Reprint *(***with corrections***):*
The New Font Family Selection — User Interface to Standard LaTeX

Frank Mittelbach
Rainer Schöpf

Contents

Abstract

In this article we describe the use of the new font selection scheme in the standard LaTeX environment. The main characteristics are:

- The possibility to change family, series, shape and sizes independently of one another.
- The existence of a style file to process older documents without any changes to their layout and their input files.
- A macro setup which is consistent with existing standard document styles.[1]

It is planned to incorporate this font selection scheme into LaTeX version 2.10.

[1] However, small changes in the document styles would make font changes a bit faster.

1 General remarks

In *TUGboat*10#2 we presented a new scheme to select fonts in TEX macro packages. This article describes the use of this new scheme in the LaTeX environment. The technical parts of the interface (which are of some interest to readers who plan to use our scheme with other fonts or with other macro packages) will be published in a separate article.

The necessary macros are distributed by the AMS together with the `amstex.sty` option which was announced in *TUGboat*10#3. The availability of the new font selection scheme at the usual servers will be announced separately in TEXhax, etc. Please refrain from asking for personal distribution.

To get a better understanding of this LaTeX interface, some words on the organisation of font families are in order. Readers of our article about the basic macros will notice that our understanding of these matters increased while working on this interface and the `amstex.sty` project; in some regards we have changed our point of view rather drastically. Surprisingly, only a few internal details within the basic macros needed adjustment; it seems that even without the real understanding, we instinctively got most of the things right when we designed them. (But probably we are still ignorant of the underlying concepts.)

In his book about "Methods of Book Design" Hugh Williamson writes [1]

> [...] To the printer, an alphabet is a set of twenty-six letters of a certain design and body, together with a few additional combinations of letters. A *fount* is usually made up of a set of alphabets of one size and based on one design. It may consist of one alphabet only, if no more alphabets exist in that design and size. Usually however a text fount will comprise five alphabets — roman and italic upper and lowercase, and small capitals. [...] A *series* is a set of founts closely related to each other in design, and usually very similar to each other, but graded in size. If only one alphabet has been made in a certain design, that alphabet alone may be a series. A *family* is a group of series compatible for composition, but loosely related in design. A family may include excerpts from more than one series.

Since TEX's physical fonts (which is the American word for fount) all contain exactly two alphabets, namely the upper and lower-case alphabets of a certain design, we will use the word font for physical TEX fonts, and fount for bundles of TEX fonts consisting for example of roman (upright or normal), italic and small capitals shapes.

The above quotation gives a good clue how to organize fonts in our font selection scheme. Hence we use the `\shape` command from the basic macros to distinguish between normal (n), italic (it), small caps (sc), sloped or slanted (sl) and upright italic (u) typefaces within one fount. Founts of different sizes form a series, so we use `\size` to access these. We think that the weight and the width of a series are good candidates to distinguish between individual series, therefore we combine them in the `\series` command. Again we use one and two letter abbreviations as shown in table 1. One or more of these series form a family which is accessed via the `\family` command.

To give some practical example, we arranged the most important families of the Computer Modern fonts according to this classification in table 2. Please note that some families like 'computer modern funny roman' (cmff) or 'computer modern sans serif quotation' (cmssq) are unclassified. These special purpose fonts are not accessible in the standard distribution of the new font selection scheme, although they could be added easily in a style file.

Given this overview about the classification of fonts it should be clear how to select a specific font with the primitive commands `\family`, `\series`, `\shape`, `\size`, and `\selectfont`. As described in [5], the `\size` macro takes two arguments: the size in printer's points as a numeral (i.e. without the dimension) and the corresponding `\baselineskip` value (with a dimension). `\selectfont` finally selects the font using values of the surrounding environment if some of the commands are missing. For example statements like "Concrete roman condensed slanted font at 9pt with 11pt leading"[2] will be translated into command sequences of the form:

```
\family{ccr}\series{c}\shape{sl}%
      \size{9}{11pt}\selectfont
```

This will explicitly load the font mentioned above, provided the necessary font shapes are known to the system.[3]

[2] This is pronounced "ledding" and measures the distance between the baselines of succeeding lines. To TEX users it is known as `\baselineskip`.

