A package for using the MathTime and MathTimePLUS fonts*

Drahoslav Lím
DrahoslavL@seznam.cz

2002/04/05

Abstract

`mt11p` is a package to relatively painlessly get access to everything available in the MathTime and MathTimePLUS commercial math fonts in LaTeX 2ε. It is compatible with the AMS packages. This package is self--contained and only the Type 1 font programs need to be copied off of the distribution media.

Contents

*This file has version number 0.4a, last revised 1997/01/27. This file has no connection with the `mathtime` package commisioned by Y&Y.

1 Introduction

The use of Computer Modern fonts for equations has been one of the tell-tale signs indicating a document was typeset by TeX. The reasons for this are that there are no good freely available substitute fonts, and that even if there were, changing TeX's math fonts is a tedious and somewhat cryptic exercise. The first problem can be overcome by purchasing math fonts such as MathTime or Lucida. This package is one of the ways overcoming the second, though not the only one. It provides everything *except* the fonts themselves. (You have shell out money and buy those, SORRY).

It is assumed you are running TeX 3.x and a recent release of LaTeX, and that you have the standard 'resident' POSTSCRIPT fonts installed for use as text fonts in TeX ("PSNFSS").

2 Installation

This package is useless unless you buy the fonts called *MathTime* and *MathTimePLUS* from Y&Y; the address is listed in Sec. 5.

Prerequisites The "8r-based" version of PSNFSS must be installed on your system to use this package, because the VFs refer to the 8r "raw" fonts. Search on CTAN for the string "psnfss", consult the *LaTeX Companion*, and/or the LaTeX 2_ε distribution on how to do this. You can check if your system already has the relevant files by searching for the TeX metric file "`ptmr8r.tfm`" in the `tfm` file area, and for the file "`T1ptm.fd`" (or maybe "`t1ptm.fd`") in the TeX input file area; if both are present you are probably OK and everything will work.

This `dtx` file should have come with several sets of accompanying files, whose installation should be routine:

Run TeX on `mt11p.ins`; that will generate the package "style" file, `mt11p.sty`. Copy that file to the TeX input file area. Run LaTeX on `mt11p.drv` to get commented source code (that's what you're reading now).

The **`.tfm` metric files** should be copied to the area where TeX and the `dvi` driver program search for font metrics. The **`.vf` definition files** should be copied to the area where the `dvi` driver program searches for virtual font definitions. Note that the metric files supplied are *not* identical to the ones distributed with the MathTime fonts.

The file `mt11p.map` should be "passed" to the `dvi` driver program to tell it that the MathTime fonts are downloadable POSTSCRIPT fonts and what the relevant invocation commands are, or perhaps the file should be passed to a POSTSCRIPT rasterizer ("PS–to–PK") program if the `dvi` driver must use PK bitmaps. For Rokicki's `dvips` driver, the contents of `mt11p.map` can be appended to the "`psfonts.map`" file.

Copy the Type 1 font "programs"—files with extension `pfa` or `pfb`—from Y&Y's distribution media to the area where your `dvi` driver or rasterizer searches for downloadable fonts. Some fonts may be provided in both `pfa` and `pfb` formats; just pick one, most systems can deal with either. (The `pfb` format is more compact, albeit binary. The `dvi` driver can probably convert `.pfb` to `.pfa` "on the fly".) There are two versions of the normal weight symbol font, `mtsy` and `mtsyn`. The difference is minor, but you might as well use the newer `mtsyn`. See the paragraph called "Symbols" in the documented source code.[1]

The names and types (`pfa`/`pfb`) of the downloadable files in `mt11p.map` should be made consistent with the actual names and types under which the Type 1 font programs are stored on your system. Either rename and/or convert the Type 1 fonts, or simply edit the contents of `mt11p.map`; beware of case sensitivity.

The `dvi` driver program will probably not need the `afm` files from the distribution medea, but a "PS–to–PK" rasterizer might; consult the rasterizer documentation. Note that `afm` files are ASCII files, convert line endings as appropriate.

At least for this package, no other files from the distribution medea are required.

2.1 Previewing

Whether or not you can preview `dvi` files which use MathTime fonts depends on your setup. If you can preview `dvi` files containing other PostScript fonts, chances are you can preview with MathTime as well, perhaps after running some utility to generate bitmaps, resolve `VF` references, etc. Otherwise, it is possible to view the final PostScript output of the `dvi` driver on practically all platforms using viewers based on the GhostScript rasterizer, or perhaps using DisplayPostScript.

3 Usage & Package Options

To switch to Times text and math, it should suffice to say `\usepackage{mt11p}` in the preamble. The oder of packages should not matter, with one exception: `mt11p` should be loaded after $\mathcal{AMS}$ math packages. Any reasonably cleanly written LaTeX input file will then be typeset with MathTime being used for all standard mathematical symbols that would have otherwise been typeset in Computer Modern. There are two notable exceptions, see Sec. 3.2.

Expect a few dozen lines of warnings as this package is being read, and a notice about substituted fonts at the end of the run. I think it is better to live with these NFSS warnings rather than do some hacks to turn them off.

At the end of the run expect a warning about size substitutions if the `resize` option was in effect (it is by default). They are are due to the Computer Modern fonts usually having a set of sizes available rather than being declared scalable in the `.fd` file, i.e., a `\DeclareFontShape` argument of something like `<10><10.95><12><14.4>` rather than `<->`. Whether or not something should be done about this depends on what Computer Modern glyphs were used. It may be the case that, in fact, none were—the warnings come from NFSS having loaded fonts for. e.g., `\oldstylenums` for math, eventhough you never actually used them. You can change the `.fd` files if you system is capable of generating bitmaps on-the-fly or if you're using PostScript versions of the Computer Modern fonts. Consult the *LaTeX Companion* or the standard documentation file `fntguide.tex`.

[1] Run LaTeX on the file `mt11p.drv` to get the commented source code; it follows after Sec. 5.

3.1 List of options

This packages has lots of options controlling some detailed aspects of font and math character handling. In general, use the defaults, unless you have some idea of what the implications of a particular option are.

Options in effect by default are indicated by a double dagger (‡).

‡T1 reset the `\encodingdefault` (applicable to text fonts and some math operators, symbols and accents) to `T1`.

OT1 reset the `\encodingdefault` to the archaic `OT1` encoding; upper case Greek letters are never taken from the text font, so the only reasons to do this are human inertia and legacy code.

noenc assume the current `\encodingdefault` is the desired one; see the commented source code if you use something other than `T1` or `OT1`.

‡opsafe use symbols and accents from the `operators` math family only when it is safe to do so.

opnone do not use the `operators` math family for accents or symbols, except when this can't be helped; if `\encodingdefault` is not recognized, this option is auto-invoked.

opmax take as much from the `operators` math family as possible. Doing so can produce problematic output in some circumstances, but it will make the maximum use of printer resident fonts; don't use this option unless you know why you are doing so. See the paragraph "Symbols from either `operators` or from elsewhere" in the documented source code.

‡opaccents take math accents from the `operators` math family, and make them follow math alphabet changes where possible; see Sec. 3.3.

symaccents take math accents from the a fixed (within a math version) font; if `\encodingdefault` is not recognized, this option is auto-invoked; see Sec. 3.3.

‡mtcal make the MathTıme script font the math alphabet `\mathcal` and make `\mathscr` and synonym of `\mathcal`; see Sec. 3.4.

cmcal make the MathTıme script font the math alphabet `\mathscr` and the Computer Modern symbol font the `\mathcal` alphabet; see Sec. 3.4.

nomtms use the Computer Modern symbol font as the `\mathcal` alphabet and do not load the MathTıme script font at all; see Sec. 3.4.

‡dohbar overwrite any existing definition of the `\hbar` ($\hbar$) symbol with one synthesized from an "h" and a bar accent. The definition here is actually rather decent, unlike the simple one usually used. (At least try it before resorting to the glyph in the $\mathcal{AMS}$ fonts. . .)

nohbar do nothing with the `\hbar` definition, retain the existing one (if any); e.g., the $\mathcal{AMS}$ fonts have `\hbar` as a single Computer Modern-like glyph.

‡subs the `bold` and `heavy` math versions in principle need some 'ultra' and 'poster' weight fonts; usually no such fonts are available, so define substitutes.

nosubs assume suitable fonts are available and will be loaded, e.g., using a configuration file `mt11p.cfg`; do not declare any font substitutions. If should be apparent from the commented source code or from `mt11p.sty` what *family*/*series*/*shape* NFSS declaration the extra fonts should get. Search for the string "SUBS" in the `.dtx` or `.sty` file.

‡do8r use the raw `8r` fonts for the four symbols §, ¶, †, ‡. The math italic fonts use the italic `8r` raw fonts, so it's not unreasonable to assume the upright ones are also available; no support for using the `8r` encoding need be installed on the system, though we make use of it if it is. `8r` is not used for anything other than these four symbols; see Sec. 3.5.

no8r do not use the raw fonts, and take the four symbols §, ¶, †, ‡ from a Computer Modern font; see Sec. 3.5.

nodag don't touch the definitions of §, ¶, †, ‡ at all; assume suitable definitions have been or will be supplied, e.g., as part of a `TS1` setup. See Sec. 3.5.

‡bsy load the $\mathcal{AMS}$ packages needed to define the `\boldsymbol` macro, and define analogous `\heavysymbol` and `\normalsymbols` macros.

nobsy don't load the $\mathcal{AMS}$ packages for `\boldsymbol` and define `\boldsymbol` and `\heavysymbol` to produce messages and behave as no-ops.

‡mathgr get upright Greek letters as the math alphabet `\mathgr`; see Sec. 3.9.

nomathgr do not load the math alphabet `\mathgr`; this is only useful if you are running out of math families, in which case you can use `\greekshape` in an `\hbox` with the obvious limitations. See ec. 3.9.

‡mathOS get oldstyle digits as a math alphabet; Computer Modern oldstyle digits are always available in text.

nomathOS do not load Computer Modern oldstyle digits as a math alphabet. See the `nomathgr` option.

‡resize change the subscript and text font sizing to better account for the fact that we are using linearly (not optically) scaled fonts.

noresize don't touch the math subscript and text font sizing.

‡activesb to improve the spacing in certain subscript situations, the subscript character "`_`" (an underscore) is made active; this *might* break some macros; see Sec. 3.8

noactivesb don't do anything with the underscore.

‡**times** assume the package (style file) which changes the roman, sans and typewriter text families to "Times & Helvetica & Courier" is called "`times.sty`"; this package is loaded via `\RequirePackage`.

* any other option will be interpreted as the name of the style file to use instead of `times`; e.g., the option "`btdutch`" will attempt to 'require' the `btdutch` package instead of the `times` package.

3.2 Interaction with $\mathcal{AMS}$ macros and other math packages

This package does nothing terribly anti-social and can co-exist with $\mathcal{AMS}$ packages, e.g., `amsmath`, `amsfonts`, etc., provided those packages are read *before* (i.e., prior to) reading this package. Some types of macros are hardwired to Computer Modern fonts, e.g., the "$\mathcal{AMS}$" logo and would have to be redefined. The fonts `msam` and `msbm` will, or course, still be used for the more uncommon math symbols. See below about math accents.

The default TeX math setup treats upper case Greek letters as mathematical "letters" and lower case Greek letters as mathematical "symbols". Hence `\mathbf{\Gamma\gamma}` produces a bold upper case $\mathbf{\Gamma}$, but does not affect the lower case γ; similarly for `\mathit`. This package uses the same conventions as the $\mathcal{AMS}$ macros: names like "`\varGamma`" are used for italic upper case Greek ($\varGamma$), and math version switching is needed (`\boldsymbol`) to get bold Greek. The 'legacy' input code `\mathbf{\Delta}\mathit{\Gamma}` ("$\mathbf{\Delta}\mathit{\Gamma}$") *does not work* with the $\mathcal{AMS}$ LaTeX macros nor with this package. See Sec. 3.3 about accents in math.

It is likely that other math font related packages can co-exist with `mt11p`, particularly if they merely add a math alphabet or symbol font, e.g., `mathbbol`, `stmaryrd` or `wasysym`. The chances of running out of math symbol font families are slim unless you also load the `oldlfont` package. Note that the $\mathcal{AMS}$ symbol packages need two math families. Of course it makes no sense to use `mt11p` together with another complete math setup package like `euler`.

One package that *will* clash with `mt11p` is `yhmath`. If you load that package after `mt11p`, all `largesymbols` in all math versions will be taken from the fixed size font `yhcmex10`; loading `yhmath` before `mt11p` will break `yhmath`'s wide math accents. If you only need the extra-wide accents, change all instances of the word "largesymbols" in the file `yhmath.sty` to something else, e.g., YHlargesymbols, and input that altered file. If you want the extra delimiter sizes, you're on your own, but see the paragraph "Declaring additional math delimiters" towards the end of the commented source code.

3.3 Math Accents

MathTıme provides an additional accent called `\widebar` or `\overbar`. It's about 1em long. If the $\mathcal{AMS}$ macros `\Hat`, etc., are defined, we also define the similar `\Overbar` and `\Widebar`.

The Computer Modern layout causes some encoding-related problems with math accents and math alphabets. NFSS has hooks to fix the problems, but does not actually do it; the $\mathcal{AMS}$ package `amsmath` does fix them by redefining most math accents.

`mt11p` incorporates fixes similar to the $\mathcal{AMS}$ ones, and overwrites any existing math accent definitions (it pretty much has to), so that any combination of math alphabets and accents (and accents options: `opaccents` or `symaccents`) will output the expected glyphs. Some curious combinations will, however, lead to odd-looking accent positioning. In a construct such as `\mathit{\hat{x}}` the accent positioning will always be "wrong", because the accent is a *text* accent—the (text) italic font implied by `\mathit` doesn't have the appropriate font metrics because it's assumed that `\mathit` will be used when a group of letters (a 'word') is to be treated as a math symbol, e.g., $x_{initial}$ or $\widehat{final}$. However, "x" is a mathematical symbol by itself. If you want an italic hat on the symbol x you have to say `\mathit{\hat{\mathnormal{x}}` (assuming it's a big deal to have an italic `\hat`...). Even then the spacing will not be optimal and will usually require some fiddling with the `\skew` macro.

3.4 Script and calligraphy math fonts

This package makes the distinction between two "calligraphy" math alphabets: *calligraphy* and *script*. This is necessitated by the fact that the Computer Modern fonts only have upper case "calligraphy" letters, and that you perhaps may not like the MathTime `mtms*` fonts or that you may want to use letters from both fonts.

