Typographers' Inn

Peter Flynn

Font packages: Symbats, Swashes

Like many LaTeX users, I often check the LaTeX Font Catalogue [5] for a suitable typeface before deciding to buy or download one. Getting one of the listed fonts means there is often a package to support it, which makes life easier, especially when there are many variants.

Just as often, though, I need a font that isn't in the Catalogue, so I usually buy it — I'm happy to download fonts that the designer has explicitly made available free of charge and free of restrictions, but I'm not a great fan of downloading from sites that steal the work of designers and pass the fonts off as 'free'. Call me old-fashioned but font design is *hard*, and font designers need to pay the rent like everyone else.

Symbats. However, fonts made freely available by their designer are an act of generosity, and I came across one that was recently updated and released under the SIL Open Font License. I hadn't seen the font before, despite its having been available for 25 years! It's a font of religious and astrological symbols, not a text font, although it does also include Runic and Ogham alphabets (see Figure 1).

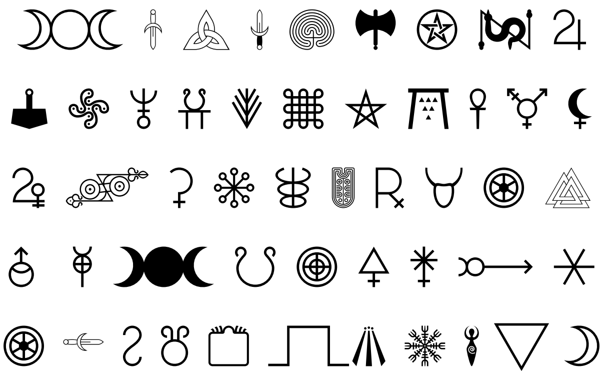


Figure 1: *Symbats 3.0* font (sample)

Those who remember the earlier days of fonts will recall that symbol font designers tended to put the symbols in the positions of the keyboard characters, so that A–Z, a–z, 0–9, and punctuation could easily be used to get the characters printed. But with so many symbols now having a standard Unicode name and codepoint, and with the big improvements in font file technology, this positioning is no longer needed in synchronous typographical editors. It's not needed for LaTeX editors either; for example ♒ (Aquarius) or ⚳ (Ceres) are both normal Unicode characters and can be typed as such if your editor has font support for them. However, some do not, so it seemed like a good step to write a package that would provide LaTeX commands for all the symbols in this font.

The first thing to do was to extract all the font character metadata into a spreadsheet using a combination of the *otfinfo* utility[1] and the *FontForge* scripting language. This provided one line per glyph, giving the actual character, the hexadecimal Unicode codepoint, and the Unicode or other name from the font file. I added a fourth column to provide a LaTeX command name. From this, it was possible to output a set of declarations like the one in Figure 2 which equated a command internal to the package (beginning with `SYMBATS@`[2]) with the `TU` font encoding and the codepoint of the symbol. As this was an OTF font, I was using XeLaTeX and the fontspec package, for which `TU` is the appropriate encoding (using XeLaTeX or LuaLaTeX is therefore a requirement for using the symbats3).

```
\DeclareTextSymbol{\SYMBATS@ceres}{TU}{"26B3}
```

Figure 2: Example of implementing an OTF glyph as a LaTeX internal command

I could have just used the LaTeX commands I had created in the spreadsheet, but some experiments had already shown that the positioning of the glyph with respect to the baseline of surrounding text did not take account of the depth (descenders) of the surrounding font. To allow for this, an additional definition would be needed for each character to expose the user command with code for the appropriate repositioning. I wrote a utility command `\SYMBATS@getdesc` to get the depth of the descenders of the current font and lower the baseline of the character by that amount (see Figure 3).

```
\newcommand{\ceres}[1][\relax]{%
  \SYMBATS@getdesc{#1}%
  \raisebox{\SYMBATS@baselineadjust}
          {\symbats\SYMBATS@ceres}%
}
```

Figure 3: Example of implementing an internal command as a LaTeX user command with an option to adjust the baseline (code in package)

As not everyone may want this, it is implemented by a package option `[descenders]`. Without it, a user can still add an optional argument to any of the character commands to raise or lower the symbol by an arbitrary amount.

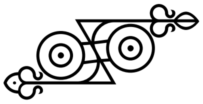
[1] Available from CTAN in the lcdf-typetools package.

[2] Thanks to Karl Berry for the change from `SY@`.

