This version of the paper is only draft! It waits to Petr Sojka's corrections.

The font management with the OFS

Petr Olšák, Petr Sojka

Faculty of Electro Engineering CVUT, Prague, Faculty of Informatics MU, Brno
Email: `petr@olsak.net, sojka@fi.muni.cz`

Abstract:

The widely used New Font Selection System (NFSS) has its pros and cons. To have uniform platform for large font set management and selection for both plain and LaTeX users, and to extend the possibilities of the NFSS, an Olšák's Font System (OFS) has been designed and implemented.

The user interface of this macro is the same for plain and LaTeX, with two independent implementations. First implementation is based on plain.tex macros only; the second one for LaTeX users is implemented as an additional layer over NFSS macros, and is slightly less elaborate (support for math font families is missing).

The main features of the system:

- The OFS defines declarative language to define the mapping from the original full names of fonts to the tfm names (in plain) or to NFSS short names of the font families (in LaTeX). The user do not need to remember the short names of tfm files and/or the short names of NFSS font families. This allows for user-defined logical font markup: one can use the real font (foundry) names from font (catalog).
- The user of the OFS can use a font separated into two TeX metrics (basic and extended tfm). The font is from the user perspective seen as one font only. The mapping works even for fonts with more than 256 characters (if one does not want to kern between all characters).
- One can choose the TeX internal encoding of fonts for your language at the beginning of your document. This feature is commonly used for Czech and Slovak languages: there are TeX fonts which encode the alphabets of these languages by Cork (T1 encoding) or by ISO-8859-2 (IL2 encoding), or even others.
- There are at least four independent parameters of fonts (family, size, encoding, variant). Possibility to switch one font parameter independently on the others is important under plain.tex where NFSS is not implemented.
- The OFS for plain includes the support of the math fonts manipulation when the PostScript fonts and/or fonts at the different sizes are used.
- One can use the `\fontusage` command which lists the short usage about OFS macros on the screen and to the log file.
- The `\showfonts` macro lists all available font families on the screen and to the log file.

- One can print the short catalogue of the all installed fonts by using `\input ofscatal.tex`.

The macro is available at ftp://math.feld.cvut.cz/pub/olsak/ofs . The declaration and support OFS files for all of the fonts offered by the Storm Type Foundry (www.stormtype.com), and e.g. MathTimes fonts have been created. The macro is in use by CSTUG users since 2001. The English documentation (in addition to the current Czech one) will be released in January 2003.

The full paper will discuss in detail the features of OFS listed above, will give examples of usage and pros and cons over current macro packages for font switching and handling in TEX.

Historical notes, plainTEX and LATEX version of OFS

In 2000, Petr Olšák got whole set of fonts from Storm Type Foundry (made by František Štorm) in order to make the TEX support for these fonts. The number of fonts was very large (many hundreds) and the names used in Typo-catalog was a bit different from pfb file names and they was *very* different from new tfm names used in TEX support because these names follow the Karl Berry's recommendation [8].

There was a presumption that the user will want to choose individual fonts and to make his/her own font collections for document design. The simple solution like `\usepackage{times}` is not possible because the user have to choose the sans serif variants, serif variants etc. from a big offer by Storm Type Foundry. There are big number of such combinations—we are not able to support a style file for each such combination. Moreover, there are a great font collection for titles and posters in Štorm's package. The conclusion is: the user needs to manage with individual fonts with the same names as in Typo-catalog provided by Storm Type Foundry. The usage of tfm names or short names for font families provided by NFSS are not acceptable at user level. The support for managing of large sets of fonts implemented at macro level is needed. This was an inspiration of OFS.

The OFS was written by Petr Olšák primary for plainTEX because he is a plainTEX user. He never used LATEX nor NFSS. Thus the first version of OFS was usable only in plainTEX environment without NFSS. This version substitutes some features solved in NFSS already: an independency of encoding/size/family/variant font setting at user level, declaration of encoding-dependent commands, more comfortable declaration of math fonts (but plainTEX approach of math fonts declarations is kept because this is more understandable for plainTEX users than NFSS approach).

