

Some thoughts on typescripts

Mojca Miklavec, the 4th ConT_EXt meeting

When writing or adapting typescripts for the excellent Polish fonts (or any other font for that matter), I sometimes lack a few features. I will try to highlight some of them and do a quick comparison between ConT_EXt (traditional, simplefonts) and L^AT_EX (NFSS, fontspec) in a short talk, followed by discussion.

Disclaimer:

- This presentation may not be “politically correct” at all the places: I’m not confident with all the font-related terms and please feel free to report any errors that you might find. I would like you to consider this presentation as an introduction to brainstorming about what could be improved.
- It would probably require modifications of some parts of ConT_EXt core in order to support the functionality mentioned here.
- There is NO sample code available.

Motivation

For both novice and some experienced T_EX users there are two areas where GUI programs are much easier to use:

- TABLES: partially covered in another talk
- FONTS: I will try to cover some aspects of that here

What makes fonts difficult to use?

- user has no simple overview of available fonts
in Word it's just a drop-down menu
- in pre-X_fT_EX era one needed complicated procedures for font conversion
 - zillions of files and encodings that had to be put on proper place
 - 256-character limitation
 - clumsy namesin Word it's ready-to-use
- in Word **bold** and *italic* work out of the box
in T_EX one needs to spend an afternoon before all the names get properly assigned

What revolution did X_YTeX bring?

- a lot of nice functionality that will not be our main focus here (like multidirectional writing, Unicode, support for exotic scripts, ...)
- in my opinion THREE aspects revolutionized & simplified font selection enormously:
 1. support for standard font formats: OpenType, AAT, ... & advanced features
 2. support for system fonts (C:\Fonts, fontconfig) – works out of the box, no need for any advanced knowledge about TEXMF trees
 3. automatic recognition of Regular / Bold / Italic / Bold Italic & Small Caps

and as a consequence:

1. no conversion, no additional files
2. no need to move files around once you install the font
3. complex .fd file/typescript replaced by oneliner:

```
\setmainfont{TeX Gyre Bonum} (fontspec)
```

```
\definetypface[bonum][rm][Xserif][TeX Gyre Bonum] (ConTeXt with XYTeX)
```

Differences between X_YTeX and LuaTeX

- LuaTeX also knows a lot (including names) about fonts in TEXMF trees
- X_YTeX only knows filenames of those fonts (but could use some info provided by LuaTeX)
- X_YTeX is slightly better integrated into system
(knows about special folders where users 'hide' their fonts)
- LuaTeX should be capable of extracting a bit more heuristics about fonts (extended etc.)

Different approaches for font handling

	pdfTeX	X _Y TeX	LuaTeX
plain TeX	<code>\font\ a=ec-lmr12</code>	<code>\font\ a="Times;+smcp"</code>	lua-based
L ^A TeX	NFSS	fontspec	fontspec
ConTeXt	typescripts	[Xserif] [Times]	simplefonts

NFSS – New Font Selection Scheme

<http://www.ctan.org/tex-archive/macros/latex/doc/fntguide.pdf>

- used in 8-bit engines in L^AT_EX
- five attributes: **encoding**, **family**, **series**¹, **shape**², **design size**
- **series** is a combination of **weight**³ and **width**⁴ with predefined values⁵
- I like the system & its flexibility to select fonts:

```
\fontfamily{ccr}\fontseries{c}\fontshape{sl}\fontsize{9}{11pt}\selectfont
```

but not the complexity of writing .fd files (example from t1anttlc.fd):

```
\DeclareFontShape{T1}{anttlc}{ebx}{n}{<->ec-anttb}{}  
... (32 lines)
```

```
\DeclareFontShape{T1}{anttlc}{sbx}{scit}{<->ec-anttricap}{}  
... (32 lines)
```

Antykwa Toruńska contains 1504 such lines (64 files) to cover 32 fonts in 11 encodings.

¹ **series**: ‘medium weight’, ‘bold extended’, ...

² **shape**: regular, italic, slanted, upright

³ **weight**: ul-Ultralight, el-Extralight, l-Light, sl-Semilight, m-Medium, sb-Semibold, b-Bold, eb-Extrabold, ub-Ultrabold

⁴ **width**: uc-Ultracondensed, ec-Extracondensed, c-Condensed, sc-Semicondensed, m-Medium, sx-Semiexpanded, x-Expanded, ex-Extraexpanded, ux-Ultraexpanded

⁵ To combine the abbreviations, weight is used first and any instance of medium (m) is dropped except when weight and width are both medium. In this case one single m is used.

