

# OpenType version of yfonts for Old German

Daniel FLIPO

19th July 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 `oldgerm-otf.sty` is included to load these fonts easily; it is meant as a replacement for LuaLaTeX and XeLaTeX of `oldgerm.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 (0.30).

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

## 1 Usage

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

All three families are loaded with all ligatures activated, an `s` prints a long *ſ* (initial and middle form), the final round *ſ* can be coded `s`<sup>1</sup>. Coding `"a`, `"e`, `"o`, `"u`, `"s`, produces *ǎ*, *ǣ*, *ǫ*, *ū*, *ſ*, as expected. Nowadays, most keyboards give access to the Unicode characters *ä*, *ë*, *ö*, *ü* and *ß*, so typing them directly is another possibility.

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

The OpenType feature `Alternate=0` turns *ǎ*, *ǣ*, *ǫ*, *ū* into *ǎ̇*, *ǣ̇*, *ǫ̇*, *ū̇*. It works for the *yfrak* and *yswab* families but not for *ygoth* (variant not available in the original version).

Those variants *ǎ̇*, *ǣ̇*, *ǫ̇*, *ū̇*, can also be accessed by coding `*a`, `*e`, `*o`, `*u` (as in `oldgerm`) without activating the `Alternate=0` feature.

Used with the *ygoth* family, the `Alternate=1` feature provides variants for the long *s* and the derived ligatures: *ſ*, *ſſ*, *ſſi*, *ſſt* are turned into *ẛ*, *ſẛ*, *ſſi̇*, *ſſṫ*.

---

<sup>1</sup>See section 2 for details.

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

## 2 Coding the long/round s

As stated in [1], the initial and middle *s* should be long (ſ) while the final form should be round (ſ̈). The OpenType `calt` feature can deal with this kind of requirement.

`oldgerm-otf.sty` enables the `calt` feature by default, so that an *s* followed by a punctuation mark or a space *character* is automatically printed as a round one (ſ̈ instead of ſ). The problem is that TeX turns spaces into *glues* which are ignored by the `calt` feature.

Conclusion: the `calt` feature works fine for the final *s* *only if* it is followed by a punctuation mark (point, comma, semicolon, etc.). This is true for LuaTeX and XeTeX.

For LuaTeX a workaround has been added in `oldgerm-otf.sty`. A null space character (U+200B) is added automatically after every *s* followed by a *glue* or a *penalty* (unbreakable spaces) or a *discretionary* (compound words), thus enabling the `calt` feature to deal also with TeX spaces and compound words. With LuaTeX, the *s* should always be printed out properly, the `s:` coding should no longer be required<sup>2</sup>.

Please note that the possibility of coding `s:` to get the round ſ̈ has been kept for compatibility, this implies that `aus:` has to be coded `aus::` with a double colon, the first one being swallowed by the ligature `s:` which prints ſ̈.

With XeTeX, the `s:` coding is still required inside compound words and before spaces f.i. `das: Aus:land.`

---

<sup>2</sup>I am not sure about compound words: is checking *discretionaries* correct?

### 3 List of optional ligatures

Some ligatures may be deactivated if necessary with the command `\addfontfeature{RawFeature=--ligname}`

	Name	Default (+)	Optional (–)
<code>\frakfamily:</code>	liga	ff	ff
	dlig	ch, cf	ch, cf
	hlig	ff, ff, ft	ff, ff, ft
	Name	Default (+)	Optional (–)
<code>\swabfamily:</code>	liga	ff	ff
	dlig	ch, cf	ch, cf
	hlig	ff, ff, ft	ff, ff, ft
	Name	Default (+)	Optional (–)
<code>\gothfamily:</code>	liga	ff, ffi, ffi, fi, fi, ij	ff, ffi, ffi, fi, fi, ij
	dlig	ch, ch, ct, qz, tz	ch, ch, ct, qz, tz
	hlig	ff, ff, ff, ffi	ff, ff, ff, ffi
	alig	ba, be, bo, da, de, do, ha, he, ho, ll, pa, pe, po, pp, qq, va, ve, vi	ba, be, bo, da, de, do, ha, he, ho, ll, pa, pe, po, pp, qq, va, ve, vi

### 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.

### References

- [1] Typesetting Old German: Fraktur, Schwabacher, Gotish and Initials, *Yannis Haralambous*, [TUGboat 12#1 \(1991\)](#), pages 129–138.
- [2] The `oldgerm` package for use with  $\text{\LaTeX}$  2 $\epsilon$ , *Frank Mittelbach*, (2020).