The LATEX users asked the similar environment as OFS for plainTEX after OFS for plain was released. This was a reason why Petr Olšák made the same user environment of OFS for LATEX (but never used it:-). This work was done by totally different macro code because the NFSS features was used in this situation. The OFS for LATEX is only the new macro level over NFSS. Roughly speaking, the

OFS for LATEX implements only the dictionary from long names of fonts used in Font-catalog to the short family names used in NFSS and defines the same "user level environment" as in OFS for plainTEX.

Because the OFS for plainTEX is more powerful and includes more interesting subjects in its implementation, we will describe only the plainTEX version of OFS in this paper.

The OFS macro is released with TEX-like license. It means, that the names of the files and its content cannot be changed if this macro is re-distributed. Only Petr Olšák can do changes. There are no more restrictions for distributors and there are no restrictions for users.

The macro is released in [??] and the same version is included in TEX support of fonts from Storm Type Foundry [3]. The TEX font metrics and fd files are included in this support. The real fonts (implemented in pfb files) are not free. If you take up your interest in these fonts you have to buy them. See [1] for more details. One family called Lido is offered as free for non-commercial use by Storm Type Foundry. There is the special TEX support only for this family [4] including the OFS.

The first release of OFS macro (September 2000) had only Czech documentation. The English version of the documentation was released in February 2003 with help of Matěj Cepl. The concept of OFS was not changed since its first release and Petr Olšák does not plan any changes to the future. He is ready only to remove bugs, if any occur.

The documentation can be found at the same place as the OFS macro. It is a bit more technical than this paper.

2 Basics of user environment

The user environment is the same in both: OFS for plainTEX and LATEX. Of course, you have to include the OFS macro in plainTEX by little bit different way:

```
\input ofs [fonts, collections]
```

than in LATEX:

```
\usepackage [fonts, collections] {ofs}
```

All other user-level commands have the same syntax and meaning.

You can use the command `\fontusage`. This command prints to the terminal and to the log file the basic usage of OFS:

```
$ tex ofs \\fontusage
This is TeX, Version 3.14159 (Web2C 7.3beta5)
(/usr/local/share/texmf/tex/csplain/ofs.tex
OFS (Olsak's Font System) based on plain initialized. <Oct. 2002>
(/usr/local/share/texmf/tex/csplain/ofsdef.tex))
\fontusage: ============= Olsak's Font System, usage: =================
\input ofs [sjannon, sdynamo, a35] ... for example
\showfonts  ... shows all loaded font families (by previous \input)
\setfonts [Family/] ... local switch to the new family, after this, the
  \rm, \bf, \it, bi will switch to the variants. The current size is used.
```

```
\setfonts [/size] ... local switch to the new size of fonts, the family is
  not changed. The "size" has the following possible formats:
    at<dimen>       ... the same as \font\something=file at<dimen>
    <dimen>         ... the same as at<dimen>
    <number>        ... the same as at<number>pt
    scaled<number>  ... the same as \font\something=file scaled<number>
    mag<decimal-number> fonts will be magnified by given coefficient
                        depend on current size of the fonts.
\setfonts [Family/size] ... switch to the new family at given size
\setfonts [Family-vr/]  ... switch to the specified font, the current size
  is used. The "vr" is acronym for variant (bf for example).
\setfonts [Family-vr/size] ... switch to the specified font.
\fontdef\name [Family/size] ... same as \gdef\name{\setfonts[Family/size]}
  The "Family" or "size" parameter may be empty.
\fontdef\name [Family-vr/size] ... \name is fixed-font switch iff:
  "size" is no empty and no mag<dec-number>.
  Fixed-font switch "\name" is implemented as \global\font\name=file.
\setmath [size/size/size] ... set math it/rm as current it/rm + use PS Symbol
\nofontmessages, \logfontmessages, \displayfontmessages, \detailfontmessages
  ... the levels of log.
*
```

The `\showfonts` prints the used font collections. You can include the OFS macro without additional font collections in square brackets (it means use only `\input ofs`). The basic font collection is initialised in such situation:

```
*\showfonts
OFS (1.0): The list of known font families:
defaults:
   [CMRoman/]           \rm, \bf, \it, \bi, \sl
   [CMSans/]            \rm, \bf, \it,  -
   [CMTypewriter/]      \rm,  - , \it,  - , \sl
   [Times/]             \rm, \bf, \it, \bi
   [Helvetica/]         \rm, \bf, \it, \bi, \nrm, \nbf, \nit, \nbi
   [Courier/]           \rm, \bf, \it, \bi
```

The family names are printed in square brackets here and followed by varinat switches usable for such font family.