Fontspec

<http://www.ctan.org/tex-archive/macros/latex/contrib/fontspec/fontspec.pdf>

<http://github.com/wspr/fontspec/>

- probably the most important package for X_YL^AT_EX, also for Lua^AT_EX
- revolutionary **powerful & easy to use**
- written with users, not with developers in mind (like NFSS)
- support calling font names with automatic bold/italic assignments, file names, setting path where fonts reside, explicitly defining which font files to use, advanced and non-criptic dealing with OpenType features, full support for optical sizes, ...
- some examples:
 - `\fontspec {Cambria}`
 - `\fontspec [Path = /Users/will/Fonts/ ,`
 - `UprightFont = *-regular ,`
 - `BoldFont = *-bold ,`
 - `...]`
 - `{texgyrepagella}`
- Anyone feels like writing `m-fontspec.tex`? :)

Simplefonts

(they deserve more attention, but I ran out of time . . .)

What I miss in ConTEXt font handling?

- no easy way to switch to Bold Italic Caps
I would like to use `\section{Titles with {\sc Small Caps}}` reliably even when `\setuphead[section][style=\bsb]` (might already be possible?)
- no standard way to use Condensed, Extended, Semi Bold, ...
 - thus there are no predefined typescript to simplify life
 - lengthy new definitions are needed
 - or we would end up with exponential number of predefined ones
- writing typescripts without Wolfgang's module is way too cumbersome:
I would be glad to have less overhead
- Wolfgang's module only works with MKIV and would not handle messy fonts properly:
human brain is still needed from time to time to help ConTEXt know more about fonts

Proposal – Hypercubes (font properties in higher dimensions)

Mandatory & with known meaning:

- **shape** (Regular / Italic / Slanted) – italic & slanted falling back to each other if one undefined
- **weight** (Light / Regular / Medium / Bold / Heavy / ...) – arbitrary names; user just selects a pair to be associated with `rm` & `bf` at runtime (like: Medium-Heavy)

Optional & with known meaning:

- **encoding** – only for MKII (if supported at all)
- **caps** (Regular / Small Caps) – in MKIV this should work automatically

Fully optional, with arbitrary names & arbitrary number of them:

- **size** (Caption / Regular / Subhead / Display) or (6pt / 8pt / 10pt / 12pt / 17pt) – the meaning might be known in case that automatic optical sizes are to be supported
- **width** (Condensed / Semi Condensed / Regular / Semi Extended / Extended / ...)
- **anything** (Any / Name / That / Comes / To / Mind) – as long as keys and values are predefined in typescript with `[anything={any,name,that,comes,to,mind}]`.

JMN Antykwa Półtawskiego

weight	width	shape
4	5	2
Light Regular Medium Bold	Condensed Semi Condensed Regular Semi Extended Extended	Regular Italic

Total number of fonts:

$$4 \cdot 5 \cdot 2 = 40$$

```

\startsonewfonttypescriptcommand % complete typescript
[antykwa poltawskiego] % font name
[width={condensed,semicondensed,*regular,semiextended,extended}] % optional (1)
[weight={light,*regular,medium,*bold}] % mandatory (2)
[shape={regular,italic}]{ % mandatory
  { % condensed
    {AntPoltLtCond-Regular.otf, AntPoltLtCond-Italic.otf}, % light
    {AntPoltCond-Regular.otf, AntPoltCond-Italic.otf}, % regular
    {AntPoltMdCond-Bold.otf, AntPoltMdCond-BoldItalic.otf}, % medium
    {AntPoltCond-Bold.otf, AntPoltCond-BoldItalic.otf} % bold
  },{ % semicondensed
    {AntPoltLtSemiCond-Regular.otf, AntPoltLtSemiCond-Italic.otf}, % light
    {AntPoltSemiCond-Regular.otf, AntPoltSemiCond-Italic.otf}, % regular
    {AntPoltMdSemiCond-Bold.otf, AntPoltMdSemiCond-BdItalic.otf}, % medium
    {AntPoltSemiCond-Bold.otf, AntPoltSemiCond-BoldItalic.otf} % bold
  },{ % regular
    {AntPoltLt-Regular.otf, AntPoltLt-Italic.otf}, % light
    {AntPolt-Regular.otf, AntPolt-Italic.otf}, % regular
    {AntPoltMd-Bold.otf, AntPoltMd-BoldItalic.otf}, % medium
    {AntPolt-Bold.otf, AntPolt-BoldItalic.otf}, % bold
  },{ % semiextended
    {AntPoltLtSemiExtd-Regular.otf, AntPoltLtSemiExtd-Italic.otf}, % light
    {AntPoltSemiExtd-Regular.otf, AntPoltSemiExtd-Italic.otf}, % regular
    {AntPoltMdSemiExtd-Bold.otf, AntPoltMdSemiExtd-BdItalic.otf}, % medium
    {AntPoltSemiExtd-Bold.otf, AntPoltSemiExtd-BoldItalic.otf}, % bold
  },{ % extended
    {AntPoltLtExtd-Regular.otf, AntPoltLtExtd-Italic.otf}, % light
    {AntPoltExtd-Regular.otf, AntPoltExtd-Italic.otf}, % regular
    {AntPoltMdExtd-Bold.otf, AntPoltMdExtd-BoldItalic.otf}, % medium
    {AntPoltExtd-Bold.otf, AntPoltExtd-BoldItalic.otf}, % bold
  }
}}
\stopsonewfonttypescriptcommand

```

Usage

```
\switchtofontdimension[width=semicondensed,weight={light,medium}]  
{\rm This is Semi Condensed Light (AntPolLtLtSemiCond-Regular.otf)}  
{\it This is Semi Condensed Light Italic (AntPolLtLtSemiCond-Italic.otf)}  
{\sl This is the same as italic.}  
{\bf This is Semi Condensed Medium (AntPolMdSemiCond-Bold.otf)}  
{\bi This is Semi Condensed Medium Italic (AntPolMdSemiCond-BdItalic.otf)}  
{\bs This is the same as bold italic.}
```

Pro et contra

Advantages:

- full order
- arbitrary number of dimensions
- once the order is there, any font switching scheme on top of that could be built
- only a single typescript \implies many different combinations possible with one-liners
- uses human brain to order fonts, everything else is done by computer
 - close-to-zero overhead
 - no vendor-specific heuristics are needed: less errors in ‘just guessing’

Disadvantages:

- holes in fonts need to be filled completely (with fallbacks)

Not covered by the example:

- applying different font features to different fonts
 - *one could use* AntPoltLt-Regular.otf*default
- intermediate names
 - *one could use* AntykwaPoltawskiegoRegular=AntPoltLt-Regular.otf
 - *Serif could be deduced automatically*
- optical sizes as in LM (I can try to elaborate that)

‘Quick’ typescripts

If there is no predefined typescript, there must be an extra-quick way to set up the font:

```
\definesomenewfonttypescript % 'quick' typescript
[my-antykwa-poltawskiego] % font name
[rm=AntPoltLtSemiCond-Regular.otf,
it=AntPoltLtSemiCond-Italic.otf,
bf=AntPoltMdSemiCond-Bold.otf,
bi=AntPoltMdSemiCond-BdItalic.otf]
```

The following is FONTSPEC’s syntax (I like the idea a lot):

```
\fontspec
[BoldFont=AntPoltMdSemiCond-Bold.otf,
ItalicFont=AntPoltLtSemiCond-Italic.otf,
BoldItalicFont=AntPoltMdSemiCond-BdItalic.otf]
{AntPoltLtSemiCond-Regular.otf}
```

