

# An essential guide to $\text{\LaTeX}$ 2 $_{\epsilon}$ usage

l2tabuen v1.8.3

Original German version\*  
by Mark Trettin<sup>†</sup>

English translation  
by Jürgen Fenn<sup>‡</sup>

December 20, 2004

## Abstract

This is the English version of l2tabu, focusing on obsolete commands and packages, and demonstrating the most severe mistakes most  $\text{\LaTeX}$  users are prone to make. You should read this guide if you want to improve on your  $\text{\LaTeX}$  code.

Please see page 2 for legal notice.

## Acknowledgements

Reading the German-language  $\text{\TeX}$  newsgroup [de.comp.text.tex](http://de.comp.text.tex) one of us (Mark Trettin) found that most discussions were about obsolete or, say, ‘bad’ packages, and commands. So he decided to write a brief summary to supply a practical guide to  $\text{\LaTeX}$ . His paper was called `altepakete.pdf` in the first place and soon it was praised by senior developers writing in the group. It is recommended for reading ever since. Later it was renamed by vote of participants in [de.comp.text.tex](http://de.comp.text.tex) to l2tabu, corresponding to l2kurz, the German title of lshort [12], and the German translation of ‘taboo’. This was about two years ago.<sup>1</sup> I (Jürgen Fenn) joined Mark later for translating this paper into English in order to help it spread to those users who do not speak German.

In this article we give a demonstration of the most common mistakes in using  $\text{\LaTeX}$ . We also explain how to avoid them. This overview is neither meant to replace introductions such as lshort [12] nor the De- $\text{\TeX}$ -FAQ [8, version 72] nor the UK FAQ [3, version 3.11]. Our goal is just to give a small overview of how to write ‘good’  $\text{\LaTeX}$  2 $_{\epsilon}$  code.

---

\* Based on the German version 1.8 of l2tabu.

<sup>†</sup> email: [Mark.Trettin@gmx.de](mailto:Mark.Trettin@gmx.de)

<sup>‡</sup> email: [juergen.fenn@gmx.de](mailto:juergen.fenn@gmx.de)

<sup>1</sup> altepakete was first announced on 18 February 2003 on [de.comp.text.tex](http://de.comp.text.tex).

## More translations of this paper

Please note that besides the German original ‘Das L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Sündenregister oder Veraltete Befehle, Pakete und andere Fehler. Tipps zu L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>’ and this English version, there are more translations of this paper. They all can be found in the respective subdirectories at

[CTAN:info/l2tabu/](http://CTAN:info/l2tabu/)

So far l2tabu has been translated into English, French, and Italian.

## How to get in touch with the authors

We are grateful for any suggestions, improvements, or comments. Please address your emails directly to the translator of the English<sup>2</sup>, the French<sup>3</sup>, or the Italian<sup>4</sup> version respectively.

Please tell us whether you have found l2tabu useful. We rely on your feedback for improving our guide.

## Thanks to...

... Ralf Angeli, Christoph Bier, Christian Faulhammer, Jürgen Fenn, Ulrike Fischer, Yvon Henel, Yvonne Hoffmüller, David Kastrup, Markus Kohm, Thomas Lotze, Frank Mittelbach, Heiko Oberdiek, Walter Schmidt, Stefan Stoll, Knut Wenzig, Emanuele Zannarini, and Reinhard Zierke for tips, remarks, and corrections of the German original version.

## Contributors to the English translation

Barbara Beeton, Christoph Bier, Klas Elmgren, Yvon Henel, Hendrik Maryns, José Carlos Santos, Walter Schmidt, Maarten Sneepe, Stefan Ulrich, Knut Wenzig, Bruno Wöhrer, and Federico Zenith have contributed to the English version, making suggestions, or encouraging development.

If we have forgotten anyone please send an email to the maintainer of the respective language version.

## Legal notice

Copyright © 2004 by Mark Trettin and Jürgen Fenn.

This material may be distributed only subject to the terms and conditions set forth in the *Open Publication License*, v1.0 or later. The latest version is available at

<http://www.opencontent.org/openpub/>

---

2 email: [juergen.fenn@gmx.de](mailto:juergen.fenn@gmx.de). – Download of l2tabu from: [CTAN:info/l2tabu/english/](http://CTAN:info/l2tabu/english/)

3 See the French translation l2tabufr by Yvon Henel at [CTAN:info/l2tabu/french/](http://CTAN:info/l2tabu/french/)

4 See the Italian translation l2tabuit by Emanuele Zannarini at [CTAN:info/l2tabu/italian/](http://CTAN:info/l2tabu/italian/)