The basic fonts from Computer Moder family (by Donald Knuth) are collected into three families here: `CMRoman`, `CMSans` and `CMTypewriter`. This is sel explanatory. Note that the common varinat `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 of another families are stored in “declaration files” (the extension `tex` is used in OFS for plainTEX and `sty` in OFS for LATEX. It is recommended to manage the file `allfonts.tex` in all TEX distribution. This file includes the `\inputs` to the declaration files of all fonts installed in such distribution. The `allfonts.tex` file may look like:

```
$ cat ‘kpsewhich allfonts.tex‘
%%% All OFS families declared on this TeX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                          Petr Olsak
```

```
\input a35       % PostScript 35
\input ffonts    % Another free fonts
\input btfonts   % Bitstream fonts
\input skatalog  % Stromtype foundry, 89 families
```

When we write `$ tex allfonts \\showfonts \\end | less`, we get the listing of all font families available on TEX distribution. This listing includes cca three hundred font families on the computer of one of the author of this article. He need not remember the tfm file names of thousands of fonts. The part of the listing follows:

```
OFS (1.0): The list of known font families:
defaults:
   [CMRoman/]             \rm, \bf, \it, \bi, \sl
   [CMSans/]              \rm, \bf, \it,  -
   [CMTypewriter/]        \rm,  - , \it,  - , \sl
   [Times/]               \rm, \bf, \it, \bi
   [Helvetica/]           \rm, \bf, \it, \bi, \nrm, \nbf, \nit, \nbi
   [Courier/]             \rm, \bf, \it, \bi
a35.tex:
   [AvantGarde/]          \rm, \bf, \it, \bi
   [Bookman/]             \rm, \bf, \it, \bi
   [NewCentury/]          \rm, \bf, \it, \bi
   [Palatino/]            \rm, \bf, \it, \bi
   [ZapfChancery/]        \rm,  - , \it,  -
   [ZapfDingbats/]        \rm,  - ,  - ,  -
   [Symbol/]              \rm,  - , \it,  -
ffonts.tex:
   [Charter/]             \rm, \bf, \it, \bi
...
sjannon.tex:
   [JannonAntikva/]       \rm, \bf, \it, \bi, \mr, \mi
   [JannonText/]          \rm, \bf, \it, \bi, \mr, \mi
   [JannonCaps/]          \rm, \bf, \it, \bi
...
sdynamo.tex:
...
   [DynaGroteskLE/]       \rm, \bf, \it, \bi
   [DynaGroteskD/]        \rm, \bf, \it, \bi
   [DynaGroteskR/]        \rm, \bf, \it, \bi
...
stitul.tex:
   [Alcoholica/]          \rm, \bf, \it,  -
   [Monarchia/]           \rm, \bf,  - ,  -
   [MonarchiaText/]       \rm, \bf,  - ,  -
   [Clichee/]             \rm, \bf, \it, \bi
   [Regula/]              \rm,  - , \it,  -
   [Splendid/]            \rm, \bf, \it,  - , \script, \sans
   [Cobra/]               \rm, \bf,  - ,  -
   [ExcelScript/]         \rm,  - ,  - ,  - , \ext
   [ExcelScriptText/]     \rm,  - ,  - ,  - , \ext
   [Zeppelin/]            \rm, \bf,  - ,  - , \lr, \coll
   [Negro/]               \rm,  - ,  - ,  -
   [Farao/]               \rm, \bf,  - ,  - , \kr, \coll
```

```
...
slido.tex:
   [Lido/]              \rm, \bf, \it, \bi, \crm, \cbf
```

If you need recall what the special variant switches means (`\mr` and `\mi` in JannonText 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):

```
*\displayfontmessages \setfonts [JannonText/]
OFS (1.0): Font family JannonText at10pt (enc=8z) activated:
OFS (1.0):  \rm () \bf (Bold) \it (Italic) \bi (BoldItalic)
OFS (1.0):  \mr (Medium) \mi (MediumItalic)
*
```