The old way

How many lines are needed to do the same in MKII or MKIV at the moment?

```
\starttypescript [serif] [my-antykwa-poltawskiego] % AP=AntykwaPoltawskiego, SC=SemiCondensed
  \definefontsynonym [AP-SC-Light] [file:AntPoltLtSemiCond-Regular] [features=default]
  \definefontsynonym [AP-SC-LightItalic] [file:AntPoltLtSemiCond-Italic] [features=default]
  \definefontsynonym [AP-SC-Medium] [file:AntPoltMdSemiCond-Bold] [features=default]
  \definefontsynonym [AP-SC-MediumItalic] [file:AntPoltMdSemiCond-BdItalic] [features=default]

  \definefontsynonym [AP-SC-CapsRegular] [file:AntPoltLtSemiCond-Regular] [features=smallcaps]
  \definefontsynonym [AP-SC-CapsItalic] [file:AntPoltLtSemiCond-Italic] [features=smallcaps]
  \definefontsynonym [AP-SC-CapsMedium] [file:AntPoltMdSemiCond-Bold] [features=smallcaps]
  \definefontsynonym [AP-SC-CapsMediumItalic] [file:AntPoltMdSemiCond-BdItalic] [features=smallcaps]
\stoptypescript

\starttypescript [serif] [my-antykwa-poltawskiego] [name]
  \definefontsynonym [Serif] [AP-SC-Light]
  \definefontsynonym [SerifBold] [AP-SC-Medium]
  \definefontsynonym [SerifItalic] [AP-SC-LightItalic]
  \definefontsynonym [SerifSlanted] [AP-SC-LightItalic]
  \definefontsynonym [SerifBoldItalic] [AP-SC-MediumItalic]
  \definefontsynonym [SerifBoldSlanted] [AP-SC-MediumItalic]
  \definefontsynonym [SerifCaps] [AP-SC-CapsRegular]
  % And how to use Bold Italic Small Caps?
\stoptypescript
```

I find this method too cumbersome to use ...

JMN Antykwa Półtawskiego (for pdfTeX)

weight	width (opt. size)	shape	encoding	caps
4	5	2	7	2
Light Regular Medium Bold	Condensed (17) Semi Condensed (12) Regular (10) Semi Extended (8) Extended (6)	Regular Italic	cs ec l7x qx t5 texnansi (ts1)	Regular Small Caps

Total number of fonts⁶:

$$4 \cdot 5 \cdot 2 \cdot 6 \cdot 2 \\ + 4 \cdot 5 \cdot 2 \cdot 1 \cdot 1 = 520$$

⁶ ts1 encoding doesn't need small caps


```

\startsonewfonttypescriptcommand % complete typescript
[antykwa poltawskiego][cs,ec,l7x,qx,t5,texnansi] % TODO % font name, encodings
[width={condensed,semicondensed,*regular,semiextended,extended}] % optional (1)
[weight={light,*regular,medium,*bold}] % mandatory (2)
[caps={no,yes}] % optional, but known
[shape={regular,italic}]{ % mandatory
  { % condensed
    {{\t2-antpl17, \t2-antpli17}, {\t2-antpl17-sc, \t2-antpli17-sc}}, % light
    {{\t2-antpr17, \t2-antpri17}, {\t2-antpr17-sc, \t2-antpri17-sc}}, % regular
    {{\t2-antpm17, \t2-antpmi17}, {\t2-antpm17-sc, \t2-antpmi17-sc}}, % medium
    {{\t2-antpb17, \t2-antpbi17}, {\t2-antpb17-sc, \t2-antpbi17-sc}} % bold
  },{ % semicondensed
    {{\t2-antpl12, \t2-antpli12}, {\t2-antpl12-sc, \t2-antpli12-sc}}, % light
    {{\t2-antpr12, \t2-antpri12}, {\t2-antpr12-sc, \t2-antpri12-sc}}, % regular
    {{\t2-antpm12, \t2-antpmi12}, {\t2-antpm12-sc, \t2-antpmi12-sc}}, % medium
    {{\t2-antpb12, \t2-antpbi12}, {\t2-antpb12-sc, \t2-antpbi12-sc}} % bold
  },{ % regular
    {{\t2-antpl10, \t2-antpli10}, {\t2-antpl10-sc, \t2-antpli10-sc}}, % light
    {{\t2-antpr10, \t2-antpri10}, {\t2-antpr10-sc, \t2-antpri10-sc}}, % regular
    {{\t2-antpm10, \t2-antpmi10}, {\t2-antpm10-sc, \t2-antpmi10-sc}}, % medium
    {{\t2-antpb10, \t2-antpbi10}, {\t2-antpb10-sc, \t2-antpbi10-sc}} % bold
  },{ % semiextended
    {{\t2-antpl8, \t2-antpli8}, {\t2-antpl8-sc, \t2-antpli8-sc}}, % light
    {{\t2-antpr8, \t2-antpri8}, {\t2-antpr8-sc, \t2-antpri8-sc}}, % regular
    {{\t2-antpm8, \t2-antpmi8}, {\t2-antpm8-sc, \t2-antpmi8-sc}}, % medium
    {{\t2-antpb8, \t2-antpbi8}, {\t2-antpb8-sc, \t2-antpbi8-sc}} % bold
  },{ % extended
    {{\t2-antpl6, \t2-antpli6}, {\t2-antpl6-sc, \t2-antpli6-sc}}, % light
    {{\t2-antpr6, \t2-antpri6}, {\t2-antpr6-sc, \t2-antpri6-sc}}, % regular
    {{\t2-antpm6, \t2-antpmi6}, {\t2-antpm6-sc, \t2-antpmi6-sc}}, % medium
    {{\t2-antpb6, \t2-antpbi6}, {\t2-antpb6-sc, \t2-antpbi6-sc}} % bold
  }
}
\stopsonewfonttypescriptcommand

