OFS – Macro Package To Manage Your Fonts

Petr Olšák
Faculty of Electrical Engineering, CTU, Prague
petr@olsak.net

Abstract

OFS (Olsak’s Font System) gives you a possibility to keep track of your fonts; especially if you have many fonts. It provides tools for making font catalogues, a comfortable user environment for font selection etc. The OFS was presented in EuroTeX 2003 (Brest, France) [5] but many new features were implemented in 2004. This article presents the latest version of this package. OFS is freely available on [1].

1 What fonts are installed?

Interactive typesetting systems have one advantage: there is a simple answer to the question cited in title of this section. Everybody can call up a menu of such an interactive system and pick the item “fonts” (or something similar), click the mouse and he/she can see names of all available fonts. If the system is more powerful then samples of fonts are listed, too. The user can simply choose any font from this listing for his/her document: just click the mouse...

On the other hand, a TeX user has a more problematic situation. Imagine that you are sitting in front of a (somebody else’s) computer where TeX is installed and you have to make a simple poster with some attractive typeface. Your first question is: what fonts are available? You can do “kpsewhich cmr10.tfm” to find the font path, then you can “cd” to font path and “ls” font metric files. Definitely, this is not a good idea...

On the other hand, if the OFS is correctly installed on the TeX system, then you can type the following (the user input is marked by rectangles):

```
$ texinfo
This is TeX, Version 3.14159 (Web2C 7.3.7a) emTeX v. Jun. 2004, the reencoding enabled.
/usr/tex/texmf/tex/ofs/ofstest.tex (/usr/tex/texmf/tex/ofs/ofstest.tex)
OFS (Olsak’s Font System) based on plain initialized. <Jul 2004>
```

You can read more declaration files by \decl

```
*** Type family name without brackets (or ? or *)
\decl
```

```
*** What to do with family AntykwaTorunska ?
```

```
\decl
```

```
```

XIV Ogólnopolska Konferencja Polskiej Grupy Użytkowników Systemu TeX 19
All fonts installed on your TeX system are listed here. You can use the ofstest macro as an interactive tool to print more information about available fonts. The listing prints the long human readable font family names and the variant switches \rm, \bf, \it, \bi, etc. available for each font family. You can write one font family name on the prompt, the \list command then prints more information about it: long names for all variant switches, the metric names (these are usually cryptic names), encodings here. You can use the \lenames for such fonts: basic metric and extra metric file listed in our example. OFS handles these couples of metric files as a single one. We’ll return to this problem later in section 6.

Fonst have typically more than 256 glyphs. This is a reason why OFS records two metric filenames for such fonts: basic metric and extra metric (see \exp\-antr metric file listed in our example). OFS handles these couples of metric files as a single one. The \list command then prints more information about it: long names for all variant switches, the metric names (these are usually cryptic names), encodings available for this family etc.

Fonts are collected in packages of fonts. Each font family has one or more font variants. The allfonts.tex file has to be managed on TeX systems in order to keep track of all font packages available for this family. Fonts have typically more than 256 glyphs. This is a reason why OFS records two metric filenames for such fonts: basic metric and extra metric (see \exp\-antr metric file listed in our example). OFS handles these couples of metric files as a single one. We’ll return to this problem later in section 6.

2 Interactive macro ofstest.tex

I’ll present a supplementary macro ofstest.tex first, not the ofstex macro itself. I hope it would be no problem for reader.

All fonts are collected in packages of fonts. Each package can consist of one or more font families and each font family has one or more font variants. The allfonts.tex file has to be managed on TeX systems in order to keep track of all font packages actually installed. These font packages are saved in allfonts.tex in a simple way:

```
$ cat 'ksources allfonts.tex'
XXX All fonts installed on your TeX system

\input a35 \ X adobe 35 base font collection (PS level2)
\input a117 \ X adobe 117 base font collection (PS level3)
\input ffonts \ X free fonts (Charter etc.)
\input skatalog \ X fonts from BitStream
\input skatalog \ X fonts from Storm Type Foundry, typocatalog 3
\input pantyk \ X Polish Antykwa
```