Aha, `\mr` means MediumItalic variant of JannonText family. The `\it` means sometimes `Italic` variant and sometimes `Oblique` variant. You can try:

```
*\setfonts[Helvetica/]
OFS (1.0): Font family Helvetica at10pt (enc=8z) activated:
OFS (1.0):  \rm () \bf (Bold) \it (Oblique) \bi (BoldOblique)
OFS (1.0):  \nrm (Narrow) \nbf (NarrowBold) \nit (NarrowItalic)
            \nbi (NarrowBoldItalic)
*
```

The declaration file name is mentioned in the listing of all families (see above). For example the file `a35.tex` declares the font families from standard PostScript “35 fonts collection” from Adobe (no all ten families are declared here because three families of them are declared in basic OFS font collection).

You can write `\input a35` instead of `\input ofs [a35]`. The second alternative is recommended if more font collections are used together, for example: `\input ofs [a35, sdynamo, slido]`. You can use

```
\input ofs [allfonts]
```

(or simply `\input allfonts`) in your document if all font collections are needed or if you need not use the names of individual font collections.

OFS provides the printing of simple font catalog. For example, you can write on your command line:

```
$ pdfcsplain ofscatal [sjannon, sdynamo, stitul] ; acroread ofscatal.pdf
```

3 The setfonts command

You can select the font family and/or font size by `\setfonts` command. Two parameters are separated by slash 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 this aspect stay unchanged. After the font family is selected, you can use the variant switches. Most common switches are `\rm`, `\bf`, `\it`, `\bi`, but another switches can be available for some font family. 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:

```
\setfonts [JannonText/10.5]  % the normal font
\setfonts [/14]\bf           % used for titles
\setfonts [/8]\rm            % for footnotes
\setfonts [DynagroteskR/]\it % for citations
\setfonts [CMTypewriter/]    % monospaced font
       % if you use \setfonts [CMTypewriter/] when
       % it is active then Typewriter italics is used.
```

The main advantage is that you can use the same names of font families as in Typo-catalog and you need not remember the cryptical 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 `\setfonts` command prints warning plus all available families to log and to the terminal (the same as the `\showfonts` command).

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

```
\def\smaller{\setfonts [/mag0.8]}
This text is {\smaller typeset by small \smaller and more small
\smaller and more more small characters} and the normal size is used
here.
```

yields to:

This text is typeset by small and more small and more more small characters and the normal size is used here.

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

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

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

The another usage of mag keyword is to make a corrections of not perfectly the same visual ex height of used font. This problem arises if you combine the CMTypewriter family with some common PostScript font families for instance. 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 another aspect of fonts: the font weight. The main reason of this feature is to boldify all text (in titles, for example) including its italics part.

I decided that this feature is not needed in OFS because plain user can define titles in the following way:

```
\def\chapterfont {\setfonts[/14]\bf \let\it=\bi}
```

Moreover you can simply define the macros which keeps more than five aspects (NFSS keeps exactly five aspects). You can find the examples in TEX support of Štorm's font where the big Dynagrotesk family implements the special font selector which keeps the level of "condensation of the font" in addition to the weight, variant (normal/italics), size and encoding.

You can declare you own font-switches in OFS by `\fontdef` command. The code `\fontdef\fontswitch [Family/size]` declares the `\fontswitch` similar as `\gdef\fontswitch{\setfonts[Family/size]}`. Example:

```
\input ofs [ffonts]                % font Charter is in free fonts
\setfonts [Charter/10pt]           % default family
\fontdef\tt [CMTypewriter/mag1.1]  % monospaced font
\fontdef\verbtt [CMTypewriter/8]   % monospaced font in verbatim
\fontdef\headfont [Charter/9]      % the font+size for headers
```

It would be more convenient to write the word "Charter" only once (this word seems to be redundant in declaration of `\headfont`). But the macro code `\fontdef \headfont [/9]` cannot work everywhere because the `\headfont` can be used for headers in output routine and this header have to be in Charter every time, no in CMTypewriter, if CMTypewriter family is the current when output routine is called out.

