

# The `hepparticles` package for L<sup>A</sup>T<sub>E</sub>X\*

Andy Buckley, [andy@insectnation.org](mailto:andy@insectnation.org)

February 22, 2005

## Abstract

This package provides macros for typesetting high energy physics particle names in a consistent, semantic and aesthetically pleasing manner, as well as fixing problems with math-mode boldness problems in section titles. You may also be interested in the `heppennames` and `hepnicensames` packages, which use this one to provide a large set of pre-existing particle names.

## 1 Motivation

Typesetting the names of high-energy fundamental particles (and their elementary composites) is well-defined by a set of rules:

- The basic particle name consists of a large Roman or Greek symbol with optional following sub- and super-scripts.
- Depending on convention, the symbols for particles are either italic or upright. Using the latter convention, the symbols should be upright when describing a specific particle; if describing a generic class of particles they should be italicised (where possible). In italic and bold contexts the symbols should adapt by becoming bold and italic themselves where possible.
- Anti-particles are written with a bar on top of the main symbol (but for aesthetic reasons the bar does not extend above the sub- and super-scripts).

---

\*This document describes `hepparticles` as of version 1.4

- Supersymmetric partners of Standard Model particles are written as for anti-particles but with a tilde in place of the bar. SUSY anti-particles (though the use of symbols to represent them is currently uncommon) may be written with a bar above the tilde.
- Resonant states may sport an extra resonance specifier consisting of a value in parentheses and optional following sub- and super-scripts. This follows the main particle name.

Several issues arise when typesetting these particle names in standard  $\text{\LaTeX}$ : for starters the requirement of sub- and super-scripts and the need to use Greek symbols forces us into math mode. But math mode does not usually follow the surrounding text context for boldness and italicism: this has been fixed in this specific case by use of the `maybemath` package. Secondly, the positions of super-scripts with overlines and tildes are affected: this is also corrected by this package. By request, the particle typesetting conventions may be specified as a package option.

## 2 Commands

The commands available are divided into two groups: those which provide appropriate typeset output for a given semantic description and those which are purely designed to implement that typesetting, several of which are exposed to public use for convenience.

### 2.1 Semantic commands

- For generic particle names e.g. all positively charged leptons:
 

```
\HepGenParticle{main}{subscript}{superscript}
```

```
\HepGenAntiParticle{main}{subscript}{superscript}
```
- For concrete particle names:
 

```
\HepParticle{main}{subscript}{superscript}
```

```
\HepAntiParticle{main}{subscript}{superscript}
```
- For supersymmetric “sparticles”:
 

```
\HepGenSusyParticle{main}{subscript}{superscript}
```

```
\HepSusyParticle{main}{subscript}{superscript}
```

```
\HepGenSusyAntiParticle{main}{subscript}{superscript}
```

```
\HepSusyAntiParticle{main}{subscript}{superscript}
```

- For resonance specifiers (just the extra terms):

```
\HepResonanceMassTerm{mainterm}{subscript}{superscript}
\HepResonanceSpecTerm{mainterm}{subscript}{superscript}
```

- For a full particle name and resonance specification:

```
\HepParticleResonance{name}{mass}{massSub}{massSup}
\HepParticleResonanceFull{main}{sub}{sup}{mass}{massSub}{massSup}
\HepParticleResonanceFormal{name}\
{mass}{massSub}{massSup}{spec}{specSub}{specSup}
\HepParticleResonanceFormalFull{main}{sub}{sup}\
{mass}{massSub}{massSup}{spec}{specSub}{specSup}
(hurrah, we hit the TEX 9-argument limit!)
```

- And finally, for containing processes describing particle evolution;

```
\HepProcess{iParticles \HepTo fParticles}
```

where `\HepTo` is a normal `\to` arrow with a bit of extra space.<sup>1</sup>

giving typeset particle names as follows:

- Normal particles:  $B_d^0, \bar{B}_d^0$
- Generic particles:  $q_d, \bar{\ell}_\mu$
- SUSY particles:  $\tilde{\chi}_1, \tilde{q}_2$
- Resonances:  $J/\psi(1S)^*$
- Process:  $B_d^0 \rightarrow K^- \pi^+$ .