Some packages can have sub-packages. For example the skatalog package includes alido, stitul, sjannon, styfa, sdynamo, etc.¹ All these packages or sub-packages are implemented by declaration files (a35.tex, ffonts.tex, slido.tex etc.). Long names of font families are declared here. A user can work only with one or more declaration files, not (directly) with all fonts. In the following example, we turn attention to ffonts.tex declaration file, namely the Charter font family.

```
# explain ofstest *ffonts*
This is TeX, Version 3.14159 (Web2C 7.3.7x)
(Part of) MikTeX v. Feb. 2003, the reencoding enabled.
The cs-fonts are preloaded and A4 size implicitly defined.
$KPS-E/F/err/text/ofstest.tex
GFS (Glask’s Font System) based on plain initialized. <May 2004>
$KPS-E/F/err/text/ofstest.tex
Czech hyphenation used (language=cs), reencoding is set on.
$KPS-E/F/err/text/ofstest.tex This is ofstest macro, version

*** What to do with family Charter ?
(type command or \help)  help
commands:
\list ..... List all variants of the family Charter
\abet ..... One line alphabet/digits sample for each variant
\chars ..... Print list of available characters including TeX sequences
\text ..... One paragraph in all variants of the family Charter
\mixed ..... The same as \list \table \abet \chars \text \mixed \math \setsize ...
\remdecl .. Remove all families of given declaration file from familylist
\famlist ... The same as \remdecl
\from ..... Remove current family or specified family from familylist
\remdecl .. Remove all families of given declaration file from familylist
\famlist ... Show list of all declared families (as \showfonts)
\decl ..... Input next declaration file
\help ..... This text
\morehelp . Show more help information
\fontusage The help screen of the OFS
\end ..... End of this session

*** What to do with family Charter ?
(type command or \help)  help
\abet \chars \mixed \math \text

$cat 'ksources allfonts.tex'
XXX All fonts installed on your TeX system

The \claplain is a Czech alternative to plain TeX. This format is used in this example because I prefer to have \eighty (Czech encoding) as default and the samples will be printed in Czech language (on the other hand, if \tex command is used, the English samples are printed and \eighty encoding is used).

The \help command gives a self-explanatory result. In the example above, we printed short lines with font samples, a list of all TeX sequences to ac-

¹ The skatalog package includes all fonts from Typocatalog 3 by Storm Type Foundry, see [2, 3].
Příklad. Nyní zjistíme, zda je možné kombinovat základní různý (resp. polotucí) řez s krúzovou v variantu Charter. Pro vyznačování je vhodné použít kurzívou a už méně těchto variant je prakticky užitnou vyhovuje kurzívou, ale ne všechny rodinách písm a je zdálity k užitnou s pokročilým vyznačováním podtržením nebo prostrkáním. To lide s dobrým vychováním nedělají. Vyznačení má být takové, aby při čtení bylo ovládáno místě zvětšené odlišné, ale při pohledu z dálky zůstal text odstavec stejněm řezem.

Tomu nejlépe vyhovuje kurzívou, ale ne ve všech rodinách písm a je zdárná k užitnou s podtržením. 

Poznámka. Funkci Gamma v bodě x značíme \( \Gamma(x) \) a počítáme ji podle vzorce:

\[
\Gamma(x) = \int_0^\infty e^{-t}t^{x-1}dt \quad (x > 0).
\]

Speciálne pro \( x = n \in \mathbb{N} \) je \( \Gamma(n) = (n-1)! \) a pro \( a \in (0, 1) \) je

\[
\Gamma(a) \Gamma(1-a) = \frac{\pi}{\sin \pi a}.
\]

Definice. Nechť \( A \) je čtvercová matice s n sloupci a řádky a s prvky \( a_{ij} \). Pak číslo

\[
det A = |A| = \sum_{\pi = (i_1, \ldots, i_n)} \text{sgn} \pi \cdot a_{i_1j_1}a_{i_2j_2}\cdots a_{i_nj_n}
\]

nazýváme determinantem matice \( A \).

Picture 1. A test of Charter font family

cess the glyphs of the tested font, more interesting samples which mix the normal variant of the tested font with bold or italic and a complex math typesetting sample of the tested font. The result (viewed by \texttt{xdi ofstest}) is shown in picture 1. Note that the dimensions mentioned on margins are not true because the picture is scaled down in this article.

Now we'll initialize \texttt{ofstest} without any declaration file:
as default. We have chosen CMRoman family and

*** What to do with family CMRoman ?

(type command or \help): \list

```
[CMRoman/]
\rm, \bf, \it, \bi, \val, \bsel
[CMSans/]
\rm, \bf, \it, \bi
[CMTypeWriter/]
\rm, \bf, \it, \bi, \val
[Helivetica/]
\rm, \bf, \it, \bi, \umsr, \bsdf, \bsit, \bsbi
[Courier/]
\rm, \bf, \it, \bi
```

... you can read more declaration files by \decl

*** What to do with family CMRoman ?

(type command or \help): \list