You can solve this problem by using the "`!`" character for the family parameter in `\fontdef`. This character is substituted by the current family in the time of `\fontdef` declaration:

```
\fontdef\headfont [!/9] \addcmd\small {\rm} % the font for headers
```

Moreover the `\addcmd` is used here. This command adds the additional code to the macro `\headfont` thus, the example above has the same effect as:

```
\gdef\headfont {\setfonts[Charter/9]\rm} % the font for headers
```

but the word Charter is not used here. Now, you can change the default family only on one place and the new default family will be used in headers too.

The whole font declaration for your plainTEX document can be localized in one place in your macro code and it can be look like the following code:

```
\input ofs [ffonts]                   % font Charter is in free fonts
\setfonts [Charter/10pt]              % default family
\fontdef \tt [CMTypewriter/mag1.1]    % monospaced font
\fontdef \verbtt [CMTypewriter-rm/8]  % monospaced font in verbatim
\fontdef \headfont [!/9]     \addcmd \headfont {\rm}
\fontdef \abstractfont [!/9] \addcmd \abstractfont {\baselineskip11pt\rm}
\fontdef \sectionfont [!/12] \addcmd \sectionfont {\bf \let\it=\bi}
\fontdef \titlefont [!-bf/14.4]       % for the title of the document
```

If the "`-vs`" (where `\vs` is the variant switch) is used in parameter of `\fontdef` or `\setfonts` then the whole family is not initialized but only the single font is used as the current font. The `\fontdef\name` is the equivalent of `\global\font\name` in such situation. This convention is used in the example above for title font because no variant switches is used in text of title.

4 Font encoding

OFS for plainTEX[1] initializes the CSfont encoding by default. This does no 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 macro `\fotenc` includes a part of used metric file names. If this macro is not defined then OFS define it as `8z`—it means CSfont encoding.

If `\fotenc` has the value `8z` then the families `CMRoman`, `CMTypewriter` and `CMSans` will work with CSfonts and the font metrics with `8z` encoding is used for another font families. If `\fotenc` is set to `8t` then DC fonts are used in `CMRoman`, `CMTypewriter` and `CMSans` families. The EC fonts can be used instead DC fonts, but you have to write the declaration file for OFS and for EC fonts. This file does not exist now, sorry.

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

```
\input ofs
\setfonts [Times/]  text 1 % used metric: ptmr8z, CSfont encoding
\def\fotenc{8t}
\setfonts [/]       text 2 % used metric: ptmr8t, T1 encoding
```

If you are using another font encoding with metric names `*8x` (for example) then you can do `\def\fotenc{8x}`. No problem.

You can find the files `ofs-8z.tex` and `ofs-8t.tex` in OFS package. The accents declaration and another encoding-dependent macros are included here. By default, no of these files is read, it means that the accent macros keep its original meaning from plainTEX. You can use `\input` of one of these files or `\input` both files (there are no conflicts in these files). Assume that both files are read. Then accent macros (such as `\v`, `\'` etc.) followed by a character expands to the new character code of CSfonts (if `\fotenc` is `8z`) or to the new code of T1 encoding (if `\fotenc` is `8t`). Another encoding dependent macros (such as `\promile`) work in the chosen encoding by `\fotenc`.

We can look into the parts of `ofs-8z.tex` and `ofs-8t.tex` files:

```
%%% Default accents in CM
\accentdef  \` *    8z  {\accent 18 }                   % grave
\accentdef  \' *    8z  {\accent 19 }                   % acute
\accentdef  \v *    8z  {\accent 20 }                   % caron
\accentdef  \u *    8z  {\accent 21 }                   % breve
...
```