Irrespective of the options in use, the two math alphabet commands `\mathcal` and `\mathscr` will be defined. (The old "`\mit`" is not defined.)

The default (`mtcal`) is to make both alphabets refer to a `mtms*` font. The variant forms of some letters are available with the commands `\varA`, `\varE`, `\varG`, `\varvarG`, `\varI`, `\varL`, `\varQ`, `\varS`, `\varr`, `\varz`, `\iscript` and `\jscript`. (`\i` and `\j` are invalid in math mode and `\imath` and `\jmath` cannot be redefined.) Both upper and lower case letters will work in `\mathcal`.

Another possibility, `cmcal`, is to have both Computer Modern and MathTime fonts available; `\mathscr` will be the latter, and `\mathcal` the former. `\mathcal` will then not have lower case letters.

A third setting, `nomtms`, avoids using the MathTime `mtms*` fonts entirely. `\mathcal` is then the Computer Modern symbols font, and `\mathscr` gives a warning before calling `\mathcal`.

A certain amount of "reasonableness" is assumed with respect to the usage of "calligraphic" alphabets. The `mtcal` option does not give any warning about the fact that `\mathcal` and `\mathscr` are identical. With the `cmcal` or `nomtms` options in effect you will get warnings if you try to use a `\var`*X* letter in `\mathcal`, but `\mathcal` will never try to check its argument to verify that it consists of *upper* case letters only, even when no font with "calligraphy" lower case letters is available. Consider using `\mathcal` when the argument is known to contain nothing but upper case letters and use `\mathscr` when using lower case letters as well; that way you will at least get a warning pertaining to script math alphabets if the options are set to avoid the use of `mtms*` or if the document is reset without MathTime fonts.

The MathTime manuals describe a `\script` macro; `mt11p` defines an equivalent macro. Note that `\script` should only be followed by white space, an optional "`*`", and then a single letter, with no space between the asterisk and the letter, e.g., `\script␣*AB` is equivalent to `\mathscr{\varA}B`, *not* `\mathscr{AB}`, and `\script*␣AB` is an error.

3.5 The symbols †, ‡, §, and ¶

The standard LaTeX setup takes these symbols, in both text and math from the Computer Modern symbols font. The MathTime symbols font does not have these glyphs, so something must be done to have them available for footnote markers, etc.

There is a fairly standard PostScript text symbol font setup, which uses the `TS1` encoding, but it seems to not (yet) be in general use.

This package offers three relevant options. The option `nodag` does absolutly nothing with the four symbols, but this can only be used if they've already been or are about to be redefined (perhaps in the `mt11p.cfg` file), because the default definitions assume the use of Computer Modern fonts and *will* produce incorrect glyphs after switching to MathTime.

The default option `do8r` takes these four symbols (and *only* these four symbols) from the "raw" `8r` encoded fonts, upon which the virtual `T1` and `OT1` PostScript fonts are built. If `.fd` files or an `8r.sty` are available, we use them; if they're not, we supply just enough code to be able to use the symbols in NFSS. It is not unreasonable to do so, because the `8r` metric file `ptmri8r.tfm` must be on the system (the `mtmi.vf` virtual font uses it), so the upright font `ptmr8r.tfm` is probably available too. Note that `8r` is an "unsupported" encoding.

The option `no8r` overwrites any existing definition and forces the use of Computer Modern for these four symbols (without affecting anything else) and avoids the direct use of `8r` raw fonts.

3.6 "New" symbols

MathTime has a few mathematical symbols that aren't available in Computer Modern. Here is the list, with alternate names in parentheses: `\overbar` (`\widebar`), `\openclubsuit`, `\shadedclubsuit`, `\openspadesuit`, `\shadedspadesuit`, `\triangleleft`, `\triangleright`, `\cupprod`, `\capprod`, `\varcirc` (`\comp`).

3.7 Avoiding the use of bitmap fonts

Often the point of buying the MathTime fonts is not just to get something other than Computer Modern math, but also to avoid the use of bitmap (Type 3) fonts and use scalable (Type 1) fonts nstead. The default option are set so that this condition is satisfied, as long as you don't use glyphs that are not available in either the resident fonts nor in MathTime, i.e., oldstyle digits, or you force the use of Computer Modern or other MetaFont fonts in some situations (the `cmcal`, `nomtms`, or `no8r` options, $\mathcal{AMS}$ packages, etc.) and you do not have substitute outline fonts. See the MathTime licence agreement about "exporting" files containing outline fonts.

3.8 Subscript spacing

The sidebearings of math italic letters with descenders are such that a spacing adjustment is required in certain contexts, in particular in the very common case of j subscripts. The MathTime distribution suggests using an active subscript character, which checks if the first thing in a subscript is one of the letters `f`, `j` or `p`, and inserts a small negative kern if is it. The macro `\jadjust` can be used if the correction needs to be done by hand; see Appendix B.2 of the MathTime *Reference Manual*.

Changing category codes is akin to Russian roulette: we can try to make the magazine larger, but it will still hold a bullet *somewhere*. It is unreasonable to assume that other people's packages will take steps to work even when the `underscore` is a macro. `underscore`s can crop up in font names or reference labels, for example.

The macro we use is slightly different from the original one in the MathTime distribution. The `\catcode` of the `underscore` ends up being 12, i.e., "other" rather than 13 ("active"), which is much more forgiving. We rely on the fact that the `underscore`'s `\mathcode` is `"8000` to get subscripts in math mode. This seems to be OK when used in most contexts. In text mode this will produce whatever the current font has in slot `` `\_ ``, but you shouldn't be using a 'naked' `underscore` in text anyway...

Nothing is done with the `underscore` unless its `\catcode` is "subscript" (8), *and* its `\mathcode` is `"8000`; if both conditions are not fulfilled, someone has presumably already fiddled with the `underscore`, so we don't touch it.

The option `noactivesb` keeps the underscore as it was, in order to quickly ascertain whether the active `underscore` is braking some macros. Reports of such cases are welcome.

3.9 Upright Greek lower case letters

MathTime contains an upright Greek letters (text) which are used in some disciplines to designate units or particles. Such letters are normally not used as math *symbols*. We provide three relevant commands: **`\greekshape`**, which is a font switch like `\itshape`), **`\textgr`**, which takes an argument similarly to `\textit`, and the math alphabet command **`\mathgr`**.

Alternate forms of some letters can be obtained as ligatures—see Section 8 of the *MathTime*PLUS reference manual. However, in math (`\mathgr`) it is not possible to use the ligatures with the asterisk character (`*`) suggested in the manual, because the asterisk will not be interpreted as a character from the same font as the Greek letters. The TFM metric files accompanying this package contain extra ligatures which produce the variants: in lieu of `O*` or `o*`, write `vO` or `vo`, respectively; in place of `e*` write `vev`.

This package does not define any `\gr` font switch. `\greekshape` performs the same function in text; use the `\mathgr` form in math.

MathTimePLUS comes with two "hybrid" fonts, `mtmub` and `mtmuh`, which are math `letters` fonts with upright (instead of sloping) lower case Greek letters. The two fonts contain both upright and italic *upper case* Greek letter. For reasons listed in the reference manual, there is no font `mtmu`. This package comes with a `mtmu` font, created as a `VF` using `mtmi` and `mtgu`, in case you decide to use such a font after all.

See the commented source about declaring upright Greek letters as math *symbols* and about using `mtgu`.

3.10 Oldstyle Digits

None of the "standard 35" PostScript fonts has oldstyle digits (1234567890) and neither do any of the MathTime fonts. The standard TeX math `letters` font has such digits, and some packages make use of the glyphs. `mt11p` defines an `\oldstylenums` macro which is essentially equivalent to the one provided in LaTeX, but note that the Computer Modern oldstyle digits don't go well with Times text fonts. A `mt11p.cfg` file or a subsequent package could redefine the macro to use a more suitable font.

3.11 Mixing math versions

Like both the standard TeX math setup and the $\mathcal{AMS}$ math setup, we assume that it will be relatively rare to require bold (unbold) *symbols*[2] in a formula set in the unbold (bold) math version. We load the $\mathcal{AMS}$ `amsbsy` package which defines a `\boldsymbol` macro, that sets its argument using the bold math version, and define corresponding corresponding `\heavysymbol` and `\normalsymbol` macros.

However, it's awkward to use the `\boldsymbol`, etc., macros in situations where alignment is critical (accents) or for large delimiters. For those cases one could declare additional symbol fonts and define things like `\boldalpha`, `\boldhat` or `\Lheavybrace`, and then an input text like

`\[\left\Lheavybrace 2\boldalpha` ⟨*some big formula*⟩ `\right\Rheavybrace\]`

would do what would otherwise take unreadable input code (`\vphantom` constructs) to produce. This requires some knowledge of the details of TeX's math workings, but nothing too esoteric. For a simple example the commented source code. Such a setup could be placed in `mt11p.cfg`. Consult also the standard LaTeX documentation file `fntguide.tex`. Note that the $\mathcal{AMS}$ macros do not provide such a setup, because it's somewhat rare to actually need it, but it can be done.

3.12 Local Configuration

After setting everything up but before 'undefining' internal auxiliary macros and flags, and before changing the subscript sizing this package inputs the file `mt11p.cfg`, if it can find one. Further definitions (bold or upright Greek letters as symbols, etc.) could be placed in such a file.

4 Bugs

Any reports about bugs, suspected bugs or package clashes are welcome provided you also send just enough TeX code to reproduce the bug and your TeX `.log` file—use the e-mail address given on the title page of this document and at various places in the files. Report of *usage* are also welcome`;-)`.

[2]The arguments of `\mathbf` are not considered 'symbols'. The input text `\mathbf{a}` produces a bold *upright* "**a**" not the bold math symbol "$\boldsymbol{a}$".

5 Acknowledgment & Misc.

This section has been put here, in case you don't print out the commented source code.

Thanks to the authors and maintainers of LaTeX, and particularly to the authors of NFSS.

Portions of this package were stolen from the `mathtimy` package by Aloysius Helminck, and from the TeX files provided with the MathTime distribution.

The "operative" part of this package was written before the release of the `mathtime` package by Y&Y.

MathTime is a trademark of Publish or Perish, Inc.

A bunch of other names are trademarks or registered trademarks, as everybody knows…

The contact address for Y&Y is

Y&Y, Inc.
106 Indian Hill
Carlisle, MA 01741
USA
http://www.yandy.com

The author has no association with Y&Y, Inc., other than having purchased the MathTime fonts.

---

6 Implementation

We start by the usual identifications. We need a few commands not present in early versions of NFSS 2, so check for the date.

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{mt11p}%
        [\filedate: setup\space for\space Times\space
         +\space MathTime\space +\space MathTimePlus]
```

Before we do anything real, check the catcodes of a few characters in case some fanatic changed them already. These are the characters I've seen activated in various packages.

We hope the the single left quote wasn't…

```
\chardef\mt@quotedbl    \catcode`\"
\chardef\mt@quotesinglr\catcode`\'
\chardef\mt@colon       \catcode`\:
\chardef\mt@question    \catcode`\?
\chardef\mt@semicolon   \catcode`\;
\chardef\mt@exclam      \catcode`\!
\@makeother\"
\@makeother\'
\@makeother\:
\@makeother\?
\@makeother\;
\@makeother\!
```

6.1 Option Declarations

We will need the option setting in conditionals at various points, so use integers (`\chardef`s) as flags.

There are three options relevant to text font encodings: either make the default `T1`, or make the default `OT1`, or don't touch `\encodingdefault`. The upright "roman" (`\rmdefault`) is used for the `operators` math family, so the choice of text encoding has implications for math. This package can handle (in as much as it needs to) with any of the following: `T1`, `OT1` or a superset thereof, `8r` and `8a`. See the commented source code if you need more information, Sec. 6.8.

```
\DeclareOption{T1}{\def\mt@enc{\def\encodingdefault{T1}}}
\DeclareOption{OT1}{\def\mt@enc{\def\encodingdefault{OT1}}}
\DeclareOption{noenc}{\let\mt@enc\relax}
```

Next the options for dealing with what math symbols to take from the `operator` family. Either take everything from elsewhere, or take 'safe' things from `operators`, or take as much as possible from `operators`, which could be dangerous in some circumstances; see Sec. 6.8.

```
\DeclareOption{opnone}{\chardef\mt@ops0␣}
\DeclareOption{opsafe}{\chardef\mt@ops1␣}
\DeclareOption{opmax}{\chardef\mt@ops2␣}
```

Accents can either come from `operators` and follow math alphabets (usual case), or be always taken from `symbols`.

```
\DeclareOption{opaccents}{\chardef\mt@accs0␣}
\DeclareOption{symaccents}{\chardef\mt@accs1␣}
```

We need options to control the handling of script/calligraphy fonts.

```
\DeclareOption{mtcal}{\chardef\mt@cal0␣}
\DeclareOption{cmcal}{\chardef\mt@cal1␣}
\DeclareOption{nomtms}{\chardef\mt@cal2␣}
```

In order to cooperate with $\mathcal{AMS}$ macros, we provide a switch to *not* define `\hbar`, i.e., retain the present definition (if any). You may want to use this to keep `\hbar` pointing to the glyph in the `msbm` fonts, rather than use the faked one we define here, though then you'll have a Computer Modern "h" which looks too light among Times, and the one provided herein isn't too shabby...

```
\DeclareOption{dohbar}{\chardef\mt@hbar1␣}
\DeclareOption{nohbar}{\chardef\mt@hbar0␣}
```

The `bold` and `heavy` math versions will probably need some font substitutions, but if you have the additional fonts, you can skip the substitutions contained herein.

```
\DeclareOption{subs}{\chardef\mt@sub1␣}
\DeclareOption{nosubs}{\chardef\mt@sub0␣}
```

For historical reasons the `\dag`, `\ddag`, `\S` and `\P` symbols have to be dealt with. The `mtmi` virtual fonts use `ptmi8r` raw fonts, so at least the TFM files should be available, and we include just enough code to use them for these four symbols. Provide an option to explicitly avoid using `8r` for these symbols; in that case we'll have to use `OMS`. A subsequent package could redefine them to use the `TS1` encoding or something. Also provide an option that does nothing at all with any of the relevant definitions.