```
[CMRoman/at12pt], encoding: 6a, variants:
\rm () car10 at12pt + tcrm100 at12pt
\bf (Bold) cabs10 at12pt + tcbh100 at12pt
\it (Italic) csit10 at12pt + tcit100 at12pt
\bi (BoldItalic) csbti10 at12pt + tctbi100 at12pt
\sl (Slanted) cssl10 at12pt + tcsl100 at12pt
\bxsl (BoldSlanted) csbxsl10 at12pt + tcsbl1200 at12pt
\rm () csr10 at12pt + tcrm1200 at12pt
\bf (Bold) csbx10 at12pt + tcbx1000 at12pt
\it (Italic) csti10 at12pt + tcsti1000 at12pt
\bi (BoldItalic) csbti10 at12pt + tctbi1000 at12pt
\sl (Slanted) cssl10 at12pt + tcsl1000 at12pt
\bxsl (BoldSlanted) csbxsl10 at12pt + tcsbl1200 at12pt
\rm () larm1200 at12pt + tcrm1200 at12pt
\bf (Bold) lbx1200 at12pt + tcbx1200 at12pt
\it (Italic) liti1200 at12pt + tcti1200 at12pt
\bi (BoldItalic) lbx1200 at12pt + tcbx1200 at12pt
\sl (Slanted) cssl1200 at12pt + tcsl1200 at12pt
\bxsl (BoldSlanted) csbxsl1200 at12pt + tcsbl1200 at12pt
```

Registered font encodings: 6a, 8t, 8z. Extra: 8c.

Modifications: { 8z:csfont }

You can see the result in picture 2.

I have printed for myself such a short catalogue of all fonts installed on my computer. There are hundreds families and thousands variants listed. This short catalogue is my “reference book” when I need to select some interesting font for my work. Let me circulate this “reference book” as an example around this lecture room.

It is irrelevant whether you are a plain\TeX{} user, a \LaTeX{} user, a Con\TeX{}t user or whatever else user because you can do this font test outside of your document. I hope that the ofstest.tex macro described in this section would be useful for you even if you have to use tex or csplain command in order to run the ofstest.tex macro. I mean that it is not important for you. The goal is achieved: you have a good control over your fonts. You can \list your chosen font family and read the metric name. Then you can use a primitive \font in your document. Of course it is not a very good idea...

### 3 Basics about OFS itself

The OFS was designed and finely tuned for plain\TeX{} because I am a plain\TeX{} user. Nevertheless there exists OFS for \LaTeX{} too with the same user environment because \LaTeX{} users ask me to do it. I never use \LaTeX{} because I have no control over all aspects of my document in \LaTeX{}.

OFS implements (apart from others things) features similar to NFSS: font selection independent on encoding/size/family/variant. The code for plain\TeX{} does not use NFSS and implements all features by itself. On the other hand, the \LaTeX{} variant of OFS is based on NFSS. Roughly speaking, the \LaTeX{} version of OFS implements only a user environment and dictionary which converts the family names from human readable form to NFSS cryptic names (such as pbk for Bookman).

The differences between OFS for plain\TeX{} and OFS for \LaTeX{} are explained in detail in document-
All other user-level commands have the same syntax than in L\textsc{a}\textsc{t}e\textsc{x}.

\texttt{inpicture}: \begin{verbatim}
\begin{verbatim}
\setfonts{[Family]/[size]} ...
\end{verbatim}
\end{verbatim}
\begin{verbatim}
\setfonts{[/size]} ...
\end{verbatim}
\begin{verbatim}
\fontdef \name{[Family]/[size]} ...
\end{verbatim}
\begin{verbatim}
\fontdef \name{[Family-vr]/[size]} ...
\end{verbatim}
\begin{verbatim}
\setfonts{[Family-bf]/[size]} ...
\end{verbatim}
\begin{verbatim}
\fontusage ...  
\end{verbatim}
\end{verbatim}

\texttt{\fontusage}: \begin{verbatim}
\fontusage: ============== Olsak's Font System, usage: ============== 
\end{verbatim}

Picture 2. A short catalogue of Antykwa Toruńska

\input{ofs/ofs} \loadingenc=1 ...
\end{verbatim}
\begin{verbatim}
\setfonts{[/size]} ...
\end{verbatim}
\begin{verbatim}
\fontdef \name{[Family]/[size]} ...
\end{verbatim}
\begin{verbatim}
\fontdef \name{[Family-vr]/[size]} ...
\end{verbatim}
\begin{verbatim}
\setfonts{[Family-bf]/[size]} ...
\end{verbatim}
\begin{verbatim}
\fontusage ...  
\end{verbatim}
\end{verbatim}

\texttt{\fontusage}

2 Note that some parts of the article about the use of OFS have similar (or somewhere the same) text as article [5].
The \showfonts prints the initialised font families. You can include the OFS macro without additional declaration files in square brackets (that is, use only \input ofs). In such situation, the basic six font families (as mentioned in previous section) are initialized.

\showfonts

SFS (1.0): The list of known font families:

defaults:
[CMTypewriter/]
[CMRoman/]
[CMSans/]
[Times/]
[Helvetica/]
[Courier/]

This is the same listing as shown in the previous section. The family names are printed in square brackets here and followed by variant switches available for each font family.

The basic fonts from Computer Modern family (by Donald Knuth) are collected into three families here: CMRoman, CMSans and CMTypewriter. This is self explanatory. Note that the common variant BoldItalic (\bi) is missing in CMSans and CMTypewriter. On the other hand, the “special” variant \sl (slanted) is available in CMRoman and CMTypewriter families.

The declaration files (other font families are declared here) can be included in your document using syntax with square brackets or you can \input them explicitly. Then you need not use \input ofs because declaration files are able to \input ofs automatically if this macro is not included before.

When you write

\$ tex allfonts \showfonts \end | less

you get the listing of all font families available in your TeX distribution. This is another way to get this listing without using the cfdtest.tex macro.