Peter Flynn
doi.org/10.47397/tb/43-3/tb135inn

A final step was to put the `\newfontface` font load command into `\AtBeginDocument` and add the font load option `[Scale=MatchUppercase]` so that symbols would be approximately the same size as the uppercase letters of the main font.

There are 325 symbols in the symbats3 package, which is available from a CTAN server near you (`https://ctan.org/pkg/symbats3`); it's also included in TEX Live and MiKTEX.



Swashes. Encouraged by this, I decided to repeat the process for another font that had been recommended to me. This is Super Tramp,[3] from designer and photographer Vivian Dehning.[4] It's a quite slender display font with a large collection of swash and variant characters, including a lot of extra ligatures, some on capitals, and a number of different upper- and lower-case asterisks.

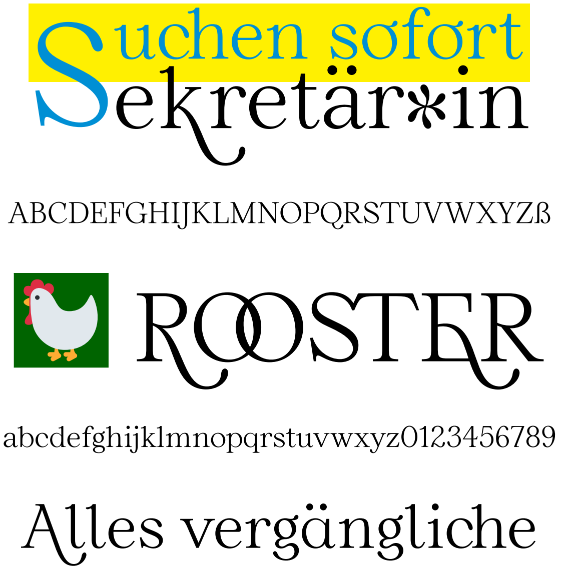

Suchen sofort
Sekretär*in
ABCDEFGHIJKLMNOPQRSTUVWXYZß
ROOSTER
abcdefghijklmnopqrstuvwxyz0123456789
Alles vergängliche

Figure 4: Vivian Dehning's *Super Tramp* font (sample, with chicken from the twemojis package).

Again, I thought it best to defer loading of the fonts until the last thing, so it could be adjusted to the main font size, although this time it uses `[Scale=MatchLowercase]`. Many of the additional characters use a StylisticSet option (up to eleven of them) in the OTF font, and there are ligatures and case-specific variants as well, so the font is currently loaded 13 times, which is probably wrong, but it works.

```
\newcommand{\Rswash}{{\ST@i R}}
\newcommand{\OOlig}{{\ST@xii OO}}
\newcommand{\Eswash}{{\ST@i E}}
```

Figure 5: Generated definitions for the *Super Tramp* font.

LATEX command names were incorporated into a spreadsheet as for the Symbats3 font, attempting to strike a balance between following the designer's choice of codepoint names and the conventions to which LATEX users have become accustomed. This resulted in some 200 commands, but this time they all seemed to sit on the baseline quite happily, so there was no need for adjustment, only a direct equivalence between LATEX command and the character from the correct StylisticSet or ligatured font load.

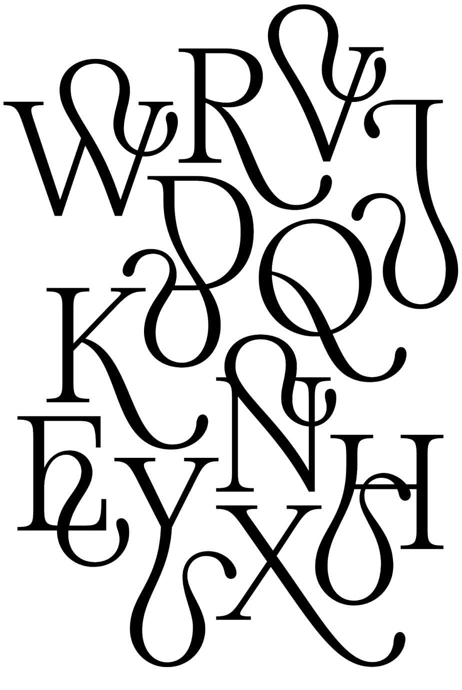


Figure 6: More sample swash characters from Vivian Dehning's *Super Tramp* font (designer's image, used with permission).

