

# Automatic suppression of unwanted ligatures

## The ConT<sub>E</sub>Xt LMTX approach

*Denis Maier*

### 1. Introduction

The usage of typographic ligatures is often considered to be a hallmark of high quality typesetting.<sup>1</sup> Not surprisingly then, some advocates of T<sub>E</sub>X and its derivatives emphasize T<sub>E</sub>X's ability to use ligatures automatically, and consider this, among other features, to be responsible for its superior typographic output [e.g. 15]. However, it's also well known that ligatures should be avoided in certain cases. Therefore, the T<sub>E</sub>Xbook mentions that the word "shelfful" should indeed be rendered without the ff-ligature [9, p. 19]. But especially when typesetting German, ligatures have proven to be tricky beasts, and it's hard to get them right automatically. The *Duden*, the main authority on German language usage, declares that ligatures should only be used if both letters belong to the same stem. Instead they should be avoided between the stem and certain endings, and if the letters belong to different parts of a compound word [4, p. 97] – e.g. Auflage, Schilfinsel, Kaufleute. This seems to affect indeed so many words that an influential German handbook on typography recommends disabling automatic ligatures globally, and to re-insert them manually as required [5, p. 194].<sup>2</sup>

To be sure, one could be tempted to get rid of this problem by just not using them at all, but ligatures exist to solve a real issue. Basically, according to Forssmann and de Jong, they serve three purposes [5, p. 194]:

- to avoid collisions between adjacent letters e.g., the lowercase letter f tends to clash with the letters i and l due to its overhanging shape.
- to combine certain letter pairs that are usually articulated as a single sound: ch, ch; ck, ck; Qu, Qu; Th, Th.
- as a means of decoration: st, st; ct, ct.

Those ligatures that fall into the second and third group are indeed optional, a personal stylistic preference, and their usage may be admired by some [15], and frowned upon by others [e.g. 3]. But those ligatures falling into the first group are different as they solve a real problem. So, if one is serious about avoiding ligatures

<sup>1</sup> Ligatures are combinations of letters rendered as a single glyph, such as fi rendered as fi, fl rendered as ff, or ff rendered as ff, but there exist many others.

<sup>2</sup> To quote from Forssmann and de Jong: "Im Deutschen muß man in denjenigen Programmen, die automatischen Ligaturen-Satz anbieten, diesen deaktivieren und jeden Text mit ‚Suchen und Ersetzen‘ durcharbeiten – wir warten noch auf automatischen Ligaturen-Satz mit Worterkennung via ‚OpenType‘."

at all cost, one has to use fonts where this problem doesn't occur, such as those recommended by Robert Bringhurst: Aldus, Melior, Mendoza, Palatino, Sabon, Trajanus and Trump Mediäval [2, p. 52].<sup>3</sup> This wouldn't entail bad typography, but switching fonts might not be a problem in every project.

## 2. Existing methods in T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt, and LuaL<sup>A</sup>T<sub>E</sub>X

Traditionally, users of T<sub>E</sub>X and its derivatives had a couple of methods at their disposal to deal with unwanted ligatures. First manually: in the T<sub>E</sub>Xbook [9, p. 306], Knuth recommends to use groups (`{shelfful}` or `shelf{}ful`),<sup>4</sup> an italic correction (`shelf\ful`), or explicit kerns (`shelf{\kern0pt}ful`) to suppress unwanted ligatures. Additionally, L<sup>A</sup>T<sub>E</sub>X users can use the shortcut `|` provided by the `babel` package [1, p. 2], while in ConT<sub>E</sub>Xt one can use `\noligature` (`shel\noligature{ff}ul`).

Such manual methods are obviously inconvenient. They are tedious and error-prone, and adding commands to each case where ligatures have to be suppressed doesn't really increase readability of the source code either. So, people often used preprocessing scripts to add these commands [8]. However, preprocessors have their own drawbacks as they add some complexity to the document processing workflow by adding additional steps and external dependencies.<sup>5</sup> For users of LuaL<sup>A</sup>T<sub>E</sub>X the easiest solution today is to load the `selnolig` package [11], which suppresses unwanted ligatures using a pattern-based approach.<sup>6</sup>

## 3. Language options in ConT<sub>E</sub>Xt LMTX

Since March 31, 2021 ConT<sub>E</sub>Xt LMTX supports a new mechanism to suppress unwanted ligatures, called *language options* [7]. I'm not the right person to write about technical details and the inner workings of that mechanism, so suffice it to say that the new mechanism plugs into the hyphenation mechanism, which minimizes the performance hit.

