OpenType version of yfonts for Old German

Daniel Flipo

16th November 2022

This bundle provides OpenType versions of the Old German fonts *yfrak*, *ygoth* and *yswab* designed by Yannis Haralambous in Metafont (1990).

A style file `yfonts-otf.sty` is included to load these fonts easily; it is meant as a replacement for LuaLaTeX and XeLaTeX of `yfonts.sty`.

A Type 1 version of these fonts has been provided by Thorsten Bronger (2002).

The *yinit* font (initials) is already available as in OpenType format, thanks to Élie Roux.

Please beware of the *experimental* status of the current version (0.43).

All three fonts are covered by OFL licence, style file and documentation are under LPPL-1.3 licence.

1 Usage

`yfonts-otf.sty` loads `fontspec` and mimics what the `yfonts` package does for the Type 1 version: it defines three font families `\frakfamily`, `\gothfamily` and `\swabfamily` and the corresponding commands with arguments `\textgoth{}`, `\textfrak{}`, and `\textswab{}`.

All three families are loaded with all ligatures activated, an `s` automatically prints a long ſ (initial and middle form) or a round ſ (final form)[1]. Coding `"a`, `"e`, `"o`, `"u`, `"s` is only supported through Babel's German shorthands to produce ä, ë, ö, ü, ß. Nowadays, most keyboards give access to the Unicode characters ä, ë, ö, ü and ß, so typing them directly is a better alternative.

In the *yfrak* family, the command `\etc` prints &c a variant of the &c abbreviation while `\Jvar` prints J a variant of J (suggestion of Daniel Sanders, mentioned by Yannis).

The OpenType feature `Alternate=0` turns ä, ë, ö, ü into ä, ë, ö, ü. It works for the *yfrak* and *yswab* families but not for *ygoth* (variant not available in the original version).

The *yswab* family offers `CharacterVariant=1` (`+cv01`) and `CharacterVariant=2` (`+cv02`) which respectively change the exclam and question marks: ! into ! and ? into ?.

Used with the *ygoth* family, the `StylisticSet=1` (`+ss01`) feature provides variants for the long s and the derived ligatures: ſ, ſſ, ſſi, ſt are turned into ſ, ſſ, ſſi, ſt.

[1] See section 2 for details.

These features can be added locally anywhere in the document body, f.i.:
`\frakfamily\addfontfeature{Alternate=0}`
or using `yfonts-otf.sty`'s options `varumlaut` and `gothvarlongs`, f.i.:
`\usepackage[varumlaut]{yfonts-otf}`
which applies globally to both *yfrak* and *yswab* families.

It is also possible to use these fonts without loading `yfonts-otf.sty`, then I recommend to call them by *file name*, as XeTeX cannot find fonts in the texmf tree by *font name*[2], f.i.: `\setmainfont{yfrak.otf}[<options>]` or `\fontspec{yswab.otf}[<options>]`, this will work with both LuaTeX and XeTeX.

2 Coding the long/round s

The traditional German rules for long (ſ) and round (s) are somewhat complex, a summary can be found in the Unifraktur Maguntia Manual (`Dokumentation_en_fraktur.pdf`, [3]).

`yfonts-otf` borrows the automatic choice from the Unifraktur Maguntia fonts. It uses OpenType features (`ss11`), according to the authors it fails in less than 1 % of the occurences. When the algorithm fails, it is possible to force a round s (coding `s=` or `\shorts`) a long ſ (coding `ſ`[3] or `\longs`).

An alias is provided for this feature: `Style=longs` is the same as `StylisticSet=11` or `RawFeature=+ss11`.

Experts might want to type ſ (U+17F) or s (U+073) to keep the full control of the s form; this requires either to deactivate the `ss11` feature after loading the `yfonts-otf` package, or to use a direct `\setmainfont{}[]` or `\fontspec{}[]` call.

3 List of optional ligatures

Ligatures are split into three groups which may be deactivated globally or inside a group with the command `\addfontfeature{RawFeature=-`*ligname*`}`[4]

`\frakfamily`:

Name	Default (+)	Optional (−)
rlig	ch, ck, ſt, ß	ch, ck, ſt, tz
liga	ff, fi, fl, ffi, ffl, ff, ſſ	ff, fi, fl, ffi, ffl, ff, ſſ

`\swabfamily`:

Name	Default (+)	Optional (−)
rlig	ch, ck, ſt, tz	ch, ck, ſt, tz
liga	ff, fi, fl, ffi, ffl, ff, ſſ	ff, fi, fl, ffi, ffl, ff, ſſ

[2] Unless they have been declared as *System* fonts…
[3] On Unix systems the `Compose` key can be used: `Compose f s`.
[4] `yonts-otf` specifically defines `\ZWNJ` (`\char"200C`) to break unwanted ligatures: entziffern (no ß lig) can be coded `ent\ZWNJ ziffern` or `ent\ZWNJ{}ziffern`.

\gothfamily:

Name	Default (+)	Optional (−)
rlig	ch, ck, ſt, ſt, tz	ch, ck, ſt, ſt, tz
liga	ct, ff, fi, fl, ffi, ffl, ij, ll,	ct, ff, fi, fl, ffi, ffl, ij, ll,
	ſt, ſi, ſti, ſſ, ſi, ſſi,	ſt, ſi, ſti, ſſ, ſi, ſſi,
hlig	ba, be, bo, da, de, do, ha, he, ho,	ba, be, bo, da, de, do, ha, he, ho,
	pa, pe, po, pp, qq, ba, be, bu	pa, pe, po, pp, qq, ba, be, bu

4 Samples

A practical usage of these fonts can be found in file `Erlkonig.ltx` to be compiled with `lualatex`. It shows the beginning of Goethe's Erlkönig poem typeset with each of them.

5 Compatibility with other packages

microtype is compatible with `yfonts-otf` (protusion, expansion and letter spacing) but as we have no specific `mt-*.cfg` config file yet for the yfonts[5], adding
```
\DeclareMicrotypeAlias{yfrak.otf}{TU-basic}
\DeclareMicrotypeAlias{yswab.otf}{TU-basic}
\DeclareMicrotypeAlias{ygoth.otf}{TU-basic}
```
after loading `microtype` is recommended to avoid (lots of) warnings about missing characters.

soul is old (2003) and not recommended for OpenType fonts. Its command `\so{}` brakes ligatures (f.i. `\so{Wasser}`), for letter spacing `microtype`'s command `\textls{}` should be preferred. With LuaTeX, `lua-ul` is a much better choice for striking or underlining.

6 Acknowledgements

Great thanks to Keno Wehr for carefully testing the initial version and making valuable suggestions for improvements.

References

[1] Typesetting Old German: Fraktur, Schwabacher, Gotisch and Initials, *Yannis Haralambous*, TUGboat 12#1 (1991), pages 129–138.

[2] The `yfonts` package for use with LaTeX 2ε, *Walter Schmidt*, (2019).

[3] The Unifraktur Maguntia TrueType fonts (2017).

[5] Contributions welcome!