## Contents

<b>1</b>	<b>‘Deadly sins’—The most severe mistakes in using <math>\text{\LaTeX}2_{\epsilon}</math></b>	<b>4</b>
1.1	<i>a4.sty</i> , <i>a4wide.sty</i> . . . . .	4
1.2	Modifying page layout . . . . .	4
1.3	Changing packages and document classes . . . . .	4
1.4	Changing inter-line space using <code>\baselinestretch</code> . . . . .	5
1.5	Parindent and the spread between paragraphs ( <code>\parindent</code> , <code>\parskip</code> ) . .	5
1.6	Separating maths formulae from continuous text using <code>\$\$\dots\$\$</code> . . . . .	6
1.7	<code>\def</code> vs. <code>\newcommand</code> . . . . .	6
1.8	Should I use <code>\sloppy</code> ? . . . . .	6
<b>2</b>	<b>Some obsolete commands and packages</b>	<b>7</b>
2.1	Commands . . . . .	7
2.1.1	Changing font style . . . . .	7
2.1.2	Mathematical fractions ( <code>\over</code> vs. <code>\frac</code> ) . . . . .	8
2.1.3	Centring text using <code>\centerline</code> . . . . .	8
2.2	Class files and packages . . . . .	9
2.2.1	<i>scrlettr.cls</i> vs. <i>scrlettr2.cls</i> . . . . .	9
2.2.2	<i>epsf.sty</i> , <i>psfig.sty</i> , <i>epsfig.sty</i> vs. <i>graphics.sty</i> , <i>graphicx.sty</i> . . . . .	9
2.2.3	<i>doubleSPACE.sty</i> vs. <i>setspace.sty</i> . . . . .	9
2.2.4	<i>fancyheadings.sty</i> , <i>scrpage.sty</i> vs. <i>fancyhdr.sty</i> , <i>scrpage2.sty</i> . . . . .	10
2.2.5	The <i>caption.sty</i> family of packages . . . . .	10
2.2.6	<i>isolatin.sty</i> , <i>umlaut.sty</i> vs. <i>inputenc.sty</i> . . . . .	10
2.2.7	<i>t1enc.sty</i> vs. <i>fontenc.sty</i> . . . . .	11
2.2.8	<i>natdin.bst</i> vs. <i>dinat.bst</i> . . . . .	11
2.3	Fonts . . . . .	12
2.3.1	<i>times.sty</i> . . . . .	12
2.3.2	<i>mathptm.sty</i> . . . . .	12
2.3.3	<i>pslatex.sty</i> . . . . .	12
2.3.4	<i>palatino.sty</i> . . . . .	12
2.3.5	<i>mathppl.sty</i> . . . . .	13
2.3.6	Typesetting upright greek letters . . . . .	13
2.3.7	<i>euler.sty</i> vs. <i>eulervm.sty</i> . . . . .	14
<b>3</b>	<b>Miscellaneous</b>	<b>14</b>
3.1	Floats—‘figure’, ‘table’ . . . . .	14
3.2	The appendix . . . . .	14
3.3	Mathematical typesetting . . . . .	14
3.4	How to use <code>\graphicspath</code> . . . . .	15
3.5	Language-specific macros – <code>\*name</code> . . . . .	16
<b>A</b>	<b>An example illustrating the effect of the <code>\sloppy</code> command</b>	<b>17</b>
	<b>References</b>	<b>18</b>

## 1 ‘Deadly sins’—The most severe mistakes in using L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

In this section we probably have gathered together the most severe mistakes that appear again and again in `de.comp.text.tex`, leaving regulars either with a flush of anger or weeping with tears in their eyes. ; –)

### 1.1 *a4.sty*, *a4wide.sty*

Do not use these ‘two’ packages any longer. You should delete them without replacement from your L<sup>A</sup>T<sub>E</sub>X source. Use the class option `a4paper` instead. Speaking in terms of typography these packages, or others similar to these do not provide good layout. What is even worse, there is more than one version of these packages around, and different versions of those packages are incompatible with one another, providing deviating settings for page margins. So you may not trust that your document will look the same – or just as bad? – when being compiled on someone else’s system when exchanging L<sup>A</sup>T<sub>E</sub>X source.

Replace: *a4.sty*, or *a4wide.sty* by `class option a4paper`

### 1.2 Modifying page layout

Page margins produced by the standard classes (*article.cls*, *report.cls*, *book.cls*) are often deemed too wide by European users printing on A4 paper. They should use the corresponding classes from the KOMA-Script bundle instead (*scrartcl.cls*, *scrreprt.cls*, *scrbook.cls*). These classes have been made with a European point of view on typography in mind. You can also use *typearea.sty* which is part of KOMA-Script with any other document class. The documentation included in the bundle provides some more information. Indeed, this very paper was typeset using *scrartcl.cls*.

If you really need to use page margins altogether different from the ones produced by *typearea.sty* use *geometry.sty*, or *vmargin.sty* because these packages provide reasonable proportions in setting page margins. Do not use `\oddsidemargin` or similar commands for modifying page layout.

Under no circumstances change `\hoffset`, or `\voffset`, unless you really understand what T<sub>E</sub>X is doing here.

### 1.3 Changing packages and document classes

Never modify L<sup>A</sup>T<sub>E</sub>X class files (*e.g.*, *article.cls*, *scrbook.cls*) or packages (style files, *e.g.*, *varioref.sty*, *color.sty*) directly! If you do not want to make yourself a ‘container class’, or a `.sty` file of your own you should *copy* the class, or style files, edit *the copy*, and save it as a *different* file using a *different* file name.

On how to create container classes see the De-TeX-FAQ [8, question 5.1.5].

**Note:** Install any additional files, or packages in the local texmf tree in your `$HOME` directory. Otherwise these changes will be overwritten when upgrading your T<sub>E</sub>X distribution. Styles or packages you only need in one particular project or which you may want to hand on to someone you wish to share your project with may as well be saved in the current working directory. See

the De-TeX-FAQ [8, question 5.1.4], or the UK FAQ [3, ‘Installing  $\LaTeX$  files’, section O, ‘Where to put new files’, question 131].

## 1.4 Changing inter-line space using `\baselinestretch`

As a rule of thumb, parameters should be set on the highest possible level within a user interface. So if you want to reset inter-line space you can do so on three levels:

1. Either by using the *setspace.sty* package;
2. or by using the  $\LaTeX$  command `\linespread{<factor>}`;
3. or by redefining `\baselinestretch`.

