

The **mathspec** package

Font selection for mathematics with X \LaTeX

version 0.2

Andrew Gilbert Moschou

andmos@gmail.com

WEDNESDAY, 30 SEPTEMBER 2009

TABLE OF CONTENTS

I	PREAMBLE	I	4.5 Shorthands	6
2	INTRODUCTION	2	4.6 A further example	6
3	IMPLEMENTATION	2	5 GREEK SYMBOLS	7
4	SETTING FONTS	3	6 GLYPH BOUNDS	9
4.1	Letters and Digits	3	7 COMPATABILITY	11
4.2	Symbols	4	8 THE PACKAGE	12
4.3	Examples	4	9 LICENSE	32
4.4	Declaring alphabets	5		

I PREAMBLE

This document describes the `mathspec` package, a package that provides an interface to select ordinary text fonts for typesetting mathematics with X \LaTeX . It relies on `fontspec` to work and familiarity with `fontspec` is advised. I thank Will Robertson for his useful advice and suggestions!

The package is developmental and later versions might to be incompatible with this version. This version is incompatible with earlier versions. The package requires at least version 0.9995 of X \TeX .

Should you be using this package? If you are using another \LaTeX package for some mathematics font, then you should not (unless you know what you are doing). If you

want to use Asana Math or Cambria Math (or the final release version of the STIX fonts) then you should be using `unicode-math`.

Some paragraphs in this document are marked **ADVANCED**. Such paragraphs may be safely ignored by basic users.

2 INTRODUCTION

Since Jonathan Kew released X_YTEX, an extension to T_EX that permits the inclusion of system wide Unicode fonts and modern font technologies in T_EX documents, users have been able to easily typeset documents using readily available fonts such as Hoefler Text and Times New Roman (This document is typeset using Sabon LT Std). Will Robertson's X_YL^AT_EX package `fontspec` provides an automatic font selection process for such fonts. Still, mathematics typesetting has not benefited from this development to the same extent as text typesetting, which is not surprising given the font requirements that are demanded.

Will Robertson has in development a package `unicode-math` that is used to typeset mathematics using some font that has an OpenType mathematics table. Currently, the only two that are available are Cambria Math, a new design, by Microsoft and Asana Math, ultimately based on Hermann Zapf's Palatino, by Apostolos Syropoulos. It is expected that the STIX fonts, based on Times, may also be used.

Thus, it is evident that a book designer is confronted with such narrow variety in mathematics typefaces, even considering already established L^AT_EX fonts such as AMS Euler (and Computer Modern, of course!) that are dedicated to providing mathematics typefaces; and the book designer is challenged to find a suitable pair of matching typefaces for mathematics and body text.

This package provides mathematics alphabets in any of the same typefaces that are available to X_YTEX, using `fontspec` as a back end. Other mathematics symbols such as arrows and operators, whose designs are largely independent of an alphabetic typeface, can be taken from collections like MnSymbol or Computer Modern and are not covered by the scope of this package.

3 IMPLEMENTATION

`\usepackage[mathspec and fontspec options]{mathspec}`

To use the package, put `\usepackage{mathspec}` in the preamble of the document. It is not necessary to put `\usepackage{fontspec}` because `mathspec` will ensure that `fontspec` is loaded anyway. To load `fontspec` with some options, parse them through `mathspec`, for example, `\usepackage[quiet]{mathspec}` is equivalent to `\usepackage[quiet]{fontspec}\usepackage{mathspec}`.

Actually, `mathspec` ordinarily loads `fontspec` with the `no-math` option. To cancel this, explicitly use the `math` option: e.g. `\usepackage[math]{mathspec}`.

4 SETTING FONTS

4.1 Letters and Digits

```
\setmathsfont(<sets>)[<shapes, font features>]{<font name>}
\setmathfont(<sets>)[<shapes, font features>]{<font name>}
```

This single command is used to entirely describe the desired font for some use. For each character set (Digits, Latin, Greek), there is a (possibly shared) `\setmathsfont` command. The command can be used only in the preamble. Then, there can be only one typeface for each character set in a single document.

For basic use, `(<sets>)` is mandatory. `[<shapes, font features>]` is optional and may be omitted.

<sets> A comma separated value list of any of the following: **Digits, Latin, Greek**.

ADVANCED *<sets>* may also take the value `Special`. *(<sets>)* is actually optional and if it is omitted, then `(Special)` is assumed. For basic use, it is safe to ignore `Special`, so *(<sets>)* would effectively be a mandatory argument.

If the value of <sets> is Special, then `\eu@ScopeSet@Special[]{}` is executed, but this command is provided by `mathspec` to gobble its arguments (that is, it does nothing). The command `\eu@ScopeSet@Special` maybe predefined before `mathspec` is loaded (or redefined after `mathspec` is loaded) to cause the command to do something else. This effectively means that `\setmathsfont(<sets>)[]{}` is a `mathspec` command, but `\setmathsfont[]{}` is an external command which another package might use.

<shapes> A comma separated value list of keys and their values. The permitted keys and their values are:

Uppercase=Regular, Italic, Plain

Lowercase=Regular, Italic, Plain

Arabic=Regular, Italic, Plain

To set the symbols in an upright font, choose **Regular**. To set the symbols in an italic font, choose **Italic**. Choose **Plain** to indicate that no font is assignment is to be performed by `mathspec`, so that the symbols will remain with their default font (usually Computer Modern). If a value's key is omitted, its default value, which depends on the *<set>*, as shown in table 1 is chosen.

The default values are very good and you would be wise not to change the shapes for the **Latin** and **Digit** sets. The Greek shapes may be changed, if needed, to adhere to some particular style, for example all **Italic** or all **Regular**.

TABLE I The default shapes for the symbol sets.

Set	Key	Default value
Digits	Arabic	Regular
Latin	Uppercase	Italic
Latin	Lowercase	Italic
Greek	Uppercase	Regular
Greek	Lowercase	Italic

⟨font features⟩ *⟨font features⟩* and *⟨font name⟩* follow directly from `fontspec` to select the font and its features. See `fontspec`'s documentation for details.

Note that if *⟨sets⟩* contains multiple sets, the command is iterated over each set. Each of these iterations share the same *⟨shapes and font features⟩* and *⟨font name⟩*. If you require that the different sets have individual options and font names, they must be specified in separate commands. However, if a value's key is omitted, the individual default values are still applied.

These same command options follow through to the other commands of this package.

4.2 Symbols

There is currently no way to set the font for general mathematical symbols such as:

$$=, \times, \mapsto, \partial, \emptyset, \in, \int, \subset$$

You can try the package `MnSymbol`, which has greater (and more uniform) coverage, with the package option `MnSymbol`, e.g. `\usepackage[MnSymbol]{mathspec}`. Note that `MnSymbol` is a third party package by Achim Blumensath, which can only be used if additionally installed.

4.3 Examples

```
\setmathsfont(Digits, Latin, Greek)
  [Numbers={Lining, Proportional}]{Minion Pro}
```

is equivalent to:

```
\setmathsfont(Digits)[Numbers={Lining, Proportional}]{Minion Pro}
\setmathsfont(Latin)[Numbers={Lining, Proportional}]{Minion Pro}
\setmathsfont(Greek)[Numbers={Lining, Proportional}]{Minion Pro}
```

and indicates that all digits, Latin and Greek characters are to be set in Minion Pro with lining, proportional digits, with digits and uppercase Greek in regular and Latin and lowercase Greek in italic, which are the default shapes.

```
\setmathsfont(Digits,Greek)
  [Uppercase=Plain,Lowercase=Regular,Scale=MatchLowercase]
  {GFS Porson}
```

is equivalent to:

```
\setmathsfont(Digits)
  [Uppercase=Plain,Lowercase=Regular,Scale=MatchLowercase]{GFS Porson}
\setmathsfont(Greek)
  [Uppercase=Plain,Lowercase=Regular,Scale=MatchLowercase]{GFS Porson}
```

and indicates that all digits and lowercase Greek characters are to be set in GFS Porson, scaled so that its x-height matches the main font's, in regular. Uppercase Greek characters remain unchanged (probably from Computer Modern).

4.4 *Declaring alphabets*

The mathematics alphabets like `\mathrm` and `\mathcal` can be set using these commands.

```
\setmathrm[<font features>]{<font name>}
```

This command defines `\mathrm`, `\mathit`, `\mathbf` and the font for operators like `\sin` and `\log`.

```
\setmathsf[<font features>]{<font name>}
```

This command defines `\mathsf`.

```
\setmathtt[<font features>]{<font name>}
```

This command defines `\mathtt`.

```
\setmathcal[<font features>]{<font name>}
```

This command defines `\mathcal`.

```
\setmathbb[<font features>]{<font name>}
```

This command defines `\mathbb`.

`\setmathfrak[⟨font features⟩]{⟨font name⟩}`

This command defines `\mathfrak`.

4.5 *Shorthands*

These commands are useful to save typing the same information multiple times, if the same font is used for different purposes.

`\setallmainfonts(⟨sets⟩)[⟨shapes, font features⟩]{⟨font name⟩}`

This command is equivalent to:

```
\setmainfont[⟨font features⟩]{⟨font name⟩}
\setmathsf(⟨sets⟩)[⟨shapes, font features⟩]{⟨font name⟩}
\setmathrm[⟨font features⟩]{⟨font name⟩}
```

If `(⟨sets⟩)` is omitted, then `(Digits, Latin, Greek)` is assumed.

`\setprimaryfont[⟨shapes, font features⟩]{⟨font name⟩}`

This command is equivalent to:

```
\setallmainfonts(Digits, Latin)[⟨shapes, font features⟩]{⟨font name⟩}
```

`\setallsansfonts[⟨font features⟩]{⟨font name⟩}`

This command is equivalent to:

```
\setsansfont[⟨font features⟩]{⟨font name⟩}
\setmathsf[⟨font features⟩]{⟨font name⟩}
```

`\setallmonofonts[⟨font features⟩]{⟨font name⟩}`

This command is equivalent to:

```
\setmonofont[⟨font features⟩]{⟨font name⟩}
\setmathtt[⟨font features⟩]{⟨font name⟩}
```

4.6 *A further example*

This document is typeset with the following:

```
\setmainfont[Numbers=OldStyle]{Sabon LT Std}
\setallsansfonts[Numbers={OldStyle, Proportional}, Scale=MatchLowercase]{Candara}
\setallmonofonts[Numbers=OldStyle, Scale=MatchLowercase]{Consolas}
\setmathsf(Digits, Latin)[Scale=MatchLowercase]{Bembo MT}
\setmathsf(Greek)[Scale=MatchLowercase]{STIXGeneral}
\setmathrm{Sabon LT Std}
```

```
\exchangeforms{phi}
\setminwhitespace[750]
```

The main text font is Sabon LT Std with old style figures. The sans serif font is Candara with old style, proportional figures and the monospaced font is Consolas with old style figures, both scaled to match Sabon LT Std in x-height.

