

A New LaTeX User Interface 0.1

Martin Hensel
mail@MartinHensel.de

4 July 2003

Introduction

This document explains my suggestion for a new user interface for $\text{\LaTeX}3$, i. e. a new syntax. It is not indented as an argument for a new syntax, although I strongly believe that one is needed. I hope that all those, who think a new user interface would be a good idea, will join into a discussion in order to improve this suggestion.

You will recognise that I will write in detail about problems, where I am not quite happy with my solution, while I will only summarise others.

The Space Problem

Let us directly start with one of the most difficult topics: the spaces. \LaTeX is a combined text mark-up and programming language. Therefore we can find three main elements in it: text, commands and programming sequences. Unfortunately these are not clearly distinguishable and spaces in the text cannot be treated differently compared to spaces in programming sequences.

In languages like HTML, XML, and most programming languages spaces are treated as following: Line breaks are considered as spaces, two or more spaces are considered as a single space. If programming, many people want to use indentation and line breaks to structure their code. This is problematic in \LaTeX as unwanted space may be inserted in the output.

Line Breaks

I found the following solutions:

- A-1) Comments before a line break prevent the occurrence of spaces in the output, but separate command (like currently in $\text{\LaTeX}2e$).
- A-2) Line breaks do not produce spaces in general, but separate commands.

A problem of solution 1 is the fact that people will not accept this way as a standard. They will not generally write a comment symbol at the end of each line in programme sequences, but only when it is necessary. This already caused many problems (and questions in support groups) with $\text{\LaTeX}2e$.

Solution 2 may lead to unwanted results in text passages if words at the end of the line and the beginning of the following line are not separated. This could be solved by indenting all text passages by at least one space.

Indentation

I found the following solutions:

- B-1) All spaces at the beginning of a line are ignored (like currently in $\text{\LaTeX}2e$).
- B-2) Special gap symbols are used to indent commands but do not produce spaces in the output (e. g. TAB or underscore).

Solution?

There are two major solutions.

C-1) Combining the solutions A-1 and B-1 and making it work like currently in $\text{\LaTeX}2\text{e}$. As mentioned above, this system already causes some trouble.

C-2) Combining the other solutions A-2 and B-2 would arise the question if there was an appropriate character and if available editors could handle this system (in particular indenting all text passages by one space).

Both solutions are not satisfactory. I would tend to C-1.

$\text{\LaTeX}2\text{e}$

Commands

In $\text{\LaTeX}2\text{e}$ there are several possible command syntaxes.

```
\"a
\command
\command []
\command {}
\command [] {}
\command {} {}
etc.
```

Problems here are

- the possibility that the following character is treated as an argument is not obvious
- special treatment of all spaces after `\command` is necessary
- the number of arguments of `\command{}` is unknown (the last `{...}` may well be normal text)

Environments

```
\begin{environment}
... \end{environment}
\begin{environment}{}
... \end{environment}
```

```
\begin{environment} [] {}
... \end{environment}
etc.
```

Problems are

- the unknown number of arguments
- for beginners the confusion where to put the optional argument (`[]`)

Parameters

There are many different, nonuniform, and confusing possibilities to specify parameters

```
titlepage, twoside
bb=0 0 100 100
BCOR11mm
\desclabelstyle{\nextlinelabel}
etc.
```

The New Syntax

Although I considered several different possibilities I will not list them all in this section, but simply present my solution.

The orders

```
command parameters and
begin-environment parameters
... end-environment
```

are very useful and should not be changed.

Commands

A symbol that signals the end of a command is very important for several reasons (e.g. to avoid problems with different spacing, to allow an easy recognition in having a closed gestalt, etc). Commands therefore have the following syntax:

```
[command]
```

That is to say they are enclosed in brackets. For the time being, this applies to all commands that currently begin with ‘\’. A command name can consist of more different characters than before. I suggest including numbers to the allowed characters (if it is technically possible and does not interfere with other supported user interfaces).