```
\DeclareOption{no8r}{\chardef\mt@eightR0␣}
```

```
\DeclareOption{do8r}{\chardef\mt@eightR1␣}
\DeclareOption{nodag}{\chardef\mt@eightR2␣}
```

It may be convenient to get the `\boldsymbol` macro from the $\mathcal{AMS}$ style files.

```
\DeclareOption{nobsy}{\chardef\mt@bsy0␣}
\DeclareOption{bsy}{\chardef\mt@bsy1␣}
```

Before exiting this package should reset the subscript sizing, because we're using linearly scaled fonts. Provide an option to switch that off.

```
\DeclareOption{resize}{\def\mt@exit{\MathTimeSizes}}
\DeclareOption{noresize}{\let\mt@exit\relax}
```

This is mainly to test if the active underscore is causing any problems.

```
\DeclareOption{activesb}{\chardef\mt@sb1␣}
\DeclareOption{noactivesb}{\chardef\mt@sb0␣}
```

Now a few options to save math families. The default leaves room for four, which may be too little if you also load $\mathcal{AMS}$, etc.

```
\DeclareOption{nomathgr}{\chardef\mt@mgr0␣}
\DeclareOption{mathgr}{\chardef\mt@mgr1␣}
\DeclareOption{nomathOS}{\chardef\mt@osdg0␣}
\DeclareOption{mathOS}{\chardef\mt@osdg1␣}
```

The package to get "Times/Helvetica/Courier" might have some wiered name:

```
\DeclareOption{times}{\def\times@style{times}}
\DeclareOption*{\edef\times@style{\CurrentOption}}
```

The defaults are

- `T1` encoding;
- assume the text font package is called `times.sty`;
- use `operators` for safe glyphs and accents;
- use `mtms` as the `\mathcal` font;
- declare substitutions for the bold and heavy math versions;
- overwrite any existing definition of `\hbar`;
- use `8r` for `\dag`, etc.;
- fetch the `\boldsymbol` $\mathcal{AMS}$ macro;
- get upright Greek and oldstyle digits;
- switch to different subscript sizing; and
- use the active underscore character.

Then read the option list.

```
\ExecuteOptions{T1,times,opsafe,opaccents,mtcal,%
                 dohbar,subs,do8r,bsy,resize,mathgr,mathOS,activesb}
\ProcessOptions*
```

6.2 Text Encoding

We have to know that the text `\encodingdefault` is, because accents in math mode may get taken from the `operators` font. We make `\encodingdefault` either `T1` (default, and preferred), or `OT1`, or do nothing with `\encodingdefault`.

```
\mt@enc
```

6.3 Text font setup

Here we fetch "Times + Helvetica + Courier" using the `times` package, or whatever we were told to use, or just do it by hand. On exit `\rmdefault`, `\sfdefault` and `\ttdefault` should be OK.

```
\IfFileExists{\times@style.sty}%
       {\RequirePackage{\times@style}}%
       {\PackageWarning{mt11p}{No file ‘\times@style.sty’ found.%
                              \MessageBreak
                              Guessing what the text
                              font familes \MessageBreak
                              are called}
        \renewcommand{\sfdefault}{phv}
        \renewcommand{\rmdefault}{ptm}
        \renewcommand{\ttdefault}{pcr}
        \renewcommand{\bfdefault}{b}}
```

6.4 Font Sizing

The documentation accompanying MathTime suggests different subscript sizing. We do it the same way the `mathtimy` package does it, by redefining things like `\@ixpt` to actually be something other than what the name implies. Then we reset all math sizes, from `\tiny` to `\Huge`. This is the sizing used in `mathtimy`, which appeares to be taken from the files in the MathTime distribution. This macro gets called just before we exit. Switching between MathTime sizing and the default Computer Modern sizing on the fly makes little sense, so we don't try to preserve the original settings. The standard sizes are, in sequence, 5, 6, 7, 8, 9, 10, 10.95, 12, 14.4, 17.28, 20.74 and 24.88 points, i.e., we decrease the 'spread' somewhat.

```
\def\MathTimeSizes{%
  \def\@vpt{6.0}%
  \def\@vipt{6.8}%
  \def\@viipt{7.6}%
  \def\@viiipt{8.4}%
  \def\@ixpt{9.2}%
  \def\@xpt{10.0}%
  \def\@xipt{10.8}%
  \def\@xiipt{11.6}%
  \def\@xivpt{13.2}%
  \def\@xviipt{15.6}%
  \def\@xxpt{18.0}%
  \def\@xxvpt{22.0}%
  \DeclareMathSizes\@vpt   \@vpt   \@vpt   \@vpt
  \DeclareMathSizes\@vipt  \@vipt  \@vpt   \@vpt
  \DeclareMathSizes\@viipt \@viipt \@vpt   \@vpt
  \DeclareMathSizes\@viiipt\@viiipt\@vipt  \@vpt
```

```
\DeclareMathSizes\@ixpt  \@ixpt  \@vipt  \@vpt
\DeclareMathSizes\@xpt   \@xpt   \@viipt \@vpt
\DeclareMathSizes\@xipt  \@xipt  \@viipt \@vpt
\DeclareMathSizes\@xiipt \@xiipt \@viiipt\@vipt
\DeclareMathSizes\@xivpt \@xivpt \@xpt   \@viipt
\DeclareMathSizes\@xviipt\@xviipt\@xiipt \@xpt
\DeclareMathSizes\@xxpt  \@xxpt  \@xivpt \@xiipt
\DeclareMathSizes\@xxvpt \@xxvpt \@xxpt  \@xviipt
\PackageInfo{mt11p}{Adjusting math subscript sizing}}
```

6.5 Fetch the MathTime and MathTimePLUS fonts.

We don't create `.fd` files, since the fonts are not for general use. We declare them as separate encodings, called `Lmt*`, because they differ quite a bit from the `OM*` encodings, which might lead to trouble if, e.g., someone assumed the `letters` font has oldstyle digits or that the `symbols` font has calligraphic letters.

There will be a *heavy* series for some fonts, which we denote by the macro `\hvdefault`, and an upright italic shape, `\uidefault`. For the bold and heavy math versions we would need additional bold, *ultra* (`\uldefault`) and *poster* (`\psdefault`) weight fonts; pretend they exist, and then define substitutes. For the off chance that these are already defined, we "provide" them, rather than define them. Only the macros are used in the rest of this file, i.e., "`{ui}`" never appears again, so no font designations are hardwired into this file (except those for common Computer Modern fonts).

```
\providecommand{\hvdefault}{h}
\providecommand{\uidefault}{ui}
\providecommand{\uldefault}{u}
\providecommand{\psdefault}{p}
```

Letters Declare the `letters` fonts. Include `\noaccents@` so math accents don't break in `\mathnormal`.

```
\DeclareFontEncoding{LmtL}{}{\noaccents@}
\DeclareFontFamily{LmtL}{mtl}{\skewchar\font=45␣}
\DeclareFontShape{LmtL}{mtl}{\mddefault}{\itdefault}{<->mtmi}{}
\DeclareFontShape{LmtL}{mtl}{\bfdefault}{\itdefault}{<->mtmib}{}
\DeclareFontShape{LmtL}{mtl}{\hvdefault}{\itdefault}{<->mtmih}{}% see dtx  SUBS
```

Note that the font `mtmih` is a virtual font that uses `ptmbi8r`, so even if you have a "heavy" weight Times italic font, you can't use it as a math `letters` font unless you know how to create a suitable raw font and change `mtmih.vf` and `mtmih.vf`. The same applies to `mtmuh`.

Declare the `letters` font with upright lower case Greek letters too; we won't do anything with them, but we will describe how to set them up, if someone wants to use them as symbols (Sec. 6.12).

```
\DeclareFontShape{LmtL}{mtl}{\mddefault}{\uidefault}{<->mtmu}{}
\DeclareFontShape{LmtL}{mtl}{\bfdefault}{\uidefault}{<->mtmub}{}
\DeclareFontShape{LmtL}{mtl}{\hvdefault}{\uidefault}{<->mtmuh}{}
```

Symbols Now the `symbols` font:

```
\DeclareFontEncoding{LmtS}{}{\no@accents}%
\DeclareFontFamily{LmtS}{mts}{}% the fonts have no skewkerns
```

Why do we say `\no@accents` when this is the *one* font that *will* have accents if we don't find any elsewhere?—because they are in non-standard slots. There is no "`\mathsy`" alphabet, but someone could try to be clever and define one. Since we said `\noaccents@`, we might as well do the following re-declarations ($\mathcal{AMS}$ does them too).

```
\DeclareFontEncoding{OML}{}{\noaccents@}
\DeclareFontEncoding{OMS}{}{\noaccents@}
```

`mtsyn` and the older `mtsy` differ only in the `radical1` glyph (`C132`, *'204*), which is a surd/radical at the baseline (as opposed to below it), otherwise identical to the other radical sign at *'160*; see Sec. 6.13.

```
\DeclareFontShape{LmtS}{mts}{\mddefault}{\updefault}{<->mtsyn}{}
\DeclareFontShape{LmtS}{mts}{\bfdefault}{\updefault}{<->mtsyb}{}
\DeclareFontShape{LmtS}{mts}{\hvdefault}{\updefault}{<->mtsyh}{}
```

Script Letters The letters-only `mtms*` fonts, which we use for `\mathcal` and/or `\mathscr`.

```
\DeclareFontEncoding{LmtW}{}{\noaccents@}
\DeclareFontFamily{LmtW}{mtw}{\skewchar\font=42␣\hyphenchar\font=-1␣}
\DeclareFontShape{LmtW}{mtw}{\mddefault}{\updefault}{<->mtms}{}
\DeclareFontShape{LmtW}{mtw}{\bfdefault}{\updefault}{<->mtmsb}{}
```

Math Extention The `largesymbols` and math extension fonts `mtex*` are compatible with `OMX`; the only difference is an extra radical in slot *'237*, which we don't use, because it is identical to the one in the `symbols` font (*'160*).

```
\DeclareFontFamily{OMX}{mtx}{}% this really is OMX, thank you
\DeclareFontShape{OMX}{mtx}{\mddefault}{\updefault}{<->mtex}{}
\DeclareFontShape{OMX}{mtx}{\bfdefault}{\updefault}{<->mtexb}{}
\DeclareFontShape{OMX}{mtx}{\hvdefault}{\updefault}{<->mtexh}{}
```

Greek Text MathTıme includes a font with upright Greek letters, upper and lower case, meant for use as *letters* (as opposed to *symbols*). The fonts have ligatures with empty slots to get alternate forms—see Section 8 of the *MathTıme*PLUS reference manual.

```
\DeclareFontEncoding{LmtG}{}{\noaccents@}
\DeclareFontFamily{LmtG}{mtg}{}
\DeclareFontShape{LmtG}{mtg}{\mddefault}{\updefault}{<->mtgu}{}
\DeclareFontShape{LmtG}{mtg}{\bfdefault}{\updefault}{<->mtgub}{}
```

Lastly, declare default for the new encodings, so NFSS is happy.

```
\DeclareFontSubstitution{LmtL}{mtl}{\mddefault}{\itdefault}
\DeclareFontSubstitution{LmtS}{mts}{\mddefault}{\updefault}
\DeclareFontSubstitution{LmtW}{mtw}{\mddefault}{\updefault}
\DeclareFontSubstitution{LmtG}{mtg}{\mddefault}{\updefault}
```

6.6 Math declarations

We assume the usual math setup was loaded into the format and change things around to use MathTıme. If for some reason you don't read a standard math setup into the format, you'll have to read the setup in before loading this package. Note that it is not necessary to "preload" the standard Computer Modern font metrics, if font memory is a problem.

Math Versions We use three math versions, *normal*, *bold*, and *heavy*, and keep everything separate, as if there really were three sets of fonts available for everything. Actually, they're not, but one could use the hooks to supply such fonts if one had them.

Give a notice so people don't freak when they see all the warnings messages NFSS emits.

```
\PackageWarning{mt11p-Font}{Expect LOTS of NFSS warnings}
```

Some of the code that follows is redundant.

```
\DeclareMathVersion{normal}
\DeclareMathVersion{bold}
\DeclareMathVersion{heavy}
\def\normalmath{\@nomath\normalmath\mathversion{normal}}
\def\boldmath{\@nomath\boldmath\mathversion{bold}}
\def\heavymath{\@nomath\heavymath\mathversion{heavy}}
\def\unboldmath{\normalmath}
\let\unheavymath\unboldmath
```

This is a private joke...

```
\let\unnormalmath\relax
```

Declare Math Font Families The following does not depend on any options:

```
\DeclareSymbolFont{operators}{\encodingdefault}%
                             {\rmdefault}{\mddefault}{\updefault}
\DeclareSymbolFont{letters}{LmtL}%
                           {mtl}{\mddefault}{\itdefault}
\DeclareSymbolFont{symbols}{LmtS}%
                           {mts}{\mddefault}{\updefault}
\DeclareSymbolFont{largesymbols}{OMX}%
                                {mtx}{\mddefault}{\updefault}
\DeclareSymbolFontAlphabet \mathnormal {letters}
\DeclareSymbolFontAlphabet \mathrm     {operators}
\DeclareMathAlphabet \mathbf {\encodingdefault}%
                             {\rmdefault}{\bfdefault}{\updefault}
\DeclareMathAlphabet \mathsf {\encodingdefault}%
                             {\sfdefault}{\mddefault}{\updefault}
\DeclareMathAlphabet \mathit {\encodingdefault}%
                             {\rmdefault}{\mddefault}{\itdefault}
\DeclareMathAlphabet \mathtt {\encodingdefault}%
                             {\ttdefault}{\mddefault}{\updefault}
```

The calligraphy stuff depends on what '`cal`' option is being used; we either use `mtms` as the `\mathcal` font, OR we make `cmsy` the `\mathcal` font and make `mtms` a "script" font `\mathscr`, OR we make `cmsy` the `\mathcal` font and don't use `mtms` for anything.

```
\ifcase\mt@cal % case 0
   \DeclareMathAlphabet \mathcal {LmtW}{mtw}{\mddefault}{\updefault}
\or            % case 1
   \DeclareMathAlphabet \mathscr {LmtW}{mtw}{\mddefault}{\updefault}
   \DeclareMathAlphabet \mathcal {OMS}{cmsy}{m}{n}
\or            % case 2
   \DeclareMathAlphabet \mathcal {OMS}{cmsy}{m}{n}
\else % oops
   \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@cal}%
```

```
                  {The \string\mathcal/\string\mathscr setup is hosed.}
\fi
```

We'll have to deal with more `\mathcal`-related stuff later.

Note that things defined via `\DeclareSymbolFontAlphabet` switch math versions automatically with `\SetSymbolFont`; these include `\mathnormal` and `\mathrm`.

Normal Math Version This is largely redundant.