If you need to recall what the special variant switches mean (\nrm in Helvetica family, for instance), you can try to switch to this family and look to the log file (or to the terminal if \displayfontmessages is set):

\showfonts

SFS (1.0): Font family Helvetica at10pt (enc=8z) activated:
SFS (1.0): \nrm (Narrow) \nbf (NarrowBold) \it (Oblique) \bi (NarrowBoldOblique)

\nrm (Narrow) \nbf (NarrowBold) \it (Oblique) \bi (NarrowBoldOblique)
\nbi (NarrowBoldOblique)

Oh yes, \nrm means Narrow variant of Helvetica family. You can see that \it sometimes means the Italic variant and sometimes the Oblique variant.

4 The setfonts command

You can select the font family and/or font size by the \setfonts command. Two parameters separated by a slash are in square brackets. The first parameter is the font family name and the second one is the font size. If one of the parameters is missing then its aspect stays unchanged. After the font family is selected, you can use the variant switches. Most common switches are \rm, \bf, \it, \bi, but other switches can be available for some font families. See the listing produced by \showfonts for more detail.

The \setfonts command keeps the variant from previous family unchanged after setting of the new family if the current variant is available in new font family. If not then the \rm variant is initialized. All families have to support at least the \rm variant.

Examples:

\input ofs [pantyk]

\setfonts [AntykwaTorunska/10.5] % the normal font
\setfonts [/14]\bf % used for titles
\setfonts [10]\rm % for footnotes
\setfonts [Helvetica]/\it % for citations
\setfonts [CMTypewriter/] % monospaced font

The main advantage is that you can use the same names of font families as in Typo-catalog and you need not remember the cryptic names of tfm files or abbreviations of family names in NFSS.

If the family name is not present in the internal OFS dictionary (perhaps misspelling), then the \setfonts command prints a warning plus all available families to log and to the terminal (just like the \showfonts command).

The font size can be specified as a decimal number without unit (the unit pt is appended automatically) or you can type number with an arbitrary TeX unit (“mm” for example). Moreover, you can use the keyword “scaled” before number with the same meaning as in the \font primitive. OFS introduces the new keyword “mag” followed by a decimal number (decimal point is required). This number denotes the fraction for the current font size. For example:

\setfonts [\it[mag=0.8]]

This text is \it smaller by small \smaller and smaller \smaller and still smaller characters and the normal size is used here.

yields:
This text is typeset by small and smaller and still smaller characters and the normal size is used here.

We can use this feature in the LaTeX logo, for example:

\def\LaTeX{L\kern-.2em\raise.45ex\hbox{\setfonts[1/7]\kern-.05em\TeX}}

This solution works in titles (the raised “A” is bold in such a situation), in normal text, footnotes, italics etc. and in all font families. This feature is not implemented in NFSS and so the LaTeX logo has its “A” implemented as superscript math font in \LaTeX kernel. I think that this is not the best idea.

Another usage of mag keyword is to make corrections of not perfectly the same visual ex height of used font. This problem arises for instance if you combine the CMTypewriter family with some common PostScript font families. The CMTypewriter font seems to be smaller if you use exactly the same design size. This is no problem: you can define \tt as \setfonts CMTypewriter/mag1.1 and the ex height is balanced. This definition works in all sizes and font variants.

The NFSS keeps track of another aspect of fonts: the “font weight”. The main reason of this feature is to keep boldface text (in titles, for example) including its italics part. I decided that this feature is not needed in OFS because users can define titles in the following way:

\def\chapterfont\setfonts[1/14]\bf \let\it=\bi

Moreover, you can simply define macros which keep more than five aspects (NFSS keeps exactly five aspects). You can find examples in the \TeX support of Storm’s font, where the big DynaGrotesk family implements a special font selector that keeps the level of “condensation of the font” in addition to the weight, variant (normal/italics), size and encoding.

There exist another task with font variants which cannot be solved by a simple command \let\it=\bi (sounds like a famous song from Beatles). Imagine that you need to implement the footnote pointers as raised and smaller text which cannot be solved by a simple command.

This problem arises for instance if you combine the CMTypewriter family with some common PostScript font families. The CMTypewriter font seems to be smaller if you use exactly the same design size. This is no problem: you can define \tt as \setfonts CMTypewriter/mag1.1 and the ex height is balanced. This definition works in all sizes and font variants.

The NFSS keeps track of another aspect of fonts: the “font weight”. The main reason of this feature is to keep boldface text (in titles, for example) including its italics part. I decided that this feature is not needed in OFS because users can define titles in the following way:

\def\chapterfont\setfonts[1/14]\bf \let\it=\bi

Moreover, you can simply define macros which keep more than five aspects (NFSS keeps exactly five aspects). You can find examples in the \TeX support of Storm’s font, where the big DynaGrotesk family implements a special font selector that keeps the level of “condensation of the font” in addition to the weight, variant (normal/italics), size and encoding.