The mathematics font for digits and Latin symbols is Bembo MT and for Greek symbols STIXGeneral, both scaled to match Sabon LT Std in x-height.

The `\mathrm`, `\mathit` and `\mathbf` alphabets are set in Sabon LT Std.

Finally, the normal and variant forms of Greek lowercase phi are exchanged, see section 5 Greek symbols, and the minimum white space on each side of a spaced character is 750 mu, see section 6 Glyph bounds.

5 GREEK SYMBOLS

For reference, the Greek alphabet and variant letter forms are given in table 2. Despite its name, `\varsigma` ς is not a variant form of `\sigma` σ , it is the final form. Digamma \digamma is an obsolete letter of the alphabet, originally placed between epsilon ϵ and zeta ζ .

ADVANCED *Initially, `mathspec` defines control sequences for the Greek characters that are absent in Computer Modern, because they are similar in form to Latin characters, from ASCII, thus control sequences like `\Alpha` and `\omicron` are defined. Any variant forms that are absent become equivalent to the normal forms. The meanings of these control sequences are preserved by other control sequences of the same names prefixed with `eu@cm@` (e.g. `\eu@cm@alpha`), which may be recalled after the Greek mathematics font is changed.*

`\exchangeforms{<list>}`

<list> A comma separated value list of any of the names for the Greek symbols which have variant forms: beta, epsilon, theta, kappa, pi, rho, phi, Theta.

Some authors might prefer the normal and variant forms of a symbol to be exchanged. For example, I prefer `\phi` to print the orthotic phi ‘ ϕ ’ and `\varphi` the cursive phi ‘ φ ’, contrary to many text fonts. To exchange the forms of any symbol, include its name in the list.

`\normalisevarforms[<list>]`

`\normalizevarforms[<list>]`

<list> As above, a comma separated value list of any of the names for the Greek symbols which have variant forms: beta, epsilon, theta, kappa, pi, rho, phi, Theta.

If [*<list>*] is omitted, then [beta,epsilon,theta,kappa,pi,rho,phi,Theta], that is every such symbol, is assumed.

TABLE 2 The Greek alphabet and variant letter forms with control sequences.

Uppercase		Lowercase		Uppercase		Lowercase	
A	<code>\Alpha</code>	α	<code>\alpha</code>	Ξ	<code>\Xi</code>	ξ	<code>\xi</code>
B	<code>\Beta</code>	β	<code>\beta</code>	Ο	<code>\Omicron</code>	$ο$	<code>\omicron</code>
Γ	<code>\Gamma</code>	γ	<code>\gamma</code>	Π	<code>\Pi</code>	π	<code>\pi</code>
Δ	<code>\Delta</code>	δ	<code>\delta</code>	Ρ	<code>\Rho</code>	ρ	<code>\rho</code>
Ε	<code>\Epsilon</code>	ϵ	<code>\epsilon</code>	Σ	<code>\Sigma</code>	ς	<code>\varsigma</code>
Z	<code>\Zeta</code>	ζ	<code>\zeta</code>			σ	<code>\sigma</code>
Η	<code>\Eta</code>	η	<code>\eta</code>	Τ	<code>\Tau</code>	τ	<code>\tau</code>
Θ	<code>\Theta</code>	θ	<code>\theta</code>	Υ	<code>\Upsilon</code>	υ	<code>\upsilon</code>
Ι	<code>\Iota</code>	ι	<code>\iota</code>	Φ	<code>\Phi</code>	ϕ	<code>\phi</code>
Κ	<code>\Kappa</code>	κ	<code>\kappa</code>	Χ	<code>\Chi</code>	χ	<code>\chi</code>
Λ	<code>\Lambda</code>	λ	<code>\lambda</code>	Ψ	<code>\Psi</code>	ψ	<code>\psi</code>
Μ	<code>\Mu</code>	μ	<code>\mu</code>	Ω	<code>\Omega</code>	ω	<code>\omega</code>
Ν	<code>\Nu</code>	ν	<code>\nu</code>	Ϝ	<code>\Digamma</code>	φ	<code>\digamma</code>
Normal form	Variant form	Normal form	Variant form	Normal form	Variant form	Normal form	Variant form
β	β	<code>\varbeta</code>		π	ϖ	<code>\varpi</code>	
ϵ	ϵ	<code>\varepsilon</code>		ρ	ϱ	<code>\varrho</code>	
θ	θ	<code>\vartheta</code>		ϕ	φ	<code>\varphi</code>	
κ	κ	<code>\varkappa</code>		Θ	Θ	<code>\varTheta</code>	

Since not all fonts contain all variant forms, there might be the odd variant letter that remains in Computer Modern while other Greek letters have changed font. For these characters, the command `\normalisevarforms` (or `\normalizevarforms`) will cause the listed symbols that are absent from the font to be equivalent to their corresponding normal forms. If a symbol is listed which is included in the font, then it is ignored.

Note that this command only adjusts the characters that are absent in the font. Of course, this might mean that, for example, `\vartheta` and `\theta` would print the same glyph, which would be dubious if they are to be used in the same document, with different meanings.

It should be noted that `\epsilon` prints the lunate epsilon ‘ε’ and `\varpepsilon` prints the usual (double loop) epsilon ‘ε’ in Plain T_EX and L^AT_EX. This is in contrast to Unicode text fonts, which contain the lunate style epsilon separately from the Greek alphabet among variant letter forms and symbols (where Unicode assigns ‘Greek Lunate Epsilon Symbol’). The definition of `\LaTeXe`, which prints ‘L^AT_EX₂ε’ uses `\varepsilon` and care should be taken to ensure that `\LaTeXe` still uses the usual epsilon either by using one of the methods that were presented in this section or by

TABLE 3 Computer Modern Italic in text and mathematics.

Text	<i>abcdefghijklmnopqrstuvwxy</i>
Mathematics	<i>abcdefghijklmnopqrstuvwxy</i>

redefining `\LaTeXe` (possibly using my `metalogo` package), because ‘`LATEX 2ε`’ is not as good as ‘`LATEX 2ε`’.

6 GLYPH BOUNDS

In using this package to set text fonts for mathematics, there are issues with glyph bounds (similar to italic correction), as illustrated in the equations:

```
\begin{align}
f(x) &= \sum_{n = -\infty}^{\infty} c_n e^{jnx} & \backslash\backslash \\
c_n &= \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-jnx} dx & \backslash, \mathrm{d}x \\
\end{align}
```

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{jnx} \quad (1)$$

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-jnx} dx \quad (2)$$

where the function f is too close to the parenthesis (and the exponent jnx is too close to its base e that there is in fact a collision. The reason is that the font has metrics that are suitable for use in text, but not for mathematics.

Comparing the alphabets typeset in Computer Modern italic, in table 3, it is shown that many characters have greater space around them in the mathematics version, which does not exist in the text version.

"*character*

It is necessary to indicate exactly where additional space needs to be inserted. By putting " before a character, the character will be typeset with additional space inserted on both sides of it.

\ "*string*"

For multiple adjacent characters, each need not have a " in front of it, but the characters may be surrounded by \ " before and " after. That is, for example `\ "abcde"` is equivalent to `"a"b"c"d"e`.

TABLE 4 All valid characters for use with " and \"...\".

0123456789
ABCDEFGHIJKLMN OP QRSTUVWXYZ
<i>abcdefghijklmnopqrstuvwxyz</i>
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩϜ
<i>αβγδεζηθικλμνξοπρςστυφχψωϝ</i>
ϐϑϕϗχϘϙϠϡ

TABLE 5 Bembo MT in text and spaced mathematics.

Text	<i>abcdefghijklmnopqrstuvwxyz</i>
Mathematics	<i>abcdefghijklmnopqrstuvwxyz</i>

Note that *<character>* must be one of the symbols that are listed in table 4. *<string>* must one or more such characters adjacent to each other. So `\xyz` and `\sin x` is okay, but `\sin x` is not because `\sin` is not a valid character. Also note that with Greek symbols, they can only be used if typed literally, not via control sequences, so `\cos θ` is okay but `\cos \theta` is not. Use of control sequences for Greek symbols is still perfectly acceptable because the spacings are built into the definitions of each (for example `\alpha` is defined to expand to `{α}` automatically).

Use of " and \ in mathematics mode does not interfere with their use in text mode for quotation marks and umlauts or diæreses: " " coöperates, even with "Mapping=te^x-text".

`\setminwhitespace[<number>]`

Use this command to change the minimum allowed white space around such a spaced character. The unit of *<number>* is millimu (mmu), that is $\frac{1}{1000}$ mu. Recall 18 mu = 1 em.

The default *<number>* is 500 which corresponds to $500 \text{ mmu} = \frac{500}{1000} \text{ mu} = 0.5 \text{ mu} = \frac{1}{36} \text{ em}$.

Note that this value corresponds to the inserted spaces on both sides for each spaced character. If two adjacent characters are spaced, then the total minimum white space between the two characters is twice this value. The effect is shown in table 5.

Now, the spacing in equations (1, 2) are improved by:

```
\begin{align}
```

```
"f\left("x\right) &= \sum_{n = -\infty}^{\infty} c_{"n}"e^{"jnx"} \\  
"c_{"n}" &= \frac{1}{2\pi} \int_{-\pi}^{\pi} f\left("x\right) e^{-"jnx"} \, \mathrm{d}x  
\end{align}
```

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{jnx} \quad (3)$$

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-jnx} dx \quad (4)$$

Also note that the spacing is improved by replacing (x) with $\left(x\right)$ (which you should be doing anyway).

7 COMPATABILITY

If `amsmath` is required, it must be loaded earlier than `mathspec`.