---

[1] OFS for LATEX does not solve font encoding, use NFSS tools for font encoding manipulation

```
%%% Standard characters in plain (redefined here)
\def\aa{\r a}
\def\AA{\r A}
\characterdef \i    8z  16
\characterdef \j    8z  17
\characterdef \SS   8z  {SS}
\characterdef \AE   8z  29
...
%%% Extra characters from CS fonts
\characterdef \promile      8z  141
\characterdef \varhyphen    8z  156
\characterdef \flqq         8z  158
\characterdef \frqq         8z  159
\characterdef \clqq         8z  254
\characterdef \crqq         8z  255
...
%%% Accented letters from CS fonts
\accentdef  \‘ A      8z  152
\accentdef  \’ A      8z  193
\accentdef  \" A      8z  196
\accentdef  \‘ a      8z  184
\accentdef  \’ a      8z  225
...
%%% Default accents in Cork
\accentdef  \‘ *    8t  {\accent 0 }
\accentdef  \’ *    8t  {\accent 1 }
\accentdef  \^ *    8t  {\accent 2 }
\accentdef  \  *    8t  {\accent 3 }
\accentdef  \" *    8t  {\accent 4 }
...
%%% Standard characters in plain (redefined here)
\def\aa{\r a}
\def\AA{\r A}
\characterdef \i    8t  25
\characterdef \j    8t  26
\characterdef \SS   8t  223
\characterdef \AE   8t  198
...
\characterdef \promile              8t {\%\char 24 }
\characterdef \textpertenthousand   8t {\%\char 24\char 24 }
...
%%% Accented letters from T1 encoding
\accentdef  \. i    8t ‘\i
\accentdef  \u A    8t 128
\accentdef  \k A    8t 129
\accentdef  \’ C    8t 130
\accentdef  \v C    8t 131
...
```

This example is self explanatory and illustrates the language for accents and encoding-dependent macro declarations. You can find more information in documentation [6].

The LaTeX user can be familiar with the commands with similar meaning: `\DeclareTextSymbol`, `\DeclareTextComposite`, etc.

OFS calculates with the possibility of existence of a special metric with extra characters (such us `\euro`). Each basic metric can be connected with such “extra metric” to the couple. These couples are used in TEX support of Storm Type Foundry because these fonts have more than 256 characters. There is prepared an extra metric `*6s` for each basic metric `*8z` or `*8t` in TEX support of Štorm’s fonts. The `*6s`-encoding specific macros are declared in `stormenc.tex` by `\accentdef` and `\characterdef` commands. If the current metric has declared an extra metric and user calls the macro declared by `\characterdef` or `\accentdef` and this macro is not declared for current encoding but only for the encoding of the extra metric then the extra metric is temporary used without any user intervene. Example:

```
\characterdef \euro           6s  136
Now the \euro{} works in all fonts with extra metric *6s
```

5 Declaration files

Look into `a35.tex` now for an example of language of the declaration files:

```
%%% Times, Helvetica, Courier is in OFS defaults
\ofsdeclarefamily [AvantGarde] {% -------------------- AvantGarde
   \loadtextfam (Book)         pagk\fotenc;%    \rm
                (Demi)         pagd\fotenc;%    \bf
                (BookOblique)  pagko\fotenc;%   \it
                (DemiOblique)  pagdo\fotenc;;%  \bi
   \def\TeX{T\kern-.08em\lower.3333ex\hbox{E}\kern-0.09emX}%
}
\ofsdeclarefamily [NewCentury] {% -------------------- NewCenturySchlbk
   \loadtextfam (Roman)        pncr\fotenc;%    \rm
                               pncb\fotenc;%    \bf
                               pncri\fotenc;%   \it
                               pncbi\fotenc;;%  \bi
   \def\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX}%
}
...
```

The mapping between font family names and metric files is defined here.[2] Each family declares four metrics for four common variants `\rm`, `\bf`, `\it` and `\bi`. Empty parameter means that the variant is missing. The metric names are saved by `\fotenc` macro in order to possibility of changing the used encoding. The extra metric can be written before the last semicolon (not used in this example). The commands from parameter of the `\ofsdeclarefamily` are processed when `\setfonts` is used. You can see the alternative definition of TEX logo for each family in `a35.tex` file on order to optimize the visual aspect of this logo.

The optional parameters of `\loadtextfam` macro is written in brackets and these parameters declares mapping of short variant switches and the full variant

---

[2] OFS for LATEX uses other declaration files: the mapping between family names and NFSS short names are defined here.

names printed to the log and terminal. For example, BookOblique is a `\it` variant in AvantGarde font family. If the common name is used (Bold/Italics/BoldItalics) then this parameter can be missing (see NewCentury family, for instance).