There exist another task with font variants which cannot be solved by a simple command \let\it=\bi (sounds like a famous song from Beatles). Imagine that you need to implement the footnote pointers as raised and smaller text which cannot be solved by a simple command.

This problem arises for instance if you combine the CMTypewriter family with some common PostScript font families. The CMTypewriter font seems to be smaller if you use exactly the same design size. This is no problem: you can define \tt as \setfonts CMTypewriter/mag1.1 and the ex height is balanced. This definition works in all sizes and font variants.

The NFSS keeps track of another aspect of fonts: the “font weight”. The main reason of this feature is to keep boldface text (in titles, for example) including its italics part. I decided that this feature is not needed in OFS because users can define titles in the following way:

\def\chapterfont\setfonts[1/14]\bf \let\it=\bi

Moreover, you can simply define macros which keep more than five aspects (NFSS keeps exactly five aspects). You can find examples in the \TeX support of Storm’s font, where the big DynaGrotesk family implements a special font selector that keeps the level of “condensation of the font” in addition to the weight, variant (normal/italics), size and encoding.

There exist another task with font variants which cannot be solved by a simple command \let\it=\bi (sounds like a famous song from Beatles). Imagine that you need to implement the footnote pointers as raised and smaller text which cannot be solved by a simple command. You can see that the last line of our example declares the title font by \fontdef as “Charter-bf/14.4”. On the other hand the \sectionfont is declared as “[Charter/12]bf”. Where is difference? The \titlefont is equivalent to a single font described as “[Charter-bf/14.4]”. It means that the \titlefont is the same font selector as if \fontdef... is used. On
the other hand, \sectionfont is a font family selector: it initializes the whole font family and sets the \textbf{bf} variant as default. There are consequences of this difference: variant switches will work incorrectly in the title of a document but they will work correctly in titles of sections. The author of these macros assumes that the title of a document will be printed only by a single font (for example he knows the text of this title).

You can see that plain\TeX users have to set the \baselineskip for each font size manually. The previous example shows that footnotes will be typeset by a different \baselineskip. The section titles are supposed to be short (one line), thus \baselineskip is not set here.

Why the \footnotefont and \headfont is declared by an explicit font family name? Why \fontdef\footnotefont[/8] (or \headfont[/9]) is not sufficient? The answer is simple: the footnote or \output routine (which makes headers) can be run while the \tt font family is active. In such a situation the solution \fontdef\footnotefont[/8] gives bad result.

OFS provides \knownfam and \ifknownfam commands. You can test by these commands whether given font family is installed in OFS. This feature gives you the possibility to interchange the document source between many users. Each of them can own a different font collection. The \knownfam would be useful if given font size manually. You can see that plain\TeX users have to set the \baselineskip for each font size manually.

The list of families in \selectfam parameter should be ended by a family from collection of six default families which are installed on each OFS-\TeX system.

6 Font encoding

OFS for plain\TeX\textsuperscript{3} initializes the CSfont encoding by default. This does not matter for English users because CSfonts have absolutely the same encoding(s) as Computer Modern in slots 0 to 127. The users of T1 encoded fonts have to set the default font encoding by the code: \def\fotenc{8t} before OFS is loaded. The encoding names are inspired by [7] by Karl Berry.

Theoretically, you can switch between encoding inside the document but this is not a common practice:

\input ofs
\setfonts [Times/]
text 1 \% used metric: ptmr8t, OFSfont encoding
\fontdef\footnotefont{8t}
\setfonts [7]
text 2 \% used metric: ptm8t, T1 encoding

If you are using another font encoding with metric names *xy (for example) then you can do \def\fotenc{xy}, no problem there.

You can find the files ofs-8z.tex, ofs-8t.tex, ofs-8c.tex etc. in the OFS package. The accent declarations and other encoding-dependent macros are situated there. By default, none of these files is read. It means that the accent macros keep their original meaning from plain\TeX. You can use \input on one or more of these files (there are no conflicts in these files). After an encoding file is read then the original meaning of plain\TeX macros like \textbackslash{\textbackslash{v}}, \textbackslash{\textbackslash{v}}, \textbackslash{\textbackslash{ss}} is lost, of course.

If you set \loadingenc=1 then OFS reads the files ofs-\{encoding\}.tex automatically when first \setfonts with given \fotenc occurs. The declarations are then stored globally and files are read in special mode where spaces at end of lines and empty lines are ignored. Default is \loadingenc=0 (users have to load encoding files manually) but \loadingenc=1 is highly recommended.