8 THE PACKAGE

The package style file is printed in this section.

```
1 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
2 \ProvidesPackage{mathspec}
3 [2009/09/30 v0.2 LaTeX Package (Mathematics font selection for XeLaTeX)]
4
5 %% Require etoolbox for convenience and amstext for well sized text in maths
6 %% mode. ifxetex is obvious.
7 \RequirePackage{etoolbox}
8 \RequirePackage{amstext}
9 \RequirePackage{ifxetex}
10
11 %% Require a recent XeTeX version.
12 \RequireXeTeX
13 \ifcsundef{XeTeXGlyphbounds}
14 {\PackageError{mathspec}
15 {mathspec requires a more recent version of XeTeX}
16 {Your current version of XeTeX is \the\XeTeXversion\XeTeXrevision.\MessageBreak
17 Update your version of XeTeX to at least 0.9995.}}
18 {\relax}
19
20 %% Booleans are created automatically on demand.
21 \newcommand\eu@booltrue[1]{\providebool{#1}\booltrue{#1}}
22 \newcommand\eu@gbooltrue[1]{\providebool{#1}\global\booltrue{#1}}
23 \newcommand\eu@boolfalse[1]{\providebool{#1}\boolfalse{#1}}
24 \newcommand\eu@ifbool[1]{\providebool{#1}\ifbool{#1}}
25
26 %% Some extras:
27 %% \eu@ifbooltrue{<bool>}{<true>}
28 %% \eu@ifboolfalse{<bool>}{<false>}
29 %% \eu@ifsomebooltrue{<bool_1>, ..., <bool_n>}{<true>}
30 %% (if <bool_i> is true for some i)
31 %% \eu@ifnumis{x}{x_1, ..., x_n}{<true>}{<false>} (if x = x_i for some i)
32 %% \eu@ifnumin{x}{a, b}{<true>}{<false>} (if a ≤ x ≤ b)
33 \newcommand\eu@ifbooltrue[2]{\eu@ifbool{#1}{#2}\relax}
34 \newcommand\eu@ifboolfalse[2]{\eu@ifbool{#1}\relax}{#2}
35 \newcommand\eu@ifsomebooltrue[1]{%
36 \eu@boolfalse{temp}%
37 \def\do##1{\eu@ifbooltrue{##1}{\eu@booltrue{temp}}}%
38 \docsvlist{#1}%
39 \eu@ifbooltrue{temp}}
40 \newcommand\eu@ifnumis[2]{%
41 \providebool{temp}%
42 \boolfalse{temp}%
43 \def\do##1{\ifnumcomp{#1}{=}{##1}{\booltrue{temp}}\relax}%
44 \docsvlist{#2}%
45 \ifbool{temp}}
46 \newcommand\eu@ifnumin[2]{\@eu@ifnumin{#1}#2@nil}
47 \def\@eu@ifnumin #1#2,#3@nil{%
```

```

48 \providebool{temp}%
49 \booltrue{temp}%
50 \ifnumcomp{#1}{<}{#2}{\boolfalse{temp}}{\relax}%
51 \ifnumcomp{#3}{<}{#1}{\boolfalse{temp}}{\relax}%
52 \ifbool{temp}}
53
54 %% Options.
55 \DeclareOption{normalskips}
56   {\PackageWarning{mathspec}
57     {Package option `normalskips' is deprecated}}
58 \def\eu@zf@math{no-math}
59 \DeclareOption{math}{\def\eu@zf@math{math}}
60 \DeclareOption{no-math}{\relax}
61 \DeclareOption{MnSymbol}{\eu@booltrue{MnSymbol}}
62 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{fontspec}}
63 \ProcessOptions\relax
64
65 %% Requires packages:
66 %% fontspec, xkeyval, mathstyle, etoolbox and maybe MnSymbol
67 \RequirePackage[\eu@zf@math]{fontspec}[2008/08/09]
68 \RequirePackage{xkeyval}
69 \eu@ifbooltrue{MnSymbol}{\RequirePackage{MnSymbol}}
70
71 \providecommand\currentmathstyle{\relax}
72
73 %% Check whether a package is Loaded.
74 %% \eu@package@Loaded<package>
75 \newcommand\eu@package@loaded[1]{
76   \@ifpackageloaded{#1}
77     {\eu@booltrue{eu@#1@loaded}}
78     {\eu@boolfalse{eu@#1@loaded}}}
79 %% Test for amsmath. Do I need to do this here or at all?
80 \eu@package@loaded{amsmath}
81
82 %% Define the undefined Greek Letters. Include all variant forms (same as normal
83 %% forms if variant forms are unavailable). Latin F for digamma is better than
84 %% nothing. If Digamma is available, then Capital and Lowercase the are same.
85 \DeclareMathSymbol{\Alpha}{\mathalpha}{operators}{"41}
86 \DeclareMathSymbol{\Beta}{\mathalpha}{operators}{"42}
87 \DeclareMathSymbol{\Epsilon}{\mathalpha}{operators}{"45}
88 \DeclareMathSymbol{\Zeta}{\mathalpha}{operators}{"5A}
89 \DeclareMathSymbol{\Eta}{\mathalpha}{operators}{"48}
90 \DeclareMathSymbol{\Iota}{\mathalpha}{operators}{"49}
91 \DeclareMathSymbol{\Kappa}{\mathalpha}{operators}{"4B}
92 \DeclareMathSymbol{\Mu}{\mathalpha}{operators}{"4D}
93 \DeclareMathSymbol{\Nu}{\mathalpha}{operators}{"4E}
94 \DeclareMathSymbol{\Omicron}{\mathalpha}{operators}{"4F}
95 \DeclareMathSymbol{\Rho}{\mathalpha}{operators}{"50}
96 \DeclareMathSymbol{\Tau}{\mathalpha}{operators}{"54}
97 \DeclareMathSymbol{\Chi}{\mathalpha}{operators}{"58}
98 \DeclareMathSymbol{\omicron}{\mathord}{letters}{"6F}

```

```

99 \let\varbeta\beta
100 \ifdef{\varkappa}
101   {\relax}
102   {\let\varkappa\kappa}
103 \ifdef{\varTheta}
104   {\let\eu@cm@varTheta\varTheta}
105   {\relax}
106 \let\varTheta\Theta
107 \ifdef{\digamma}
108   {\let\Digamma\digamma}
109   {\DeclareMathSymbol{\Digamma}{\mathalpha}{operators}"46}
110   {\DeclareMathSymbol{\digamma}{\mathord}{letters}"46}}
111
112 %% Preserve old definitions of all Greek Letters. \eu@cm@alpha etc.
113 \def\do#1{\csletcs{eu@cm@#1}{#1}}
114 \docsvlist{
115   Alpha,Beta,Gamma,Delta,Epsilon,Zeta,Eta,Theta,Iota,Kappa,Lambda,Mu,Nu,
116   Xi,Omicron,Pi,Rho,Sigma,Tau,Upsilon,Phi,Chi,Psi,Omega,Digamma,alpha,
117   beta,gamma,delta,epsilon,zeta,eta,theta,iota,kappa,lambda,mu,nu,xi,
118   omicron,pi,rho,varsigma,sigma,tau,upsilon,phi,chi,psi,omega,digamma,
119   varTheta,varbeta,varepsilon,vartheta,varkappa,varpi,varrho,varphi}
120 %% varTheta is done separately because amsmath defines it differently.
121
122 %% Can define/redefine any command using the syntax of
123 %% \newcommand/\renewcommand without error. Some helpers.
124 \newcommand\renewcommand[1]{\ifdef{#1}{\renewcommand{#1}}{\newcommand{#1}}}
125 \providecommand\expanded[1]{\edef@tempa{#1}\@tempa}
126 \newcommand\eu@setkeys[3][]{
127   \ifblank{#1}{\relax}{\presetkeys[eu]{#2}{#1}{}}
128   \expanded{\noexpand\setkeys*[eu]{#2}{\expandonce#3\@empty}}}
129 \newcommand\eu@fontspec[2]{
130   \expanded{\noexpand\zf@fontspec{\expandonce#1\@empty}{\expandonce#2\@empty}}}
131
132 %% Deprecated commands.
133 %% \+: Use \dots or " instead.
134 %% \plaindigits: Digits are no longer selected automatically from either
135 %% the Latin or the Greek mathematics font. Instead, they
136 %% must explicitly be stated. If you want plain digits, just
137 %% don't say you want digits.
138 %5 \normalvarforms: Now called \normalisevarforms
139 %% \varforms: Now called \exchangeforms
140 %% Eventually, documents that use these will compile with error.
141 \newcommand\+{
142   \PackageWarning{mathspec}
143   {\protect\+\space is deprecated, recommend to use\MessageBreak\protect\"..."}
144   or "}
145   \eu@plus}
146 \newcommand\setsansfonts{
147   \PackageWarning{mathspec}
148   {\protect\setsansfonts\space is deprecated, recommend to use \protect
149   \setallsansfonts.}

```

```

I50 \setallsansfonts}
I51 \@onlypreamble\setsansfonts
I52 \newcommand\setmonofonts{
I53 \PackageWarning{mathspec}
I54 {\protect\setmonofonts\space is deprecated, recommend to use \protect
I55 \setallmonofonts.}
I56 \setallsansfonts}
I57 \@onlypreamble\setmonofonts
I58 \newcommand\plaindigits{
I59 \PackageWarning{mathspec}{\protect\plaindigits\space is deprecated}}
I60 \@onlypreamble\plaindigits
I61 \newcommand\normalvarforms{
I62 \PackageWarning{mathspec}
I63 {\protect\normalvarforms\space is deprecated, recommend to use
I64 \protect\normalisevarforms\space or \protect\normalizevarforms}
I65 \normalisevarforms}
I66 \@onlypreamble\normalvarforms
I67 \newcommand\varforms{
I68 \PackageWarning{mathspec}
I69 {\protect\varforms\space is deprecated, recommend to use
I70 \protect\exchangeforms}
I71 \exchangeforms}
I72 \@onlypreamble\varforms
I73
I74 %% I don't want this clogging up my sty file. It will be gone eventually.
I75 \def@ifnext#1#2#3{%
I76 \let\@tempd=#1%
I77 \def\@tempa{#2}%
I78 \def\@tempb{#3}%
I79 \futurelet\@tempc\@ifnexta}
I80 \def@ifnexta{%
I81 \ifx\@tempc\@tempd%
I82 \let\@tempb\@tempa%
I83 \fi\@tempb}
I84 \def\eu@DeclareRobustCommand{\@star@or@long\eu@declare@robustcommand}
I85 \def\eu@declare@robustcommand#1{%
I86 \ifx#1\@undefined\else\ifx#1\relax\else
I87 \@latex@info{Redefining \string#1}%
I88 \fi\fi
I89 \edef\reserved@a{\string#1}%
I90 \def\reserved@b{#1}%
I91 \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
I92 \edef#1{%
I93 \ifx\reserved@a\reserved@b
I94 \noexpand\x@protect
I95 \noexpand#1%
I96 \fi
I97 \noexpand\protect
I98 \expandafter\noexpand\cname\expandafter\@gobble\string#1\endcsname}%
I99 \let\@ifdefinable\@rc@ifdefinable
I200 \expandafter\eu@new@command\cname\expandafter\@gobble\string#1\endcsname}