As the name suggests, these language options are language-specific and can be defined like this (to suppress the ff-ligature in "Auffassung"):

```
\startlanguageoptions[de]
  Auf|fassung
\stoplanguageoptions
```

As one would probably not want to add a huge number of words here one could also load a goodies file:

<sup>3</sup> Interestingly, Ari Rafaeli has described Sabon as a type, "which conforms to German standards" without "concessions to English tastes". And: "The Germanism of Sabon is particularly distinguishable by its lowercase f" [14, pp. 128–29].

<sup>4</sup> But these don't work with LuaT<sub>E</sub>X.

<sup>5</sup> See as well [12, p. 2].

<sup>6</sup> For more details see [12, pp. 6–11].

```
\setuplanguage[de][goodies={lang-de.llg}]
```

For convenience, ConTeXt currently includes two prebuilt goodies files that cover common use cases, `lang-en.llg` for English and `lang-de.llg` for German.<sup>7</sup>

#### 4. Structure of a language goodies file

The basic structure of a goodies file is as follows:

```
return {
  name      = "whatever",
  options   = {
    {
      ...
    }
  }
}
```

Under options one will find replacement definitions. Let's look at a couple of examples. The simplest case is just a list of words equivalent to using `\startlanguageoptions` inside one's document:

```
{
  words = [[
    Auf|lauf|bremse
    Auf|lauf|bremsen
  ]],
}
```

However, there are also more complex variants, such as applying patterns to a group of words:

```
{
  patterns = {
    ff = "f|f",
    fl = "f|l",
    fi = "f|i",
    ffi = "f|fi",
    ffl = "f|fl",
  },
  words = [[
```

<sup>7</sup> `lang-en.llg` has been compiled by Dave Jarvis based on a word list collected by Mico Loretan in a stack-exchange post [10]. As for the German word list, as a basis we used the test word list from the *selnolig* package, which in turn appears to be a slightly extended version of the word lists used by *rmligs*. In a second step, we extended the word list by applying the patterns used by the *selnolig* package [12, pp. 41–63] to three corpora provided by the “Projekt Deutscher Wortschatz / Leipzig Corpora Collection” available at [wortschatz.uni-leipzig.de/de](http://wortschatz.uni-leipzig.de/de)[6]: Mixed-typical, 2011, 1M words; News, 2020, 1M words; Wikipedia, 2016, 1M words. This resulted in the current word list of about 3900 words.

## contextgroup > context meeting 2021

```
Auffahren
Auffahrtgefahr
Auflage
Chefideologe
]],
}
```

If we want to block certain ligatures globally, we can use substitutions. For example, if one uses Minion, but doesn't want the *Th*-ligature:

```
{
  substitutions = {
    th = "t|h",
    Th = "T|h",
  },
}
```

As one would probably not want to add such a definition to the goodies files provided by ConT<sub>E</sub>Xt, one can also save them in a dedicated goodies file:

```
return {
  name = "blockfancylics",
  options = {
    {
      substitutions = {
        th = "t|h",
        Th = "T|h",
      },
    },
  },
}
```

This even works from within a ConT<sub>E</sub>Xt source file:

```
\startluacode
local blockfancylics = {
  name = "blockfancylics",
  options = {
    {
      substitutions = {
        th = "t|h",
        Th = "T|h",
      },
    },
  },
}
table.save("blockfancylics.llg",blockfancylics)
\stopluacode
```

In both cases, one can then load the additional goodies files:

```
\setuplanguage[en][goodies={lang-en.llg, blockfancyllg.llg }]
```

As English makes ample use of prefixes and suffixes to form new words (think of: unselfishness), it's also possible to use the keys `prefixes` and `suffixes`:

```
{
  patterns = {f|i},
  words = [[
    selfish
  ]],
  suffixes = [[
    ness
    ly
  ]]
},
{
  patterns = {f|i},
  words = [[
    proofing
  ]],
  prefixes = [[
    fire
    water
  ]]
}
```

## 5. Extending the word lists

Of course, the word lists provided by ConT<sub>E</sub>Xt are not complete, and probably never will be. Having to check a complete file for incorrect ligatures would put us into more or less the same situation as before, but luckily there is a convenient way to check whether incorrect ligatures have been touched, and which words went through unmodified.

