Fonts with complex OpenType tables

Karel Píška

Institute of Physics, Academy of Sciences
Prague, Czech Republic

17 September 2010

Introduction

Probably not many people are designing OpenType fonts with such technical work as creating OpenType tables is.
Is somebody going to produce OT fonts and OT tables?
I think if someone will be interested in the matter (s)he can read my proceeding article and discuss after the presentation. Anyway, please report about my mistakes.
In my presentation I want to describe principles of OT tables and overview development of the fonts with its problems more shortly and then demonstrate "How OT works" on several examples.

My starting point were two METAFONT fonts covering Czech and Georgian handwriting:
the font [slabikar] designed and created in METAFONT by Petr Olšák (1997)
and my [Georgian] fonts from 1998.
And may aim was to create the OT versions of that fonts.
Now only one common comment: I have not solved the problems with linespacing and unification: Czech and Armenian handwriting are oblique and Georgian is upright

OpenType tables principles

Tables (names) in OTF (PS/Type1 flavoured OT)
and in TTF* (TrueType flavoured OT)

head	
hhea	
maxp	
OS/2	
hmtx	
cmap	
loca	
CFF	glyf*
name	
post	
gasp	
FFTM	
GDEF	
GPOS	are
GSUB	important

The tables in OTF/TTF are binary. Human readable form is provided by TTX [Just van Rossum] converting OT into XML and back: [ttx], [ttx-GSUB], [ttx-GPOS] is difficult for writing and editing.
Therefore there are tools to define the tables in a source textual form and then compile them into binaries.
VTP – VOLT project format: [hwu-vtp]
FEA – Feature file specification: [hwu-fea]
VOLT (Microsoft) can import VTP files
makeotf (Adobe AFDKO), FontForge and commercial FontStudio (FontLab) can import FEA files and generate OT binaries. These programs allow also enter and edit OT data in interactive way. But I did not use that facilities because filling the form for hundreds rules is inefficient.
I will show some of the examples I prepared in my [proceedings article].

Overview of development

Creating (semi-manual editing) of source files (VTP and FEA) defining OT tables (glyph lists, features, lookups: substitution and positional rules, . . .)

Generating binaries (final OTF/TTF) with

VOLT (from VTP), FontForge and AFDKO (from FEA)

Proofing with VOLT, FontForge

Testing with XƎTEX, ConTEXt

Another attempts, e.g. conversions

Problems during development

Creating and editing of source files has be done in a semi-manual way.
Interactive tools (in VOLT, FontForge) to entry hundreds rules is not efficient.
FontForge generated wrong GSUB according ConTEXt
AFDKO (Adobe) generated wrong GPOS according XƎTEX
I has not been successful with any conversions (METAFONT, VOLT based TTF, FEA based OTF)

Demonstrations

OT tables are instructions . . .
and their interpretation and rendering is solved by text layout engines connect with operating systems or application programs
VOLT and FontForge proofing tools have their own internal engines
XƎTEX delegates font handling to libraries (ICU, ATSUI, Graphite, . . .) for rendering and positioning

Proofing with VOLT and FontLab

I use free software and do not have the commercial FontLab Studio.
VOLT is available only for MS Windows and, unfortunately, proofing Georgian is impossible in FontForge under Windows; I use FontForge on Linux.

Proofing with FontForge

I have prepared several [words] for proofing and demonstration “How OT (features and lookup rules) works.”

(switching keyboard layout/mapping:
setxkbmap us ; setxkbmap cz ; setxkbmap ge ;)

XƎTEX

XƎTEX can read Unicode input stream, use OT fonts directly and produce PDF with the OT fonts inside: [XƎLATEX source][XƎLATEX samples]

ConTeXt

Also ConTeXt (Mark IV) offers the usage of OT fonts:
[Font definitions] [ConTeXt source] [pdf]

ConTeXt testing: showotfcomposition

ConTeXt allows to trace processing of OTF features and lookups step by step: [source][pdf]

Conclusion

I am able to produce OT fonts working properly (i.e. according my expectations) with XƎTEX (`xelatex`) and ConTEXt Mark IV (`context`)

We have two different versions of OT fonts producing the same results:

VOLT based TTF

FEA based OTF

The fonts generated from the same FEA sources by *makeotf* (AFDKO) and by FontForge are different, they may contain (different!) errors or are not compatible with XƎTEX and ConTEXt.

Automatic conversion of substition and positioning tables between METAFONT/TFM, VOLT based TTF, FEA based OTF is impossible or does not work properly with tools available (to me).