```

```

201 \def\eu@new@command#1{\eu@@testopt{\eu@@newcommand#1}{}
202 \def\eu@@newcommand#1[#2]{\@ifnextx{\@xargdef#1[#2]}{\@argdef#1[#2]}}
203 \long\def\eu@@testopt#1#2{\@ifnextx[#1]{#1[#2]}}
204 \eu@DeclareRobustCommand\eu@plus[1][[]]{
205
206 %% The main user command (comes in two spellings)
207 %% \setmathsfont(<sets>)[<shapes, font features>]{<font name>}
208 %% \setmathfont(<sets>)[<shapes, font features>]{<font name>}
209 %%
210 %% <set> is a CSV list of any of: Special, Latin, Greek, Digits, Symbols. If
211 %% (<set>) is omitted, then (Special) is assumed. Special is provided to
212 %% hook to some external code (e.g. potentially unicode-math). That is,
213 %% for mathspec purposes, (<sets>) is mandatory. Omit it so that it
214 %% behaves like a different command. See below.
215 %% <shapes> is the keyval list of font shapes for the subset of <set>.
216 %% Valid keys are: Uppercase, Lowercase, Arabic.
217 %% Valid values are: Regular, Italic, Plain.
218 %% <font features>, <font name> follow directly from fontspec.
219 \providecommand\setmathsfont{\eu@setmathsfont}
220 \let\setmathfont\setmathsfont
221 \newcommand\eu@setmathsfont{
222 \@@ifnextchar(
223 {\eu@setmathsfont}
224 {\eu@setmathsfont(Special)}}
225 \def\@@eu@setmathsfont(#1){
226 \edef\eu@setmathsfont@Set{#1}
227 \@@eu@setmathsfont}
228 \newcommand\@@eu@setmathsfont[2][[]]{
229 \@for\i@for:=\eu@setmathsfont@Set\do
230 {\csname eu@ScopeSet@\i@for\endcsname[#1]{#2}}}
231
232 %% The FIRST branch \setmathsfont(Special)
233 %% Predefine this command before mathspec is loaded (or redefine it) to get
234 %%
235 %% \setmathsfont[<font features>]{<font name>}
236 %%
237 %% to do something else.
238 %%
239 %% Essentially, \setmathsfont(<sets>)[<font features>]{<font name>} is a mathspec
240 %% command, but \setmathsfont[<font features>]{<font name>} is an external
241 %% command.
242 \providecommand\eu@ScopeSet@Special[2][[]]{\relax}
243
244 %% The SECOND branch \setmathsfont(Digits)
245 \newcommand\eu@ScopeSet@Digits[2][[]]{
246 \eu@setkeys[Arabic=Regular]{Digits}{#1}
247 \eu@fontspec{XKV@rm}{#2}
248 \ifcase\eu@DigitsArabic@@value %% If Digits Regular
249 \ernewcommand\eu@DigitsArabic@symfont{Digits:m:n}
250 \let\eu@Digitmathsfont\zf@family
251 \eu@booltrue{Digits}

```

```

252 \or %% If Digits Italic
253 \ernewcommand\eu@DigitsArabic@symfont{Digits:m:it}
254 \let\eu@Digitsmathsfont\zf@family
255 \eu@booltrue{Digits}
256 \or %% If Digits Plain
257 \eu@boolfalse{Digits}
258 \fi
259 \eu@ifsomebooltrue{Digits}
260 {\DeclareSymbolFont{Digits:m:n}{EU1}{\eu@Digitsmathsfont}{m}{n}}
261 \eu@ifbooltrue{Digits}
262 {\fontfamily\eu@Digitsmathsfont\selectfont
263 \DeclareMathSymbol{0}{\mathord}{\eu@DigitsArabic@symfont}{`0}
264 \DeclareMathSymbol{1}{\mathord}{\eu@DigitsArabic@symfont}{`1}
265 \DeclareMathSymbol{2}{\mathord}{\eu@DigitsArabic@symfont}{`2}
266 \DeclareMathSymbol{3}{\mathord}{\eu@DigitsArabic@symfont}{`3}
267 \DeclareMathSymbol{4}{\mathord}{\eu@DigitsArabic@symfont}{`4}
268 \DeclareMathSymbol{5}{\mathord}{\eu@DigitsArabic@symfont}{`5}
269 \DeclareMathSymbol{6}{\mathord}{\eu@DigitsArabic@symfont}{`6}
270 \DeclareMathSymbol{7}{\mathord}{\eu@DigitsArabic@symfont}{`7}
271 \DeclareMathSymbol{8}{\mathord}{\eu@DigitsArabic@symfont}{`8}
272 \DeclareMathSymbol{9}{\mathord}{\eu@DigitsArabic@symfont}{`9}}}}
273
274 %% The THIRD branch \setmathsfont(Latin)
275 \newcommand\eu@ScopeSet@Latin[2][ ]{
276 \eu@setkeys[Uppercase=Italic,Lowercase=Italic]{Latin}{#1}
277 \eu@fontspec{XKV@rm}{#2}
278 \ifcase\eu@LatinUppercase@@value %% If Latin Uppercase Regular
279 \ernewcommand\eu@LatinUppercase@symfont{Latin:m:n}
280 \let\eu@Latinmathsfont\zf@family
281 \eu@booltrue{LatinUppercase}
282 \or %% If Latin Uppercase Italic
283 \ernewcommand\eu@LatinUppercase@symfont{Latin:m:it}
284 \let\eu@Latinmathsfont\zf@family
285 \eu@booltrue{LatinUppercase}
286 \or %% If Latin Uppercase Plain
287 \eu@boolfalse{LatinUppercase}
288 \fi
289 \ifcase\eu@LatinLowercase@@value %% If Latin Lowercase Regular
290 \ernewcommand\eu@LatinLowercase@symfont{Latin:m:n}
291 \let\eu@Latinmathsfont\zf@family
292 \eu@booltrue{LatinLowercase}
293 \or %% If Latin Lowercase Italic
294 \ernewcommand\eu@LatinLowercase@symfont{Latin:m:it}
295 \let\eu@Latinmathsfont\zf@family
296 \eu@booltrue{LatinLowercase}
297 \or %% If Latin Lowercase Plain
298 \eu@boolfalse{LatinLowercase}
299 \fi
300 \eu@ifsomebooltrue{LatinUppercase,LatinLowercase}
301 {\DeclareSymbolFont{Latin:m:n}{EU1}{\eu@Latinmathsfont}{m}{n}}
302 \DeclareSymbolFont{Latin:m:it}{EU1}{\eu@Latinmathsfont}{m}{it}}

```

```

303 \eu@ifbooltrue{LatinUppercase}
304   {\fontfamily\eu@Latinmathsfnt\selectfont
305     \DeclareMathSymbol{A}{\mathalpha}{\eu@LatinUppercase@symfont}{`A}
306     \DeclareMathSymbol{B}{\mathalpha}{\eu@LatinUppercase@symfont}{`B}
307     \DeclareMathSymbol{C}{\mathalpha}{\eu@LatinUppercase@symfont}{`C}
308     \DeclareMathSymbol{D}{\mathalpha}{\eu@LatinUppercase@symfont}{`D}
309     \DeclareMathSymbol{E}{\mathalpha}{\eu@LatinUppercase@symfont}{`E}
310     \DeclareMathSymbol{F}{\mathalpha}{\eu@LatinUppercase@symfont}{`F}
311     \DeclareMathSymbol{G}{\mathalpha}{\eu@LatinUppercase@symfont}{`G}
312     \DeclareMathSymbol{H}{\mathalpha}{\eu@LatinUppercase@symfont}{`H}
313     \DeclareMathSymbol{I}{\mathalpha}{\eu@LatinUppercase@symfont}{`I}
314     \DeclareMathSymbol{J}{\mathalpha}{\eu@LatinUppercase@symfont}{`J}
315     \DeclareMathSymbol{K}{\mathalpha}{\eu@LatinUppercase@symfont}{`K}
316     \DeclareMathSymbol{L}{\mathalpha}{\eu@LatinUppercase@symfont}{`L}
317     \DeclareMathSymbol{M}{\mathalpha}{\eu@LatinUppercase@symfont}{`M}
318     \DeclareMathSymbol{N}{\mathalpha}{\eu@LatinUppercase@symfont}{`N}
319     \DeclareMathSymbol{O}{\mathalpha}{\eu@LatinUppercase@symfont}{`O}
320     \DeclareMathSymbol{P}{\mathalpha}{\eu@LatinUppercase@symfont}{`P}
321     \DeclareMathSymbol{Q}{\mathalpha}{\eu@LatinUppercase@symfont}{`Q}
322     \DeclareMathSymbol{R}{\mathalpha}{\eu@LatinUppercase@symfont}{`R}
323     \DeclareMathSymbol{S}{\mathalpha}{\eu@LatinUppercase@symfont}{`S}
324     \DeclareMathSymbol{T}{\mathalpha}{\eu@LatinUppercase@symfont}{`T}
325     \DeclareMathSymbol{U}{\mathalpha}{\eu@LatinUppercase@symfont}{`U}
326     \DeclareMathSymbol{V}{\mathalpha}{\eu@LatinUppercase@symfont}{`V}
327     \DeclareMathSymbol{W}{\mathalpha}{\eu@LatinUppercase@symfont}{`W}
328     \DeclareMathSymbol{X}{\mathalpha}{\eu@LatinUppercase@symfont}{`X}
329     \DeclareMathSymbol{Y}{\mathalpha}{\eu@LatinUppercase@symfont}{`Y}
330     \DeclareMathSymbol{Z}{\mathalpha}{\eu@LatinUppercase@symfont}{`Z}}
331 \eu@ifbooltrue{LatinLowercase}
332   {\fontfamily\eu@Latinmathsfnt\selectfont
333     \DeclareMathSymbol{a}{\mathalpha}{\eu@LatinLowercase@symfont}{`a}
334     \DeclareMathSymbol{b}{\mathalpha}{\eu@LatinLowercase@symfont}{`b}
335     \DeclareMathSymbol{c}{\mathalpha}{\eu@LatinLowercase@symfont}{`c}
336     \DeclareMathSymbol{d}{\mathalpha}{\eu@LatinLowercase@symfont}{`d}
337     \DeclareMathSymbol{e}{\mathalpha}{\eu@LatinLowercase@symfont}{`e}
338     \DeclareMathSymbol{f}{\mathalpha}{\eu@LatinLowercase@symfont}{`f}
339     \DeclareMathSymbol{g}{\mathalpha}{\eu@LatinLowercase@symfont}{`g}
340     \DeclareMathSymbol{h}{\mathalpha}{\eu@LatinLowercase@symfont}{`h}
341     \DeclareMathSymbol{i}{\mathalpha}{\eu@LatinLowercase@symfont}{`i}
342     \DeclareMathSymbol{j}{\mathalpha}{\eu@LatinLowercase@symfont}{`j}
343     \DeclareMathSymbol{k}{\mathalpha}{\eu@LatinLowercase@symfont}{`k}
344     \DeclareMathSymbol{l}{\mathalpha}{\eu@LatinLowercase@symfont}{`l}
345     \DeclareMathSymbol{m}{\mathalpha}{\eu@LatinLowercase@symfont}{`m}
346     \DeclareMathSymbol{n}{\mathalpha}{\eu@LatinLowercase@symfont}{`n}
347     \DeclareMathSymbol{o}{\mathalpha}{\eu@LatinLowercase@symfont}{`o}
348     \DeclareMathSymbol{p}{\mathalpha}{\eu@LatinLowercase@symfont}{`p}
349     \DeclareMathSymbol{q}{\mathalpha}{\eu@LatinLowercase@symfont}{`q}
350     \DeclareMathSymbol{r}{\mathalpha}{\eu@LatinLowercase@symfont}{`r}
351     \DeclareMathSymbol{s}{\mathalpha}{\eu@LatinLowercase@symfont}{`s}
352     \DeclareMathSymbol{t}{\mathalpha}{\eu@LatinLowercase@symfont}{`t}
353     \DeclareMathSymbol{u}{\mathalpha}{\eu@LatinLowercase@symfont}{`u}