[3] Among the AMS distribution an example style option '`concrete.sty`' is provided which makes the Concrete roman as well as the Euler math fonts available. These fonts were used to typeset [4] and this article.

However, in the normal case, the LATEX user can safely rely on the standard LATEX font selection commands defined in terms of these primitive commands. These standard commands are discussed in the next sections.

2 Choosing a new text font

In standard LATEX, different fonts (of the same size) are selected by commands like `\rm`, `\bf`, `\it`, etc. These commands, however, select a specific font regardless of surrounding conditions. E.g., if you write `\sf\bf` you don't get the 'bold extended' *series* of the 'sans serif' *family* (i.e. cmssbx), instead you get the 'bold extended' *series* of the 'roman' *family* (i.e. cmbx). In our implementation this will be different.

Commands like `\bf` (or `\sf`) are now implemented to switch to the wanted *series* (or *family*, respectively), but to leave the other font characteristics untouched. However, this concept has one drawback in the current LATEX version: commands like `\footnote` might switch to a smaller size but will inherit other characteristics for the font to use from the environment where they are used. E.g., a footnote appearing in the scope of a theorem environment will erroneously be typeset in italic shape. This problem will vanish in LATEX version 2.10. Until then all font characteristics in such special circumstances must be reset by hand using the commands given below. However, this is not necessary if one uses the '`oldlfont`' style option described in sections 4 and 5. This option defines the font selection commands to behave in the same way as they do now in LATEX 2.09.

2.1 Choosing a new family

To switch to another *family* one may use `\rm`, `\sf` or `\tt` denoting the 'cmroman', 'cmsansserif' or 'cmtypewriter' *family*, respectively. The new font is selected without changing the current *series*, *shape*, and *size*. E.g., if the current font is cmbx10 (that is *family* 'computer modern roman', *series* 'bold extended', *shape* 'normal', and *size* '10pt') then `\sf` will change to cmssbx10 (that is *family* 'computer modern sansserif', *series* 'bold extended', *shape* 'normal', and *size* '10pt'). Using, e.g., `\tt` afterward will produce a warning and switch to cmtt10 because the 'computer modern typewriter' *family* does not contain a 'bold extended' *series*; therefore the default ('medium') *series* is tried.

Weight Class		Width Class		
Ultralight	ul	Ultracondensed	50%	uc
Extralight	el	Extracondensed	62.5%	ec
Light	l	Condensed	75%	c
Semilight	sl	Semicondensed	87.5%	sc
Medium (normal)	m	Medium	100%	m
Semibold	sb	Semiexpanded	112.5%	sx
Bold	b	Expanded	125%	x
Extrabold	eb	Extraexpanded	150%	ex
Ultrabold	ub	Ultraexpanded	200%	ux

Table 1: Weight and width classification for fonts. The percent values are derived from [2]. To combine the abbreviations in the `\series` command, weight is used first and any instance of medium (`m`) is dropped except when weight and width are both medium. In this case one single `m` is used. So bold expanded would be `bx` whereas medium expanded would be `x`.

Computer Modern families

family	**series**	**shape(s)**	**Example of external names**
Computer modern roman			
cmr	m	n, it, sl, sc, u	`cmr10, cmti10, cmsl10, cmcsc10, cmu10`
cmr	bx	n, it, sl	`cmbx10, cmbxti, cmbxsl`
cmr	b	n	`cmb10`
Computer modern sans serif			
cmss	m	n, sl	`cmss10, cmssi10`
cmss	bx	n	`cmssbx10`
cmss	sbc	n	`cmssdc10`
Computer modern typewriter			
cmtt	m	n, it, sl, sc	`cmtt10, cmitt10, cmsltt, cmtcsc10`
Computer modern fibonacci			
cmfi	m	n	`cmfib8`

Table 2: Classification of the Computer modern fonts. You will notice that not all possible combinations of family, series and shape are available. E.g. there is no small capitals shape in the medium series of the computer modern sans serif. However, Philip Taylor announced recently that he has filled some of the holes. It might be a good idea to include such additional parameter files for METAFONT into the general distributions.