The naming of multiple swashes for the same base character led to the adoption of the same mechanism used for step-size fonts (`\large`, `\Large`, and `\LARGE`) so we have `\Jswash` (J), `\JSwash` (J), and `\JSWASH` (J), etc, in increasing order of complexity.

[3] Nothing to do with the rock band, movie, or trampoline manufacturer.

[4] This is a commercial font which can be bought from the designer via `https://viviandehning.com/super-tramp-typeface/`.

Characters that can be generated from the keyboard, especially the common diacritics, have *not* been given command names. For example, to get Ř you just type `Ř` using whatever mechanism your operating system and keyboard provides: there is no `\Rcaron` command; similarly for the Ẅ, but the crossed-W variant with diaeresis Ẅ is provided for with `\Wuml`.

There is still work to do in tidying this up before I can package it up for CTAN but it should be ready by the autumn.

To publish or not to publish

There have been some discussions recently among TeX consultants about providing a better service for self-publishers. These are authors — of all kinds of material, not just math and science — who for one reason or another are not using traditional publishers, so they have to edit, typeset, proof, bind or package, and publish their PDF or EPUB themselves.

There are of course lots of useful web sites, services, and other resources (including books!) to help people do this, many of them providing very good quality — at a cost — and very sound advice, often starting with 'Don't . . . '

The conventional path — a wordprocessor, often Microsoft Word — is perfectly capable, in expert hands, and with a lot of effort and reconfiguration, of producing passable typesetting. LaTeX can do the job just as easily if suitable classes and packages are used, and can produce better quality output.

I have always considered that LaTeX's key feature was automation: define a pattern once, and it can be used consistently throughout the whole document, and even across documents. Anyone who has ever taught LaTeX knows that avoiding repetitive manual formatting provides an instant productivity boost to the bottom line in terms of time saved. Newcomers have to rely on experienced users to create these patterns (the definitions of commands and environments), so for this project to succeed it is essential that we can open and maintain a dialogue between the potential users (authors with a book to typeset) and the potential contributors (experienced users).

The discussions include a TeX Live book publishing resource, and one of the tasks will be to promote it among self-publishers who care about publishing trade-quality books. One of the participants, the author Lloyd R Prentice [6], has said 'the big challenge is how to simply and clearly explain why LaTeX is a better book styling and typesetting option relative to word processing'. I think that's something everyone can contribute to.

Afterthought: Translations

I am always happy to be able to announce when an article from the Inn is translated into another language so that it reaches a wider audience. To be able to announce two translations and a reprint of the same article at the same time is even nicer.

My article 'To print or not to print' from this column [2] is now available in French as 'Imprimer ou ne pas imprimer' in the French typographical journal *Graphê* 85 [1], and was reprinted in *Cahier* 58 of GUTenberg (the journal of the French-speaking TeX users' group), translated by Patrick Bideault [4]; and in collaboration with Bernd Raichle in German as 'Druck oder Nichtdruck' in *Die TeXnische Komödie* (the journal of the German-speaking TeX users' group) [3].

I was remiss, however, in not RTFM and referring them to the copyright and translation notice inside the front cover of issues of *TUGboat* to ensure that *TUGboat* itself was notified of the publication. Thank you, Barbara Beeton, for reminding me.

References

[1] P. Flynn. Imprimer ou ne pas imprimer. *Graphê* 85:13, Dec 2020. Tr. Patrick Bideault. `https://typo-graphe.com/produit/numero-85/`

[2] P. Flynn. Typographers' Inn: To print or not to print. *TUGboat* 41(3), Dec 2020. `https://tug.org/TUGboat/tb41-3/tb129inn.pdf`

[3] P. Flynn. Druck oder Nichtdruck. *Die TeXnische Komödie* 2022(2), May 2022. Tr. Patrick Bideault and Bernd Raichle.

[4] P. Flynn. Imprimer ou ne pas imprimer. *Cahier* 58, Apr 2022. Tr. Patrick Bideault. `https://www.gutenberg-asso.fr/Cahier-numero-58-Septembre-2021`

[5] P. Jørgensen. The LaTeX Font Catalogue, May 2021. `https://tug.org/FontCatalogue/`

[6] L. Prentice, V. Novotný. *Publish Beautiful Books with Markdown.* Writersglen Publications, Marshfield, MA, 2021.

◇ Peter Flynn
Textual Therapy Division,
Silmaril Consultants
Cork, Ireland
Phone: +353 86 824 5333
`peter (at) silmaril dot ie`
`blogs.silmaril.ie/peter`