```
\SetSymbolFont{operators}{normal}{\encodingdefault}%
                                  {\rmdefault}{\mddefault}{\updefault}
\SetSymbolFont{letters}{normal}{LmtL}%
                                {mtl}{\mddefault}{\itdefault}
\SetSymbolFont{symbols}{normal}{LmtS}%
                                {mts}{\mddefault}{\updefault}
\SetSymbolFont{largesymbols}{normal}{OMX}%
                                     {mtx}{\mddefault}{\updefault}
\SetMathAlphabet \mathsf {normal}%
                  {\encodingdefault}{\sfdefault}{\mddefault}{\updefault}
\SetMathAlphabet \mathtt {normal}%
                  {\encodingdefault}{\ttdefault}{\mddefault}{\updefault}
\SetMathAlphabet \mathit {normal}%
                  {\encodingdefault}{\rmdefault}{\mddefault}{\itdefault}
\SetMathAlphabet \mathbf {normal}%
                  {\encodingdefault}{\rmdefault}{\hvdefault}{\updefault}
```

`\mathcal` and `\mathscr` have several cases.

```
\ifcase\mt@cal % case 0
   \SetMathAlphabet \mathcal {normal}{LmtW}{mtw}{\mddefault}{\updefault}
\or            % case 1
   \SetMathAlphabet \mathscr {normal}{LmtW}{mtw}{\mddefault}{\updefault}
   \SetMathAlphabet \mathcal {normal}{OMS}{cmsy}{m}{n}
\or            % case 2
   \SetMathAlphabet \mathcal {normal}{OMS}{cmsy}{m}{n}
\fi
```

Bold Math Version Same story as above.

```
\SetSymbolFont{operators}{bold}{\encodingdefault}%
                                {\rmdefault}{\bfdefault}{\updefault}
\SetSymbolFont{letters}{bold}{LmtL}%
                              {mtl}{\bfdefault}{\itdefault}
\SetSymbolFont{symbols}{bold}{LmtS}%
                              {mts}{\bfdefault}{\updefault}
\SetSymbolFont{largesymbols}{bold}{OMX}%
                                   {mtx}{\bfdefault}{\updefault}
\SetMathAlphabet \mathsf {bold}{\encodingdefault}%
                                {\sfdefault}{\bfdefault}{\updefault}
\SetMathAlphabet \mathtt {bold}{\encodingdefault}%
                                {\ttdefault}{\bfdefault}{\updefault}
\SetMathAlphabet \mathit {bold}{\encodingdefault}%
                                {\rmdefault}{\bfdefault}{\itdefault}
\SetMathAlphabet \mathbf {bold}{\encodingdefault}%
                                {\rmdefault}{\uldefault}{\updefault}
```

And the cases

```
\ifcase\mt@cal % case 0
   \SetMathAlphabet \mathcal {bold}{LmtW}{mtw}{\bfdefault}{\updefault}
\or            % case 1
   \SetMathAlphabet \mathscr {bold}{LmtW}{mtw}{\bfdefault}{\updefault}
   \SetMathAlphabet \mathcal {bold}{OMS}{cmsy}{b}{n}
\or            % case 2
   \SetMathAlphabet \mathcal {bold}{OMS}{cmsy}{b}{n}
\fi
```

Heavy Math Version Still more of the same.

```
\SetSymbolFont{operators}{heavy}{\encodingdefault}%
                                 {\rmdefault}{\hvdefault}{\updefault}
\SetSymbolFont{letters}  {heavy}{LmtL}%
                                 {mtl}{\hvdefault}{\itdefault}
\SetSymbolFont{symbols}  {heavy}{LmtS}%
                                 {mts}{\hvdefault}{\updefault}
\SetSymbolFont{largesymbols}{heavy}{OMX}%
                                    {mtx}{\hvdefault}{\updefault}
\SetMathAlphabet \mathsf {heavy}{\encodingdefault}%
                                 {\sfdefault}{\hvdefault}{\updefault}
\SetMathAlphabet \mathtt {heavy}{\encodingdefault}%
                                 {\ttdefault}{\hvdefault}{\updefault}
\SetMathAlphabet \mathit {heavy}{\encodingdefault}%
                                 {\rmdefault}{\hvdefault}{\itdefault}
\SetMathAlphabet \mathbf {heavy}{\encodingdefault}%
                                 {\rmdefault}{\psdefault}{\updefault}%
```

And the cases for script,

```
\ifcase\mt@cal % case 0
   \SetMathAlphabet \mathcal {heavy}{LmtW}{mtw}{\hvdefault}{\updefault}
\or            % case 1
   \SetMathAlphabet \mathscr {heavy}{LmtW}{mtw}{\hvdefault}{\updefault}
   \SetMathAlphabet \mathcal {heavy}{OMS}{cmsy}{\hvdefault}{\updefault}
\or            % case 2
   \SetMathAlphabet \mathcal {heavy}{OMS}{cmsy}{\hvdefault}{\updefault}
\fi
```

6.7 Bold & heavy font substitutions

We pretended there are all sorts of weights available, but, in fact, they're not. Declare substitutions, unless we were told not to. NFSS will complain if we use an as-of-yet undeclared font as a substitute, so we explicitly read in the `.fd` files (this happens later for other fonts too). No harm will be done if they have already been read. Read in even `OMScmsy.fd`, in case the format has zilch preloaded.

```
\ifcase\mt@sub % case 0
   \PackageWarning{mt11p}{No math font substitutions set}
```

Check for and read in the files ⟨`\encodingdefault`⟩⟨`\`*xy*`default`⟩`.fd` for the three text families. Be nice and check for the all-lower case name too.

```
\or % case 1
   \def\mt@fd@check#1{%
          \edef\reserved@a{\lowercase{\noexpand
```

```
                               \InputIfFileExists{\encodingdefault #1.fd}}}%
          \reserved@a\relax{\InputIfFileExists{\encodingdefault #1.fd}\relax
                         {\PackageError{mt11p}%
                           {No file `\encodingdefault#1.fd' found}%
                           {Your text \string#1 setup looks inconsistent.}}}}
    \mt@fd@check\rmdefault
    \mt@fd@check\sfdefault
    \mt@fd@check\ttdefault
    \edef\reserved@a{\lowercase{\noexpand\InputIfFileExists{omscmsy.fd}}}%
    \reserved@a\relax{\InputIfFileExists{OMScmsy.fd}\relax
                 {\PackageError{mt11p}{No OMScmsy.fd file}%
                           {Your are missing a basic distribution file. %
                                        Check the LaTeX installation.}}}
    \DeclareFontShape{\encodingdefault}%
                      {\rmdefault}{\hvdefault}{\updefault}% heavy \rm SUBS
                      {<->ssub*\rmdefault/\bfdefault/\updefault}{}
    \DeclareFontShape{\encodingdefault}%
                      {\sfdefault}{\hvdefault}{\updefault}% heavy \sf SUBS
                      {<->ssub*\sfdefault/\bfdefault/\updefault}{}
    \DeclareFontShape{\encodingdefault}%
                      {\ttdefault}{\hvdefault}{\updefault}% heavy \tt SUBS
                      {<->ssub*\ttdefault/\bfdefault/\updefault}{}
    \DeclareFontShape{\encodingdefault}%
                      {\rmdefault}{\psdefault}{\updefault}% poster \rm SUBS
                      {<->ssub*\rmdefault/\bfdefault/\updefault}{}
    \DeclareFontShape{LmtW}%
                      {mtw}{\hvdefault}{\updefault}%      heavy mtms SUBS
                      {<->ssub*mtw/\bfdefault/\updefault}{}
    \DeclareFontShape{OMS}{cmsy}{\hvdefault}{\updefault}%
                      {<->ssub*cmsy/b/n}{}%              heavy cmsy SUBS
\else % oops
    \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@sub}%
                 {No math font substitutions set.}
\fi
```

6.8 Mathcodes

The MathTime fonts are *nearly* replacements for Computer Modern, but there are a few things to fix up. We also make it possible to take some symbols and accents from either `operators` or from elsewhere, depending on the options; this will be done later.

`letters` Font Upright upper case Greek no longer comes from the `operators` font, but from the `letters` font, so upper and lower case Greek letters now behave identically, namely as math symbols. This means that something like `$\mathit{\Delta\Gamma}\mathbf{\Xi}$` no longer works—upper case Greek letters are not affected by math alphabet changes. There is no simple fix.

Upright upper case Greek:

```
\DeclareMathSymbol\Gamma       {\mathord}{letters}{'060}
\DeclareMathSymbol\Delta       {\mathord}{letters}{'061}
\DeclareMathSymbol\Theta       {\mathord}{letters}{'062}
\DeclareMathSymbol\Lambda      {\mathord}{letters}{'063}
\DeclareMathSymbol\Xi          {\mathord}{letters}{'064}
```

```
\DeclareMathSymbol\Pi          {\mathord}{letters}{'065}
\DeclareMathSymbol\Sigma       {\mathord}{letters}{'066}
\DeclareMathSymbol\Upsilon     {\mathord}{letters}{'067}
\DeclareMathSymbol\Phi         {\mathord}{letters}{'070}
\DeclareMathSymbol\Psi         {\mathord}{letters}{'071}
\DeclareMathSymbol\Omega       {\mathord}{letters}{'177}
```

Italic upper case Greek; $\mathcal{AMS}$ calls these `\var`⟨*Greek Letter*⟩.

```
\DeclareMathSymbol\varGamma    {\mathord}{letters}{'000}
\DeclareMathSymbol\varDelta    {\mathord}{letters}{'001}
\DeclareMathSymbol\varTheta    {\mathord}{letters}{'002}
\DeclareMathSymbol\varLambda   {\mathord}{letters}{'003}
\DeclareMathSymbol\varXi       {\mathord}{letters}{'004}
\DeclareMathSymbol\varPi       {\mathord}{letters}{'005}
\DeclareMathSymbol\varSigma    {\mathord}{letters}{'006}
\DeclareMathSymbol\varUpsilon  {\mathord}{letters}{'007}
\DeclareMathSymbol\varPhi      {\mathord}{letters}{'010}
\DeclareMathSymbol\varPsi      {\mathord}{letters}{'011}
\DeclareMathSymbol\varOmega    {\mathord}{letters}{'012}
```

Lower case Greek letters are OK, except for

```
\DeclareMathSymbol\varkappa    {\mathord}{letters}{'176}
\DeclareMathSymbol\omicron     {\mathord}{letters}{`\o}% for completeness
```

`symbols` Font There are a few new symbols, and the fonts no longer contain calligraphic letters. Easy things (those where no alternatives are available) first; the new names are those used in the MathTime distribution.

```
\DeclareMathSymbol\openclubsuit   {\mathord}{symbols}{"78}
\DeclareMathSymbol\shadedclubsuit {\mathord}{symbols}{"79}
\DeclareMathSymbol\openspadesuit  {\mathord}{symbols}{"7A}
\DeclareMathSymbol\shadedspadesuit{\mathord}{symbols}{"7B}
\DeclareMathSymbol\triangleleft   {\mathbin}{symbols}{"47}
\DeclareMathSymbol\triangleright  {\mathbin}{symbols}{"46}
\DeclareMathSymbol\cupprod        {\mathbin}{symbols}{"59}
\DeclareMathSymbol\capprod        {\mathbin}{symbols}{"5A}
```

The `\varcirc` symbol is slightly smaller than the 'open bullet'. `mathtimy` calls it "`\comp`".

```
\DeclareMathSymbol\varcirc{\mathbin}{symbols}{"42}
\let\comp=\varcirc
```

`\vec` used to be in `letters`, now it's in `symbols`.

```
\DeclareMathAccent\vec     {\mathord}{symbols}{"45}
```

`mtsy` has an accent glyph called `overbar`, which `mathtimy` calls `\widebar`.

```
\DeclareMathAccent\overbar{\mathord}{symbols}{"53}
\let\widebar\overbar
```

The glyph called `backslash` in the AFM file of `mtsyn` is the one usually used for `\setminus`, and the format file has a definition for it (`symbols`, slot `"6E`). There is a `setminus` in the AFM file, but it looks somewhat like a tilted minus sign. $\mathcal{AMS}$ packages call that glyph "`\smallsetminus`".

```
\DeclareMathSymbol\varsetminus{\mathbin}{symbols}{"58}
\let\smallsetminus\varsetminus
```

Calligraphic letters and their variants Irrespective of the options, both `\mathcal` and `\mathscr` get defined; warn only when the `nomtms` option is used.

```
\ifcase\mt@cal % case 0
   \let\mathscr\mathcal % don't warn
\or            % case 1
\or            % case 2
   \def\mathscr#1{\mathcal{#1}%
              \PackageWarning{mt11p}{no \string\mathscr\space available;
                           \MessageBreak using OMScmsy \string\mathcal}}
\fi
```

We don't try to trap cases like `\mathcal{a}` when the `\mathcal` alphabet is `OMS` (and hence has no lowercase letters!). Presumably by writing `\mathcal{a}` you know that `\mathcal` is `LmtW`, i.e., the `mtcal` option is in effect. Writing `\mathscr{a}` when `\mathscr` is essentially a synonym for `\mathcal` will at least produce a warning.

`mtms`* has alternate forms of a few letters. We define math commands to fetch them, assuming we have an `mtms`* font as `\mathscr`. (The script letters have sidebearings which make them unsuitable for text; use math mode for logos and such.) The auxiliary macro will be defined later.

```
\def\varA   {\mt@al@hexsy\mathscr{30}}
\def\varE   {\mt@al@hexsy\mathscr{31}}
\def\varG   {\mt@al@hexsy\mathscr{32}}
\def\varvarG{\mt@al@hexsy\mathscr{33}}
\def\varI   {\mt@al@hexsy\mathscr{34}}
\def\varL   {\mt@al@hexsy\mathscr{35}}
\def\varQ   {\mt@al@hexsy\mathscr{36}}
\def\varS   {\mt@al@hexsy\mathscr{37}}
\def\varr   {\mt@al@hexsy\mathscr{38}}
\def\varz   {\mt@al@hexsy\mathscr{39}}
\def\iscript{\mt@al@hexsy\mathscr{7B}}% can't use \imath
\def\jscript{\mt@al@hexsy\mathscr{7C}}% can't use \jmath
```

The MathTime distribution defines the macro `\script`, which is supposed to produce the script letter "A" and its alternate form with "`\script A`" and "`\script*A`", respectively. The argument of `\script` had better be a single token (or an asterisk and a single token), else we get a mess; in order to get the expected output that token should be a single letter$_{11}$, one of those that have variant forms.