Redefining parameters such as `\baselinestretch` works on the lowest level available—which should better be left to packages. The `\linespread` command is provided for this, so it is a better way to get more inter-line space than fiddling with `\baselinestretch`. It is even better, though, to use *setspace.sty* which also takes care of space in footnotes and list environments that you usually don’t want to change when modifying inter-line space.

So if you just need some more spacing between lines, say, you would like to set spacing to one half or to double spacing, *setspace.sty* provides the easiest way to achieve this. However, if you only want to use fonts other than Computer Modern you may use `\linespread{<factor>}`. For example, when using Palatino `\linespread{1.05}` would be appropriate.

## 1.5 Paragraph indent and the spread between paragraphs (`\parindent`, `\parskip`)

It may make sense to change the indent of the first line in paragraphs (`\parindent`). However, if you do so, please note the following:

- Never use absolute sizes (*e.g.*, ‘mm’) to modify paragraph indent. Use sizes that depend on font size, such as ‘em’, *e.g.* The latter does *not* mean that indent adapts automatically when changing the font size. Rather, the value that goes with the font currently activated is used.
- Always use  $\LaTeX$  commands. For example, this may make it easier to parse<sup>5</sup> a  $\LaTeX$  file through an external program, or script. Your code will be easier to maintain, too. So problems concerning compatibility with other packages can be avoided as well (*calc.sty*, *e.g.*).

Replace: `\parindent=1em` by `\setlength{\parindent}{1em}`

In case you prefer some additional space between paragraphs to paragraph indent for marking the start of a new paragraph (‘zero paragraph indent’) do *not* use

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{\baselineskip}
```

---

<sup>5</sup> That is to say, analyse syntactically, or split up.

The `\parskip` macro should not be used as it will also modify settings for list environments, table of contents etc., and headings.

*parskip.sty*, however, as well as the KOMA-Script classes go to some lengths to avoid these side effects. On how to use these KOMA-Script class options (`parskip`, `halfparskip`, etc.) see scruien [4]. When using one of the KOMA-Script classes you do *not* need to load *parskip.sty*.

## 1.6 Separating maths formulae from continuous text using $\$ \$ . . . \$ \$$

Please don’t do this!  $\$ \$ . . . \$ \$$  is a Plain  $\TeX$  command. It will modify vertical spacing within formulae, rendering them inconsistent. This is why it should be avoided in  $\LaTeX$  (see section 3.3 on page 14; note the warning concerning `displaymath` along with the *amsmath.sty* package). What’s more, class option `fleqn` won’t work any more.

Replace:  $\$ \$ . . . \$ \$$  by `\[ . . . \]`

or

```
\begin{displaymath}
...
\end{displaymath}
```

## 1.7 `\def` vs. `\newcommand`

Always use `\newcommand{\<name>}{ . . . }` for defining macros.<sup>6</sup>

Never use `\def\<name>{ . . . }`. The main problem with `\def` is that no check is done on whether there already exists another macro of the same name. So a macro defined earlier may be overwritten without any error warning.

Macros may be re-defined using `\renewcommand{\<name>}{ . . . }`.

If you know *why* you need to use `\def` you will probably know about the pros and cons of this command. Then, you may as well ignore this subsection.

## 1.8 Should I use `\sloppy`?

Frankly speaking, the `\sloppy` switch should not be used at all. Most notably you shouldn’t use it in the preamble of a document. If line breaks appear in single paragraphs you should

1. check whether the right hyphenation patterns, *e.g.*, *(n)german.sty*, and T1 fonts have been loaded (see De-TeX-FAQ [8, section 5.3]), or the UK FAQ [3, ‘Hyphenation’, section Q.7];
2. put your text in other words. You do not necessarily need to change the sentence the line break problem appears in. It may suffice to change the preceding, or the next sentence;
3. slightly change some parameters  $\TeX$  uses in its line-breaking mechanism. Axel Reichert suggested the following solution<sup>7</sup> on [de.comp.text.tex](mailto:de.comp.text.tex):<sup>8</sup>:

---

<sup>6</sup> See [5, section 2.7.2], [7, section 3.4].

<sup>7</sup> Of course you may change these values according to taste, but beware of fiddling with `\emergencystretch`. Otherwise you’ll get quite sloppy justified text as you would get with a rather well-known text processor.

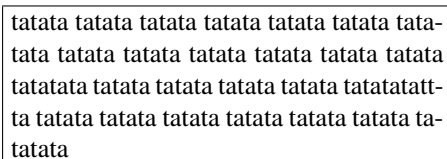
<sup>8</sup> The posting may be found as [Message-ID: <a84us0\\$plqcm\\$7@ID-30533.news.dfncis.de>](mailto:Message-ID: <a84us0$plqcm$7@ID-30533.news.dfncis.de>)

## 2 Some obsolete commands and packages