After encoding files are read then accent macros (such as \textbackslash{\textbackslash{v}}, \textbackslash{\textbackslash{v}} etc.) followed by a character, or glyph macros (such as \textbackslash{promile}) expand to character codes depending on the actual content of \fotenc macro.

You can look at some parts of ofs-8z.tex and ofs-8t.tex files:

\verbatim
%
% Default accents in CM
\accentdef \textbackslash{\textbackslash{v}} = 9z \{accent 18 \} \% grave
\accentdef \textbackslash{\textbackslash{v}} = 8z \{accent 19 \} \% acute
\accentdef \textbackslash{\textbackslash{v}} = 8z \{accent 20 \} \% caron
\accentdef \textbackslash{\textbackslash{v}} = 8z \{accent 21 \} \% breve

\endverbatim

\textsuperscript{3} OFS for \LaTeX does not solve font encoding, use NFSS tools for font encoding manipulation
This example is self-explanatory and illustrates the language for accents and encoding-dependent macro declarations. You can find more information in documentation [4].

OFS takes into account the possibility of existence of a special metric with extra characters (like \euro). Each basic metric can be connected with such an “extra metric” into a couple. These couples are used (for example) for CM Roman family, where basic metric depends on \fotenc and extra metric is available then the character from extra metric will be used, of course.

The Euro is lost. OFS prints the following warning:

OFS uses Euro symbol from an extra metric in the CM Roman family, but when Times is activated then the Euro is lost. OFS prints the following warning:

Now the missing \euro will be substituted by text “Euro” for all font families where \euro is unavailable. If \euro is available then the character from basic or extension metric will be used, of course.

A similar feature is implemented for \accentdef characters. You can find more information about exceptions from standard encodings in [4].

I think that Knuth’s \mathhexbox macro for text characters (like $\S$) is a somewhat bad idea because the result is dependent on setting of math fonts (no text fonts), it is independent of text font size and text font variant. This is the reason why OFS defines \oshexbox macro which respects font size and font variant (if the current variant is a “standard” one $\rm, \it, \bf, \bi$). First, you can declare the “special font family” by:

\oshexboxdef \langle\rangle \langle\rangle \langle\rangle \langle\rangle
where \textit{name} is a name of this special family, \texttt{rm}, \texttt{bf}, \texttt{it}, \texttt{bi} are metric files for given variants. Then the command \texttt{\textbackslash{ofsdeclarefamily} \textit{name}(\textit{hexa code}) prints the glyph from slot of \textit{hexa code} from one of four declared metric files. It keeps the current variant and current font size (the metric files are internally loaded by \texttt{\textbackslash{font}) .... (metric file) at(current fontsize)). For example:

\begin{verbatim}
\ofsdeclarefamily [AvantGarde] \% -------------- AvantGarde
\loadtextfam (Book) pagk\fotenc; \% rm
(Demi) pagk\fotenc; \% bf
(BookOblique) pagk\fotenc; \% it
(DemiOblique) pagk\fotenc;8c; \% bi
\end{verbatim}

is another solution of “missing Euro” problem for families where \texttt{euro} is unavailable.\footnote{The metric files in this example are loaded from \texttt{eurosym} package.}

\section{Declarations files}

Look into \texttt{a35.tex} now for an example of language of declaration files:

\begin{verbatim}
XXX Times, Helvetica, Courier is in OFS defaults
\ofsdeclarefamily [AvantGarde] \% -------------- AvantGarde
\loadtextfam (Book) pagk\fotenc; \% rm
(Demi) pagk\fotenc; \% bf
(BookOblique) pagk\fotenc; \% it
(DemiOblique) pagk\fotenc;8c; \% bi
\end{verbatim}

The mapping between font family names and metric files is defined here.\footnote{OFS for \LaTeX uses other declaration files \texttt{.sty}. The mapping between family names and NFSS short names are defined there.} Each family declares four metric files for four common variants \texttt{rm}, \texttt{bf}, \texttt{it} and \texttt{bi}. Empty parameter means that the variant is missing. The metric names include \texttt{\textbackslash{font}} in order to get names \texttt{pagk8z}, \texttt{pagk8t}, etc. after expansion. The extra metric \texttt{8c} is declared before the last semicolon. The commands from parameter of the \texttt{\ofsdeclarefamily} are processed when \texttt{\setfonts} is used. You can see the alternative definition of \TeX logo here. It optimizes the visual aspect of this logo specially for each font family.

The optional parameters of \texttt{\loadtextfam} macro are written in brackets and these parameters declare mapping from short variant switches to the full variant names printed to the log and terminal. For example, \texttt{\textbackslash{it} variant} is BookOblique in AvantGarde font family. If the common name is used (Bold/Italics/BoldItalics) then this parameter can be missing.

The \texttt{\modifyenc} commands make connections of the family to exception lists from standard encodings. For example, the \texttt{8c:poor} list was mentioned in the previous section.

Finally, the \texttt{\registerenc} commands say that the declared family is ready to be used in specified encodings. The AvantGarde and Bookman family have registered only \texttt{8z} and \texttt{8t} basic encodings. It means that these families are not available in other encodings.

How is the \texttt{CMRoman} family declared? This family includes two special features. First: the metric file names do not include the \texttt{8z} or \texttt{8t} acronym for encoding. Second: different metric names are used for different font sizes. Both problems are solved by the command \texttt{\registerertfm} (see the \texttt{ofsdef.tex} file):