```
\DeclareRobustCommand\script{\@ifstar\@varscript\@script}
\ifnum\mt@cal<2 % mtms available
   \def\@script#1{\mathscr{#1}}
   \def\@varscript#1{%
               \ifx#1A\varA
          \else\ifx#1E\varE
          \else\ifx#1G\varG
          \else\ifx#1g\varvarG
          \else\ifx#1I\varI
          \else\ifx#1L\varL
          \else\ifx#1Q\varQ
          \else\ifx#1S\varS
          \else\ifx#1r\varr
          \else\ifx#1z\varz
```

```
          \else\ifx#1i\iscript
          \else\ifx#1j\jscript
          \else       \PackageWarning{mt11p}{#1: no such variant letter}%
                                   \mathscr{#1}%
          \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi}
\else % OMS only
    \def\@script#1{\mathscr{#1}}
    \def\@varscript#1{\PackageWarning{mt11}%
                             {no variant letters available in OMS}%
                             \@script{#1}}
\fi
```

The above will produce *two* warnings if you say `$\mathcal{a}$` with the `nomtms` option in effect.

Symbols from either `operators` or from elsewhere Many symbols and punctuation marks can be taken from either the `operators` math family, or from `letters`/`symbols`. Most of the glyphs are virtually identical in either font. The 'optimal' choice depends on several things, e.g., do you wish to use as few glyphs from the MathTime fonts as possible, do you prefer 'thinner' symbols, etc. The `opmax` option will take as many things from `operators` as possible, but risk, e.g., that the = sign will not line up with double arrows, etc. The `opsafe` option uses `operators` glyphs only when absolutly nothing can go wrong; the `opnone` option *avoids* `operators` whenever possible and uses MathTime glyphs instead.

The `operators` font is uses the encoding `\encodingdefault`. We *must* recognize that encoding in order to take any symbols from `operators`. The only encodings I know of that someone might possibly be using with LaTeX 2_ε are `T1`, `OT1` or a superset thereof, `8r` and `8a`. It's easy to hack in another encoding, if required, as would be the case with a superset or `OT1`. If we don't recognize the encoding, stick to using `symbols` and `letters`.

NFSS has a "`\DeclareMathDelimiter`" but nothing to assign `\delcode`s. We try to make a general macro and insist on saying "`\symletters`" rather than just saying "`1`", eventhough it's pretty much out of the question that `\symoperators`, `\symletters`, `\symsymbols` and `\symlargeoperators` are something other than 0, 1, 2 and 3, respectively, since TeX itself checks for extra `fontdimen`s of `\fam2` and `\fam3` before it agrees to typeset math. (The second and fourth arguments should be hexadecimal numbers without the "`"`".)

```
\def\mt@assign@del#1#2#3#4#5{%
          \edef\reserved@a{\expandafter\the\csname sym#2\endcsname}%
          \edef\reserved@b{\expandafter\the\csname sym#4\endcsname}%
          \delcode#1="\reserved@a#3\reserved@b#5 \relax}
```

This code is read once and not stored, so we just write the whole thing out case by case. `\colon` is defined in some of the $\mathcal{AMS}$ packages, so don't overwrite an existing definition.

Here's the overly conservative option:

```
\ifcase\mt@ops % case 0
    \DeclareMathSymbol{(}{\mathopen}{letters}{"2E}
    \mt@assign@del{`\(}{letters}{2E}{largesymbols}{00}
    \DeclareMathSymbol{)}{\mathclose}{letters}{"2F}
    \mt@assign@del{`\)}{letters}{2F}{largesymbols}{01}
    \DeclareMathSymbol{+}{\mathbin}{symbols}{"43}
```

```
  \DeclareMathSymbol{=}{\mathrel}{symbols}{"44}% identical one at "48
  \DeclareMathSymbol{;}{\mathpunct}{symbols}{"49}
  \DeclareMathSymbol{[}{\mathopen}{symbols}{"54}
  \mt@assign@del{`\[}{symbols}{54}{largesymbols}{02}
  \DeclareMathSymbol{]}{\mathclose}{symbols}{"55}
  \mt@assign@del{`\]}{symbols}{55}{largesymbols}{03}
  \DeclareMathSymbol{:}{\mathrel}{symbols}{"56}
  \ifx\colon\undefined
     \DeclareMathSymbol{\colon}{\mathpunct}{symbols}{"56}
  \fi
  \DeclareMathSymbol{!}{\mathclose}{symbols}{"57}
  \DeclareMathSymbol{.}{\mathord}{letters}{"3A}
  \DeclareMathSymbol{\period}{\mathpunct}{letters}{"3A}
  \DeclareMathSymbol{,}{\mathpunct}{letters}{"3B}
  %\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}% format defaults
  %\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
  %\DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
  %\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
  %\DeclareMathSymbol{-}{\mathbin}{symbols}{"00}
```

Now the safe version; the stuff we take from operators never gets used in “synthesized” or extensible symbols.

```
\or           % case 1
  \DeclareMathSymbol{(}{\mathopen}{operators}{`\(}
  \mt@assign@del{`\(}{operators}{28}{largesymbols}{00}
  \DeclareMathSymbol{)}{\mathclose}{operators}{`\)}
  \mt@assign@del{`\)}{operators}{29}{largesymbols}{01}
  \DeclareMathSymbol{+}{\mathbin}{symbols}{"43}
  \DeclareMathSymbol{=}{\mathrel}{symbols}{"44}% identical one at "48
  \DeclareMathSymbol{;}{\mathpunct}{operators}{`\;}
  \DeclareMathSymbol{[}{\mathopen}{operators}{`\[}
  \mt@assign@del{`\[}{operators}{5B}{largesymbols}{02}
  \DeclareMathSymbol{]}{\mathclose}{operators}{`\]}
  \mt@assign@del{`\]}{operators}{5D}{largesymbols}{03}
  \DeclareMathSymbol{:}{\mathrel}{operators}{`\:}
  \ifx\colon\undefined % AMS defines it differently; don't overwrite that
     \DeclareMathSymbol{\colon}{\mathpunct}{operators}{`\:}
  \fi
  \DeclareMathSymbol{!}{\mathclose}{operators}{`\!}
  \DeclareMathSymbol{.}{\mathord}{operators}{`\.}
  \DeclareMathSymbol{\period}{\mathpunct}{operators}{`\.}
  \DeclareMathSymbol{,}{\mathpunct}{operators}{`\,}
  %\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}% format defaults
  %\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
  %\DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
  %\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
  %\DeclareMathSymbol{-}{\mathbin}{symbols}{"00}
```

Now `opmax`. This is risky, or more exactly some glyphs *will* be flawed, because the minus and equal signs won't line up with arrows, etc. On the other hand, this will minimize the use of non-resident fonts, so it may make the final output marginally smaller if you have partial font downloading or if you include the MathTıme fonts as bitmaps.

```
\or           % case 2
  \PackageWarning{mt11p}{opmax enabled}
```

```
\DeclareMathSymbol{(}{\mathopen}{operators}{`\(}
\mt@assign@del{`\(}{operators}{28}{largesymbols}{00}
\DeclareMathSymbol{)}{\mathclose}{operators}{`\)}
\mt@assign@del{`\)}{operators}{29}{largesymbols}{01}
\DeclareMathSymbol{+}{\mathbin}{operators}{`\+}
\DeclareMathSymbol{=}{\mathrel}{operators}{`\=}% dangerous
\DeclareMathSymbol{;}{\mathpunct}{operators}{`\;}
\DeclareMathSymbol{[}{\mathopen}{operators}{`\[}
\mt@assign@del{`\[}{operators}{5B}{largesymbols}{02}
\DeclareMathSymbol{]}{\mathclose}{operators}{`\]}
\mt@assign@del{`\]}{operators}{5D}{largesymbols}{03}
\DeclareMathSymbol{:}{\mathrel}{operators}{`\:}
\ifx\colon\undefined
   \DeclareMathSymbol{\colon}{\mathpunct}{operators}{`\:}
\fi
\DeclareMathSymbol{!}{\mathclose}{operators}{`\!}
\DeclareMathSymbol{.}{\mathord}{operators}{`\.}
\DeclareMathSymbol{\period}{\mathpunct}{operators}{`\.}
\DeclareMathSymbol{,}{\mathpunct}{operators}{`\,}
\DeclareMathSymbol{/}{\mathord}{operators}{`\/}
\mt@assign@del{`\/}{operators}{2F}{largesymbols}{0E}
```

Now we have to be careful: `<`, `>` and `|` can be taken from `operators` if `\encodingdefault` is anything *except* `OT1`. We use an `endash` instead of a hyphen for the minus sign if we recognize the encoding to get at least something close to a real minus sign. That's a sloppy kludge, but choosing the `opmax` option implies you've lowered your standards somewhat.

```
\def\reserved@a{OT1}\edef\reserved@b{\encodingdefault}
\ifx\reserved@a\reserved@b % unavailable in OT1...
   %\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}% format defaults
   %\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
   %\DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
   \DeclareMathSymbol{-}{\mathbin}{operators}{"7B}% dangerous
\else
   \def\reserved@a{T1}%
   \ifx\reserved@a\reserved@b
      \DeclareMathSymbol{-}{\mathbin}{operators}{"15}% dangerous
   \else
      \def\reserved@a{8r}%
      \ifx\reserved@a\reserved@b
         \DeclareMathSymbol{-}{\mathbin}{operators}{150}% dangerous
      \else
         \def\reserved@a{8a}%
         \ifx\reserved@a\reserved@b
            \DeclareMathSymbol{-}{\mathbin}{operators}{"B1}% dangerous
         \else
            % keep symbols, slot "00
         \fi % 8a
      \fi % 8r
   \fi % T1
   \DeclareMathSymbol{<}{\mathrel}{operators}{`\<}% ascii positions
   \DeclareMathSymbol{>}{\mathrel}{operators}{`\>}
   \DeclareMathSymbol{|}{\mathord}{operators}{`\|}
   \mt@assign@del{`\|}{operators}{7C}%  ascii bar
```

```
                     {largesymbols}{0C}
     \DeclareMathDelimiter\backslash\mathord{operators}{"5C}% ascii backslash
                                         {largesymbols}{"0F}
     \DeclareMathDelimiter\lbrace   \mathopen{operators}{`\{}% ascii `{'
                                         {largesymbols}{"08}
     \DeclareMathDelimiter\rbrace   \mathclose{operators}{`\}}% ascii `}'
                                         {largesymbols}{"09}
  \fi
\else % oops
  \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@ops}%
               {The mathsymbol setup is hosed; expect bad output.}
\fi
```

Math Accents Accents in math can be made to either use a fixed font, or to obey math alphabets. In order to have accents of the latter type, we *must* recognize the `\encodingdefault` (which the `operators` math family is in) and set the corresponding accent codes up for that encoding. If we don't recognize the encoding we have to use fixed accents; MathTime provides suitable accents in the `symbols` font.

For some reason the ring accent "å" is always left out, so we define it, but don't warn if we're stuck with `symbols` accents which have no ring accent glyph and just make `\ring` a no-op.

The code below is written out long hand style, to keep it legible. It does not get stored, it's read only once, and most of it is skipped, so it would be only a marginal improvement to merge everything.

The `opaccents` case:

```
\ifcase\mt@accs % case 0
```

We do something similar to what $\mathcal{AMS}$ does, but rather simple-mindedly: choose between 'fixed' (class `0`) or 'variable' (class `7`) based on the value of `\mt@acc@class`, whose value should be either `0` or `7`, depending on whether the (encoding of the) current math alphabet has accents *in the expected positions*. We could use the $\mathcal{AMS}$ hook (`\accentclass@`), but we ought to make our own… Actually, this is slightly naughty; we could have gone through the `\DeclaterMathAccent` command, declare two accents, and have the user-level accent command pick the right one based on something `\noaccents@` says, but that's kind of pointless.

All of this works as long as we recognize the encoding. If we don't, use fixed accents from `symbols` and complain. The slot specification must be in hex.