```

```

354 \DeclareMathSymbol{v}{\mathalpha}{\eu@LatinLowercase@symfont}{`v}
355 \DeclareMathSymbol{w}{\mathalpha}{\eu@LatinLowercase@symfont}{`w}
356 \DeclareMathSymbol{x}{\mathalpha}{\eu@LatinLowercase@symfont}{`x}
357 \DeclareMathSymbol{y}{\mathalpha}{\eu@LatinLowercase@symfont}{`y}
358 \DeclareMathSymbol{z}{\mathalpha}{\eu@LatinLowercase@symfont}{`z}}
359
360 %% The FOURTH branch \setmathsfont(Greek)
361 \newcommand\eu@ScopeSet@Greek[2][]{
362 \eu@setkeys[Uppercase=Regular,Lowercase=Italic]{Greek}{#1}
363 \eu@fontspec{\XKV@rm}{#2}
364 \ifcase\eu@GreekUppercase@@@value %% If Greek Uppercase Regular
365 \ernewcommand\eu@GreekUppercase@symfont{Greek:m:n}
366 \let\eu@Greekmathsfont\zf@family
367 \eu@booltrue{GreekUppercase}
368 \or %% If Greek Uppercase Italic
369 \ernewcommand\eu@GreekUppercase@symfont{Greek:m:it}
370 \let\eu@Greekmathsfont\zf@family
371 \eu@booltrue{GreekUppercase}
372 \or %% If Greek Uppercase Plain
373 \eu@boolfalse{GreekUppercase}
374 \fi
375 \ifcase\eu@GreekLowercase@@@value %% If Greek Lowercase Regular
376 \ernewcommand\eu@GreekLowercase@symfont{Greek:m:n}
377 \let\eu@Greekmathsfont\zf@family
378 \eu@booltrue{GreekLowercase}
379 \or %% If Greek Lowercase Italic
380 \ernewcommand\eu@GreekLowercase@symfont{Greek:m:it}
381 \let\eu@Greekmathsfont\zf@family
382 \eu@booltrue{GreekLowercase}
383 \or %% If Greek Lowercase Plain
384 \eu@boolfalse{GreekLowercase}
385 \fi
386 \eu@ifsomebooltrue{GreekUppercase,GreekLowercase}
387 {\DeclareSymbolFont{Greek:m:n}{EU1}{\eu@Greekmathsfont}{m}{n}
388 \DeclareSymbolFont{Greek:m:it}{EU1}{\eu@Greekmathsfont}{m}{it}}
389 \eu@ifbooltrue{GreekUppercase}
390 {\fontfamily\eu@Greekmathsfont\selectfont
391 \XeTeXDeclareMathSymbolA{\mathalpha}{\eu@GreekUppercase@symfontA}{` }[\Alpha]
392 \XeTeXDeclareMathSymbolB{\mathalpha}{\eu@GreekUppercase@symfontB}{` }[\Beta]
393 \XeTeXDeclareMathSymbolGamma{\mathalpha}{\eu@GreekUppercase@symfontGamma}{` }[\Gamma]
394 \XeTeXDeclareMathSymbolDelta{\mathalpha}{\eu@GreekUppercase@symfontDelta}{` }[\Delta]
395 \XeTeXDeclareMathSymbolEpsilon{\mathalpha}{\eu@GreekUppercase@symfontEpsilon}{` }[\Epsilon]
396 \XeTeXDeclareMathSymbolZeta{\mathalpha}{\eu@GreekUppercase@symfontZeta}{` }[\Zeta]
397 \XeTeXDeclareMathSymbolEta{\mathalpha}{\eu@GreekUppercase@symfontEta}{` }[\Eta]
398 \XeTeXDeclareMathSymbolTheta{\mathalpha}{\eu@GreekUppercase@symfontTheta}{` }[\Theta]
399 \XeTeXDeclareMathSymbolIota{\mathalpha}{\eu@GreekUppercase@symfontIota}{` }[\Iota]
400 \XeTeXDeclareMathSymbolKappa{\mathalpha}{\eu@GreekUppercase@symfontKappa}{` }[\Kappa]
401 \XeTeXDeclareMathSymbolLambda{\mathalpha}{\eu@GreekUppercase@symfontLambda}{` }[\Lambda]
402 \XeTeXDeclareMathSymbolMu{\mathalpha}{\eu@GreekUppercase@symfontMu}{` }[\Mu]
403 \XeTeXDeclareMathSymbolNu{\mathalpha}{\eu@GreekUppercase@symfontNu}{` }[\Nu]
404 \XeTeXDeclareMathSymbolXi{\mathalpha}{\eu@GreekUppercase@symfontXi}{` }[\Xi]

```

```

405 \XeTeXDeclareMathSymbol0{\mathalpha}{\eu@GreekUppercase@symfont0}{` }[\Omicron]
406 \XeTeXDeclareMathSymbol1{\mathalpha}{\eu@GreekUppercase@symfont1}{` }[\Pi]
407 \XeTeXDeclareMathSymbolP{\mathalpha}{\eu@GreekUppercase@symfontP}{` }[\Rho]
408 \XeTeXDeclareMathSymbolSigma{\mathalpha}{\eu@GreekUppercase@symfontSigma}{` }[\Sigma]
409 \XeTeXDeclareMathSymbolT{\mathalpha}{\eu@GreekUppercase@symfontT}{` }[\Tau]
410 \XeTeXDeclareMathSymbolY{\mathalpha}{\eu@GreekUppercase@symfontY}{` }[\Upsilon]
411 \XeTeXDeclareMathSymbolPhi{\mathalpha}{\eu@GreekUppercase@symfontPhi}{` }[\Phi]
412 \XeTeXDeclareMathSymbolX{\mathalpha}{\eu@GreekUppercase@symfontX}{` }[\Chi]
413 \XeTeXDeclareMathSymbolPsi{\mathalpha}{\eu@GreekUppercase@symfontPsi}{` }[\Psi]
414 \XeTeXDeclareMathSymbolOmega{\mathalpha}{\eu@GreekUppercase@symfontOmega}{` }[\Omega]
415 \XeTeXDeclareMathSymbolF{\mathalpha}{\eu@GreekUppercase@symfontF}{` }[\Digamma]
416 \XeTeXDeclareMathSymboltheta{\mathalpha}{\eu@GreekUppercase@symfonttheta}{` }[\varTheta]
417 \eu@fixgreekcsA{Alpha}
418 \eu@fixgreekcsB{Beta}
419 \eu@fixgreekcsGamma{Gamma}
420 \eu@fixgreekcsDelta{Delta}
421 \eu@fixgreekcsE{Epsilon}
422 \eu@fixgreekcsZ{Zeta}
423 \eu@fixgreekcsH{Eta}
424 \eu@fixgreekcsO{Theta}
425 \eu@fixgreekcsI{Iota}
426 \eu@fixgreekcsK{Kappa}
427 \eu@fixgreekcsLambda{Lambda}
428 \eu@fixgreekcsM{Mu}
429 \eu@fixgreekcsN{Nu}
430 \eu@fixgreekcsXi{Xi}
431 \eu@fixgreekcsOmicron{Omicron}
432 \eu@fixgreekcsPi{Pi}
433 \eu@fixgreekcsRho{Rho}
434 \eu@fixgreekcsSigma{Sigma}
435 \eu@fixgreekcsTau{Tau}
436 \eu@fixgreekcsUpsilon{Upsilon}
437 \eu@fixgreekcsPhi{Phi}
438 \eu@fixgreekcsChi{Chi}
439 \eu@fixgreekcsPsi{Psi}
440 \eu@fixgreekcsOmega{Omega}
441 \eu@fixgreekcsF{Digamma}
442 \eu@fixgreekcstheta{varTheta}
443 \eu@ifbooltrue{GreekLowercase}
444 { \fontfamily\eu@Greekmathsfont\selectfont
445 \XeTeXDeclareMathSymbolalpha{\mathalpha}{\eu@GreekLowercase@symfontalpha}{` }[\alpha]
446 \XeTeXDeclareMathSymbolbeta{\mathalpha}{\eu@GreekLowercase@symfontbeta}{` }[\beta]
447 \XeTeXDeclareMathSymbolgamma{\mathalpha}{\eu@GreekLowercase@symfontgamma}{` }[\gamma]
448 \XeTeXDeclareMathSymboldelta{\mathalpha}{\eu@GreekLowercase@symfontdelta}{` }[\delta]
449 \XeTeXDeclareMathSymbolepsilon{\mathalpha}{\eu@GreekLowercase@symfontepsilon}{` }[\epsilon]
450 \XeTeXDeclareMathSymbolzeta{\mathalpha}{\eu@GreekLowercase@symfontzeta}{` }[\zeta]
451 \XeTeXDeclareMathSymboleta{\mathalpha}{\eu@GreekLowercase@symfonteta}{` }[\eta]
452 \XeTeXDeclareMathSymboltheta{\mathalpha}{\eu@GreekLowercase@symfonttheta}{` }[\theta]
453 \XeTeXDeclareMathSymboliota{\mathalpha}{\eu@GreekLowercase@symfontiota}{` }[\iota]
454 \XeTeXDeclareMathSymbolkappa{\mathalpha}{\eu@GreekLowercase@symfontkappa}{` }[\kappa]
455 \XeTeXDeclareMathSymbollambda{\mathalpha}{\eu@GreekLowercase@symfontlambda}{` }[\lambda]