```

Or an even faster way to define 480 fonts at once (can you beat it?):

```
\startsonewfonttypescriptcommand
  [antykwa-poltawskiego]
  [encoding={cs,ec,l7x,qx,t5,texnansi},replace=self] % 2
  [width={condensed,semicondensed,*regular,semiextended,extended},replace={17,12,10,8,6}] % 3
  [weight={light,*regular,medium,*bold},replace={l,r,m,b}] % 4
  [caps={no,yes},replace={,-sc}] % 5
  [shape={regular,italic},replace={,i}] % 6
    {{\t2}-antp{\t4}{\t6}{\t3}{\t5}}
\stopsonewfonttypescriptcommand
```

The \t2 stand for what is currently known as \typescripttwo etc.

Adobe Kepler® Std Opticals

weight	width	opt. size	shape
6	4	4	2
Light Regular Medium Semibold Bold Black	Condensed Semicondensed Regular Extended	Caption Regular Subhead Display	Regular Italic

Total number of fonts⁷:

$$6 \cdot 3 \cdot 4 \cdot 2 \\ + 6 \cdot 1 \cdot 2 \cdot 2 = 168$$

⁷ Condensed width has only two optical sizes (Subhead and Display – in implementation Caption and Regular would fall back on Subhead).

```

\startsonewfonttypescriptcommand
[kepler]
[width={condensed,semicondensed,*regular,extended}]
% caption < 10pt <= regular < 12pt <= subhead < 17pt <= display
% without switch=... one needs to switch manually with [size=display]
[size={caption,*regular,subhead,display},switch={10pt,12pt,17pt}]
[weight={light,*regular,medium,semibold,*bold,black}]
[shape={regular,italic}]{
  { % condensed
    ...
  },{ % semicondensed
    { % caption
      {KeplerStd-LightScnCapt.otf, KeplerStd-LightScnItCapt.otf}, % light
      {KeplerStd-ScnCapt.otf, KeplerStd-ScnItCapt.otf}, % regular
      {KeplerStd-MediumScnCapt.otf, KeplerStd-MediumScnItCapt.otf}, % medium
      {KeplerStd-SemiboldScnCapt.otf, KeplerStd-SemiboldScnItCapt.otf}, % semibold
      {KeplerStd-BoldScnCapt.otf, KeplerStd-BoldScnItCapt.otf}, % bold
      {KeplerStd-BlackScnCapt.otf, KeplerStd-BlackScnItCapt.otf} % black
    },{ % regular
      ...
    },{ % subhead
      ...
    },{ % display
      ...
    }
  },{ % regular
    ...
  },{ % extended
    ...
  }}
\stopsonewfonttypescriptcommand

```

Linotype Neue Helvetica®

weight	width	shape
9	3	2
Ultra Light Thin Light Regular Medium Bold Heavy Black Extra Black	Condensed Regular Extended	Regular Oblique/Italic

Total number of fonts⁸: $8 \cdot 3 \cdot 2 + 1 \cdot 1 \cdot 2 = 50$.

