si opsat. Protože kódování zcela jistě nebude odpovídat tomu, které bychom potřebovali pro práci s  $\TeX$ em, musíme provést překódování fontu, ke kterému máme dvě příležitosti. Jednak můžeme použít PostScript a jeho kódový vektor, a nebo zajistit změnu kódů jednotlivých písmen pomocí virtuálních fontů. Jednodušší se mi zdá být cesta použití virtuálních fontů, ačkoliv předpokládá, že máme *dvi* ovladače, které si s virtuálními fonty rozumějí. Naštěstí ovladače Eberharda Mattese mezi ně patří, jakož i program DVIPS od Tomase Rokického.

Teď již přistoupíme k vlastní konverzi metrik. Vytvoříme si soubor `czencode.enc`, do kterého zapíšeme něco takového:

```
/CzEncoding [
/Gamma /Delta /Theta ...
%další řecká písmena pokud nějaká v našem fontu jsou
/Phi /Psi /Omega /ff /fi /fl /ffi /ffl /dotlessi /dotlessj /grave
/acute /caron /breve /macron /ring /cedilla /germandbls /ae /oe
/oslash /AE /OE /Oslash /.notdef /exclam /quotedblright
/numbersign /dollar /percent /ampersand /quoteright /parenleft
/parenright /asterisk /plus /comma /hyphen /period /slash /zero
```

---

<sup>1</sup> Nebo FontMonger (oba programy existují pro MS Windows). Nedávno se objevil ve volném prodeji CD-ROM s českými verzemi písem z programu Corel 3.0, na němž najdeme i odpovídající afm soubory. Na rozdíl od jiného balíku těchto fontů však zcela chybějí některé standardní znaky jako ligatury apod. Pokud je mi známo, jsou fonty pro tento program počestovány automaticky (asi podobně jako když použijeme accents), tuto verzi jsem však ještě nestihl prozkoumat. (*Pozn. red.*)

```

/one /two /three /four /five /six /seven /eight /nine /colon
/semicolon /guilsinglleft /equal /guilsinglright /question
/at /A /B /C /D /E /F /G
/H /I /J /K /L /M /N /O
/P /Q /R /S /T /U /V /W
/X /Y /Z
/bracketleft /quotedblleft /bracketright /circumflex /dotaccent
/quoteleft /a /b /c /d /e /f /g
/h /i /j /k /l /m /n /o
/p /q /r /s /t /u /v /w
/x /y /z
/endlash /emdash /hungarumlaut /tilde /dieresis
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /perthousand
/.notdef /.notdef /.notdef /Agrave /.notdef /.notdef /.notdef
/hyphen /ogonek /guillemotleft /quillemotright /.notdef /.notdef
/.notdef /.notdef /.notdef /Lcaron /.notdef /.notdef
% nejsem si jist jestli se L jmenuje Lcaron, podobně l, t, d
/.notdef /Scaron /.notdef /Tcaron /.notdef /.notdef /Zcaron /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /lcaron /.notdef
/.notdef /agrave /scaron /.notdef /tcaron /.notdef /.notdef /zcaron
/.notdef /Racute /Aacute /.notdef /.notdef /Adieresis /Lacute
/.notdef /.notdef /Ccaron /Eacute /.notdef /.notdef /Ecaron /Iacute
/.notdef /Dcaron /.notdef /.notdef /Ncaron /Oacute /Ocircumflex
/.notdef /Odieresis /.notdef /Rcaron /Uring /Uacute /.notdef
/Udieresis /Yacute /.notdef /.notdef /racute /aacute /.notdef
/.notdef /adieresis /lacute /.notdef /.notdef /ccaron /eacute
/.notdef /.notdef /ecaron /iacute /.notdef /dcaron /.notdef /.notdef
/ncaron /oacute /ocircumflex /.notdef /odieresis /.notdef /rcaron
/uring /uacute /.notdef /udieresis /yacute /quotedblbase
/quotedblleft ]
% LIGKERN hyphen hyphen =: endash ; endash hyphen =: emdash ;
% LIGKERN quoteleft quoteleft =: quotedblleft ;
% LIGKERN quoteright quoteright =: quotedblright ;
% LIGKERN guilsinglleft guilsinglleft =: guillemotleft ;
% LIGKERN guilsinglright guilsinglright =: guillemotright ;
% frenchdblquotes
% LIGKERN comma comma =: quotedblbase ;
% czquoteleft
% LIGKERN space {} * ; * {} space ; zero {} * ; * {} zero ;
% LIGKERN one {} * ; * {} one ; two {} * ; * {} two ;
% LIGKERN three {} * ; * {} three ; four {} * ; * {} four ;
% LIGKERN five {} * ; * {} five ; six {} * ; * {} six ;
% LIGKERN seven {} * ; * {} seven ; eight {} * ; * {} eight ;
% LIGKERN nine {} * ; * {} nine ;
% LIGKERN question {} quoteleft ; exclam {} quoteleft ;
% obrácený otazník a vykřičník nejsou pro běžné použití třeba,
% lomené závorky se hodí víc