```
\tolerance 1414
\hbadness 1414
\emergencystretch 1.5em
\hfuzz 0.3pt
\widowpenalty=10000
\vfuzz \hfuzz
\raggedbottom
```

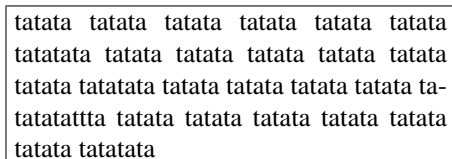
Note that warnings appearing with the above settings *really* should be taken seriously. You *should* consider putting your text in other words, then.

Only if this fails you may try to typeset the following paragraph more ‘loosely’ using the `sloppypar` environment.



tatata tatata tatata tatata tatata tatata tata-  
tata tatata tatata tatata tatata tatata tatata  
tatatata tatata tatata tatata tatata tatatatatt-  
ta tatata tatata tatata tatata tatata tatata ta-  
tatata

**Figure 1:**  $\text{\LaTeX}$ 's default settings



tatata tatata tatata tatata tatata tatata  
tatatata tatata tatata tatata tatata tatata  
tatata tatatata tatata tatata tatata tatata ta-  
tatatatatta tatata tatata tatata tatata tatata  
tatata tatatata

**Figure 2:** This demonstrates the effect of `\sloppy`

In figures 1 and 2 I have tried to show the effect of `\sloppy`. This also depends on the font employed. When using Times the negative effects of `\sloppy` do not show as extremely as with, say, Computer Modern. The effect in principle, however, should become clear.

In `comp.text.tex` Markus Kohm has posted an example that shows this effect even better. With his kind permission I quote his code appendix A on page 17.

## 2 Some obsolete commands and packages

Markus Kohm has written a Perl script you can test your files online with for the most common mistakes. See <http://kohm.de.tf/markus/texidate.html>. Please note, however, that this script is not a complete  $\text{\TeX}$  parser. This is why it will only check for the most common mistakes. Please test your file first, then post for help to a newsgroup, or to a mailing list.

### 2.1 Commands

#### 2.1.1 Changing font style

Table 1 on the next page shows obsolete and ‘proper’ commands in  $\text{\LaTeX}$  2<sub>ε</sub> side by side for changing font style. Macros called ‘local’ only apply to their own argument whereas those called ‘global/switch’ will apply to all following text till the end of the document.

## 2 Some obsolete commands and packages

**Table 1:** Commands for changing font style

obsolete	Replacement in L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub>	
	local	global/switch
<code>{\bf ...}</code>	<code>\textbf{...}</code>	<code>\bfseries</code>
—	<code>\emph{...}</code>	<code>\em<sup>a</sup></code>
<code>{\it ...}</code>	<code>\textit{...}</code>	<code>\itshape</code>
—	<code>\textmd{...}</code>	<code>\mdseries</code>
<code>{\rm ...}</code>	<code>\textrm{...}</code>	<code>\rmfamily</code>
<code>{\sc ...}</code>	<code>\textsc{...}</code>	<code>\scshape</code>
<code>{\sf ...}</code>	<code>\textsf{...}</code>	<code>\sffamily</code>
<code>{\sl ...}</code>	<code>\textsl{...}</code>	<code>\slshape</code>
<code>{\tt ...}</code>	<code>\texttt{...}</code>	<code>\ttfamily</code>
—	<code>\textup{...}</code>	<code>\upshape</code>

<sup>a</sup> May be useful when defining macros. In continuous text `\emph{...}` should be preferred to `\em`.

**Why not use obsolete commands?** Obsolete commands do not support L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'s new font selection scheme, or NFSS. `{\bf foo}`, for example, resets all font attributes which had been set earlier before it prints *foo* in bold face. This is why you cannot simply define a bold-italics style by `{\it \bf Test}` only. (This definition will produce: **Test**.) On the other hand, the new commands `\textbf{\textit{Test}}` will behave as expected producing: **Test**. Apart from that, with the former commands there is no 'italic correction', cf. for instance *halfhearted* (`{\it half}hearted`) to *halfhearted* (`\textit{half}hearted`).

For an overview of NFSS see [6].

### 2.1.2 Mathematical fractions (`\over` vs. `\frac`)

Avoid the `\over` command. `\over` is a T<sub>E</sub>X command which due to the syntax differing from L<sup>A</sup>T<sub>E</sub>X's is even more complicated to parse or which cannot be parsed at all. The *amsmath.sty* package redefines `\frac{}{}{}` which will result in error messages when using `\over`. Another point in favour of using `\frac{}{}{}` is that it is easier to fill in both the fraction's numerator and denominator, especially with more complex fractions.

Replace: `$a \over b$` by `$(\frac{a}{b})$`

### 2.1.3 Centring text using `\centerline`

The `\centerline` command is another T<sub>E</sub>X command you should not use. On the one hand `\centerline` is incompatible with some L<sup>A</sup>T<sub>E</sub>X packages, such as *color.sty*. On the other hand the package may yield unexpected results. *E.g.*:

```
\begin{enumerate}
\item \centerline{An item}
\end{enumerate}
```

An item
1.



Replace: `\centerline{...}` by `{\centering ...}`  
or  
`\begin{center}`  
`...`  
`\end{center}`

**Note:** On how to centre graphics and tables see section 3.1 on page 14.

## 2.2 Class files and packages

### 2.2.1 *scrlettr.cls* vs. *scrlettr2.cls*

*scrlettr.cls* class from the KOMA-Script bundle is obsolete. It was replaced by *scrlettr2.cls*. In order to produce a layout *similar* to the former KOMA-Script letter class use class option `KOMAold` which provides a compatibility mode.

Replace: `\documentclass{scrlettr}` by `\documentclass[KOMAold]{scrlettr2}`

**Note:** For new templates and letters use the new interface. It is definitely more flexible.

It is not possible to elaborate on the differences between the two user interfaces in this overview. See scruien [4] for details.

### 2.2.2 *epsf.sty*, *psfig.sty*, *epsfig.sty* vs. *graphics.sty*, *graphicx.sty*

The *epsf.sty* and the *psfig.sty* packages have been replaced by *graphics.sty* and *graphicx.sty*. *epsfig.sty* is just a wrapper<sup>9</sup> for processing old documents which had been done using *psfig.sty* with the *graphicx.sty* package.

As *epsfig.sty* uses *graphicx.sty* internally *epsfig.sty* still *may* be used. You should not use it, though, for new documents. *graphics.sty* or *graphicx.sty* should be preferred, then. *epsfig.sty* is mainly provided for reasons of compatibility, as mentioned above.

For the differences between *graphics.sty*, and *graphicx.sty* see grfguide [2]. For hints on centring graphics see section 3.1 on page 14.

Replace: `\usepackage{psfig}` by `\usepackage{graphicx}`  
`\psfig{file=image,...}` `\includegraphics[...]{image}`

### 2.2.3 *doublestpace.sty* vs. *setstpace.sty*

For changing inter-line space use the *setstpace.sty* package. *doublestpace.sty* is obsolete. It was replaced by *setstpace.sty*. Cf. section 1.4 on page 5.

Replace: `\usepackage{doublestpace}` by `\usepackage{setstpace}`

---

<sup>9</sup> A ‘wrapper’ here denotes a style file which itself loads another one or more style files, hence modelling functions.

### 2.2.4 *fancyheadings.sty*, *scrpage.sty* vs. *fancyhdr.sty*, *scrpage2.sty*

The *fancyheadings.sty* package was replaced by *fancyhdr.sty*. Another way to modify headings is provided by the *scrpage2.sty* package from the KOMA-Script bundle. Do not use *scrpage.sty* for it is obsolete. For documentation on *scrpage2.sty* see scruien [4].

Replace: `\usepackage{fancyheadings}` by `\usepackage{fancyhdr}`

Replace: `\usepackage{scrpage}` by `\usepackage{scrpage2}`

### 2.2.5 The *caption.sty* family of packages

The *caption2.sty* package should no longer be used because there is a new version (v3.x) of *caption.sty*. Please make sure to use the latest version of this package by loading *caption.sty* like this:

Replace: `\usepackage{caption}` by `\usepackage{caption}[2004/07/16]`

In case you used *caption2.sty* before, please have a look into the package documentation *anleitung* [13, section 8].

### 2.2.6 *isolatin.sty*, *umlaut.sty* vs. *inputenc.sty*

**Some general notes:** Basically there are four ways to input German *umlauts* and other non-ASCII characters:

1. `H{"u}lle`: This will work on any given system anytime.

The main disadvantages, however, are that kerning<sup>10</sup> between letters is disturbed badly; it is extremely complicated to input at least in a German-language text; and it is rather hard to read in source code.

So this variant should *always* be avoided due to the problems as far as kerning is concerned.

2. With `H"ulle` or `H{"u}lle` the abovementioned problems as far as kerning is concerned do not appear. It can be used on every system, too.

However it is just as tricky to input and to read the text as with the above variant.

This variant does make sense, however, when defining macros or style files for it does not require a particular text file encoding nor any additional packages.

3. With *(n)german.sty* or the *(n)german* option in *babel.sty* German *umlauts* can be input easier (`H"ulle`). Again this will work on all systems. As both *babel.sty* and *(n)german.sty* are available on all T<sub>E</sub>X systems there should be no problems as far as compatibility is concerned.

However, this again is tricky to input, and the source is comparatively hard to read.

This variant is best for use in continuous text. But it should be avoided in macro definitions and in preambles.

---

10 ‘Kerning’ means including positive or negative space between characters depending on which characters are to be typeset.

4. Direct input (`Hülle`). The advantage of this variant is obvious. You can input and read the continuous source text just as any other ‘normal’ text.

On the other hand you have to tell  $\text{\LaTeX}$  which input encoding is used. There may also be problems when exchanging files between different systems. This is *not* a problem for  $\text{\TeX}$ , or  $\text{\LaTeX}$  itself, but it may cause *problems in displaying text in editors* on different systems. For example, a Euro currency symbol encoded in iso-8859-15 (latin9) may be *displayed* in an editor on a windows box (CP1252) as  $\text{\text{€}}$ .

This variant is quite good for continuous text. It should, however, be avoided in macro definitions and in preambles.

To sum it up, in macros, in preambles, and in style files `H\ "ulle`, or `H\ "{u}lle` should be used, while in the rest of the text you should either use `H"ulle`, or `Hülle`.

**Input Encoding** Do *not* use the packages *isolatin1.sty*, *isolatin.sty*, or *umlaut.sty* for setting input encoding! Those packages are either obsolete, or they are not available on any given system.

Use *inputenc.sty*. There are four options available:

**latin1/latin9** for Unix-like systems (latin1 also works on MS Windows and Mac OS X)

**ansinew** for MS Windows

**applemac** for the Macintosh<sup>11</sup>

**cp850** for OS/2

Replace: `\usepackage{isolatin1}` by `\usepackage[latin1]{inputenc}`

Replace: `\usepackage{umlaut}` by `\usepackage[latin1]{inputenc}`

### 2.2.7 *t1enc.sty* vs. *fontenc.sty*

Generally speaking, the topic has been dealt with sufficiently in both the De-TeX-FAQ [8, questions 5.3.2, 5.3.3, 10.1.10], and the UK FAQ [3, ‘Why use *fontenc* rather than *t1enc*’, question 316]. So all that remains to be said is that *t1enc.sty* is obsolete and hence should be replaced by *fontenc.sty*.

Replace: `\usepackage{t1enc}` by `\usepackage[T1]{fontenc}`

### 2.2.8 *natdin.bst* vs. *dinat.bst*

Style file *natdin.bst* was replaced by *dinat.bst*.

Replace: `\bibliographystyle{natdin}` by `\bibliographystyle{dinat}`

---

<sup>11</sup> latin1 encoding is recommended for OS X users, too, as it is better fit for exchanging files cross-platform than applemac. If you do so you should, however, check the encoding settings of your editor first. In the long run you might like to switch to unicode, but please note that unicode support in *inputenc.sty* still is a work in progress at this point of time. Some users say they are content with *ucs.sty* from the unicode package.