```
\def\mt@accent@make#1#2{\def#1{\mathaccent"\mt@acc@class0#2␣}}
```

Here are the encodings we recognize:

```
  \edef\reserved@a{\encodingdefault}
  \def\reserved@b{T1}% a.k.a Cork or DC or EC
  \ifx\reserved@a\reserved@b
     \mt@accent@make{\grave}{00}
     \mt@accent@make{\acute}{01}
     \mt@accent@make{\hat}{02}
     \mt@accent@make{\tilde}{03}
     \mt@accent@make{\ddot}{04}
     \mt@accent@make{\ring}{06}
     \mt@accent@make{\check}{07}
```

```
    \mt@accent@make{\breve}{08}
    \mt@accent@make{\bar}{09}
    \mt@accent@make{\dot}{0A}
  \else
    \def\reserved@b{OT1}% ``Knuth roman''
    \ifx\reserved@a\reserved@b
     \mt@accent@make{\grave}{12}
     \mt@accent@make{\acute}{13}
     \mt@accent@make{\check}{14}
     \mt@accent@make{\breve}{15}
     \mt@accent@make{\bar}{16}
     \mt@accent@make{\ring}{17}
     \mt@accent@make{\hat}{5E}
     \mt@accent@make{\dot}{5F}
     \mt@accent@make{\tilde}{7E}
     \mt@accent@make{\ddot}{7F}
    \else
      \def\reserved@b{8r}% TeXBase1
      \ifx\reserved@a\reserved@b
        \mt@accent@make{\dot}{01}
        \mt@accent@make{\ring}{09}
        \mt@accent@make{\breve}{0B}}
        \mt@accent@make{\check}{10}
        \mt@accent@make{\grave}{1E}
        \mt@accent@make{\hat}{88}
        \mt@accent@make{\tilde}{98}
        \mt@accent@make{\ddot}{A8}
        \mt@accent@make{\bar}{AF}
        \mt@accent@make{\acute}{B4}
      \else
        \def\reserved@b{8a}% AdobeStandard
        \ifx\reserved@a\reserved@b
          \mt@accent@make{\grave}{C1}
          \mt@accent@make{\acute}{C2}
          \mt@accent@make{\hat}{C3}
          \mt@accent@make{\tilde}{C4}
          \mt@accent@make{\bar}{C5}
          \mt@accent@make{\breve}{C6}
          \mt@accent@make{\dot}{C7}
          \mt@accent@make{\ddot}{C8}
          \mt@accent@make{\ring}{CA}
          \mt@accent@make{\check}{CF}
```

If we don't recognize the encoding, we have to resort to `symbols`:

```
        \else % I give up...
          \DeclareMathAccent\grave{\mathord}{symbols}{"4A}
          \DeclareMathAccent\acute{\mathord}{symbols}{"4B}
          \DeclareMathAccent\check{\mathord}{symbols}{"4C}
          \DeclareMathAccent\breve{\mathord}{symbols}{"4D}
          \DeclareMathAccent\bar  {\mathord}{symbols}{"4E}
          \DeclareMathAccent\hat  {\mathord}{symbols}{"4F}
          \DeclareMathAccent\dot  {\mathord}{symbols}{"50}
          \DeclareMathAccent\tilde{\mathord}{symbols}{"51}
          \DeclareMathAccent\ddot {\mathord}{symbols}{"52}
          \def\ring#1{{#1}} % no \ring in the symbols font
```

```
              \PackageError{mt11p}%
                            {Unknown \string\encodingdefault\space
                                    \encodingdefault. \MessageBreak
                             Reverting to fixed accents
                             from symbols font}%
                            {I had to make math accents ignore alphabet
                             changes,\MessageBreak
                             because I don't recognize the
                             `\encodingdefault' encoding.}
            \fi % 8a
         \fi % 8r
      \fi % OT1
   \fi % T1
```

Add the switch to `\noaccents@` (and check that that's defined before hand) and initialize the flag.

```
   \ifx\noaccents@\@undefined\let\noaccents@\@empty\fi % be safe
   \g@addto@macro{\noaccents@}{\def\mt@acc@class{0}}
   \def\mt@acc@class{7}
```

The `symaccents` case is easy; just make everything fixed, and make sure `\noaccents@` is defined.

```
\or % case 1, accents from symbols
   \DeclareMathAccent\grave{\mathord}{symbols}{"4A}
   \DeclareMathAccent\acute{\mathord}{symbols}{"4B}
   \DeclareMathAccent\check{\mathord}{symbols}{"4C}
   \DeclareMathAccent\breve{\mathord}{symbols}{"4D}
   \DeclareMathAccent\bar  {\mathord}{symbols}{"4E}
   \DeclareMathAccent\hat  {\mathord}{symbols}{"4F}
   \DeclareMathAccent\dot  {\mathord}{symbols}{"50}
   \DeclareMathAccent\tilde{\mathord}{symbols}{"51}
   \DeclareMathAccent\ddot {\mathord}{symbols}{"52}
   \def\ring#1{{#1}} % no \ring in the symbols font
   \ifx\noaccents@\@undefined\let\noaccents@\@empty\fi % be safe
\else % oops
   \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@accs}%
               {The math accent setup is hosed; expect bad output.}
\fi
```

Now another messy part: the $\mathcal{AMS}$ package `amsmath` has the macros `\Hat`, etc., defined with hard-coded accent positions. We redo the definition to work with whatever encoding or accents we're using. The code below is basically copied from the $\mathcal{AMS}$ package. We omit this patch unless all of the following conditions are satisfied: *i*) the macro `\mathaccent@` exists; *iii*) the macros `\Hat` and `\Check` exist; *iii*) there exists a macro `\accentclass@` whose definition is "7". If all of that is OK we pretty likely have the `amsmath` package and the following should be safe. Note that `\mathaccent@` is a rather large macro that figures out what skew kern is needed to get double accents to line up.

```
\ifx\mathaccent@\undefine\else % (i)
\ifx\Hat\undefine\else         % (iia)
\ifx\Check\undefine\else       % (iib)
\def\@tempa{7}%
\ifx\@tempa\accentclass@       % (iii)
   \PackageInfo{mt11p}{amsmath detected; redefining double accents}
```

Define fancy versions of \hat, etc., called "\Hat", etc., that perform skew calculations. To get a double accent say something like \Hat{\Hat{x}}.

There are two versions of the definition macro: one for fixed accents, another for variable ones.

```
\def\@tempa#1#2{% variable 7/0
          \def#1{\relax % write it out
                 \ifmmode
                     \def\next@{\mathaccent@{"\accentclass@#2 }}%
                 \else
                     \def\next@{\nonmatherr@{#1}}%
                 \fi
                 \next@}}
\def\@tempb#1#2{% fixed
          \def#1{\relax % write it out
                 \ifmmode
                     \def\next@{\mathaccent@{"0#2 }}%
                 \else
                     \def\next@{\nonmatherr@{#1}}%
                 \fi
                 \next@}}
```

These two accents are from the symbols fonts:

```
\@tempb\Vec{245}
\@tempb\Overbar{253}
\def\Widebar{\Overbar}% synonym
```

Go through the encodings we know and define the accents again… The symaccents option affects this too.

```
\ifnum\mt@accs=0 %
   \edef\reserved@a{\encodingdefault}
   \def\reserved@b{T1}% a.k.a Cork or DC or EC
   \ifx\reserved@a\reserved@b
      \@tempa\Hat{002}
      \@tempa\Check{007}
      \@tempa\Tilde{003}
      \@tempa\Acute{001}
      \@tempa\Grave{000}
      \@tempa\Dot{00A}
      \@tempa\Ddot{004}
      \@tempa\Breve{008}
      \@tempa\Bar{009}
   \else
      \def\reserved@b{OT1}% DEK's
      \ifx\reserved@a\reserved@b
         \@tempa\Grave{012}
         \@tempa\Acute{013}
         \@tempa\Check{014}
         \@tempa\Breve{015}
         \@tempa\Bar{016}
         \@tempa\Hat{05E}
         \@tempa\Dot{05F}
         \@tempa\Tilde{07E}
         \@tempa\Ddot{07F}
      \else
```

```
            \def\reserved@b{8r}% TeXBase1
            \ifx\reserved@a\reserved@b
               \@tempa\Dot{001}
               \@tempa\Breve{00B}
               \@tempa\Check{010}
               \@tempa\Grave{01E}
               \@tempa\Hat{088}
               \@tempa\Tilde{098}
               \@tempa\Ddot{0A8}
               \@tempa\Bar{0AF}
               \@tempa\Acute{0B4}
            \else
               \def\reserved@b{8a}% AdobeStandard
               \ifx\reserved@a\reserved@b
                  \@tempa\Grave{0C1}
                  \@tempa\Acute{0C2}
                  \@tempa\Hat{0C3}
                  \@tempa\Tilde{0C4}
                  \@tempa\Bar{0C5}
                  \@tempa\Breve{0C6}
                  \@tempa\Dot{0C7}
                  \@tempa\Ddot{0C8}
                  \@tempa\Check{0CF}
               \else % beats me...
```

If we don't know the encoding use fixed accents from `symbols`. We already gave an error message before, so just warn.

```
                  \@tempb\Grave{24A}
                  \@tempb\Acute{24B}
                  \@tempb\Check{24C}
                  \@tempb\Breve{24D}
                  \@tempb\Bar{24E}
                  \@tempb\Hat{24F}
                  \@tempb\Dot{250}
                  \@tempb\Tilde{251}
                  \@tempb\Ddot{252}
                  \PackageWarning{mt11p}{using fixed accents for amsmath}
               \fi % 8a
            \fi % 8r
         \fi % OT1
      \fi % T1
   \else % symaccents option
      \@tempb\Grave{24A}
      \@tempb\Acute{24B}
      \@tempb\Check{24C}
      \@tempb\Breve{24D}
      \@tempb\Bar{24E}
      \@tempb\Hat{24F}
      \@tempb\Dot{250}
      \@tempb\Tilde{251}
      \@tempb\Ddot{252}
   \fi
\fi\fi\fi\fi
```

Upright Greek Text The `mtgu*` fonts are not meant to be used as symbols, so we provide only text-type support: text mode font changing commands, and a math alphabet. Note that this means you *can* use the fonts in math with correct subscript sizing, etc.; just say something like `\mathgr{m}` for an upright μ (Computer Modern does not have such a glyph); in text `\textgr{m}` will work. Note also that the ligatures involving nonprinting *letters* for getting variant forms will work inside of `\mathgr`, but `o*`, `O*` or `e*` will not, which is why we added some ligatures to the `tfm` file (Sec. 3.9). The text commands are non-fragile, because the components of the macros are robust. See Sec. 6.12 for making upright Greek letters into math *symbols*. Trap the `nomathgr` case.

```
\def\greekshape{\fontencoding{LmtG}\selectfont}%
\DeclareTextFontCommand{\textgr}{\greekshape}
\ifcase\mt@mgr % case 0
   \def\mathgr#1{\PackageError{mt11p}{upright Greek not loaded for math}%
                                     {Don't use the nomathgr option.}%
                 \hbox{\greekshape #1}}% as good as anything else
\or % case 1
   \DeclareMathAlphabet\mathgr {LmtG}{mtg}{\mddefault}{\updefault}
   \SetMathAlphabet    \mathgr {normal}{LmtG}{mtg}{\mddefault}{\updefault}
   \SetMathAlphabet    \mathgr {bold}  {LmtG}{mtg}{\bfdefault}{\updefault}
   \SetMathAlphabet    \mathgr {heavy} {LmtG}{mtg}{\hvdefault}{\updefault}
\else % oops
   \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@mgr}%
                {The \string\mathgr\space setup is hosed.}
\fi
\ifnum\mt@sub=1 %
   \DeclareFontShape{LmtG}{mtg}{\hvdefault}{\updefault}%
                    {<->ssub*mtg/\bfdefault/\updefault}{}% heavy mtgu SUBS
\fi
```

The MathTime distribution defines the `\gk` font switch macro, roughly equivalent to `\greekshape`; we don't.

\hbar needs to be defined a bit carefully. The plainTEX definition is just a macron (bar) accent with an 'h' backed up into it: `{\mathchar'26\mkern-9muh}`. This is pretty horrible, and doesn't work in Times, no matter what `\mkern` you use. The definition below actually looks rather decent in all math versions. We define a private bar accent to be taken from the `symbols` math family, since we don't know what the `\bar` accent is (it might follow math alphabets changes). Force the 'h' to be taken from `letters`.

```
\ifnum\mt@hbar=1 %
   \DeclareMathSymbol \mt@hbar@macron {\mathord}{symbols}{"4E}%
   \DeclareMathSymbol \mt@hbar@h      {\mathord}{letters}{`\h}
   \def\hbar@macron#1{\hbox to\z@{\m@th\raisebox{-0.16\height}%
                        {$\csname#1style\endcsname
                            \mkern.65mu\mt@hbar@macron$}\hss}}
   \def\hbar{\mathord{\mathchoice{\hbar@macron{display}}%
                                 {\hbar@macron{text}}%
                                 {\hbar@macron{script}}%
                                 {\hbar@macron{scriptscript}}%
                      \mt@hbar@h}}

\fi
```

\dag, \ddag, \S and \P First a comment: the only reason we are dealing with these symbols at all is that they historically came from math (`OMS`) fonts; it would be easier to use a text symbol font, but such a font is not yet commonly in use, so we hack around with `8r`. If you do have such a setup, the `nodag` option will skip all of the stuff below and not touch these four symbols, but check that the setup you use does not assume `\fam2` is `OMS`.

The default is to fetch the symbols from `8r`. We do not use anything else from `8r`, in particular we don't touch `\textregistered`, `\texttrademark`, etc. The `8r.sty` "emulation" included herein only deals with the four symbols.

The format files defines these to use text or math versions of the character commands by using definitions similar to[3]

```
\def\P{\ifmmode\mathparagraph\else\textparagraph\fi}
```

(The actual definition is made robust.) The text versions have declared defaults taken from `OMS`; the math versions were `\mathchar`s pointing to the (`OMS`) `symbols` math family. We do the text versions first, since the handling of the math versions depends on them.

We take the text symbols from `8r` fonts, unless we're told not to (or unless we're skipping the whole thing). The format file makes the default test encoding for these four symbols `OMS`, so we don't have to do anything. `8r` fonts.

```
\ifcase\mt@eightR % case 0
  \PackageInfo{mt11p}{Retaining CM for \string\textdagger, etc}%
```

We have to be careful here, because the use of `8r` fonts is "non-standard". So check if we have an `8r .fd` file for `\rmdefault`. If so, check for `8r.sty`, read that, and (re)declare the four text commands we need. If there's no `8r.sty`, supply just enough to get to the four symbols we need. We assume the TFM files *do exist* (they have to, because MathTime virtual fonts use them), but we try loading the `tfm` file just to make sure.

```
\or % case 1
  \IfFileExists{8r\rmdefault.fd}%
     {\IfFileExists{8r.sty}%
        {\PackageInfo{mt11p}%
                 {MT math using 8r.sty}%
         \RequirePackage{8r}%
         \DeclareTextSymbol{\textsection}{8r}{167}%   section
         \DeclareTextSymbol{\textparagraph}{8r}{182}% paragraph
         \DeclareTextSymbol{\textdagger}{8r}{134}%    dagger
         \DeclareTextSymbol{\textdaggerdbl}{8r}{135}% daggerdbl
         \DeclareTextSymbolDefault{\textsection}{8r}%
         \DeclareTextSymbolDefault{\textparagraph}{8r}%
         \DeclareTextSymbolDefault{\textdagger}{8r}%
         \DeclareTextSymbolDefault{\textdaggerdbl}{8r}%
        }% using 8r.sty
        {\PackageWarning{mt11p}%
                  {No file 8r.sty found. Winging it}%
         \DeclareFontEncoding{8r}{}{\no@accents}%
         \DeclareFontSubstitution{8r}{\rmdefault}%
```

[3]Lines not proceeded by a line number are a part of this description and do not wind up in the `sty` file!