```

S největší pravděpodobností bude třeba v závislosti na konkrétním fontu něco změnit.

Pak již můžeme pomocí programu AFM2TFM zkonvertovat soubor `.afm` na soubory `.vf` a `.tfm`. Použijeme příkaz

```
afm2tfm <type1font> -o czencode.enc -v <virtfont> <rawfont>.
```

Přípony není třeba uvádět. Program vytvoří dva soubory: 1. *raw tfm* — soubor, který obsahuje pouze metriky fontu v původním kódování bez ligatur a kerningů. 2. *virtual property list .vpl* — obsahuje metriky fontu v našem kódování včetně ligatur a kerningů a definici virtuálního fontu pomocí fontu *raw*. Pomocí programu VPTOVF překonvertujeme soubor `.vpl` do souborů čitelných T<sub>E</sub>Xem — vzniknou soubory `.vf` pro příslušný ovladač a `.tfm` pro T<sub>E</sub>X. Nyní je již stačí umístit do správných adresářů a můžeme náš nový font používat při sazbě. Používáme vždy virtuální soubor `.tfm`.\*

### Cesta druhá — písma v kódování Adobe Standard Encoding

Tyto fonty můžeme sehnat pomocí internetovské služby *ftp*, ale většinou se jedná o nepřiliš kvalitní písma. Pokud bychom chtěli jejich kvalitnější verzi zakoupit, většinou se nám vyplatí koupit českou verzi. Jediný důvod by mohl být ten, že chceme právě a jedině font, který v české verzi ani od firmy Adobe, ani od URW, případně od jiných neexistuje.

Pak musíme použít program ACCENTS (L2ACCENT) pro vytvoření českého virtuálního fontu z původního písma. Nejprve ale vytvoříme z původního `.afm` souboru soubor `.tfm`. Tento soubor však nesmí být nijak překódovaný a neměly by v něm chybět kerningy a ligatury, pokud je chceme mít ve výsledném fontu. Nejlepší je proto použít program `aftopl` od pana Clarka pro konverzi souboru `.afm` na soubor `.pl` a program `pltotf` na konverzi souboru `.pl` na `.tfm`. Program `aftopl` jsem našel ve zdrojovém tvaru na `ftp.muni.cz`. Na takto vytvořenou metriku nám již nic nebrání vypustit program ACCENTS — kódování Cork alias DC, nebo L2ACCENT — kódování Latin 2 upravené pro použití se standardní distribucí  $\zeta$ T<sub>E</sub>Xu. Použijeme příkaz `accents <rawtfm.tfm> <virtfont>`

---

\* To ovšem není nutné, zvolíme-li při zadání kódovacího souboru `-T` místo `-o`. Pak budou oba vzniklé `.tfm` soubory kódovány shodně a na identickou permutaci znaků soubor `.vf` přeci nepotřebujeme. Budeme pak samozřejmě zas používat virtuální `.tfm` soubor, jen vymažeme *raw .tfm* a `.vf`. Aby to navíc nebylo tak jednoduché, verze `afm2tfm` 7.8 — nejnovější, kterou mám k dispozici — namísto `-o` má `-p|t|T`. (Pozn. red.)