## 2.3 Fonts

‘Fonts and L<sup>A</sup>T<sub>E</sub>X’ is a troublesome topic. Most discussions in [de.comp.text.tex](#) start with the question why fonts display so ‘fuzzy’ in Adobe Acrobat<sup>®</sup> Reader. Most answers point to the *times.sty* or *pslatex.sty* packages. However, those packages use completely different sets of fonts.

For an overview of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>’s New Font Selection Scheme, or NFSS see [6].

For making Computer Modern fonts display just fine in *acroread* see De-TeX-FAQ [8, question 9.2.3], or UK FAQ [3, ‘The wrong type of fonts in PDF’, question 92].

### 2.3.1 *times.sty*

*times.sty* is obsolete (see *psnfss2e* [10]). It does set `\rmdefault` to Times, `\sfdefault` to Helvetica, and `\ttdefault` to Courier. But it does *not* use the corresponding mathematical fonts. What’s more, Helvetica is not scaled correctly which makes it appear too big in comparison. So if you want to use the combination Times/Helvetica/Courier you should use:

Replace: `\usepackage{times}` by `\usepackage{mathptmx}`  
`\usepackage[scaled=.90]{helvet}`  
`\usepackage{courier}`

**Note.** The scaling factor for *helvet.sty* together with Times should be somewhere between 0.90 and 0.92.

### 2.3.2 *mathptm.sty*

*mathptm.sty* is the predecessor to *mathptmx.sty*. So please use the latter for typesetting mathematical formulae in Times.

Replace: `\usepackage{mathptm}` by `\usepackage{mathptmx}`

### 2.3.3 *pslatex.sty*

*pslatex.sty* internally works like *mathptm.sty* + *helvet.sty* (scaled). However, it uses a Courier font scaled too narrowly. The main disadvantage in using *pslatex.sty* is that it does *not* work with T1 and TS1 encodings.

Replace: `\usepackage{pslatex}` by `\usepackage{mathptmx}`  
`\usepackage[scaled=.90]{helvet}`  
`\usepackage{courier}`

**Note on Courier for all combinations of Times/Helvetica** You do not have to load *courier.sty* at all. You may use the usual `cmtt` font for typewriter faces.

### 2.3.4 *palatino.sty*

*palatino.sty* behaves like *times.sty* – apart from setting `\rmdefault` to Palatino, of course. *palatino.sty* is obsolete, too. This is why it should not be used any more.

Replace: `\usepackage{palatino}` by `\usepackage{mathpazo}`  
`\usepackage[scaled=.95]{helvet}`  
`\usepackage{courier}`

**Note:** Scaling factor for *helvet.sty* in combination with Palatino should be set to 0.95.

Helvetica is *not* the ‘best’ sans-serif font at all for use with Palatino. It rather is the best *freely-available* one. He that possesses a CorelDraw®-CD (this may well be an older version) can use Palatino quite well along with Frutiger<sup>12</sup>, or Optima<sup>13</sup>. Walter Schmidt supplies adaptations for using some PostScript fonts with T<sub>E</sub>X on his homepage.<sup>14</sup>

### 2.3.5 *mathpple.sty*

This package was a predecessor to *mathpazo.sty*. It lacks some symbol fonts. So those fonts are taken from the Euler fonts instead. Some other symbols are not fit for use with Palatino as the font metrics are not correct. For details cf. psnfss2e [10].

### 2.3.6 Typesetting upright greek letters

The passages I have marked as red in the following are not obsolete in the sense of ‘you should not use this any more’, but now editing text is made much easier by *upgreek.sty*. For some more hints on usage please see the documentation upgreek [9].

#### The *pifont.sty* tricks

Replace: `\usepackage{pifont}` by `\usepackage{upgreek}`  
`\newcommand{\uppi}{\Pisymbol{psy}{112}}` `$\uppi$`  
`\uppi`

or

`\newcommand[1]{\upgreek}{%`  
`\usefont{U}{psy}{m}{n}#1}`  
`\upgreek{p}`

#### The *babel.sty* trick

Replace: `\usepackage[greek,...]{babel}` by `\usepackage{upgreek}`  
`\newcommand[1]{\upgreek}{%` `$\uppi$`  
`\foreignlanguage{greek}{#1}}`  
`\upgreek{p}`

---

12 Bitstream Humanist 777, bfr

13 Bitstream Zapf Humanist, bop

14 Fonts for T<sub>E</sub>X: <http://home.vr-web.de/was/fonts>

### 2.3.7 *euler.sty* vs. *eulervm.sty*

Use *eulervm.sty* instead of *euler.sty* for mathematical typesetting. *eulervm.sty* is a  $\text{\LaTeX}$  package for using the eulervm fonts. These are virtual math fonts based on both the Euler and the CM fonts, consuming less of  $\text{\TeX}$ 's resources and supplying some improved math symbols. Improved `\hslash` and `\hbar` are also supplied. Please see the package documentation eulervm [11] for details.

Replace: `\usepackage{euler}` by `\usepackage{eulervm}`

## 3 Miscellaneous

This section – apart from 3.2 – gives some more general advice than the ‘deadly sins’ section, pp. 4 ff.

### 3.1 Floats—‘figure’, ‘table’

For centring a float environment we recommend you use `\centering` instead of `\begin{center} ... \end{center}` because the latter will include an additional vertical skip you can do without in most cases.

Replace:

<code>\begin{figure}</code>	by <code>\begin{figure}</code>
<code>\begin{center}</code>	<code>\centering</code>
<code>\includegraphics{bild}</code>	<code>\includegraphics{bild}</code>
<code>\end{center}</code>	<code>\end{figure}</code>
<code>\end{figure}</code>	

**Note:** However, when centring a region within continuous text or within a `titlepage` environment this additional space may be welcome!

### 3.2 The appendix

The appendix is introduced by the `\appendix` command. Note that this is *not an environment*.

Replace: `\begin{appendix}` by `\appendix`

<code>\section{Blub}</code>	<code>\section{Blub}</code>
<code>\end{appendix}</code>	

### 3.3 Mathematical typesetting

Generally speaking, you should use *amsmath.sty* for advanced mathematical typesetting, providing a number of new environments replacing `eqnarray` in the first place. The main advantages of the package are these:

- Spacing within and around environments is more consistent.
- Equation numbering will be placed in a way so that they will not be printed over any more.
- Some new environments, e.g., `split`, provide a solution to split up long equations easily.

- It is easy to define new operators (similar to `\sin` etc.) with proper spacing.

**Warning:** When using *amsmath.sty* you should *never* use the `displaymath`, `eqnarray`, or `eqnarray*` environments because those are not supported by *amsmath.sty*. Otherwise this would lead to inconsistent spacing.

`\[...\]` is adapted correctly by *amsmath.sty*. So it may be used instead of `displaymath`, `eqnarray`, and `eqnarray*` may be replaced by `align`, or `align*`. For a complete overview of *amsmath.sty* see `amslatex` [1].

Replace: <code>\begin{eqnarray}</code>	by <code>\begin{align}</code>
<code>a &amp;= b \\\</code>	<code>a &amp;= b \\\</code>
<code>b &amp;= c \\\</code>	<code>b &amp;= c \\\</code>
<code>a &amp;= c</code>	<code>a &amp;= c</code>
<code>\end{eqnarray}</code>	<code>\end{align}</code>

### 3.4 How to use `\graphicspath`

There are several reasons why you should avoid Macro `\graphicspath`. Replace it by setting environment variable `TEXINPUTS`.<sup>15</sup>

1. There are *different* separators in path names on different platforms. While MS Windows and Unices both use a slash '/', a colon ':' is used on Macintosh systems.
2. `TeX`search takes longer than with using the `kpathsea` library (with today's fast chips this is not as important an argument as it used to be).
3. `TeX`'s memory is limited, and every picture uses part of this memory. What's more memory is not cleared during the compiling process.

In a Bourne shell use