```

```

456 \XeTeXDeclareMathSymbol\mu{\mathalpha}{\eu@GreekLowercase@symfont\mu}{\mu}
457 \XeTeXDeclareMathSymbol\nu{\mathalpha}{\eu@GreekLowercase@symfont\nu}{\nu}
458 \XeTeXDeclareMathSymbol\xi{\mathalpha}{\eu@GreekLowercase@symfont\xi}{\xi}
459 \XeTeXDeclareMathSymbol\omicron{\mathalpha}{\eu@GreekLowercase@symfont\omicron}{\omicron}
460 \XeTeXDeclareMathSymbol\pi{\mathalpha}{\eu@GreekLowercase@symfont\pi}{\pi}
461 \XeTeXDeclareMathSymbol\rho{\mathalpha}{\eu@GreekLowercase@symfont\rho}{\rho}
462 \XeTeXDeclareMathSymbol\varsigma{\mathalpha}{\eu@GreekLowercase@symfont\varsigma}{\varsigma}
463 \XeTeXDeclareMathSymbol\sigma{\mathalpha}{\eu@GreekLowercase@symfont\sigma}{\sigma}
464 \XeTeXDeclareMathSymbol\tau{\mathalpha}{\eu@GreekLowercase@symfont\tau}{\tau}
465 \XeTeXDeclareMathSymbol\upsilon{\mathalpha}{\eu@GreekLowercase@symfont\upsilon}{\upsilon}
466 \XeTeXDeclareMathSymbol\phi{\mathalpha}{\eu@GreekLowercase@symfont\phi}{\phi}
467 \XeTeXDeclareMathSymbol\chi{\mathalpha}{\eu@GreekLowercase@symfont\chi}{\chi}
468 \XeTeXDeclareMathSymbol\psi{\mathalpha}{\eu@GreekLowercase@symfont\psi}{\psi}
469 \XeTeXDeclareMathSymbol\omega{\mathalpha}{\eu@GreekLowercase@symfont\omega}{\omega}
470 \XeTeXDeclareMathSymbol\digamma{\mathalpha}{\eu@GreekLowercase@symfont\digamma}{\digamma}
471 \XeTeXDeclareMathSymbol\varbeta{\mathalpha}{\eu@GreekLowercase@symfont\varbeta}{\varbeta}
472 \XeTeXDeclareMathSymbol\varepsilon{\mathalpha}{\eu@GreekLowercase@symfont\varepsilon}{\varepsilon}
473 \XeTeXDeclareMathSymbol\vartheta{\mathalpha}{\eu@GreekLowercase@symfont\vartheta}{\vartheta}
474 \XeTeXDeclareMathSymbol\varkappa{\mathalpha}{\eu@GreekLowercase@symfont\varkappa}{\varkappa}
475 \XeTeXDeclareMathSymbol\varpi{\mathalpha}{\eu@GreekLowercase@symfont\varpi}{\varpi}
476 \XeTeXDeclareMathSymbol\varrho{\mathalpha}{\eu@GreekLowercase@symfont\varrho}{\varrho}
477 \XeTeXDeclareMathSymbol\varphi{\mathalpha}{\eu@GreekLowercase@symfont\varphi}{\varphi}
478 \eu@fixgreekcs\alpha}
479 \eu@fixgreekcs\beta}
480 \eu@fixgreekcs\gamma}
481 \eu@fixgreekcs\delta}
482 \eu@fixgreekcs\epsilon}
483 \eu@fixgreekcs\zeta}
484 \eu@fixgreekcs\eta}
485 \eu@fixgreekcs\theta}
486 \eu@fixgreekcs\iota}
487 \eu@fixgreekcs\kappa}
488 \eu@fixgreekcs\lambda}
489 \eu@fixgreekcs\mu}
490 \eu@fixgreekcs\nu}
491 \eu@fixgreekcs\xi}
492 \eu@fixgreekcs\omicron}
493 \eu@fixgreekcs\pi}
494 \eu@fixgreekcs\rho}
495 \eu@fixgreekcs\varsigma}
496 \eu@fixgreekcs\sigma}
497 \eu@fixgreekcs\tau}
498 \eu@fixgreekcs\upsilon}
499 \eu@fixgreekcs\phi}
500 \eu@fixgreekcs\chi}
501 \eu@fixgreekcs\psi}
502 \eu@fixgreekcs\omega}
503 \eu@fixgreekcs\digamma}
504 \eu@fixgreekcs\varbeta}
505 \eu@fixgreekcs\varepsilon}
506 \eu@fixgreekcs\vartheta}

```

```

507 \eu@fixgreekcsκ}{\varkappa}
508 \eu@fixgreekcsω}{\varpi}
509 \eu@fixgreekcsρ}{\varrho}
510 \eu@fixgreekcsφ}{\varphi}}
511 \newcommand\eu@fixgreekcs[2]{
512 \ifcsequal{#2}{\eu@cm@#2}
513 {\relax}
514 {\expandafter\def\csname #2\endcsname{"#1}}}}
515
516 %% The FIFTH branch \setmathsfont(Symbols)
517 %% The symbols are not for now.
518 %% \newcommand\eu@ScopeSet@Symbols[2][]{
519 %% \eu@fontspec{#1}{#2}
520 %% \def\eu@Symbols@symfont{Symbols:m:n}
521 %% \let\eu@Symbolsmathsfont\zf@family
522 %% \eu@booltrue{Symbols}}
523 %%
524 %% The FIFTH branch (REDEFINED) \setmathsfont(Symbols)
525 \newcommand\eu@ScopeSet@Symbols[2][{\eu@boolfalse{Symbols}}
526
527 %% Just so we know, by "all variant forms" I mean:
528 %% varbeta varpi cursive beta "omega" style pi
529 %% varepsilon varrho Lunate epsilon rho with tail hooked under
530 %% vartheta varphi cursive theta cursive phi (or orthotic phi)
531 %% varkappa varTheta cursive kappa Theta with horizontal bar connected
532 %%
533 %% Note that varsigma is not included in this list (varsigma is not a variant
534 %% form, it is a final form).
535
536 %% Some authors might prefer the normal and variant forms to be exchanged. For
537 %% example, I prefer \phi to print the orthotic phi and \varphi the cursive phi,
538 %% contrary to many text fonts.
539 %%
540 %% \exchangeforms{<list>}
541 %% <list> is a CSV list of any of:
542 %% beta, epsilon, theta, kappa, pi, rho, phi, Theta
543 \newcommand\exchangeforms[1]{
544 \AtBeginDocument{
545 \@for\i@for:=#1\do{\eu@booltrue{exchange\i@for forms}}
546 \eu@ifbooltrue{GreekLowercase}
547 {\eu@ifbooltrue{exchangebetaforms}
548 {\XeTeXDeclareMathSymbolβ}{\mathalpha}{\eu@GreekLowercase@symfontβ}{`}{\beta}
549 \XeTeXDeclareMathSymbolϐ}{\mathalpha}{\eu@GreekLowercase@symfontβ}{`}{\varbeta}}
550 \eu@ifbooltrue{exchangeepsilonforms}
551 {\XeTeXDeclareMathSymbolε}{\mathalpha}{\eu@GreekLowercase@symfontε}{`}{\epsilon}
552 \XeTeXDeclareMathSymbolϵ}{\mathalpha}{\eu@GreekLowercase@symfontε}{`}{\varepsilon}}
553 \eu@ifbooltrue{exchangethetaforms}
554 {\XeTeXDeclareMathSymbolθ}{\mathalpha}{\eu@GreekLowercase@symfontθ}{`}{\theta}
555 \XeTeXDeclareMathSymbolϑ}{\mathalpha}{\eu@GreekLowercase@symfontθ}{`}{\vartheta}}
556 \eu@ifbooltrue{exchangekappaforms}
557 {\XeTeXDeclareMathSymbolκ}{\mathalpha}{\eu@GreekLowercase@symfontκ}{`}{\kappa}

```

```

558     \XeTeXDeclareMathSymbolκ{\mathalpha}{\eu@GreekLowercase@symfontκ}{`}{\varkappa}}
559 \eu@ifbooltrue{exchangepiforms}
560   {\XeTeXDeclareMathSymbolπ{\mathalpha}{\eu@GreekLowercase@symfontπ}{`}{\pi}
561   \XeTeXDeclareMathSymbolϖ{\mathalpha}{\eu@GreekLowercase@symfontπ}{`}{\varpi}}
562 \eu@ifbooltrue{exchangerhoforms}
563   {\XeTeXDeclareMathSymbolρ{\mathalpha}{\eu@GreekLowercase@symfontρ}{`}{\rho}
564   \XeTeXDeclareMathSymbolϱ{\mathalpha}{\eu@GreekLowercase@symfontρ}{`}{\varrho}}
565 \eu@ifbooltrue{exchangephiforms}
566   {\XeTeXDeclareMathSymbolφ{\mathalpha}{\eu@GreekLowercase@symfontφ}{`}{\phi}
567   \XeTeXDeclareMathSymbolϕ{\mathalpha}{\eu@GreekLowercase@symfontφ}{`}{\varphi}}
568 \eu@ifbooltrue{GreekUppercase}
569   {\eu@ifbooltrue{exchangeThetaforms}
570   {\XeTeXDeclareMathSymbolΘ{\mathalpha}{\eu@GreekUppercase@symfontΘ}{`}{\Theta}
571   \XeTeXDeclareMathSymbolϑ{\mathalpha}{\eu@GreekUppercase@symfontΘ}{`}{\varTheta}}}}
572 \@onlypreamble\exchangeforms
573
574 %% Some text fonts do not contain all variant forms. For those that don't,
575 %% after defining the Greek mathematics font, the 'absent variant forms will
576 %% still be typeset in Computer Modern (usually).
577 %%
578 %% \normalisevarforms[<list>]
579 %% \normalizevarforms[<list>]
580 %% <list> is a CSV list of any of:
581 %%     beta, epsilon, theta, kappa, pi, rho, phi, Theta
582 %%
583 %% This command makes the listed variant forms of Greek symbols equivalent to
584 %% their normal forms, but only if they do not already exist in the font. For
585 %% example, if \varbeta and \eu@cm@varbeta are equivalent, then
586 %% \XeTeXDeclareMathSymbol failed to assign \varbeta to β because β does not
587 %% exist in the font.
588 \newcommand\normalisevarforms[1][beta,epsilon,theta,kappa,pi,rho,phi,Theta]{
589   \AtBeginDocument{
590     \@for\i@for:=#1\do{\eu@booltrue{normalisevar\i@for}}
591     \eu@ifbooltrue{GreekLowercase}
592       {\eu@ifbooltrue{normalisevarbeta}
593         {\ifx\varbeta\eu@cm@varbeta
594           \XeTeXDeclareMathSymbolβ{\mathalpha}{\eu@GreekLowercase@symfontβ}{`}}
595         \let\varbeta\beta
596         \fi}
597     \eu@ifbooltrue{normalisevarepsilon}
598       {\ifx\varepsilon\eu@cm@varepsilon
599         \XeTeXDeclareMathSymbolε{\mathalpha}{\eu@GreekLowercase@symfontε}{`}}
600       \let\varepsilon\epsilon
601       \fi}
602     \eu@ifbooltrue{normalisevartheta}
603       {\ifx\vartheta\eu@cm@vartheta
604         \XeTeXDeclareMathSymbolθ{\mathalpha}{\eu@GreekLowercase@symfontθ}{`}}
605       \let\vartheta\theta
606       \fi}
607     \eu@ifbooltrue{normalisevarkappa}
608       {\ifx\varkappa\eu@cm@varkappa