[`virtfont.adj`], kterým vytvoříme metriky použitelné pro naši další práci s  $\TeX$ em. Pokud by se nám postavení akcentů nad některými písmeny nelíbilo, můžeme jejich polohu upravit pomocí souboru `.adj` se syntaxí stejnou jako u souborů `.pl` a `.vp1`, kde u každého znaku, u kterého chceme upravit postavení akcentů, uvedeme direktivu `RIGHT`, `LEFT` resp. `UP` a `DOWN`. Velikost parametrů doporučuji zjistit experimentálně. Například:

```
(COMMENT Toto je komentář )
(DESIGNUNITS R 10)
(COMMENT Počet jednotek, ve kterých zadáváme parametry,)
(COMMENT na velikost písma)
(COMMENT Teď následují jednotlivé znaky, které chceme měnit)
(CCHARACTER 0 206 (RIGHT R 1.25)(UP R .5))
(COMMENT Toto je znak s kódem oktálově 206)
```

Pokud metriku vytvořenou pomocí tohoto programu začneme používat, zjistíme, že neobsahuje některé znaky, které se v českých *cmr* fontech vyskytují, patří mezi ně například francouzské a české uvozovky. Tento nedostatek se mi bohužel podařilo odstranit pouze přímou editací vzniklého souboru `.vf` a `.tfm`, samozřejmě po jejich konverzi do čitelného souboru `.vp1`. Zřejmě by bylo vhodné upravit program `ACCENTS`, aby příslušnou úpravu za nás udělal sám, pokud ovšem neexistuje jiná jednodušší cesta, jak to udělat.

## Konverze vlastního fontu — outline křivek

### Cesta první — nekonvertujeme

Touto cestou se dáme, máme-li postscriptovou tiskárnu nebo emulátor PostScriptu.

Chceme-li pro výstup na postscriptovou tiskárnu použít program DVIPS od Tomase Rokického musíme uvést do seznamu postscriptových fontů (v souboru `psfonts.map`) něco jako:

```
rptmr Times-Roman <c:\psfonts\tir____.pfb
```

`rptmr` je název našeho souboru *raw* `.tfm` bez přípony, `Times-Roman` je skutečný název našeho nového fontu — najdeme jej na začátku souboru `.afm` v položce `FontName`, `<c:\psfonts\tir____.pfb` je cesta k našemu souboru s písmem.

Pak by již mělo být možné soubor pomocí DVIPS vytisknout — DVIPS připojí náš soubor s písmem na začátek výstupního souboru. Pokud neu-

vedeme cestu k fontu, bude DVIPS považovat font za rezidentní v tiskárně a do výstupního souboru jej nepřidá.

### **Cesta druhá — konverze programem PS2PK**

Tato cesta je snad ještě jednodušší. Seženeme si program PS2PK nejlépe v co nejnovější verzi zkompileované pomocí DJGCC. Ve verzi 1.4 se tento program vyskytuje na `ftp.muni.cz`.

Příkazem

```
ps2pk -X<rozlišení v dpi> -P<velikost písma v pt> <font.pfb>
```

vygenerujeme soubory `.pk`. Maximum velikosti písma je v této verzi 60 pt pro rozlišení 1 270 dpi. Výsledný soubor stačí vhodně přejmenovat,\* přemístit do patřičného adresáře a vyzkoušet si, že skutečně funguje např. i s ovladačem `dviscr`.