```
                                                {\mddefault}{\updefault}%
                \DeclareTextSymbol{\textsection}{8r}{167}%   section
                \DeclareTextSymbol{\textparagraph}{8r}{182}% paragraph
                \DeclareTextSymbol{\textdagger}{8r}{134}%    dagger
                \DeclareTextSymbol{\textdaggerdbl}{8r}{135}% daggerdbl
                \DeclareTextSymbolDefault{\textsection}{8r}%
                \DeclareTextSymbolDefault{\textparagraph}{8r}%
                \DeclareTextSymbolDefault{\textdagger}{8r}%
                \DeclareTextSymbolDefault{\textdaggerdbl}{8r}%
               }% faked 8r.sty
          \input{8r\rmdefault.fd}%
          }% 8r\rmdefault.fd was available
          {\PackageWarning{mt11p}%
                           {No fd file for 8r-coded \string\rmdefault\space
                                found; guessing TFM file names}%
```

If there's no `.fd` file for `\rmdefault` we have to wing it. Just declare the `8r` fonts for `\rmdefault` text family. (That's all you would have with Computer Modern fonts anyway, and since we're here, no elaborate setup exists on the system.) Hopefully the TFM files are available and they have the expected names. This will work when someone has the `8r`-based fonts installed, but no support for using the raw fonts. To catch the fatal case right now, we just load the `8r` font by hand and put a somewhat illuminating comment on the same line in case TEX complains.

```
      \font\preloaded=\rmdefault r8r\relax % next time try the ‘no8r’ option
           \let\preloaded\undefined
           \DeclareFontEncoding{8r}{}{\no@accents}%
           \DeclareFontFamily{8r}{\rmdefault}{\hyphenchar\font=-1 }%
           \DeclareFontSubstitution{8r}{\rmdefault}{\mddefault}{\updefault}%
           \DeclareFontShape{8r}{\rmdefault}{\mddefault}{\updefault}%
                                               {<->\rmdefault  r8r}{}%
           \DeclareFontShape{8r}{\rmdefault}{\bfdefault}{\updefault}%
                                               {<->\rmdefault  b8r}{}%
           \DeclareFontShape{8r}{\rmdefault}{\mddefault}{\itdefault}%
                                               {<->\rmdefault  ri8r}{}%
           \DeclareFontShape{8r}{\rmdefault}{\bfdefault}{\itdefault}%
                                               {<->\rmdefault  bi8r}{}%
           \DeclareTextSymbol{\textsection}{8r}{167}%   section
           \DeclareTextSymbol{\textparagraph}{8r}{182}% paragraph
           \DeclareTextSymbol{\textdagger}{8r}{134}%    dagger
           \DeclareTextSymbol{\textdaggerdbl}{8r}{135}% daggerdbl
           \DeclareTextSymbolDefault{\textsection}{8r}%
           \DeclareTextSymbolDefault{\textparagraph}{8r}%
           \DeclareTextSymbolDefault{\textdagger}{8r}%
           \DeclareTextSymbolDefault{\textdaggerdbl}{8r}%
          }% no 8r\rmdefault.fd
```

Perhaps we're skipping this altogether.

```
\or % case 2
  \PackageInfo{mt11p}{not doing anything with \string\textdagger, etc.}
\else % oops
  \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@eightR}%
               {\string\dag, \string\ddag, \string\S\space and
                \string\P\space are probably hosed.}
\fi
```

If we're told not to use 8r fonts (\mt@eightR = 0) and if \mt@cal = 0, we have no choice but to get OMS as a 'private' math alphabet and fetch the symbols from there. The auxiliary macro \mt@al@hexsy gets defined later.

```
\ifcase\mt@eightR % 0: avoid 8r
  \ifnum\mt@cal=0 %
    \DeclareMathAlphabet\mt@sy{OMS}{cmsy}{m}{n}
    \SetMathAlphabet    \mt@sy{normal}{OMS}{cmsy}{m}{n}
    \SetMathAlphabet    \mt@sy{bold}  {OMS}{cmsy}{b}{n}
    \SetMathAlphabet    \mt@sy{heavy} {OMS}{cmsy}{\hvdefault}{\updefault}
    \def\mathsection  {\mt@al@hexsy\mt@sy{78}}
    \def\dagger       {\mt@al@hexsy\mt@sy{79}}
    \def\ddagger      {\mt@al@hexsy\mt@sy{7A}}
    \def\mathparagraph{\mt@al@hexsy\mt@sy{7B}}
```

(Substitutions have already been declared.) If \mt@cal > 0 we just use \mathcal, which will be OMS.

```
  \else % \mt@cal>0
    \def\mathsection  {\mt@al@hexsy\mathcal{78}}
    \def\dagger       {\mt@al@hexsy\mathcal{79}}
    \def\ddagger      {\mt@al@hexsy\mathcal{7A}}
    \def\mathparagraph{\mt@al@hexsy\mathcal{7B}}
  \fi
```

If it's OK to use the 8r fonts, make them a "private" math alphabet. Here we do need substitution declarations. The .fd file has already been read (or the data supplied).

```
\or % 1: 8r is ok
  \DeclareMathAlphabet\mt@sy{8r}{\rmdefault}{\mddefault}{\updefault}
  \SetMathAlphabet \mt@sy {normal}{8r}%
                          {\rmdefault}{\mddefault}{\updefault}
  \SetMathAlphabet \mt@sy {bold}  {8r}%
                          {\rmdefault}{\bfdefault}{\updefault}
  \SetMathAlphabet \mt@sy {heavy} {8r}%
                          {\rmdefault}{\hvdefault}{\updefault}
  \def\mathparagraph{\mt@al@hexsy\mt@sy{B6}}
  \def\dagger       {\mt@al@hexsy\mt@sy{86}}
  \def\ddagger      {\mt@al@hexsy\mt@sy{87}}
  \def\mathsection  {\mt@al@hexsy\mt@sy{A7}}
  \ifnum\mt@sub=1 %
    \DeclareFontShape{8r}{\rmdefault}{\hvdefault}{\updefault}% heavy \rm in 8r SUBS
                        {<->ssub*\rmdefault/\bfdefault/\updefault}{}
  \fi
```

Or skip it…

```
\or % 2: nothing
  \PackageInfo{mt11p}{not doing anything with (math) \string\dagger, etc.}
\fi
```

Actually, the code above is suboptimal in the sense that is \encodingdefault happens to be 8r, we waste a math family. Hopefully, no heavy math user is running 8r…

6.9 Auxiliary macro

In a few instances we need a macro to get a specific character from a particular math alphabet, i.e., some thing like "`\math`*alph*`{\char"`*yy*`}`". You can't do that though, you would get `\mathchar"00yy`. The TeX math class 7 and `\fam` are used to get letters and digits from math alphabets, so we can just say `\math`*alph*`{\mathchar"70`*yy*`}`. The macro below is shorthand for the operation "set character ⟨*slot*⟩ from the font of the math alphabet `\math`*alph*".

```
\def\mt@al@hexsy#1#2{#1{\mathchar"70#2}}
```

where `#1` is the math alphabet macro (e.g., `\mathcal`) and `#2` is the hexadecimal number of the slot (no double quote).

6.10 Assorted Fixes, Odds & Ends

Here are a few assorted (sordid;-)) fixes.

Active Underscore Make the underscore character active so we can kern to `j`, `f` and `p` in subscripts. This is rather messy and probably breaks on occasion. The code was basically copied from the MathTime distribution.

```
\ifcase\mt@sb % case 0
   \PackageInfo{mt11p}{not using active underscore}
\or % case 1
   \ifnum\the\catcode`\_=8 %
      \ifnum\the\mathcode`\_=32768 % "8000
         \PackageInfo{mt11p}{redefining the underscore \string\catcode}
         \PackageInfo{mt11p}{\string\catcode\space of underscore made 12 (other)}
         \def\space@.{\futurelet\space@\relax}
         \space@. %
         \let\mt@orig@sb_
         \catcode`\_=\active
         \def_{\ifmmode\expandafter\sb@\else\expandafter\mt@orig@sb\fi}
         \let\sb_
         \def\sb@#1{\mt@orig@sb{\futurelet\next\sb@@#1}}
         \def\sb@@{\ifx\next\space@\def\next@. {\futurelet\next\sb@@}%
                    \else
                       \def\next@.{\ifx\next j\jadjust
                              \else\ifx\next f\jadjust
                              \else\ifx\next p\mkern-\@ne mu%
                                   \fi\fi\fi}%
                    \fi
                    \next@.}
         \@makeother\_
      \else % \the\mathcode`\_<>"8000 !
         \PackageWarning{mt11}{underscore did not have mathcode "8000;
                               \MessageBreak Underscore not made active}
      \fi
   \else % \the\catcode`\_ <>8 !
      \PackageWarning{mt11}{underscore was not of category 8;\MessageBreak
                            Underscore not made active}
   \fi
\else % oops
   \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@sb}%
```

```
                {Underscore not made active.}
\fi
```

`\jadjust` is just a small negative kern:

```
\def\jadjust{\mkern-\tw@ mu}
```

`\vdots`, `\ddots` and `\angle` might be missing `\m@th`s in the `\hbox`es.

```
\def\vdots{\vbox{\baselineskip4\p@\lineskiplimit\z@\kern6\p@
                \hbox{$\m@th.$}\hbox{$\m@th.$}\hbox{$\m@th.$}}}
\def\ddots{\mathinner{%
           \mkern1mu\raise7\p@\vbox{\kern7\p@\hbox{$\m@th.$}}\mkern2mu
           \raise4\p@\hbox{$\m@th.$}\mkern2mu%
           \raise\p@\hbox{$\m@th.$}\mkern1mu}}
```

The plainTEX definition of `\angle` has an `\hrule` height of 0.34pt; the MathTime distribution says “0.48pt”. Both of those numbers are suspiciously close to the default rule thickness of the (scaled) extention font. We define `\angle` to use the default rule thickness, i.e., `\fontdimen 8` of the appropriate extention (family `3`) font. This does not work in `\scriptscriptstyle`, but that case is unlikely to occur. The `.tfm` files have been slightly changed to make `\fontdimen 8` really be the rule thickness implied, e.g., by the height of the root glyphs.

```
\def\mt@angle#1#2{{\vbox{\ialign{$\m@th#1##$\crcr
            \not\mathrel{\mkern14mu}\crcr
            \noalign{\nointerlineskip}%
            \mkern2.5mu\leaders\hrule height1.05\fontdimen8#2 3 %
                            \hfill\mkern2.5mu\crcr}}}}
\def\angle{\mathchoice{\mt@angle{\scriptstyle}{\scriptfont}}
                      {\mt@angle{\scriptstyle}{\scriptfont}}
                      {\mt@angle{\scriptscriptstyle}{\scriptscriptfont}}
                      {\mt@angle{\scriptscriptstyle}{\scriptscriptfont}}}
```

The Tie Accent Usually, the `\t` accent (o͡o) is taken from an `OML` font (position `"7F`). MathTime has the tie in `symbols`, so we take it from there, and make it the dafault.

```
\DeclareTextAccent{\t}{LmtS}{"41}
\DeclareTextAccentDefault{\t}{LmtS}
```

`\textcircled` The large-ish circle for constructing ©, ®, etc., is usually taken from `OMS`. The `LmtS` fonts have the same glyph in the same slot, so just redefine the default to point to `LmtS`. A text symbol font might have ©, etc., available as single characters, e.g., the raw `8r` fonts do, but it’s not our business to deal with all that.

```
\DeclareTextAccentDefault{\textcircled}{LmtS}
\DeclareTextCommand{\textcircled}{LmtS}[1]{{%
   \ooalign{\hfil\raise.07ex\hbox{\upshape#1}\hfil\crcr\char13}}}
```

Misc. text symbols Some text versions of a few symbols (`\text`⟨*symb*⟩) point to `OM*` encodings; change the defaults. Those symbols that exists in `T1` will be taken from `T1` in text if that’s the `\encodingdefault`. Try to be nice even if

\encodingdefault is OT1 and point the defaults to MathTıme, so we don't needlessly resort to Computer Modern. Noone uses these definitions directly, but macros might.

```
\DeclareTextSymbol{\textless}{LmtL}{60}% afm names
\DeclareTextSymbol{\textgreater}{LmtL}{62}
\DeclareTextSymbol{\textperiodcentered}{LmtS}{1}
\DeclareTextSymbol{\textasteriskmath}{LmtS}{3}% centered
\DeclareTextSymbol{\textopenbullet}{LmtS}{14}
\DeclareTextSymbol{\textbullet}{LmtS}{15}
\DeclareTextSymbol{\textbraceleft}{LmtS}{102}
\DeclareTextSymbol{\textbraceright}{LmtS}{103}
\DeclareTextSymbol{\textbar}{LmtS}{106}
\DeclareTextSymbol{\textbackslash}{LmtS}{110}
```

Make these the default:

```
\DeclareTextSymbolDefault{\textperiodcentered}{LmtS}
\DeclareTextSymbolDefault{\textasteriskmath}{LmtS}
\DeclareTextSymbolDefault{\textopenbullet}{LmtS}
\DeclareTextSymbolDefault{\textbullet}{LmtS}
\DeclareTextSymbolDefault{\textbraceleft}{LmtS}
\DeclareTextSymbolDefault{\textbraceright}{LmtS}
\DeclareTextSymbolDefault{\textbar}{LmtS}
\DeclareTextSymbolDefault{\textbackslash}{LmtS}
\DeclareTextSymbolDefault{\textless}{LmtL}
\DeclareTextSymbolDefault{\textgreater}{LmtL}
```

Oldstyle Digits (See Sec. 3.10). MathTıme has no oldstyle digits, so we grab the ones from OML. I'm not sure it's useful to have this command in math mode, but we might as well do that too, though not via a math family—we define a private \math@OS alphabet instead. Read the .fd file, in case the format did no preloading. nomathOS disables the math family.