How the special variants are declared? Look into the file `sjannon.tex`, for example:

```
\ofsdeclarefamily [JannonAntikva] {% -------- Jannon Antikva
   \loadtextfam sjnr\fotenc;%
                sjnb\fotenc;%
                sjnri\fotenc;%
                sjnbi\fotenc;6s;%
   \newvariant2 \mr (Medium)        sjnm\fotenc;6s;%
   \newvariant3 \mi (MediumItalic)  sjnmi\fotenc;6s;%
}
\ofsdeclarefamily [JannonText] {% ------------ Jannon Text
   \loadtextfam sjnrg\fotenc;%
                sjnbg\fotenc;%
                sjnrig\fotenc;%
                sjnbig\fotenc;6s;%
   \newvariant2 \mr (Medium)        sjnmg\fotenc;6s;%
   \newvariant3 \mi (MediumItalic)  sjnmig\fotenc;6s;%
}
...
```

Note that there is declared the extra metric with `6s` encoding here. See the documentation [6] for more information.

How the `CMRoman` etc. families are declared? These families have two special subjects. First: the metrics file names do not include the `8z` or `8t` acronym for encoding. Second: the different metrics are used for different font sizes. Both problems are solved by the OFS command `\registertfm` (see the `ofsdef.tex` file):

```
\registertfm cmr8z     -      csr10  % metric for all sizes
\registertfm cmr8z  0pt-6pt   csr5
\registertfm cmr8z  6pt-7pt   csr6
\registertfm cmr8z  7pt-8pt   csr7
\registertfm cmr8z  8pt-9pt   csr8
\registertfm cmr8z  9pt-10pt  csr9
\registertfm cmr8z  10pt-12pt csr10
\registertfm cmr8z  12pt-17pt csr12
\registertfm cmr8z  17pt-*    csr17
...
\registertfm cmr8t     -   dcr10     % metris for all sizes
...
\ofsdeclarefamily [CMRoman] {% ------------ Computer Modern Roman
   \loadtextfam cmr\fotenc;%
                cmbx\fotenc;%
                cmti\fotenc;%
                cmbxti\fotenc;;%
   \newvariant8 \sl (Slanted) cmsl\fotenc;;%
}
```

6 Math fonts

The math fonts are collected in math families (three fonts per one family) by `\textfont`, `\scriptfont` and `\scriptscriptfont` primitives. The math families with number 0, 1, 2, 3 have special meaning in math typesetting. The declaration of new math family by TEX primitives is not too comfortable. PlainTEX user can declare the new math family by OFS macro `\loadmathfam`. This command will be described below.

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

```
\setmath [//] is the same as \setmath [mag1.0/mag.7/mag.5]
```

The `\setmath` command calculates the needed sizes from given parameters and starts the `\mathfonts` macro followed by macro `\mathchars`. The plainTEX user can define these macros in his own opinion but OFS gives the reasonable default values of these macros. These macros run the variant part of the code depend on the values of the `\fomenc` and `\mathversion` macros.

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

If you use `\def\fomenc{CM}` then `\setmath` loads the math fonts from Computer Modern family and does not change the math encoding. In this case, the `\setmath[//]` command only sets the actual sizes of these fonts depend on current text font size.

If you buy the MathTimes family, you can `\input` the `ofsmtdef.tex` file (this is a part of OFS package) and you can use `\def\fomenc{MT}`. Then the `\setmath` command loads the math italics from current `\it` font and family 0 from current `\rm` font. Moreover, it loads the MathTimes fonts for all math symbols. Unlike `\def\fotens{PS}` mode there is no “rest”: no symbols needs to be realized from Computer Modern family.

You can control the math families collection loaded by `\setmath` by the value of `\mathversion` macro. OFS declares two math families collections: `\def\mathversion{normal}` and `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.

