The **type1cm** package for LaTeX*

David Carlisle†

2002/09/05

LaTeX separates its internal notion of font specification from the external fonts available on the system by means of 'Font Shape Specifications', that are normally held in 'Font Descriptor' (FD) files.

The FD files that come with LaTeX that refer to the standard Computer Modern set (and the related $\mathcal{AMS}$ set) are based on the classical bitmap fonts which are available in 'magstep' sizes only. For instance The specification of the main roman font is

```
1 \DeclareFontShape{OT1}{cmr}{m}{n}
2  {  <5> <6> <7> <8> <9> <10> <12> gen * cmr
3     <10.95> cmr10
4     <14.4> cmr12
5     <17.28><20.74><24.88>cmr17}{}
```

which says for instance that no font is available at 10.5 pt (LaTeX will substitute the nearest available size if you ask for this) and similarly the font is not available at all above 25 pt.

Such restrictions are essential with bitmap fonts to save generating huge numbers of bitmaps for any size that might be requested, however with scalable versions of the fonts these restrictions are not really needed. For instance the equivalent definition defined here is:

```
1 \DeclareFontShape{OT1}{cmr}{m}{n}{
2         <-6>    cmr5
3         <6-7>   cmr6
4         <7-8>   cmr7
5         <8-9>   cmr8
6         <9-10>  cmr9
7         <10-12> cmr10
8         <12-17> cmr12
9         <17->   cmr17
10       }{}
```

---

which means that you can (Although some may consider it bad style) go

```
\fontsize{10.7pt}{12pt}\selectfont
```

and be given a suitably scaled version of `cmr10` and a baseline setting of 12 pt. Similarly if you really want, you can go

```
\fontsize{2cm}{2.5cm}\selectfont
```

and use `cmr17` scaled to 2 cm height for a special display context.

Warning

If you use this setup and produce DVI files using these sizes, be very careful if you give the DVI files so created to someone using classical bitmap versions of the cm fonts. Depending on how their system is set up they will either not be able to print the file at all, as they will not have the fonts at the corrrect sizes, or the system will make all these strange sizes 'on demand' and fill the disk with bitmaps.

So the default way of using this package is as a LATEX package file. Install `type1cm.sty` as described below and then just add

```
\usepackage{type1cm}
```

to your document. This will override the definitions for all the fonts and so the standard FD files are not used. The disadvantage of this method is all the definitions are then loaded into memory, even for fonts you don't use in the document. (However only the size specifications are so loaded, not the fonts themselves). However the advantage is that you can always choose not to specify `\usepackage{type1cm}` and so use the same font sizes (and get the same line and page breaks) as everyone else.

Alternatively you may use this package to produce a set of fd files that *replace* the standard files (such as `ot1cmr.fd`) You will then need to remake the LATEX format by running `initex` on `latex.ltx` (as some FD files are loaded into the format). Then you will not need to use `\usepackage`. All your documents will have the freedom of scalable font choices.

1 Installation instructions

The installation is controlled by the file `type1cm.ins` which you may edit in two places.

1) If you have not got the $\mathcal{AMS}$ Font set available in scalable form remove the '`%`' from the line `%\def\ams{}`
   This will cause a more restricted font specification to be used.

   If you have an extended $\mathcal{AMS}$ font set available in scalable form (`msam6`, `msam8` and `msam9` in addition to `msam5`, `msam7` and `msam10`) then remove the '`%`' from the line `%\def\ams{ams,extra}`

2) If after reading the warning above you decide to make a set of fd files, remove the '%' from the line %\makefdtrue

Then run `tex` (or `latex`) on the file `type1cm.ins` and a package file `type1cm.sty` (and perhaps a set of FD files) will be produced which you should place in a directory where TEX looks for input files.

2 Small technical note about the Font Shape specifications used.

I consistantly specified that given a requested size, the largest available font size smaller than the requested size should be used and then enlarged to the requested size. The exception being sizes smaller than the smallest available font, which use a reduced version of that font. The rationale for this is that enlarging a small font typically produces a rather 'fat' font, but something readable, whereas shrinking a font may produce something unreadable quite quickly

So for msam (if 'extra' is not specified) I used:

```
1  \DeclareFontShape{U}{msa}{m}{n}{
2        <-7>   msam5
3        <7-10> msam7
4        <10->  msam10
5        }{}
```

That is sizes strictly below 7 pt use msam5, 7 pt to (less than) 10 pt use msam7 and sizes 10 pt and above use msam10.

This differs from the Specification that the $\mathcal{AMS}$ use for the scalable $\mathcal{AMS}$ fonts (used by the `psfonts` option to the amsfonts package) They use

```
1  \DeclareFontShape{U}{msa}{m}{n}{
2        <-6>      msam5
3        <6-8>     msam7
4        <8->      msam10
5    }{}
```

This has the advantage of minimising the scaling used, so for instance a 9 pt request is satisfied by msam7 scaled up to 9 by this package, but by msam10 scaled down to 9 by the $\mathcal{AMS}$ package.

In practice this is not likely to make a difference that anyone notices, but it could in principle affect line breaks etc, so I thought that I would mention it here.

3 History

1997/01/27 v0.02 BlueSky/Y&Y Type1 CM font definitions (DPC)

    Possibly first public release? Maybe there was a 0.01..

2002/04/01 v0.03 BlueSky/Y&Y Type1 CM font definitions (DPC)

    cmdunhill hasn't worked since the beginning: `\hypenchar`
                 ↑

    Reported by Brian Mays via the Debian bug tracking system. Also, made distribution conditions LPPL (previously unspecified)