2.2 Choosing a new series

To switch between 'medium' and 'bold extended' *series* the commands `\mediumseries` and `\bf` are provided.

2.3 Choosing a new shape

Analogously the commands `\sl`, `\it`, and `\sc` are used to switch to the *shapes* 'sloped', 'italic' and 'small-caps', this time leaving *family*, *series*, and *size* alone.

In addition, we introduce the `\normalshape` command, in case one wants to switch back to the 'normal' *shape*. If font changes are done only inside of groups this command is necessary only to reset a shape in a footnote or a similar context to avoid the problem mentioned above.

2.4 Choosing a new size

To change to another *size* the standard LaTeX commands

`\tiny`	`\scriptsize`	`\footnotesize`
`\small`	`\normalsize`	`\large`
`\Large`	`\LARGE`	`\huge` and `\Huge`

may be used. These commands also set the parameter `\baselineskip` and the `\strutbox` as well as the script and scriptscript *sizes* for the new text *size*. But once again they will not change other font characteristics. So, for example, it doesn't matter whether one writes `\large\tt` or `\tt\large`, the same font will be selected.

2.5 Doing it by hand

As mentioned before, primitive font selection commands like `\family`, `\series`, `\shape`, `\size`, and `\selectfont` are also available to carry out the change. E.g.,

```
\shape{n}\family{cmss}\selectfont
```

will switch to the 'cmsansserif' family with 'normal' shape.[4] This article was set in concrete roman type by simply writing `\family{ccr}\selectfont` following the `\begin{document}` command.[5] Of course, the `\documentstyle` command also specifies a style option ('`concrete`') which sets up the internal tables for these fonts.

2.6 Changing the meaning of `\rm`, `\sf` etc.

To make it easy to typeset documents with other font families (like Times Roman, Optima etc.) we maintain seven additional macros

```
\rmdefault \sfdefault \ttdefault \bfdefault
\itdefault \scdefault \sldefault
```

denoting the family chosen by `\rm`, `\sf`, `\tt` or the series chosen by `\bf` or the shape for `\it`, `\sc` and `\sl`, respectively.[6] If, for example, a document should be typeset in sans serif one could add in the preamble (between `\documentstyle` and `\begin{document}`) the following redefinitions:

```
\renewcommand{\rmdefault}{cmss}
\renewcommand{\itdefault}{sl}
```

The first line means that whenever `\rm` is called the family `cmss` (i.e. computer modern sans serif) is chosen and the second line redefines `\it` to switch to the slanted shape since this family hasn't got an *italic* shape. Another possible use is to say

```
\renewcommand{\bfdefault}{b}
```

This will redefine `\bf` to select the bold instead of the bold extended series which is the current default. However these commands are probably ignored by document styles for journals which decide to use their own font families in the final print.

3 Fonts for Math

The selection of a specific typeface in a math formula should not depend on the surrounding environment. Characters in math normally denote special things which should stay fixed even if the surrounding text is set in another *shape* or *series*. Therefore the strategy for selecting math fonts is somewhat different.

3.1 Simple formulas

Normal letters and standard symbols are typeset simply by using the letters directly or using a command that denotes the wanted symbol. So `$\sum A_{i}$` will produce $\sum A_i$. The typeface chosen will depend on the current ⟨*math version*⟩. You can switch between ⟨*math versions*⟩ outside of math mode,[7] thereby changing the overall layout of the following formulas.

LaTeX knows about two versions called 'normal' and 'bold'. As the name indicates, `\mathversion{normal}` is the default. In contrast, the bold version will produce bolder letters and symbols. This might be suitable in certain situations like

[4] For full details of the usage of these primitive commands see the article about the basic macros [5].

[5] Actually we also said `\size{10}{13pt}` to establish a larger `\baselineskip`.

[6] This was suggested by Sebastian Rahtz who was one of the first users of our prototype version.