```

```

609      \XeTeXDeclareMathSymbol\alpha{\mathalpha}{\eu@GreekLowercase@symfontk}{` }
610      \let\varkappa\kappa
611      \fi}
612      \eu@ifbooltrue{normalisevarpi}
613      {\ifx\varpi\eu@cm@varpi
614      \XeTeXDeclareMathSymbol\varpi{\mathalpha}{\eu@GreekLowercase@symfont\pi}{` }
615      \let\varpi\pi
616      \fi}
617      \eu@ifbooltrue{normalisevarrho}
618      {\ifx\varrho\eu@cm@varrho
619      \XeTeXDeclareMathSymbol\rho{\mathalpha}{\eu@GreekLowercase@symfont\rho}{` }
620      \let\varrho\rho
621      \fi}
622      \eu@ifbooltrue{normalisevarphi}
623      {\ifx\varphi\eu@cm@varphi
624      \XeTeXDeclareMathSymbol\phi{\mathalpha}{\eu@GreekLowercase@symfont\phi}{` }
625      \let\varphi\phi
626      \fi}}
627      \eu@ifbooltrue{GreekUppercase}{
628      \eu@ifbooltrue{normalisevarTheta}
629      {\ifx\varTheta\eu@cm@varTheta
630      \XeTeXDeclareMathSymbol\Theta{\mathalpha}{\eu@GreekUppercase@symfont\Theta}{` }
631      \let\varTheta\Theta
632      \fi}}}}
633      \@onlypreamble\normalisevarforms
634      \let\normalizevarforms\normalisevarforms
635      \@onlypreamble\normalizevarforms
636
637      %% xkeyval things
638      \define@choicekey[eu]
639      {Digits}{Arabic}[\eu@DigitsArabic@value\eu@DigitsArabic@@@value]
640      {Regular,Italic,Plain}[Regular]{\relax}
641      \define@choicekey[eu]
642      {Latin}{Uppercase}[\eu@LatinUppercase@value\eu@LatinUppercase@@@value]
643      {Regular,Italic,Plain}[Italic]{\relax}
644      \define@choicekey[eu]
645      {Latin}{Lowercase}[\eu@LatinLowercase@value\eu@LatinLowercase@@@value]
646      {Regular,Italic,Plain}[Italic]{\relax}
647      \define@choicekey[eu]
648      {Greek}{Uppercase}[\eu@GreekUppercase@value\eu@GreekUppercase@@@value]
649      {Regular,Italic,Plain}[Regular]{\relax}
650      \define@choicekey[eu]
651      {Greek}{Lowercase}[\eu@GreekLowercase@value\eu@GreekLowercase@@@value]
652      {Regular,Italic,Plain}[Italic]{\relax}
653
654      %% Also say Digits has Uppercase and Lowercase keys, and Latin and Greek both
655      %% have Arabic key. So that if a superfluous key is accidently sent through the
656      %% wrong command (possible with \setallmainfonts), then it doesn't blow up.
657      \define@key[eu]{Digits}{Uppercase}{\relax}
658      \define@key[eu]{Digits}{Lowercase}{\relax}
659      \define@key[eu]{Latin}{Arabic}{\relax}

```

```

660 \define@key[eu]{Greek}{Arabic}{\relax}
661
662 %% I hope WILL doesn't mind if I patch fontspec. Why? Because these keys are
663 %% accidently sent through fontspec (e.g. \setallmainfonts does \setmainfont,
664 %% which is a fontspec command). So I allow fontspec to understand them, but do
665 %% nothing.
666 \define@key[zf]{options}{Arabic}{\relax}
667 \define@key[zf]{options}{Uppercase}{\relax}
668 \define@key[zf]{options}{Lowercase}{\relax}
669
670 %% Fonts have metrics suitable for text, not for mathematics. Need to kern each
671 %% Letter individually. Previous version took a trial and error approach with
672 %% \+[<size>] where <size> was determined by trial and error. Now, there is a
673 %% XeTeX primitive \XeTeXGlyphbounds which is used to automatically calculate
674 %% the required mkern.
675 %%
676 %% Some register definitions. \three@digits takes a number and, if it is less
677 %% than 100, inserts leading zeroes so that it has three digits.
678 \muskipdef\muskip@z@
679 \muskipdef\muskip@i@ne
680 \dimendef\dimen@iv 4\relax
681 \newcounter{eu@}
682 \newcounter{eu@i}
683 \providecommand\three@digits[1]{\ifnum#1<100 0\ifnum#1<10 0\fi\fi\number#1}
684
685 %% \setminwhitespace{<number>}
686 %% Sets the minimum gap between adjacent characters in mathematics.
687 %% <number> is in units of mmu (1/1000 mu). So 1 em = 18000 mmu.
688 %% \eu@minwhitespace is the minimum white space. It's default value is 500
689 %% which corresponds to 0.5 mu or 1/36 em.
690 \newcommand\setminwhitespace[1][500]{\def\eu@minwhitespace{#1}}
691 \setminwhitespace
692
693 %% \eu@mkern operates over some adjacent character tokens
694 %% \@eu@mkern operates over individual character tokens
695 %%
696 %% Unless I've missed something, it is really, really hard to get information
697 %% about the font that a mathematics symbol is typeset in, so I devised a series
698 %% of tests in text mode (\text!) (otherwise the received information will be
699 %% rubbish, related to the text font outside the mathematics). It would be
700 %% neater if I do it inside a box that's not printed, instead of an empty
701 %% \text. The \@eu@mkern algorithm to calculate the mkern might be neater if I
702 %% use e-TeX's \numexpr and \glueexpr. Maybe \gluetomu might be useful? What I
703 %% have here already ought to be sufficient but perhaps inefficient. I think
704 %% optical sizes follow through automatically because \text inherits the
705 %% surrounding size, is that right?
706 \newcommand\eu@mkern[1]{%
707   \@tfor\i@tfor:=#1\do{\expandafter\@eu@mkern\i@tfor}}
708 \def\eu@get@familyseriesshape#1:#2:#3\@nil{%
709   \expandafter\fontfamily\csname eu@#1mathsfont\endcsname\selectfont
710   \ifstrequal{#2}{m}{\mdseries}{\relax}%

```

```

711 \ifstrequal{#2}{bx}{\bfseries}{\relax}%
712 \ifstrequal{#3}{n}{\upshape}{\relax}%
713 \ifstrequal{#3}{it}{\itshape}{\relax}}
714 \def\eu@get@familyseriesshape#1:#2:#3\@nil{%
715 \def\eu@family{#1}%
716 \def\eu@series{#2}%
717 \def\eu@shape{#3}}
718 \newcounter{mkern}
719 \newcommand\eu@mkern[1]{%
720 \setcounter{mkern}{-1}%
721 \eu@boolfalse{domkern}%
722 \eu@ifnumin{`#1}{`0,`9}{\setcounter{mkern}{0}}{\relax}%
723 \eu@ifnumin{`#1}{`A,`Z}{\setcounter{mkern}{1}}{\relax}%
724 \eu@ifnumin{`#1}{`a,`z}{\setcounter{mkern}{2}}{\relax}%
725 \eu@ifnuminΑΩ{`#1}{`,`}{\setcounter{mkern}{3}}{\relax}%
726 \eu@ifnuminω{`#1}{`,`}{\setcounter{mkern}{4}}{\relax}%
727 \eu@ifnumisFΘ{`#1}{`,`}{\setcounter{mkern}{3}}{\relax}%
728 \eu@ifnumisβδφωϕκρε{`#1}{`,``,``,``,``,`}{\setcounter{mkern}{4}}{\relax}%
729 \text{%
730 \ifnum\the\c@mkern>\m@ne
731 \eu@gbooltrue{domkern}%
732 \fi
733 \ifcase\the\c@mkern\relax % If Digits (0)
734 \ifdef{\eu@DigitsArabic@symfont}%
735 {\expandafter\eu@get@familyseriesshape\eu@DigitsArabic@symfont\@nil}%
736 {\global\boolfalse{domkern}}%
737 \or % If Latin Uppercase (1)
738 \ifdef{\eu@LatinUppercase@symfont}%
739 {\expandafter\eu@get@familyseriesshape\eu@LatinUppercase@symfont\@nil}%
740 {\global\boolfalse{domkern}}%
741 \or % If Latin Lowercase (2)
742 \ifdef{\eu@LatinLowercase@symfont}%
743 {\expandafter\eu@get@familyseriesshape\eu@LatinLowercase@symfont\@nil}%
744 {\global\boolfalse{domkern}}%
745 \or % If Greek Uppercase (3)
746 \ifdef{\eu@GreekUppercase@symfont}%
747 {\expandafter\eu@get@familyseriesshape\eu@GreekUppercase@symfont\@nil}%
748 {\global\boolfalse{domkern}}%
749 \or % If Greek Lowercase (4)
750 \ifdef{\eu@GreekLowercase@symfont}%
751 {\expandafter\eu@get@familyseriesshape\eu@GreekLowercase@symfont\@nil}%
752 {\global\boolfalse{domkern}}%
753 \fi
754 \ifdef{\eu@family}%
755 {\expandafter\fontfamily\csname eu@\eu@family mathsfon\endcsname\selectfont}%
756 {\relax}%
757 \ifdef{\eu@series}%
758 {\expandafter\ifstrequal\expandafter{\eu@series}{m}%
759 {\mdseries}%
760 {\relax}%
761 \expandafter\ifstrequal\expandafter{\eu@series}{bx}%

```

```

762         {\bfseries}%
763         {\relax}}%
764     {\relax}%
765     \ifdef{\eu@shape}%
766         {\expandafter\ifstrequal\expandafter{\eu@shape}{n}%
767         {\upshape}%
768         {\relax}}%
769         \expandafter\ifstrequal\expandafter{\eu@shape}{it}%
770         {\itshape}%
771         {\relax}}%
772     {\relax}%
773     \global\dimen@XeTeXglyphbounds\@ne\theXeTeXcharglyph`#1\relax
774     \global\dimen@ivXeTeXglyphbounds\thr@@\theXeTeXcharglyph`#1}
775 \dimen@ii 1em
776 \c@eu@\dimen@
777 \c@eu@i\dimen@ii
778 \multiply\c@eu@ 3000\relax
779 \divide\c@eu@i 6\relax
780 \divide\c@eu@\c@eu@i
781 \c@eu@i\c@eu@
782 \ifnum\number\c@eu@ <\eu@minwhitespace
783     \c@eu@\eu@minwhitespace
784     \fi
785     \advance\c@eu@ -\c@eu@i
786     \c@eu@i\c@eu@
787     \divide\c@eu@i\@m
788     \edef\eu@mkern@left{\number\c@eu@i}
789     \multiply\c@eu@i\@m
790     \advance\c@eu@ -\c@eu@i
791     \edef\eu@mkern@left{\eu@mkern@left.\three@digits{\number\c@eu@}\mu}
792     \dimen@\dimen@iv
793     \dimen@ii 1em
794     \c@eu@\dimen@
795     \c@eu@i\dimen@ii
796     \multiply\c@eu@ 3000\relax
797     \divide\c@eu@i 6\relax
798     \divide\c@eu@\c@eu@i
799     \c@eu@i\c@eu@
800     \ifnum\number\c@eu@ <\eu@minwhitespace
801         \c@eu@\eu@minwhitespace
802         \fi
803         \advance\c@eu@ -\c@eu@i
804         \c@eu@i\c@eu@
805         \divide\c@eu@i\@m
806         \edef\eu@mkern@right{\number\c@eu@i}
807         \multiply\c@eu@i\@m
808         \advance\c@eu@ -\c@eu@i
809         \edef\eu@mkern@right{\eu@mkern@right.\three@digits{\number\c@eu@}\mu}
810     \eu@ifbool{domkern}
811         {\mkern\eu@mkern@left#1\mkern\eu@mkern@right}
812     {\#1}}