### **Nejdůležitější oblasti, které jsem čtenáři zamlčel**

Jednak jsem se prakticky nezmínil o možnosti měnit kód písma pomocí kódového vektoru PostScriptu, neřekl jsem nic o postupu, jak využít písma, která se sice v písmu nacházejí, ale nevyskytují se v původním kódovém vektoru, ani jsem se nezabýval podrobnostmi a možnými chybami při práci s uvedenými programy. Možná, že se o to pokusím někdy příště.

### **Doporučená literatura, z níž jsem čerpal**

- [1] Dokument `aboutacc.tex` od Petra Sojky
- [2] Dokumentace k programu DVIPS a AFM2TFM — `dvips.tex` od Tomase Rokického

Oba dva dokumenty doporučuji přečíst každému vážnějšímu zájemci.

### **Poděkování**

Donaldu E. Knuthovi za jeho skvělý program  $\TeX$ .  
Eberhardu Mattesovi za jeho implementaci  $\TeX$ u pro počítače PC.  
Tomasi Rokickimu za jeho programy DVIPS a AFM2TFM.  
Pietu Tutelaersovi za jeho program PS2PK.

---

\* Žádoucí jméno (i s cestou) lze připojit na konec příkazového řádku. (*Pozn. red.*)

Všem členům  $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}\mathcal{U}$ , kteří se podíleli na tvorbě české distribuce  $\text{emT}\mathcal{E}\mathcal{X}\mathcal{U}$  —  $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ .

Jiřímu Zlatuškoví za jeho program ACCENTS.

*Tomáš Mráz*

`xmr@cslab.felk.cvut.cz`

---

---

## Záludnosti PostScriptu

ZDENĚK WAGNER

Každý, kdo se vážně zabývá typografií, dostane se dříve nebo později k PostScriptu. PostScript je jazyk pro popis stránek, který poskytuje další možnosti, jež lze  $\mathcal{T}\mathcal{E}\mathcal{X}$ em a  $\mathcal{M}\mathcal{E}\mathcal{T}\mathcal{A}\mathcal{F}\mathcal{O}\mathcal{N}\mathcal{T}\mathcal{E}\mathcal{M}$  realizovat jen s obtížemi nebo vůbec. To samozřejmě vyžaduje určité znalosti. Lze se však spokojit s přístupem uživatele, který programem  $\mathcal{D}\mathcal{V}\mathcal{I}\mathcal{P}\mathcal{S}$  vytvoří PostScriptový soubor a ten vytiskne na PostScriptové tiskárně nebo osvitové jednotce. Přesto je vhodné o PostScriptu něco vědět, aby se člověk vyhnul záludnostem, které v sobě skrývá jak PostScript tak další  $\mathcal{T}\mathcal{E}\mathcal{X}$ ovské pomocné programy.

Murphy by PostScript definoval jako nástroj, s jehož pomocí lze ve velmi krátkém okamžiku znehodnotit značné množství drahého materiálu. Experimentálně bylo totiž ověřeno, že rychlost PostScriptového zařízení je nepřímo úměrná smysluplnosti tištěného textu.

Smyslem tohoto článku je upozornit na hlavní záludnosti. Vše, co bude dále uvedeno, bylo objeveno většinou metodou pokusů a omylů a bohužel i finančně nákladných chyb. Jiná, zřejmě i lepší řešení jsou jistě možná. . .

Moderní tiskárny mají PostScript level 2, zatímco starší zařízení znají pouze level 1. Nebudeme se zde příliš zabývat rozdíly. Téměř vše, co je v tomto článku uvedeno, platí pro level 1. Odlišnosti budou zdůrazněny pouze tam, kde mohou působit problémy.

PostScript je podobně jako  $\mathcal{T}\mathcal{E}\mathcal{X}$  interpretační jazyk. Společným znakem je možnost definice uživatelských maker. Oba jazyky však mají řadu odlišností. Hlavní rozdíl spočívá v tom, že PostScript neumí lámat řádky a odstavce. V PostScriptu je nutno popsat již zlomenou stránku.