[7] This is done with the command `\mathversion{`⟨*version name*⟩`}`.

headings, but recall that changing the version means changing the appearance (and perhaps the meaning) of the whole formula. If you want to bolden only some symbols or characters within one formula you should not change the `\mathversion`. Instead you should define a special math alphabet for characters (see below) and/or use the command `\boldsymbol` which is provided by the document style option `amsbsy`. For historical reasons LATEX maintains two abbrivations to switch to its math versions: `\boldmath` and `\unboldmath`.

Other versions could be provided in special style options. For example the '`concrete`' option mentioned before sets up a version called 'euler' to typeset formulas in the same way as it was done in [4].

3.2 Special math alphabets

But simple formulas with one alphabet and a huge number of symbols are not sufficient for mathematicians to expose their thoughts properly. They tend to use every available typeface to denote special things.

To cope with this need for special alphabets in formulas,' we introduce the concept of ⟨*math alphabet identifiers*⟩. These constructs are special commands which switch to a specific typeface. They might correspond to different typefaces in different math versions but within one version they always select the same typeface regardless of surrounding conditions.

A ⟨*math alphabet identifier*⟩ can be defined according to the users' needs but standard LATEX already has a few of them built in. They are described in table 3.

When using such an ⟨*alphabet identifier*⟩ two syntax variants are available: one can understand a command like `\cal` as a switch to a different font, i.e. using a syntax `{\cal ...}` as the old LATEX does, but we prefer to view the ⟨*math alphabet identifier*⟩ as a command with one argument, i.e. to use a syntax of the form `..\cal{A}..` To select the first alternative a style option '`nomargid`' is provided. This option is automatically selected if the '`oldlfont`' option is used since this option is supposed to produce identical results for older documents.

New ⟨*math alphabet identifiers*⟩ are defined in two steps. First the identifier is made known to the system with the `\newmathalphabet` command. Then specific typefaces in some or all ⟨*math versions*⟩ are assigned by means of the `\addtoversion` command.

Let us discuss this process in detail. Suppose that you want to make a sans serif typeface available as a math alphabet. First we choose a new command name (e.g. `\sfmath`) and tell LATEX about it with the line

```
\newmathalphabet{\sfmath}
```

Then we consult table 2 to find suitable fonts to assign to this alphabet identifier. As you find out, the computer modern sans serif family consists of three series, a medium, semi bold condensed and a bold extended one. The medium and the bold extended series both contain a normal shape typeface. So we add the line:

```
\addtoversion{normal}{\sfmath}{cmss}{m}{n}
\addtoversion{bold}{\sfmath}{cmss}{bx}{n}
```

Now our alphabet identifier is ready for use in these two versions. We demonstrate this with the formula

$$\sum \mathsf{A}_i = \tan \alpha$$

which was produced by

```
\mathversion{normal}
\[  \sum \sfmath{A}_{i} = \tan \alpha \]
```

Note that we first switched back to the normal version. This was necessary since this article is typeset with a third version (Euler) in force. If we had tried to use `\sfmath` in this version we would have gotten an error message stating that this ⟨*math alphabet identifier*⟩ isn't defined for the Euler version.[8]

If we are interested in a slanted shape we have to face a problem: there is no slanted shape in the bold extended series of the Computer Modern sans serif family. So, if we make the identifier known only in the normal version then it would produce an error message when encountered in the bold (or any other) version. Of course we can get by using always the same typeface in all versions. To make this task a bit easier there is also a * variant of the `\newmathalphabet` command which takes three more arguments: the default values for family, series and shape for all math versions in which the alphabet identifier is not explicitly defined via an `\addtoversion` command. So our second example can be set up simply by stating

```
\newmathalphabet*{\sfslmath}{cmss}{m}{sl}
```

This would have the additional advantage that this math alphabet identifier is also allowed in math versions which are defined in style files or document styles

[8] Actually we cheated a bit more in this article: we had to reset the `\mathcode` of certain characters because they are in different places in the Euler version. A few more details can be found in Don Knuth's article [3]. However, this is not a real problem because such changes can be done in commands similar to `\boldmath` if such incompatible versions are to coexist in real applications.