Parameters

All parameters are in the same enclosure as their commands (or environments) to represent the affiliation. There is no distinction between optional or mandatory arguments at the interface level.

Parameters are identified by a combined system of naming and position. All commands are defined with named parameters. The names are at least two characters long.

An example: Assuming the defined arguments of the command `maketitle` were `title`, `author`, and `date` (in this order), then one could call the command with

```
[maketitle title={The New [LaTeX]
syntax}, author={Martin Hensel},
date={2003}]
```

or by

```
[maketitle {The New [LaTeX]
syntax}, {Martin Hensel},
date={2003}]
```

or

```
[maketitle title={The New [LaTeX]
syntax}, {Martin Hensel}{2003}]
```

or even by

```
[maketitle {The New [LaTeX]
syntax}, author={Martin Hensel}
date=2003]
```

The idea is that the unnamed parameters are given sequentially from left to right. Only named parameters ‘jump’ in this sequence. Assuming the definition of the parameters was ‘AA’, ‘BB’, ‘CC’, ‘DD’, ‘EE’, and ‘FF’: If, in a command call, the first parameter was not named, it would be considered to be ‘AA’. If the next parameter was not named as well, it would be considered to be ‘BB’. If the third parameter was named ‘EE’, the following unnamed parameter would represent ‘FF’.

The ‘content’ of the parameters is normally enclosed by `{...}`, but these braces can be omitted if the parameter does not contain any spaces, commas or braces, brackets etc. Unnamed arguments are always to be enclosed to avoid any collision with parameter names. This does not apply to one-character argu-

ments. This is to allow convenient use of accent commands like `["a]`.

Parameters are separated by any sequence of spaces or commas or by their enclosing braces.

The named parameters are given either by `param={value}` or `param=value` or by `param{value}`. Each parameter name has to be at least two chars long. I suggest reserving ‘#’ as an alias for the often used `text` parameter (which would be the only allowed 1-character parameter name).

It is suggested, that all switch parameters support many different input possibilities (`true`, `yes`, `on`, `1`, and `false`, `no`, `off`, `0`). To allow consistency with the current switch parameters, one can define ‘aliases’. For instance `titlepage` corresponds to `titlepage=true` and `notitlepage` corresponds to `titlepage=false`.

Environments

Environments are very similar to commands. They use the same syntax for parameters, but are enclosed differently: `[env> ... <env]`.

It should be possible to have commands and environments with the same name, e.g. `[center> ...<center]` and `[center]` (instead of `\centering`). If this is not possible because of other user interfaces, it should be at least impossible to use an environment wrongly as a command (like now with `\center`) and vice versa.

Comments

Comments are indicated by `%` (to the end of the line) and by `%> ...<%` or `[%> ...<%]` (as an environment)

Further Changes

In order to distinguish clearly between the user interfaces, two new environments should be introduced

```
[latex2e> ... <latex2e]
[teX> ... <teX]
```

in which old code would be allowed.

Further Examples

```
[newcommand {titlepage},  
  parameters={{title}  
    {author}{date}{titlehead}}]
```

```
[newcommand command=test  
  {draft, width},  
  alias={{draft}{draft=on}}  
    {{final}{draft=off}}}]
```

```
[table {hbtpt}>  
  [tabularx {1|1},  
    width=[linewidth]>  
      content & content  
  <tabularx]  
<table]
```

```
[documentclass article]  
[usepackage fontenc enc=T1]  
[document>  
  [maketitle  
    title={This is an example}  
    author={Martin Hensel}]  
  This is an example using the  
  [emph {new}] [LaTeX] syntax.  
<document]
```

```
[emph {new}]  
[emph{new}]  
[emph text={new}]  
[emph text=new]  
[emph text{new}]  
[emph #={new}]  
[emph #=new]  
[emph #{new}]  
[emph #new]
```