```
\ifcase\mt@osdg % case 0
   \def\math@OS#1{\PackageError{mt11p}{oldstyle digits not loaded for math}%
                                    {Don't use the nomathOS option.}%
                 {\hbox{\oldstylenums{#1}}}}% as good as anything else
\or % case 1
   \DeclareMathAlphabet\math@OS {OML}{cmm}{m}{it}
   \SetMathAlphabet    \math@OS {normal}{OML}{cmm}{m}{it}
   \SetMathAlphabet    \math@OS {bold}  {OML}{cmm}{b}{it}
   \SetMathAlphabet    \math@OS {heavy} {OML}{cmm}{\hvdefault}{\itdefault}
\else % oops
   \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@osdg}%
                {Oldsyle digits in math are hosed.}
\fi
\ifnum\mt@sub=1 %
   \InputIfFileExists{omlcmm.fd}\relax % grrr...
                        {\InputIfFileExists{OMLcmm.fd}\relax
                        {\PackageError{mt11p}{No OMLcmm.fd file}%
                                 {Your are missing a basic distribution file. %
                                              Check the LaTeX installation.}}}
   \DeclareFontShape{OML}{cmm}{\hvdefault}{\itdefault}%
                    {<->ssub*cmm/b/it}{}%      heavy cmmi SUBS
\fi
```

The `\oldstylenums` macro just picks between using the `OML` font as a text font, or the `\math@OS` math alphabet. If the argument contains things other than digits, the result will be amusing…

```
\def\oldstylenums#1{%
     \ifmmode
        \math@OS{#1}%
     \else
        {\usefont{OML}{cmm}{\f@series}{it}#1}%
     \fi}
```

6.11 `\boldsymbol` and `\heavysymbol`

As mentioned above, we assumed that formulæ do not require simultaneous access to fonts in different math versions, or, more precisely, that a single math list need not contain several weights of, e.g., the `symbols` fonts. In other words, that it is not worth while to *simultaneously* have several weights of, e.g., the `symbols` fonts available as "`\mathchars`". Instead we fetch the $\mathcal{AMS}$ package `amsbsy`, which defines a "`\boldsymbol`" macro and define a similar "`\heavysymbol`" macro. Don't do it if we're told not to, or if we can't find `amsbsy.sty` and don't overwrite existing definitions.

```
\ifcase\mt@bsy % case 0
   \providecommand{\boldsymbol}[1]{\PackageError{mt11p}%
                              {Get or load the package amsbsy
                               to use \string\boldsymbol}{See above.}#1\relax}
   \providecommand{\heavysymbol}[1]{\PackageError{mt11p}%
                              {Get or load the package amsbsy
                               to use \string\heavysymbol}{See above.}#1\relax}
   \providecommand{\normalsymbol}[1]{\PackageError{mt11p}%
                              {Get or load the package amsbsy
                               to use \string\normalsymbol}{See above.}#1\relax}
\or % case 1
   \IfFileExists{amsbsy.sty}%
        {\RequirePackage{amsbsy}%
         \DeclareRobustCommand{\heavysymbol}[1]{%
             \begingroup
             \let\@nomath\@gobble \mathversion{heavy}%
             \math@atom{##1}{%
             \mathchoice
               {\hbox{$\m@th\displaystyle##1$}}%
               {\hbox{$\m@th\textstyle##1$}}%
               {\hbox{$\m@th\scriptstyle##1$}}%
               {\hbox{$\m@th\scriptscriptstyle##1$}}}%
             \endgroup}%
         \DeclareRobustCommand{\normalsymbol}[1]{%
             \begingroup
             \let\@nomath\@gobble \mathversion{normal}%
             \math@atom{##1}{%
             \mathchoice
               {\hbox{$\m@th\displaystyle##1$}}%
               {\hbox{$\m@th\textstyle##1$}}%
               {\hbox{$\m@th\scriptstyle##1$}}%
               {\hbox{$\m@th\scriptscriptstyle##1$}}}%
             \endgroup}%
```

```
        }% using amsbsy.sty
        {\PackageError{mt11p}{No file ‘amsbsy.sty’ found}%
                  {Use the ‘nobsy’ option if you don’t have or \MessageBreak
                  don’t want the AMS \string\boldsymbol\space macro.}
        \let\boldsymbol\@firstofone
        \let\heavysymbol\@firstofone
        \let\normalsymbol\@firstofone
        }% no amsbsy.sty
\else % oops
   \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@bsy}%
               {The \string\boldsymbol\space macro is probably hosed.}
\fi
```

6.12 Examples of new math symbol definitions

Note that code in this section, and some of the code in the following sections is not included in the package `.sty` file. It's meant as an example of things that could be put into a `mt11p.cfg` file.

Here follow a few examples of defining more math symbols in order to easily and efficiently mix symbols normally found in different math versions. Note that it is rarely worth it perform such definitions—not even the $\mathcal{AMS}$ document classes do it. For more details about NFSS see the standard LaTeX documentation file `fntguide.tex`.

Upright lower case Greek letters as symbols. The font set includes a `letters` font with upright lower case Greek letters (l.c.G.). Usually *italic* or *slanted* l.c.G. are used as symbols, though recently it's possible see the PostScript `Symbols` font (upright; noone bothers to skew it) used a lot, but that is a more a sign of weak software. The upright l.c.G. letters are made available in both text and math (Sec. 3.9). It would be possible to make them available as math *symbols* by using the code below. (The code is not available as an option, because there is no common definition for how the symbols should be named; for illustration we call the symbols `\upalpha`, `\upbeta`, etc.)

First define a math symbol font:

```
\DeclareSymbolFont{upgreek}{LmtL}{mtl}{\mddefault}{\uidefault}
```

Don't redefine `\mathgr` to point to the `upgreek` family, because the encodings aren't even close.

Next set the math versions:

```
\SetSymbolFont{upgreek}{normal}{LmtL}{mtl}{\mddefault}{\uidefault}
\SetSymbolFont{upgreek}{bold}{LmtL}{mtl}{\bfdefault}{\uidefault}
\SetSymbolFont{upgreek}{heavy}{LmtL}{mtl}{\hvdefault}{\uidefault}
```

If this step is omitted, the `upgreek` symbol font family will be the same in all math versions.

Lastly, make the symbol declarations (choose names that don't conflict with existing symbols):

```
\DeclareMathSymbol{\upalpha}{\mathord}{upgreek}{"0B}
```

```
\DeclareMathSymbol{\upbeta}{\mathord}{upgreek}{"0C}
\DeclareMathSymbol{\upgamma}{\mathord}{upgreek}{"0D}
\DeclareMathSymbol{\updelta}{\mathord}{upgreek}{"0E}
\DeclareMathSymbol{\upepsilon}{\mathord}{upgreek}{"0F}
\DeclareMathSymbol{\upzeta}{\mathord}{upgreek}{"10}
\DeclareMathSymbol{\upeta}{\mathord}{upgreek}{"11}
\DeclareMathSymbol{\uptheta}{\mathord}{upgreek}{"12}
\DeclareMathSymbol{\upiota}{\mathord}{upgreek}{"13}
\DeclareMathSymbol{\upkappa}{\mathord}{upgreek}{"14}
\DeclareMathSymbol{\uplambda}{\mathord}{upgreek}{"15}
\DeclareMathSymbol{\upmu}{\mathord}{upgreek}{"16}
\DeclareMathSymbol{\upnu}{\mathord}{upgreek}{"17}
\DeclareMathSymbol{\upxi}{\mathord}{upgreek}{"18}
\DeclareMathSymbol{\uppi}{\mathord}{upgreek}{"19}
\DeclareMathSymbol{\uprho}{\mathord}{upgreek}{"1A}
\DeclareMathSymbol{\upsigma}{\mathord}{upgreek}{"1B}
\DeclareMathSymbol{\uptau}{\mathord}{upgreek}{"1C}
\DeclareMathSymbol{\upupsilon}{\mathord}{upgreek}{"1D}
\DeclareMathSymbol{\upphi}{\mathord}{upgreek}{"1E}
\DeclareMathSymbol{\upchi}{\mathord}{upgreek}{"1F}
\DeclareMathSymbol{\uppsi}{\mathord}{upgreek}{"20}
\DeclareMathSymbol{\upomega}{\mathord}{upgreek}{"21}
\DeclareMathSymbol{\upvarepsilon}{\mathord}{upgreek}{"22}
\DeclareMathSymbol{\upvartheta}{\mathord}{upgreek}{"23}
\DeclareMathSymbol{\upvarpi}{\mathord}{upgreek}{"24}
\DeclareMathSymbol{\upvarrho}{\mathord}{upgreek}{"25}
\DeclareMathSymbol{\upvarsigma}{\mathord}{upgreek}{"26}
\DeclareMathSymbol{\upvarphi}{\mathord}{upgreek}{"27}
\DeclareMathSymbol{\upvarkappa}{\mathord}{upgreek}{'176}
```

That is the complete set; of course, you need not define them all.

Additional **math accents** can be defined in an entirely analogous way by using `\DeclareMathAccent`. NFSS can define two types of accents: `\mathord`, which are taken from a fixed font, and `\mathalpha`, which change with math alphabets. A specially declared accent will probably be `\mathord`; do not use `\mathalpha` unles you know what you're doing. An example for a math symbol font family called `boldsyms` which has an accent in slot `"52` is

```
\DeclareMathAccent{\boldddot}{\mathord}{boldsyms}{"52}
```

Declaring additional math delimiters Although any combination of symbols from various math versions can be achieved through the use of `\boldsymbol` and `\heavysymbol`, it can be inefficient and awkward to have to do so frequently. Particularly tedious are delimiters, which either require freezing their size (`\boldsymbol{\Bigg(}`), or cryptic constructs involving `\vphantom`s. In this case it is also possible to define special math symbols that work as expected. The procedure is similar to the one for declaring upright Greek letters as symbols, an outline follows. TeX's math workings are described in D. E. Knuth's *The TeXbook* and, e.g., in *TeX by topic* by V. Eijkhout.

First declare a new symbol font, e.g.,

```
\DeclareSymbolFont{hvlargesymbols}{OMX}{mtx}{\hvdefault}{\updefault}
```

Next define the delimiter itself:

`\DeclareMathDelimiter{`⟨*cmd*⟩`}{`⟨*math type*⟩`} {`⟨*small*⟩`}{`⟨*small slot*⟩`}{`⟨*large*⟩`}{`⟨*large slot*⟩`}`

where `{`⟨*small*⟩`}` and `{`⟨*large*⟩`}` are two symbol font math families. If you're defining a new delimiter from some special font, you probably only have one variant (or more precisely one chain of linked delimiters). This command does not allow making a variant null (`000`) without assuming that `operators` is math family `0`, but it is possible to make sure the small variant "succeeds" in finding a delimiter (e.g., an extensible delimiter will always terminate a search), so that the large variant is irrelevant.

For example, assuming there is a math symbol font family called `hvlargesymbols` (see above), `\lbraceHV` and `\rbraceHV` will produce correctly sized heavy braces:

```
\DeclareMathDelimiter{\lbraceHV}{\mathopen}{hvlargesymbols}{"08}%
                                           {hvlargesymbols}{"08}
\DeclareMathDelimiter{\lbraceHV}{\mathclose}{hvlargesymbols}{"09}%
                                           {hvlargesymbols}{"09}
```

(The small and large variants are identical, but since the small one will always succeed—the last delimiter in the chain is extendible—the large one never comes into play.) `\delcode`s are not normally changed.

Running out of math families TEX with NFSS is limited to 16 math font families in any single math version, and it's not all too difficult to run into the limit. With the default options, it's possible to load *four* more math symbol fonts/alphabets. (A `\DeclareSymbolFont` and an accompanying `\DeclareSymbolFontAlphabet` take up one family.) The options `nomathgr` and `nomathOS` will each cause one less math family to be used. The `cmcal` option uses one math family more than `mtcal` or `nomtms` (as long as `8r` is OK for the four infamous symbols). If you do run out of math families, declare a new math version, or define infrequently used symbols as explicit font/character references in text mode (in an `\hbox`), possibly wrapped in `\mathchoice` and an explicit class command, e.g., `\mathbin`. See `fntguide.tex`.

6.13 Unused glyphs

This package leaves no unique MathTime glyphs inaccessible. A few glyphs are repeated more than once in a font, or are present in more than one font. It makes no difference to TEX whether a font is taken from `letters` or `symbols`.

It would be possible to take the small square root sign from the `largesymbols` font, rather than from the `symbols` font. That `largesymbols` radical (`"9F`) is, however, completely identical to the one in `symbols`, so it's rather pointless. The command would be

```
\DeclareMathRadical{\sqrtsign}{largesymbols}{"9F}{largesymbols}{"70}
```

The extra root sign in mtsyn (not present in mtsy) is identical to the usual one, except for a vertical shift. Since a ‘radical’ sign is normally of class 1 (\mathop), it gets vertically centered anyway. This alternate radical glyph could be used by defining something like

```
\DeclareMathSymbol{\varsurd}{\mathop}{symbols}{"84}
```

6.14 muskip Assignments

A little smaller than for Computer Modern. The original setting is

```
\thinmuskip=3mu
\medmuskip=4mu plus 2mu minus 4mu
\thickmuskip=5mu plus 5mu
```

and we’ll use

```
\thinmuskip=2mu␣\relax
\medmuskip=2.5mu␣plus␣1mu␣minus␣1mu␣\relax
\thickmuskip=4mu␣plus␣1.5mu␣minus␣1mu␣\relax
```

6.15 Local Configuration File

Provide a hook to load anything important or complicated enough to be inappropriate for the main input file.

```
\InputIfFileExists{mt11p.cfg}%
          {\typeout{*************************************^^J%
                    * Local config file mt11p.cfg loaded *^^J%
                    *************************************}%
           }{}
```

6.16 Cleanup

Do the resizing now:

```
\mt@exit
```

Restore any crazy catcodes.

```
\catcode`\"\mt@quotedbl
\catcode`\'\mt@quotesinglr
\catcode`\:\mt@colon
\catcode`\?\mt@question
\catcode`\;\mt@semicolon
\catcode`\!\mt@exclam
```

Make the various scratch macros, etc., we used \undefined, but keep \mt@cal and \mt@ops, in case we need to decide something based on them. A ‘real’ input file should not do such a thing, but a demo document might.

```
\let\mt@quotedbl  \undefined
\let\mt@quotesinglr\undefined
\let\mt@colon     \undefined
\let\mt@question  \undefined
\let\mt@semicolon \undefined
\let\mt@exclam    \undefined
```

```
\let\mt@exit      \undefined
\let\MathTimeSizes\undefine
\let\mt@enc       \undefined
\let\times@style  \undefined
\let\mt@fd@check  \undefined
\let\mt@accs      \undefined
\let\mt@accent@make\undefined
\let\mt@hbar      \undefined
\let\mt@eightR    \undefined
\let\mt@bsy       \undefined
\let\mt@sb        \undefined
\let\mt@mgrs      \undefined
\let\mt@osdg      \undefined
\let\mt@assign@del\undefined
 %\let\mt@cal       \undefined
 %\let\mt@ops       \undefined
```

...and we're done.