The zcm example font family

Lars Hellström*

October 2, 2010

1 The zcm font family and reldemo.tex

The zcm family of fonts consists of only two fonts and it is meant to accompany the relenc package documentation. These fonts are really just combinations of glyphs taken from other fonts, primarily the Computer Modern fonts (hence the cm), but there are also some glyphs which are taken from some LaTeX symbol fonts (lasy10 and lcircle10).

The primary reason these fonts exist is that I felt I needed an example of what can be done with the relenc package. As one really cannot expect that people will have any font that is not either a Computer Modern or a LaTeX font, such an example has to be made using these fonts. I am well aware that some of the glyphs look terrible, but the zcm family is not intended to be used for any serious typesetting.

The font family consists of two fonts: zcmr8d and zcmra. The first of these is declared as T1R/zcm/m/n and its coding scheme is identical to that of the T1 encoding. The second is declared as T1R/zcm/m/a and its coding scheme deviates from that of the T1 encoding in four slots, which have been reassigned to contain ligatures instead of accented letters.

A demonstration of these fonts is generated by typesetting reldemo.tex; this document contains a font table listing the contents of each slot, a demonstration of all the symbol commands, a demonstration of the effects of applying the accenting commands to the non-accented letters, and a comprehensive list of character pairs, for both fonts. The reason I have put it in a separate file is (i) that the relenc package will have to be installed before these fonts can be viewed (the files needed are relenc.sty, t1renc.def, and t1rzcm.fd) and (ii) that there might be some problems connected to viewing these fonts.

To begin with, both fonts are virtual fonts and some DVI drivers cannot handle these. If you only have such drivers, you should get one that can (not just for the sake of the relenc package—virtual fonts are in general neat to be able to use).

One of the fonts used to make these virtaul fonts—the lcircle10 font—occurs under two other names as well: circle10 and lcirc10. Unfortunately, one cannot expect that there is more than one of these in any given TeX installation. Thus you

*E-mail: Lars.Hellstrom@math.umu.se

will get some problems if your installation uses another name for this font. Some DVI drivers will translate the name used in the virtual font to that of the font names which actually occurs in the system, and in this case there is no problem, but if not then you will have to intervene yourself.

The easiest way to do this is simply to rename a few virtual font files, since there are such files accompanying the relenc package for using the `lcircle10` font under any one of these names. What you have to see too is that the ones which use the font under the name on your system are the ones which are named `zcmr8d.vf` and `zcmra.vf` respectively. The following table shows what the file names are and which font it uses

Virtual font file	`lcircle10` name used
`zcmr8d.vf`	`lcricle10`
`zcmra.vf`	`lcricle10`
`zcmr8d.vf2`	`cricle10`
`zcmra.vf2`	`cricle10`
`zcmr8d.vf3`	`lcric10`
`zcmra.vf3`	`lcric10`

Note that all the above files are virtual font files. The reason only two have been given the correct suffix `.vf` is that there really are no more than two fonts, so they should not take up any more font name space than that either.

Finally, there is one other use for the `T1R/zcm/m/n` font—it is the final substitution font in the `T1R` encoding. LaTeX requires that there is such a font, so it might as well be this one. This means that you should see to that the files which are this font (`zcmr8d.tfm` and `zcmr8d.vf`) are put in such locations that TeX and its helper programs (DVI drivers etc.) will find them if they are needed. You should not need to move them before you typeset `reldemo.tex` for the first time, however, as these fonts are initially in the same directory as this and TeX looks for files there first.

If you do have the `ec` family of fonts however, you may (quite understandably) want to use one of them as final substitution font for the `T1R` encoding instead. There is a package called ecsubzcm that is installed with relenc. This package declares the `zcm` family in encoding `T1R` and `ecrm1000` under the encoding/family/series/shape combination `T1R/zcm/m/n`. After loading this package, LaTeX will not bother to input the font definition file `t1rzcm.fd`, so the "real" `zcm` fonts will not get involved.