```

```

813
814 %% Redefine \" and " in maths mode only. Umlaut and quote definitions remain
815 %% in effect in text mode.
816 %%
817 %% " put the " before a character, e.g "$f$, and the character is
818 %% printed with kerns on either side.
819 %% \"...\" Surround a series of adjacent characters, e.g. $\"abcde\"$ is
820 %% equivalent to \"$a\"b\"c\"d\"e$.
821 %% Note that the tokens that \" and \" operate on MUST be characters because
822 %% they are sent through the XeTeX primitive \XeTeXcharglyph as in:
823 %%
824 %% \XeTeXglyphbounds n \the\XeTeXcharglyph`#1
825 %%
826 %% where n = 1,2,3,4 and #1 is the character (If #1 is not a character, then
827 %% \XeTeXcharglyph`#1 doesn't make sense). Higher Level tests are needed to
828 %% avoid this problem.
829 \let\eu@original@quote="
830 \let\eu@original@csquote="\
831 \mathcode`\\"="8000
832 \newcommand\eu@active@quote{%
833   \ifmmode
834     \expandafter\eu@new@quote
835   \else
836     \expandafter\eu@original@quote
837   \fi}
838 \newcommand\eu@active@csquote{%
839   \ifmmode
840     \expandafter\eu@new@csquote
841   \else
842     \expandafter\eu@original@csquote
843   \fi}
844 \begingroup
845   \catcode`\\"=\active
846   \global\let\"=\eu@active@quote
847 \endgroup
848 \let\"=\eu@active@csquote
849 \newcommand\eu@new@quote[1]{\eu@mkern{#1}}
850 \def\eu@new@csquote#1{\eu@mkern{#1}}
851
852 %% Redefine LaTeX 2e kernel macros to do Unicode characters too. Add optional
853 %% fifth argument. This is a list of control sequences which will be let equal
854 %% to the symbol if the symbol exists in the font. If the symbol does not exist
855 %% in the font, then the fifth argument is ignored.
856 %%
857 %% e.g. \XeTeXDeclareMathSymbol#1#2#3#4#5{\mathrel}{font}{"2260}[\neq\ne]
858 \def\XeTeXDeclareMathSymbol#1#2#3#4#5{%
859   \expandafter\in@\csname sym#3\expandafter\endcsname
860   \expandafter{\group@list}%
861   \ifin@
862     \begingroup
863     \if\relax\noexpand#1% is command?

```

```

864     \edef\reserved@a{\noexpand\in@{\string\mathchar}{\meaning#1}}%
865     \reserved@a
866     \ifin@
867         \expandafter\XeTeXset@mathsymbol
868         \csname sym#3\endcsname#1#2{#4}%
869         \@font@info{Redeclaring math symbol \string#1}%
870     \else
871         \expandafter\ifx
872         \csname\expandafter\@gobble\string#1\endcsname
873         \relax
874         \expandafter\XeTeXset@mathsymbol
875         \csname sym#3\endcsname#1#2{#4}%
876     \else
877         \@latex@error{Command ` \string#1' already defined}\@eha
878     \fi
879 \fi
880 \else
881     \expandafter\XeTeXset@mathchar
882     \csname sym#3\endcsname#1#2{#4}%
883 \fi
884 \endgroup
885 \def\XeTeXDeclareMathSymbol@symbol{#1}%
886 \def\XeTeXDeclareMathSymbol@slot{#4}%
887 \expandafter\XeTeXDeclareMathSymbol@option
888 \else
889     \@latex@error{Symbol font `#3' is not defined}\@eha
890     \expandafter\@gobbleoarg
891 \fi}
892 \@onlypreamble\XeTeXDeclareMathSymbol
893 \def\XeTeXset@mathchar#1#2#3#4{%
894     \global\XeTeXmathcode`#2="\mathchar@type#3#1#4\relax}
895 \@onlypreamble\XeTeXset@mathchar
896 \def\XeTeXset@mathsymbol#1#2#3#4{%
897     \global\XeTeXmathchardef#2"\mathchar@type#3#1#4\relax}
898 \@onlypreamble\XeTeXset@mathsymbol
899 \newcommand\@gobbleoarg[1][{}]{
900 \newcommand\XeTeXDeclareMathSymbol@option[1][{}]{%
901     \ifnum\the\XeTeXcharglyph\XeTeXDeclareMathSymbol@slot>\z@
902         \@tfor\i@tfor #1:=\do%
903         {\expandafter\edef\i@tfor{\expandonce\XeTeXDeclareMathSymbol@symbol}}%
904     \fi}
905
906 %% Some shorthands, so the same information isn't typed out more than once
907 %%
908 %% \setallmainfonts[<sets>][<shapes, font features>]{<font name>}
909 %% \setprimaryfont[<shapes, font features>]{<font name>}
910 %% \setallsansfonts[<shapes, font features>]{<font name>}
911 %% \setallmonofonts[<shapes, font features>]{<font name>}
912 \newcommand\setallmainfonts{
913     \@ifnextchar(
914     {\eu@setallmainfonts}

```

```

915     {\eu@setallmainfonts(Digits, Latin, Greek)}}
916 \def\eu@setallmainfonts(#1){
917   \edef\eu@setmathsfont@Set{#1}
918   \@eu@setallmainfonts}
919 \newcommand\eu@setallmainfonts[2][[]]{
920   \setmainfont[#1]{#2}
921   \setmathsfont(\eu@setmathsfont@Set)[#1]{#2}
922   \setmathrm[#1]{#2}}
923 \newcommand\setprimaryfont{\setallmainfonts(Digits, Latin)}
924 \newcommand\setallsansfonts[2][[]]{
925   \setsansfont[#1]{#2}
926   \setmathsf[#1]{#2}}
927 \newcommand\setallmonofonts[2][[]]{
928   \setmonofont[#1]{#2}
929   \setmathtt[#1]{#2}}
930
931 %% Set the particular mathematics alphabets
932 \ernewcommand\setmathrm[2][[]]{
933   \zf@fontspec{#1}{#2}
934   \let\eu@mathrm\zf@family
935   \DeclareMathAlphabet{\mathrm}{EU1}{\eu@mathrm}{m}{n}
936   \SetMathAlphabet{\mathrm}{bold}{EU1}{\eu@mathrm}{bx}{n}
937   \DeclareMathAlphabet{\mathit}{EU1}{\eu@mathrm}{m}{it}
938   \DeclareMathAlphabet{\mathbf}{EU1}{\eu@mathrm}{bx}{n}
939   \SetMathAlphabet{\mathit}{bold}{EU1}{\eu@mathrm}{bx}{it}
940   \DeclareSymbolFont{Operators:m:n}{EU1}{\eu@mathrm}{m}{n}
941   \def\operator@font{\expandafter\mathgroup\csname symOperators:m:n\endcsname}}
942 \ernewcommand\setmathcal[2][[]]{
943   \zf@fontspec{#1}{#2}
944   \let\eu@mathcal\zf@family
945   \DeclareMathAlphabet{\mathcal}{EU1}{\eu@mathcal}{m}{n}}
946 \ernewcommand\setmathsf[2][[]]{
947   \zf@fontspec{#1}{#2}
948   \let\eu@mathsf\zf@family
949   \DeclareMathAlphabet{\mathsf}{EU1}{\eu@mathsf}{m}{n}
950   \SetMathAlphabet{\mathsf}{bold}{EU1}{\eu@mathsf}{bx}{n}}
951 \ernewcommand\setmathtt[2][[]]{
952   \zf@fontspec{#1}{#2}
953   \let\eu@mathtt\zf@family
954   \DeclareMathAlphabet{\mathtt}{EU1}{\eu@mathtt}{m}{n}}
955 \ernewcommand\setmathfrak[2][[]]{
956   \zf@fontspec{#1}{#2}
957   \let\eu@mathfrak\zf@family
958   \DeclareMathAlphabet{\mathfrak}{EU1}{\eu@mathfrak}{m}{n}}
959 \ernewcommand\setmathbb[2][[]]{
960   \zf@fontspec{#1}{#2}
961   \let\eu@mathbb\zf@family
962   \DeclareMathAlphabet{\mathbb}{EU1}{\eu@mathbb}{m}{n}}
963
964 %% If amsmath is Loaded, it must be Loaded before mathspec. Checking for its
965 %% existence \AtBeginDocument is too late because the damage is already done

```

```

966 %% (It attempted to define \varTheta when mathspec already defined it). Or
967 %% should I delay the definition of \varTheta until \AtBeginDocument?
968 \let\original@RequirePackage\RequirePackage
969 \renewcommand\RequirePackage[2][]{
970   \ifstrequal{#2}{amsmath}
971     {\PackageError{mathspec}
972       {\`amsmath' must be loaded earlier than \`mathspec'}}
973     {Edit the document so that \`amsmath' is required earlier than \`mathspec'.}}
974   {\relax}
975   \original@RequirePackage[#1]{#2}}
976 \@onlypreamble\RequirePackage
977 \let\usepackage\RequirePackage
978 \@onlypreamble\usepackage
979
980 %% Any font changes that mathspec has done are reset by LaTeX ε2 at
981 %% \begin{document} using \process@table.
982
983 \endinput
984
985 %% © Andrew Gilbert Moschou 2009
986 %%
987 %% This work may be distributed and/or modified under the
988 %% conditions of the LaTeX Project Public License, either version 1.3c
989 %% of this license or (at your option) any later version.
990 %% The latest version of this license is in:
991 %%
992 %% http://www.latex-project.org/lppl.txt
993 %%
994 %% and version 1.3c or later is part of all recent distributions of LaTeX.
995 %%
996 %% This work has the LPPL maintenance status ‘maintained.’
997 %%
998 %% The Current Maintainer of this work is Andrew Gilbert Moschou.
999 %%
1000 %% This work consists of the files mathspec.sty and mathsepc4.tex.

```

Exactly 1000 lines! How about that?

9 LICENSE

© Andrew Gilbert Moschou 2009

This work may be distributed and/or modified under the conditions of the L^AT_EX Project Public License, either version 1.3c of this license or (at your option) any later version. The latest version of this license is in:

<http://www.latex-project.org/lppl.txt>

and version 1.3c or later is part of all recent distributions of L^AT_EX.

This work has the LPPL maintenance status ‘maintained’.

The Current Maintainer of this work is Andrew Gilbert Moschou.

This work consists of the files `mathspec.sty` and `mathspec.tex`.