First, we enable the relevant tracker:

```
\enabletrackers[languages.goodies]
```

This will add lines such as these to the log file:

```
languages      > goodies > properties: Auflage auflage
languages      > goodies > ignored: AuflageX => auflageX
```

The first line shows a word where a ligature has been suppressed. The word in the second line has been ignored by the mechanism. We can now examine each word and add it to the goodies file or to `\startlanguageoptions` if appropriate. Obviously, checking the log file is tedious as it contains a wealth of information, but a

## contextgroup > context meeting 2021

simple script, such as `check-ligatures.lua` [13], that extracts the relevant words can simplify this process. Running:

```
mtxrun --script check-ligatures.lua input whitelist
```

will produce a new file `input-ligatures-new.log` with words to check. We can then add them to the goodies file or to the language options (if they indeed contain problematic ligatures), or to a whitelist so we don't have to check same words over and over again.

## 6. References

- Braams, J., Raichle, B., Schmidt, W., & Spitzmüller, J. (2021). *Babel support for the German language (post-1996 orthography)*. Retrieved from <http://mirrors.ctan.org/macros/latex/contrib/babel-contrib/german/ngermanb.pdf>
- Bringhurst, R. (2019). *The elements of typographic style*. (Fourth edition (version 4.3) ed.). Seattle: Hartley & Marks, Publishers.
- Butterick, M. (n.d.). *Practical typography. Chapter: Ligatures*. Retrieved from <https://practicaltypography.com/ligatures.html>
- Dudenredaktion (Ed.) (2004). *Duden – Die deutsche Rechtschreibung*. (Vol. 1, 23 ed.). Mannheim: Dudenverlag.
- Forssman, F. & de Jong, R. (2008). *Detailtypografie: Nachschlagewerk zu allen Fragen zu Schrift und Satz*. (4 ed.). Mainz: Verlag Hermann Schmidt Mainz.
- Goldhahn, D., Eckart, T., & Quasthoff, U. (2012, 5). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA). Retrieved from [http://www.lrec-conf.org/proceedings/lrec2012/pdf/327\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2012/pdf/327_Paper.pdf)
- Hagen, H. (2021). *New upload*. Message on the ConT<sub>E</sub>Xt mailing list, 2021/03/21. Retrieved from <https://www.mail-archive.com/ntg-context@ntg.nl/msg98035.html>
- Jacke, B. (2012). *Rmligs*. Version 0.87. Retrieved from <https://www.ctan.org/pkg/rmligs>
- Knuth, D. E. (1996). *The TeXbook*. Reading, Massachusetts: Addison-Wesley.
- Loretan, M. (2012). *Answer to: When should I not use a ligature in English typesetting?*. Forum post, last edited 2012/04/24. Retrieved from <https://english.stackexchange.com/a/50957/22099>

\_\_\_\_\_ (2015a). *Selnolig*. Lua<sup>A</sup>T<sub>E</sub>Xpackage, version 0.302, 2015/10/26. Retrieved from <https://www.ctan.org/pkg/selnolig>

\_\_\_\_\_ (2015b). *The selnolig package: Selective suppression of typographic ligatures*. Manual of the selnolig package for Lua<sup>A</sup>T<sub>E</sub>X. Version from 2015/10/26. Retrieved from <http://mirrors.ctan.org/macros/luatex/latex/selnolig/selnolig.pdf>

Maier, D. (n.d.). *Check ligatures*. Retrieved from <https://gist.github.com/denismaier/eb2d6301ef273c80c98c228e1d64323c>

Rafaelli, A. (2005). *Book Typography*. New Castle, Delaware: Oak Knoll Press.

Taraborelli, D. (2015). *The Beauty of LaTeX*. Retrieved from <https://nitens.org/w/latex/>