LaTeX knows about three ⟨*math alphabet identifier*⟩s. `\cal` will select calligraphic letters like $\mathcal{ABCD}$, `\mathrm` will select upright roman letters for use in functions like $\max_i$, and finally `\mit` selects the default math italic alphabet.

Table 3: Predefined ⟨*math alphabet identifiers*⟩ in LaTeX

(like the Euler version mentioned earlier). Any explicit `\addtoversion` command overwrites the defaults given by `\newmathalphabet*`; so, it might be a good idea always to specify default values.

Here we show the same formula as above, but this time in the Euler version and with `\sfslmath` instead of `\sfmath`:

$$\sum A_i = \tan\alpha$$

4 Processing older documents

To typeset documents which are written with the old LaTeX (i.e. with a format using the old font selection scheme) only the source line containing the `\documentstyle` command has to be changed. More exactly the 'oldlfont' option must be added to the list of document style options if the new font selection scheme is in force.[9]

5 Setting up a new format

This section is written for people called 'local wizards' by the LaTeX manual, which simply refers to the (poor) guys who are always being pestered if things do not work.[10] If you are using LaTeX on your own PC you might have to read this section, too, even if you don't feel like being a wizard.

5.1 Preparations

Before generating a new format it is necessary to rename a few files. This enables you to customize the format to the special needs of your site.

`lfonts.tex` First of all you should rename the file `lfonts.tex` (supplied with the standard distribution of LaTeX); otherwise you will always end up with an old format. Call it, say, `lfonts.ori`.

`hyphen.tex` Another file which should probably be renamed is `hyphen.tex` (the original American `\patterns` from Don Knuth) because this enables you to insert your favourite `\pattern` package when IniTeX is asking for this file. This might even be useful if you use TeX version 3.0 which is multilingual (assuming that your computer has only a limited memory).

5.1.1 Preloading Fonts

Now you have to decide which fonts to preload in your format. Unlike the old font selection scheme of LaTeX, where only preloaded fonts could be used in math applications (like subscripts etc.), the new font selection scheme poses no restriction at all; documents will always come out the same. So you have to take your pick by weighing the two conflicting principles:

- Preloading often used fonts might make your TeX run a bit faster.
- Using more load-on-demand fonts will make your format much more flexible, because you can switch to different families far more easily. After all, there is an upper limit to the number of fonts TeX can use in one run and every preloaded font will count even if it is never accessed.

On the PC at home we nowadays always use formats with only 5 fonts preloaded.[11] We don't think that TeX is actually running much more slowly than before.

Together with the new font selection scheme two files `preload.min` and `preload.ori` are distributed. The first one will preload next to nothing while the second will preload the same fonts as the old `lfonts.tex`. You can copy either of these files to `preload.tex` and then change it if you want to preload some other fonts. But please make sure that you don't change one of the original files of the distribution.

5.1.2 Making more fonts available

Besides deciding which fonts to preload, you also have to tell the TeX sytem which external fonts are available and how they are organized in families, series, shapes and sizes. In short you have to set up internal tables giving informations like "family `cmr`, series `b`, shape `n`, size `10` is associated with the external font `cmb10` but there is no font with similar characteristics in size 9". This is done with the `\new@fontshape` command, either in a style file (see '`concrete.sty`' as an example)

[9] This means that it is the default (see next section).

[10] YOU might belong to this group!

[11] This is the absolute minimum. These fonts are accessed by `lplain.tex` and `latex.tex` when the format is generated.

or when dumping a format.

Again two files `fontdef.ori` and `fontdef.max` are distributed. You can copy one of them to `fontdef.tex`. The file `fontdef.ori` defines all fonts which are necessary to run standard LaTeX documents while `fontdef.max` also defines certain fonts from the AMSFonts collection. To make other font families available you can either append appropriate `\new@fontshape` definitions to `fontdef.tex` (again, leave the originals untouched!) or add them in a style file.[12] For a detailed description of how to set up new families with the `\new@fontshape` command, see [5] about the basic macros or one of the example files.

5.2 Running IniTEX

When setting up a new format one has to start IniTEX with `lplain.tex` as the input file. After displaying some progress report on the terminal, `lplain.tex` will try to `\input` the files `hyphen.tex` and `lfonts.tex`.

