

The gensymb package for LaTeX 2_ε

Walter Schmidt*

(v1.0 – 2003/07/02)

1 The problem

Typesetting units of measurement with LaTeX is sometimes difficult. Why? For instance, most (but not all) typefaces have an upright μ and also a degree symbol, but there is no obvious way to use these in math mode. On the other hand, an upright Ω is part of all math fonts for TeX, but most text fonts are lacking the corresponding `\textohm`. Thus, it is not only impossible to use the same notation in text and in formulas; depending on the fonts used, it may even be impossible to typeset units properly at all, if you are restricted to the means of ‘standard’ LaTeX.

2 The macro package

The gensymb package provides a number of ‘generic’ macros, which produce the same output in text and math mode:

<code>\degree</code>	<code>\celsius</code>	<code>\perthousand</code>	<code>\ohm</code>	<code>\micro</code>
°	°C	‰	Ω	μ

Thus, you can write, for instance:

```
$ \lambda = 10.6\,\micro\mathrm{m} $  
... a wavelength of 10.6\,\micro m
```

With the help of the package units, you can even apply exactly the same notation in text and formulas:

```
$ \lambda = \unit[10.6]{\micro m} $  
... a wavelength of \unit[10.6]{\micro m}
```

Under normal circumstances you should use gensymb in conjunction with the textcomp package. In this case the above symbols are taken from a text font using the TS1 (text companion) encoding—even in math. As a result, they will exhibit the style of the surrounding text or of the `\mathrm` alphabet, respectively.

When textcomp is *not* used (for instance, because your text fonts are not available with TS1 encoding), the gensymb package tries to emulate the above symbols using what is available in the math fonts. However, the symbols μ and ‰ cannot be faked, and the package will issue appropriate warning messages. The option `upmu` is a workaround to provide at least the μ —see below.

*w-a-schmidt@arcor.de

The symbol `\ohm` is particular, too: Loading the `textcomp` package does *not ensure* that the text fonts actually include an Ω . Unfortunately, many fonts don't. LaTeX cannot detect this situation in advance, and the command `\ohm` will print some garbage then—possibly without any error message. A workaround is to use the options `Omega` or `Upomega` described below.

The default behavior of the `gensymb` package can be modified through the following options:

upmu With this option, the `\micro` uses internally a math symbol with the name `\upmu`. You must additionally load the package `upgreek`¹ to provide it (or define `\upmu` yourself). This option is useful, when your text fonts are not available with TS1 (text companion) encoding, or when their `\textmu` is ugly or broken. The drawback is, that the style of this μ will not vary with the surrounding font. It is always upright roman, regardless of whether the surrounding text is sans serif, bold or whatever.

Omega makes `\ohm` always use the upright Greek Omega from the current math font, regardless of whether `textcomp` is loaded. This makes sense, when there is no Ω in your text font(s). The option works always, but the style of the Omega in text will not vary then.

Various packages provide an option `slantedGreek` to make uppercase Greek letters in formulas slanted. You may safely use this option in conjunction with the `Omega` option of the package `gensymb`: `\ohm` will be upright, though!

Upomega A math symbol with the name `\Upomega` will be used to for the `\ohm`. You must additionally load the package `upgreek` to define it (or define `\Upomega` yourself). Again, the style of this `\ohm` in text never changes. The option may be useful, when the text font does not include a `\textomega` of its own and the mathematical `\Omega` is not a suitable substitute, for one reason or another.

3 Examples

The optimum case

With text fonts providing μ as well as Ω in the text companion encoding, the default behavior of `gensymb` is appropriate. This holds, e.g., for the EC/TC fonts (i.e., Computer Modern with T1 and TS1 encoding), for Palatino (in conjunction with PSNFSS version 9 and later), and for most font families from MicroPress:

```
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{gensymb}
```

The normal case

Many commercial text fonts have a correct ‘micro’ symbol, but no Omega. In this case the package can be loaded with the `Omega` option, to make sure that the mathematical Omega is used instead:

¹ available from the same CTAN directory as `gensymb`

```

\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx} % for instance
\usepackage[Omega]{gensymb}

```

Using the ‘classical’ CM fonts

When you are restricted to the classical Computer Modern fonts with OT1 encoding, you can still take the Omega from the math fonts, but you need to take the upright mu from an extra font such as Euler Roman. Notice, however, that `\perthousand` is not made available. You may try `\permil` instead, which is provided in the WASY fonts.

```

\usepackage[Euler]{upgreek}
\usepackage[Omega, upmu]{gensymb}
\usepackage{wasysym}
\let\perthousand=\permil

```

A special case

Imagine that you are using the Bitstream Charter text fonts in conjunction with Euler-VM for math. Charter, like most typefaces from Bitstream, has a wrong mu symbol. The missing Omega could be taken from the math font (Euler), but it would not go well with Charter in a combination such as $M\Omega$. As a workaround, load the upgreek package and take both `\micro` and `\ohm` from the ‘Adobe Symbol’ font:

```

\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{charter,eulervm}
\usepackage[Symbol]{upgreek}
\usepackage[Upomega, upmu]{gensymb}

```

4 Using gensymb together with other packages

- `gensymb` goes well with the package units, but it cannot be used in conjunction with the package `Slunits`, because the latter has its own means to typeset prefixes.
- Using the package `mathcomp` together with `gensymb` is possible, if you need the additional ‘math companion’ symbols. Both packages will use only one common math symbol font.

5 Known deficiencies

The current version of the package does not work properly in conjunction with the LY1 font encoding. This will possibly be fixed in a future release, if there is an actual need.