```
$ TEXINPUTS=PictureDir:$TEXINPUTS latex datei.tex
```

or add to `~/.profile`

```
export TEXINPUTS=./PictureDir:$TEXINPUTS
```

In the latter case the files in `PictureDir` will be found within the current working directory. Up to MS Windows 98 the environment variable is set by adding

```
set TEXINPUTS=.\PictureDir;%TEXINPUTS%
```

to your `autoexec.bat`. On MS Windows NT-based systems according to the 'Microsoft Knowledge Base' the variable can be set by rightclicking at My Computer → System Properties → Advanced → Environment variables.<sup>16</sup>

The above are only some suggestions on how to proceed. I am well aware that `TEXINPUTS` may be set in different ways. Please see the documentation of your operating system, or of your `TeX` distribution for more.

<sup>15</sup> Cf. David Carlisle's answer on Markus Kohm's 'Bug-Report' at <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/2618>

<sup>16</sup> On Windows 2000 you may use: Start → Settings → Control Panel → System.

**Table 2:** Macros defined by *(n)german.sty* or by *babel.sty* with the *(n)german* option

Name of macro	Original definition	Usual output in German
<code>\prefacename</code>	Preface	Vorwort
<code>\refname<sup>a</sup></code>	References	Literatur
<code>\abstractname</code>	Abstract	Zusammenfassung
<code>\bibname<sup>b</sup></code>	Bibliography	Literaturverzeichnis
<code>\chaptername</code>	Chapter	Kapitel
<code>\appendixname</code>	Appendix	Anhang
<code>\contentsname</code>	Contents	Inhaltsverzeichnis
<code>\listfigurename</code>	List of Figures	Abbildungsverzeichnis
<code>\listtablename</code>	List of Tables	Tabellenverzeichnis
<code>\indexname</code>	Index	Index
<code>\figurename</code>	Figure	Abbildung
<code>\tablename</code>	Table	Tabelle
<code>\partname</code>	Part	Teil
<code>\enclname</code>	encl	Anlage(n)
<code>\ccname</code>	cc	Verteiler
<code>\headtoname</code>	To	An
<code>\pagename</code>	Page	Seite
<code>\seename</code>	see	siehe
<code>\alsoname</code>	see also	siehe auch

<sup>a</sup> In article class only.

<sup>b</sup> In report and book classes only.

### 3.5 Language-specific macros – `\*name`

From time to time the question comes up in `de.comp.text.tex` how to modify, *e.g.*, the ‘References’ heading to ‘Literaturliste’ or to something else. So I have compiled those macros in table 2. They have been taken from the *german.sty* package. Users who want to adapt macro output to other languages may as well refer to this table as an example.

So if you want to change the heading ‘List of Figures’ to, say, ‘Pictures’ you may use the following command:

```
\renewcommand*{\listfigurename}{Pictures}
```

The other macros are changed in the same way respectively. With *babel.sty* use the `\addto` macro. For more details see the De-Tex-FAQ [8].



## **A An example illustrating the effect of the \sloppy command**

This is the example Markus Kohm published earlier:

---

```
\documentclass{article}

\setlength{\textwidth}{20em}
\setlength{\parindent}{0pt}
\begin{document}
\typeout{First without \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\~ta\~tata
tatata tatata tatata tatata tatata tatata tata\~tata
tatata tatata tatata tatata ta\~tatatatt\~ta
tatata tatata tatata tatata tatata tatata ta\~ta\~ta\~ta

\typeout{done.}

\sloppy
\typeout{Second with \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\~ta\~tata
tatata tatata tatata tatata tatata tatata tata\~tata
tatata tatata tatata tatata ta\~tatatatt\~ta
tatata tatata tatata tatata tatata tatata ta\~ta\~ta\~ta

\typeout{done.}
\end{document}
```

---

From: [Message-ID: <8557097.gEimXdBtjU@ID-107054.user.dfncis.de>](#)

## References

- [1] AMERICAN MATHEMATICAL SOCIETY: *User's Guide for the amsmath Package*. December 1999, Version 2.0.  
URL: [CTAN:macros/latex/required/amslatex/](http://CTAN:macros/latex/required/amslatex/).
- [2] DAVID P. CARLISLE: *Packages in the 'graphics' bundle*. January 1999.  
URL: [CTAN:macros/latex/required/graphics/](http://CTAN:macros/latex/required/graphics/).
- [3] ROBIN FAIRBAIRNS: *The UK T<sub>E</sub>X FAQ. Your 355 Questions Answered*. WWW, Version 3.11, 14 March 2003,  
URL: <http://www.tex.ac.uk/faq>.
- [4] MARKUS KOHM, FRANK NEUKAM und AXEL KIELHORN: *The KOMA-Script Bundle*. scruien.  
URL: [CTAN:macros/latex/supported/koma-script/](http://CTAN:macros/latex/supported/koma-script/).
- [5] THE L<sup>A</sup>T<sub>E</sub>X3 PROJECT: L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for class and package writers. 1999.  
URL: [CTAN:macros/latex/doc/clsguide.pdf](http://CTAN:macros/latex/doc/clsguide.pdf)
- [6] THE L<sup>A</sup>T<sub>E</sub>X3 PROJECT: L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> font selection. 2000.  
URL: [CTAN:macros/latex/doc/fntguide.pdf](http://CTAN:macros/latex/doc/fntguide.pdf)
- [7] THE L<sup>A</sup>T<sub>E</sub>X3 PROJECT: L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for authors. 2001.  
URL: [CTAN:macros/latex/doc/usrguide.pdf](http://CTAN:macros/latex/doc/usrguide.pdf)
- [8] BERND RAICHLE, ROLF NIEPRASCHK und THOMAS HAFNER: *Fragen und Antworten (FAQ) über das Textsatzsystem T<sub>E</sub>X und DANTE, Deutschsprachige Anwendervereinigung T<sub>E</sub>X e.V.* WWW, Version 72. September 2003,  
URL: <http://www.dante.de/faq/de-tex-faq/>.
- [9] WALTER SCHMIDT: *The upgreek package for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*. May 2001, Version 1.0.  
URL: [CTAN:macros/latex/contrib/supported/was/](http://CTAN:macros/latex/contrib/supported/was/).
- [10] WALTER SCHMIDT: *Using common PostScript fonts with L<sup>A</sup>T<sub>E</sub>X*. April 2002, PSNFSS version 9.0.  
URL: [CTAN:macros/latex/required/psnfss/psnfss2e.pdf](http://CTAN:macros/latex/required/psnfss/psnfss2e.pdf)
- [11] WALTER SCHMIDT: *The Euler Virtual Math Fonts for use with L<sup>A</sup>T<sub>E</sub>X*. Januar 2004, Version 3.0a.  
URL: [CTAN:fonts/eulervm/](http://CTAN:fonts/eulervm/)
- [12] WALTER SCHMIDT, JÖRG KNAPPEN, HUBERT PARTL und IRENE HYNÄ: *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Kurzbeschreibung*. April 1999, Version 2.1.  
URL: [CTAN:info/lshort/german/](http://CTAN:info/lshort/german/). English Translation available at  
URL: [CTAN:info/lshort/english/](http://CTAN:info/lshort/english/)

## References

- [13] AXEL SOMMERFELD: *Setzen von Abbildungs- und Tabellenbeschriftungen mit dem caption-Paket*. Juli 2004, Version 3.0c.  
URL: [CTAN:macros/latex/contrib/caption/](http://CTAN:macros/latex/contrib/caption/).

\* \* \*