## 2.2 Primitive commands

A primitive command, `\HepParticleStruct`, is provided for typesetting particle-like structures with a main term and super and sub-scripts with appropriate automatic changing of the text style into bold, italic and upright fonts. It also accounts for most vertical spacing problems in the sub- and super-scripts, largely due to empty script boxes. In addition,

---

<sup>1</sup>In fact, this space will only be added when not compiling via PDF<sub>T</sub>E<sub>X</sub>. This is due to a conflict not yet understood \*sigh\*.

three “styles”: the wrapper macros that apply appropriate text stylings are available publicly. The implementation of the semantic commands is built round conditionally using these styles within the structure macro.

These macros can be used for implementing particle-like structures with custom requirements and it’s advised that you take a look at the internals of `hepparticles.sty` if you’re planning on using them.

- Style which only applies conditional boldness:

`\HepNormStyle{text}`

- Style appropriate for generic particle names:

`\HepGenStyle{text}`

- Style appropriate for concrete particle names:

`\HepConStyle{text}`

- Particle structure:

`\HepParticleStruct{main}{sub}{sup}`

### 3 Package options

By request, the package now typesets particles in italic as well as upright convention. The choice of convention can be made when the package is loaded with the `italic` and `notitalic` options, e.g.

`\usepackage[italic]{hepparticles}`. The default mode is upright (i.e. `notitalic`).

In addition, the `forceit` option will force *everything* in particle names to be italic, even if they aren’t normally italic in math mode (such as Arabic numerals). Note that the italic font that will appear here is that used by `\mathit` and so will appear more script-like than normal math mode. I can’t say that I recommend using this option, but it’s there for flexibility’s sake.

Finally, a pair of options, `maybess` and `noss`, are available: using `maybess` will allow particle names to be typeset in sans-serif if the surrounding context is sans-serif and `noss` has the converse effect. Note that since there is no italic sans-serif math font in LaTeX, generic particle names will not be typeset in italic sans font. Maybe this behaviour will change in future if there’s lots of enthusiasm for a fix. However, it looks pretty good at the moment and I suspect most people will want sans-serif particle names in sans documents, so `maybess` is set by default.

## 4 Known problems

Since this package messes around quite a bit with the sub- and super-scripts, not everything you might want to do can be done. Sorry...

- To make the macros a bit “safer”, the output is wrapped in a pair of braces — i.e. it’s intended as a packaged unit. Hence you can’t add new indices using math mode scripting... this example hopefully illustrates what I mean in that it doesn’t accomplish the intended effect of placing the  $i$  subscript directly underneath the B meson superscript.

$\$ \backslash \text{HepParticle}\{B\}\{\{+\}_i \$ \Rightarrow B^+_i$

I don’t consider this a problem — the package structures already manoeuvre the vertical positioning if the scripts so much that compliance with externally applied indices is pretty much guaranteed not to happen. If you want this sort of effect then you should use the `\HepParticleStruct` macro.

- When putting particle names in sub or superscripts it’s a good idea to wrap them in braces. I’d like to be able to avoid this requirement but wrapping everything in braces, despite removing compile errors, hasn’t done the trick of actually making the sub/sup-script work as intended. Please let me know how to do it, should the answer spring to mind...

## 5 Installation

**Requirements:** You will need to be using a  $\text{\LaTeX} 2_\epsilon$  system to use `hepparticles`. Hopefully this isn’t a problem — I wasn’t feeling up to writing a Plain  $\text{\TeX}$  version! In addition, you’ll need the `maybemath` package (get a recent version: the first release has bugs which were only discovered when writing the 1.4 version of this package).

To install, simply copy the `hepparticles.sty` file into a location in your `LATEXINPUTS` path. Tada!

Any feedback is appreciated! Email to [andy@insectnation.org](mailto:andy@insectnation.org), please.