As we said above, it seems a good idea to rename these files because, when TEX complains that it cannot find them and asks you to type in another file name, you get the chance to substitute your favourite hyphenation patterns without changing `lplain.tex` or copying something to `hyphen.tex`. The transcript file will show the name of the file used which is very useful to debug weird errors (later).

When the point is reached where TEX wants to read in `lfonts.tex`, you now have to specify '`lfonts.new`'. This file will `\input` some other files. After processing them (which will take some time), IniTEX stops once more since it cannot find the file `xxxlfont.sty`. This is intentional; in this way you may now specify the desired default by entering one of the following file names:

`oldlfont.sty` If you choose this file, your format will be identical to the standard LaTeX version 2.09 except that a few additional commands (like `\normalshape`) are available. Of course, documents or style options which explicitly refer to things like `\tentt` will produce error messages since such internal commands are no longer defined.[13] Nevertheless it is easy to fix the problem in such a case: if we know that `\tentt` referred to `cmtt10`, i.e. Computer modern typewriter normal at 10pt, we can define it as

```
\newcommand{\tentt}{\family{cmtt}
    \series{m}\shape{n}\size{10}{12pt}
    \selectfont}
```

Since we assume the '`oldlfont`' option as default, where `\tt` resets series and shape, the definition could be shortened to

```
\newcommand{\tentt}{\size{10}{12pt}\tt}
```

To get the new way of font selection as described in the previous sections (e.g. where `\tt` simply means to switch to another family) you only have to add the '`newlfont`' style option to the `\documentstyle` command in your document.

`newlfont.sty` This is just the counterpart to `oldlfont.sty`: it will make the new mechanism the default and you have to add '`oldlfont`' as a style option if you want to process older documents which depend on the old mechanism.

`basefnt.tex` This file is similar to `newlfont.sty` but does not define the LaTeX symbol fonts. These fonts contain only a few characters which are also included in the AMS symbol fonts. Therefore we provided the possibility of generating a format which doesn't unnecessarily occupy one of the (only) sixteen math groups within one math version. Using this file you can easily switch to the old scheme (adding '`oldlfont`' as an option), to the new scheme with LaTeX symbol fonts (using '`newlfont`') or to the new scheme with additional AMS fonts by using either the style option '`amsfonts`' (fonts only) or the style option '`amstex`' (defining the whole set of AMS-TEX macros in a LaTeX like syntax).

We suggest using the `basefnt.tex` file since the new font selection scheme will be incorporated into LaTeX version 2.10, but on installations with many users it might be better to switch smoothly to the new font selection scheme by first using '`oldlfont`' as a default.

Anyway, after reading the file chosen, TEX will continue by processing `latex.tex` and finally displaying the message "Input any local modifications here". If you don't dare to do so, use `\dump` to finish the run. This will leave you with a new `.fmt` file (to be put into TEX's format area) and the corresponding transcript file. It isn't a very good idea to delete this one

[12] The latter alternative might be better if you use these fonts very rarely (e.g., at sites with many users) to avoid filling TEX's memory with unnecessary definitions.

[13] By the way, such documents were at no time portable since Leslie Lamport stated that it was always permissible to customize `lfonts.tex` according to the local needs. Therefore this is *not* an incompatible change.

because you might need it later to find out what you did when you dumped the format!

6 Remarks on the development of this interface

We started designing the new font selection scheme around April 1989. A first implementation was available after one month's work and thereafter the prototype version ran successfully for some months at a few sites in Germany and the UK. Frank's visit to Stanford as well as our work on the 'amstex' style option brought new aspects to our view. The result was a more or less complete redefinition of the LaTeX interface for this font mechanism. It was a long way from the first sketch (which was about five pages in Frank's notebook) to the current implementation of the interface which now consists of nearly 2000 lines of code and about 4000 lines of internal documentation. The AMS-TeX project itself, which triggered this reimplemenation, has about the same dimensions. Surely in such a huge software package one will find typos and bugs. But we hope that most of the bugs in the code are found by now. It is planned that the new font selection scheme will replace the old one in LaTeX version 2.10. We therefore hope that this pre-release which runs in version 2.09 will help to find all remaining problems so that the switch to version 2.10 will be without discomfort to the user.