2 Implementation

2.1 Font definition file

As is the custom, the file starts by announcing itself.

```
1 ⟨*fd⟩
2 \ProvidesFile{t1rzcm.fd}[1999/01/19 Font definitions for T1R/zcm.]
```

Then the family is declared. So far, there is nothing out of the ordinary.

```
\DeclareFontFamily{T1R}{zcm}{}
```

Next the shapes are declared. There is nothing strange about this either.

```
\DeclareFontShape{T1R}{zcm}{m}{n}{
   <-> zcmr8d
}{}
\DeclareFontShape{T1R}{zcm}{m}{a}{
   <-> zcmra
}{}
```

That would be all in a normal font definition file. This file does however go on with defining some variants of commands and compositions, to compensate for the differences between the coding scheme of the `zcmra` font and the default coding scheme.

Slot number 128 is used for a ligature instead of the default 'Ă', so this has to be compensated for. In this case it is done by making another default definition of the `\u` command and defining all the other compositions of that command again.

```
\DefineTextAccentVariant{\u}{T1R}{zcm}{}{a}{8}
\DefineTextVariantComposition{\u}{T1R}{zcm}{}{a}{G}{135}
\DefineTextVariantComposition{\u}{T1R}{zcm}{}{a}{a}{160}
\DefineTextVariantComposition{\u}{T1R}{zcm}{}{a}{g}{167}
```

It should be noted at this point that this font definition file is not set up in the most efficient way. Normally, one would want to minimize the number of variants that needs to be defined—something which the above most certainly does not do—and one often achieves this by trying to affect as few accenting commands as possible. The four extra ligatures of the `zcmra` font could have been put in slots 128, 135, 160, and 167—that would have made it possible to get by with the single variant definition

```
\DefineTextAccentVariant{\u}{T1R}{zcm}{}{a}{8}
```

although a font designer who frequently use the `\u` command will probably prefer to use slots for compositions of some other command—but `zcmra` has been set up more to show what is possible than what is optimal.

The same problem occurs with slot 131, which does not contain the 'Č' glyph that the encoding-level composition of `\v` with `C` assumes it does. Another solution to the problem is to give an explicit definition of the composition at a lower level, as in

```
\DefineTextVariantCompositionCommand{\v}{T1R}{zcm}{}{a}{C}{%
   \add@accent{7}{C}%
}
```

`\add@accent` is standard LaTeX and its definition can be found in [1]. This solution gets by using only two variants (the `T1R`/`zcm`/`?`/`a` variant of `\v`, and the compositon of that variant with `C`). Had instead the first solution been used here, it would have required the code

```
\DefineTextAccentVariant{\v}{T1R}{zcm}{}{a}{7}
```

```
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{D}{132}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{E}{133}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{L}{137}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{N}{140}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{R}{144}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{S}{146}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{T}{148}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{Z}{154}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{c}{163}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{d}{164}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{e}{165}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{l}{169}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{n}{172}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{r}{176}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{s}{178}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{t}{180}
\DefineTextVariantComposition{\v}{T1R}{zcm}{}{a}{z}{186}
```

which defines another 18 control sequences. The `\v` command sure has many compositions, hasn't it?

Things are a bit different when the slot is used for a variant of a composition, instead of a composition of a variant. In the former case, there is a special command that can be used to extract the default definition of the accenting command.

```
\DefineTextUncomposedVariant{\'}{T1R}{zcm}{}{a}{C}
```

Finally, another example like `\v`.

```
\DefineTextVariantCompositionCommand{\k}{T1R}{zcm}{}{a}{a}%
   {a\llap{\char12\kern-0.07em}}
⟨/fd⟩
```

This definition demonstrates that there is absolutely no need to base the definition of a composition on the default definition; using that in this case would (i) be much longer, (ii) be unstable, as the default definition of `\k` positions the accent incorrectly if not both the height of the accent and the depth of the letter is zero or less, and (iii) probably not kern particularly good (at least this definition kerns like 'a' to the left, the default does not kern at all).

2.2 The **ecsubzcm** package

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{ecsubzcm}[1999/01/19]

\DeclareFontFamily{T1R}{zcm}{}

\DeclareFontShape{T1R}{zcm}{m}{n}{
   <-> ecrm1000
}{}
⟨/package⟩
```

References

[1] Johannes Braams, David Carlisle, Alan Jeffrey, Frank Mittelbach, Chris Rowley, Rainer Schöpf: `ltoutenc.dtx` (part of the LaTeX 2_ε base distribution).