The example from `ofsdef.tex` file illustrates the language of declarations of the math fonts:

```
\def\defaultmathfonts{\csname load\fomenc\mathversion math\endcsname}
\def\defaultmathchars{\csname set\fomenc mathchars\endcsname}
\def\mathfonts{\defaultmathfonts}
\def\mathchars{\defaultmathchars
   \let\mathchars=\relax % to protect the twice math-setting
}
\def\loadPSnormalmath{%
  \loadmathfam 0[tenrm/]%              Actual Roman font
  \loadmathfam 1[tenit/]%              Actual Italic font
  \loadmathfam 2[/cmsy]%               Standard symbols from CM
  \noindexsize\loadmathfam 3[tenex/]%  Standard extra symbols from CM
  \chardef\itfam=1 \chardef\bifam=5
  \loadmathfam \bffam [tenbf/]%        Actual Bold font
  \loadmathfam \bifam [tenbi/]%        Actual Bold Italic
  \newmathfam\symbfam
  \loadmathfam \symbfam [/psyr]%       PostScript Symbol
  \newmathfam\symbofam
  \loadmathfam \symbofam [/psyro]%     PostScript Symbol Oblique
}
\def\loadPSboldmath{%
  \loadmathfam 0[tenbf/]%              Actual Bold font
  \loadmathfam 1[tenbi/]%              Actual Bold-Italic font
...
}
\def\loadCMnormalmath{%
  \loadmathfam 0[/cmr8z]%                    Roman font
  \loadmathfam 1[/cmmi]%                     Math Italic font
  \loadmathfam 2[/cmsy]%                     Standard symbols from CM
  \noindexsize\loadmathfam 3[/cmex10]%       Standard extra symbols from CM
...
}
\def\loadCMboldmath{%
  \loadmathfam 0[/cmbx8z]%                     Roman font
  \loadmathfam 1[/cmmib10]%                    Math Italic font
...
}
\def\setPSmathchars{%
  \fontmessage{\ofsmessageheader Math codes are set for PS encoding}%
  \mathcode`\,="602C
  \mathcode`\.="002E
  \delcode`\<="\hex\symbfam E130A
  \delcode`\>="\hex\symbfam F130B
  \delcode`\|="\hex\symbfam 7C30C
  \edef\langle{\delimiter"4\hex\symbfam E130A }%
  \edef\rangle{\delimiter"5\hex\symbfam F130B }%
  \mathchardef\alpha    "0\hex\symbofam 61
  \mathchardef\beta     "0\hex\symbofam 62
  \mathchardef\gamma    "0\hex\symbofam 67
...
}
```

If you need to add next math families (math alphabets in NFSS terminology) then you can use the cod similar as the following:

```
\def\mathfonts{\defaultmathfonts
  \newmathfam\bbfam
  \loadmathfam \bbfam [/bbold12]%  Dvojitá vertikální kresba
  \def\bb{\fam\bbfam}%
}
\def\mathchars{\defaultmathchars
  \mathchardef\balpha    "0\hex\bbfam 0B
  \mathchardef\bbeta     "0\hex\bbfam 0C
  ...
}
```

We re-write the example from section 3 in order to support the math typesetting:

```
\input ofs [ffonts]                  % font Charter is in free fonts
\setfonts [Charter/10pt]             % default family
\setmath[//]                         % math initialisation
\fontdef\tt [CMTypewriter/mag1.1]    % monospaced font
\fontdef\verbtt [CMTypewriter-rm/8]  % monospaced font in verbatim
\fontdef\headfont [!/9]              % running heads
\addcmd \headfont {\baselineskip11pt\rm\def\mathversion{normal}\setmath[//]}
\fontdef\sectionfont [!/12]          % section font
\addcmd \sectionfont {\bf \let\it=\bi \def\mathversion{bold}\setmath[//]}
\fontdef\titlefont [!-bf/14.4]       % for the title of the document
```

The math fonts works in all sizes in our virtual example. The math formulae are in bold variant and in right size in section headers.

Reference

1. `www.stormtype.com`
2. `www.pismolijna.cz`,
3. `www.cstug.cz/stormtype`.
4. `www.cstug.cz/stormtype/slido.html`.
5. `ftp://math.feld.cvut.cz/pub/olsak/ofs`.
6. Petr Olšák. *OFS: Olšákův fontový systém*. 2001. The English documentation is in files `ofsdoc-e.tex`, `ofsdoc-e.pdf`
7. Petr Olšák. *Jak TEX k fontům ze Střešovic přišel*. Bulletin of CSTUG, number 4/2001, pp 153–180.
8. Karl Berry. *Fontname*, March 1999. The documentation is included in `web2c` TEX, file `fontname.texi`, `fontname.pdf`