7 Acknowledgements

During this project we got help from many people. A big 'thank you' to all of them, especially to Michael Downes from the AMS for his cooperation and help, to Stefan Lindner for his help with the Atari TeX and to Sebastian Rahtz for playing a willing guinea-pig. Finally we also want to thank Ron Whitney who did a marvelous job on all our articles so far. This time we posed some extra problems because he had to first make a new format in order to read how to make a new format.

8 List of distributed files

`lfonts.new` The new version of `lfonts.tex`, to be copied to a file of this name after the old `lfonts.tex` has been renamed.

`fontdef.ori` The font definitions for the computer modern fonts in the distribution by Donald E. Knuth. To be copied to `fontdef.tex` if this selection is to be used.

`fontdef.max` Complete font definitions for the computer modern fonts and the AMSFonts collections. To be copied to `fontdef.tex` if this selection is to be used.

`preload.ori` Preloads the same fonts as the old `lfonts.tex` does. To be copied to `preload.tex` if this is desired.

`preload.min` Preloads only the absolute minimum of fonts. To be copied to `preload.tex` if this is desired.

`fam.tex` The basic macros for the new font selection scheme. Automatically read by `lfonts.new`.

`latint.tex` LaTeX interface for the new font selection scheme. Automatically read by `lfonts.new`.

`setsize.tex` Size definitions for the LaTeX interface. Automatically read by `lfonts.new`.

`newlfont.sty` Selects new version of font selection for LaTeX.

`oldlfont.sty` Selects old version of font selection for LaTeX.

`basefnt.tex` Like `newlfont.sty`, but does not define the LaTeX symbol fonts.

`margid.sty` Style file that defines all ⟨*math alphabet identifiers*⟩ to have one argument. This is the default that is built in into the new font selection scheme. Therefore this style file is only necessary if the installation decided to load '`nomargid.sty`' at dump time.

`nomargid.sty` In contrast to `margid.sty`, defines all ⟨*math alphabet identifiers*⟩ to switch to the alphabet. This style option is necessary if you want to be compatible to the old LaTeX syntax *in math mode only*.

`tracefnt.sty` Style file that allows the tracing of font usage. Use `\tracingfonts` with values 1 to 3 and watch what happens.

`syntonly.sty` Defines the `\syntaxonly` declaration. This can be used in the preamble of a document to suppress all output.

`amsfonts.sty` Defines the commands to select symbols from the AMSFonts collection.

`amsbsy.sty` Defines the `\boldsymbol` command.

`amssymb.sty` Defines additional AMS-TeX symbols.

`amstex.sty` Defines special AMS-TeX structures (like alignments in math mode) with LaTeX syntax.

`amstext.sty` Defines the AMS-TeX `\text` command.

`euscript.sty` Contains the definitions to use the Euler script fonts.

References

[1] Hugh Williamson, *Methods of Book Design*, Yale University Press, New Haven, London, Third Edition, 1985.

[2] International Business Machines Corporation, Font Object Content Architecture Reference, First Edition, December 1988.

[3] Donald E. Knuth, "Typesetting *Concrete Mathematics*," *TUGboat* 10#1, 1989, pp. 31–36.

[4] Ronald L. Graham. Donald E. Knuth, and Oren Patashnik, *Concrete Mathematics*. Addison-Wesley, 1989.

[5] Frank Mittelbach and Rainer Schöpf, "A New Font Selection Scheme for TEX Macro Packages," *TUGboat* 10#2, 1989, pp. 222–238.

◇ Frank Mittelbach
Electronic Data Systems (Deutschland) GmbH
Eisenstraße 56 (N15)
D-6090 Rüsselsheim
Federal Republic of Germany

◇ Rainer Schöpf
Institut für Theoretische Physik der Universität Heidelberg
Philosophenweg 16
D-6900 Heidelberg
Federal Republic of Germany