\begin{verbatim}
\registerertfm cmr8z \% cmr8z \% metric for all sizes
\registerertfm cmr8t \% cmr8t \% metric for all sizes
\registerertfm cmr8c \% cmr8c \% metric for all sizes
\end{verbatim}

\section{Math fonts}

The math fonts are collected in math families (three fonts per one family) by \texttt{\textbackslash{textfont}}, \texttt{\textbackslash{scriptfont}} and \texttt{\textbackslash{scriptscriptfont}} primitives. The math fam-
ilies with number 0, 1, 2, 3 have special meaning in math typesetting. The declaration of a new math family by \TeX primitive is not too comfortable. Plain\TeX users can declare a new math family by OFS macro \texttt{\loadmathfam}. This command will be described below.

Plain\TeX user have to initialize the math fonts in OFS by \texttt{\setmath} command. The math fonts are in the same state as declared in plain\TeX macro until the \texttt{\setmath} command is used. It means that the Computer Modern at 10/7/5 pt size are used. The \texttt{\setmath} command has three parameters separated by slashes enclosed in square brackets. These parameters describe the text/script/scriptscript size of the math fonts. An empty parameter means that the mag1.0/mag.7/mag.5 (relatively to the current size of textual font) is substituted:

\texttt{\setmath [//]} is the same as \texttt{\setmath [mag1.0/mag.7/mag.5]}

The \texttt{\setmath} command calculates the needed sizes from given parameters and starts the \texttt{\mathfonts} macro followed by macro \texttt{\mathchars}. A plain\TeX user can define these macros in his own way but OFS gives the reasonable default meaning of these macros. The outcome of these default macros depends on the values of the \texttt{\fomenc} and \texttt{\mathversion} macros.

If \texttt{\def\fomenc{PS}} is used (it is default value in OFS) then \texttt{\setmath} initializes math fonts in the following way: math italic is loaded from text italics of the current text font family, family 0 is loaded from \texttt{\rm} variant of the current family. The math symbols are loaded (if it is possible) from common PostScript font Symbol. The rest (which is not included in Symbol font) is loaded from Computer Modern fonts. The math encoding is redefined (by \texttt{\mathchardef etc.} primitives) for many symbols in order to keep the accessibility of all math characters declared in plain\TeX. For example the lower letter Greek characters are loaded from slanted variant of PostScript Symbol font.

If you write \texttt{\def\fomenc{CM}} then \texttt{\setmath} loads the math fonts from Computer Modern family (like in plain\TeX) and does not change the math encoding. In this case, the \texttt{\setmath [//]} command only sets the actual sizes of these fonts depending on current text font size.

After \texttt{\input txfn.tex} you can write \texttt{\def\fomenc{TX} or {PX}. The free available TX fonts are used in such case for math typesetting. They are very similar to Times and Helvetica families and they include a large set of math glyphs.

\texttt{\def\fomenc{TX}} means that all math typesetting will be realized by TX fonts. If \texttt{\def\fomenc{PX}} is set then math italic and family 0 is copied from the current text font family.\footnote{Please, return to the section 2, picture 1. This picture does not illustrate exactly the output from ofstest example mentioned in this section. In fact, the math typesetting sample was printed by \texttt{\def\mathenc{PX} \math commands.}}

After \texttt{\input amsfn.tex} you can write \texttt{\def\fomenc{AMS}. AMS fonts will be used. If you buy the MathTimes family, you can do \texttt{\input mfn.tex} and you can use \texttt{\def\fomenc{MT}}.

You can control the math families collection loaded by \texttt{\setmath} by the value of \texttt{\mathversion} macro. OFS declares two math family collections: \texttt{\def\mathversion{normal}} and \texttt{\{bold\}. You can declare more collections if you need it. The “bold” collection is the same as “normal”, but bold variants of italics, family 0 and math symbols (if accessible) are loaded instead normal variants.

The example from \texttt{ofsddef.tex} file illustrates the language of declarations of the math fonts:

\begin{verbatim}
\def\loadNormalmath{%
  \loadmathfam 0[\mr/]% Actual Roman font
  \loadmathfam 1[\it/]% Actual Italic font
  \defaultskewchar=48
  \loadmathfam 2[\cmyx/]% Standard symbols from CM
  \chardef\symbfam 4
  \loadmathfam \symbfam [/payr] PostScript Symbol
  \chardef\symbfam 6
  \loadmathfam \symbfam [/psyro] PostScript Symbol Oblique
  \chardef\symbfam 7
  \loadmathfam \symbfam [-bi/]
  \chardef\itfam 1
  \let\slfam\undefined \let\ttfam\undefined
  \setfosize \tmpa mag1.44:
  \font \bigsymbofont=psyr \tmpa Big variant for \displaysize
  \font \bigsymbofont=psyro \tmpa

\def\loadBoldmath{%
  \loadmathfam 0[-bf/]% Actual Bold font
  \loadmathfam 1[-bi/]
  \defaultskewchar=48
  \loadmathfam 2[\cmyx/]
  \chardef\symbfam 4
  \loadmathfam \symbfam [/payr] PostScript Symbol
  \chardef\symbfam 5
  \loadmathfam \symbfam [/psyro] PostScript Symbol Oblique
  \lastfam = 7 % needs to be the same as in normal version
  \chardef\itfam 1
  \chardef\bfifam 0
  \let\slfam\undefined \let\ttfam\undefined
  \setfosize \tmpa mag1.44:
  \font \bigsymbofont=psyr \tmpa
}
\end{verbatim}
If you need to add next math families (math alphabets in NFSS terminology) then you can use the code similar as the following:

```
\addcmd\mathfonts{%
\namefontfamilybf{bb}
\namefontfamilyms{ms}
\namefontfamilyrm{rm}
\namefontfamilymi{mi}
\namefontfamilyex{ex}
}\addcmd\mathcharcodes{%
\mathchardef\bbeta "0\hex\bbfam 0C
\loadmathfam \bbfam [/bbold12]
}\addcmd\mathinitialization{%
\setmath[//]
}\addcmd\mathinitialization{%
\setmath[//]
}\addcmd\mathfonts{\addcmd\mathinitialization{\addcmd\mathfonts}}
```

The math fonts will work in all sizes (besides \textfont) in our virtual example. The math formulas are in bold variant and in right size in section titles. For normal font, \setmath[/1] is initialized on line 4 of our example. Other font selectors do not run \setmath[/1] when called, this command will be run only when \TeX enters to math mode (see everymath trick).

Let me do the last test in this article. The following code is appended to the previous example:

```
\def\section #1{\medskip{\sectionfont#1\par\nobreak}\medskip}
```

You can see that the text encoding file ofs-8z.tex and math encoding files ofs-px.tex, ofs-ams.tex, ofs-tx.tex are read automatically. A detailed report about the processing of all \font primitives is logged as the outcome of detailfontmessages command. The initialization of math fonts family at 12/8.4/6 in “bold” version is shown.

The “overflow message” prints the font identifier like \CharterRM/10pt. This is more legible than cryptic font identifiers used in NFSS.

References

1. ftp://math.feld.cvut.cz/pub/olsak/ofsl
2. www.stormtype.com
4. Petr Olsák. **OFS**: Olšákův fontový systém. 2001. The English documentation is in files ofsdocc-e.tex, ofsdocc-e.pdf