⁸ Extra Black only in Condensed width

```

\startsonewfonttypescriptcommand
[helvetica-neue]
[width={condensed,*regular,extended}]
[weight={ultralight,thin,light,*regular,medium,*bold,heavy,black,extrablack}]
[shape={regular,italic}]{
  { % condensed
    {HelveticaNeueLTStd-UltLtCn.otf, HelveticaNeueLTStd-UltLtCn0.otf}, % Ultra Light
    {HelveticaNeueLTStd-ThCn.otf, HelveticaNeueLTStd-ThCn0.otf}, % Thin
    {HelveticaNeueLTStd-LtCn.otf, HelveticaNeueLTStd-LtCn0.otf}, % Light
    {HelveticaNeueLTStd-Cn.otf, HelveticaNeueLTStd-Cn0.otf}, % Regular
    {HelveticaNeueLTStd-MdCn.otf, HelveticaNeueLTStd-MdCn0.otf}, % Medium
    {HelveticaNeueLTStd-BdCn.otf, HelveticaNeueLTStd-BdCn0.otf}, % Bold
    {HelveticaNeueLTStd-HvCn.otf, HelveticaNeueLTStd-HvCn0.otf}, % Heavy
    {HelveticaNeueLTStd-BlkCn.otf, HelveticaNeueLTStd-BlkCn0.otf}, % Black
    {HelveticaNeueLTStd-XBlkCn.otf, HelveticaNeueLTStd-XBlkCn0.otf}, % Extra Black
  },{ % regular
    {HelveticaNeueLTStd-UltLt.otf, HelveticaNeueLTStd-UltLtIt.otf}, % Ultra Light
    {HelveticaNeueLTStd-Th.otf, HelveticaNeueLTStd-ThIt.otf}, % Thin
    {HelveticaNeueLTStd-Lt.otf, HelveticaNeueLTStd-LtIt.otf}, % Light
    {HelveticaNeueLTStd-Roman.otf, HelveticaNeueLTStd-It.otf}, % Regular
    {HelveticaNeueLTStd-Md.otf, HelveticaNeueLTStd-MdIt.otf}, % Medium
    {HelveticaNeueLTStd-Bd.otf, HelveticaNeueLTStd-BdIt.otf}, % Bold
    {HelveticaNeueLTStd-Hv.otf, HelveticaNeueLTStd-HvIt.otf}, % Heavy
    {HelveticaNeueLTStd-Blk.otf, HelveticaNeueLTStd-BlkIt.otf}, % Black
    {<fallback, probably 'XBlkCn'>, <fallback, probably 'XBlkCn0'>} % Extra Black
  },{ % extended
    ...
  }}
\stopsonewfonttypescriptcommand

```

```

\startsonewfonttypescriptcommand % the same typescript, but with full style names
[helvetica-neue] % instead of cryptic file names; just for illustration
[width={condensed,*regular,extended}]
[weight={ultralight,thin,light,*regular,medium,*bold,heavy,black,extrablack}]
[shape={regular,italic}]{
  { % Condensed
    {27 Ultra Light Condensed, 27 Ultra Light Condensed Oblique}, % Ultra Light
    {37 Thin Condensed, 37 Thin Condensed Oblique}, % Thin
    {47 Light Condensed, 47 Light Condensed Oblique}, % Light
    {57 Condensed, 57 Condensed Oblique}, % Regular
    {67 Medium Condensed, 67 Medium Condensed Oblique}, % Medium
    {77 Bold Condensed, 77 Bold Condensed Oblique}, % Bold
    {87 Heavy Condensed, 87 Heavy Condensed Oblique}, % Heavy
    {97 Black Condensed, 97 Black Condensed Oblique}, % Black
    {107 Extra Black Condensed, 107 Extra Black Condensed Oblique} % Extra Black
  },{ % Regular
    {25 Ultra Light, 26 Ultra Light Italic}, % Ultra Light
    {35 Thin, 36 Thin Italic}, % Thin
    {45 Light, 46 Light Italic}, % Light
    {55 Roman, 56 Italic}, % Regular
    {65 Medium, 66 Medium Italic}, % Medium
    {75 Bold, 76 Bold Italic}, % Bold
    {85 Heavy, 86 Heavy Italic}, % Heavy
    {95 Black, 96 Black Italic}, % Black
    {<fallback>, <fallback>} % Extra Black
  },{ % Extended
    ...
  }}
\stopsonewfonttypescriptcommand

```