

The easylist package for numbered items

Paul Isambert
zappathustra@free.fr
<http://paulisambert.free.fr>

November 16, 2008

Abstract

This package is designed for typesetting lists of numbered items (like Wittgenstein's *Tractatus* or—more likely—the outline of a work yet to be done) with a single active character acting as the only command. Various options are available to achieve greater control over the general appearance of the list.

1 Choosing the symbol

Here's what the L^AT_EX code with `easylist` in its default usage looks like:

```
\begin{eList}
§ First proposition.
§§ Interesting comment.
§§§ A note on the comment.
§§§ Another note.
§§§§ By the way...
§§§§§ This is a subsub...-proposition.
§ Let's start something new...
\end{eList}
```

And it yields:

```
1. First proposition.
1.1. Interesting comment.
1.1.1. A note on the comment.
1.1.2. Another note.
1.1.2.1. By the way...
1.1.2.1.1. This is a subsub...-proposition.
2. Let's start something new...
```

Now, the section sign § might not be readily accessible on some (most?) keyboards (although it is accessible and useless on French ones). So you can choose another one instead by setting an option when calling the package (§ is default):

```
\usepackage[pilcrow]easylist to use ¶.
\usepackage[at]easylist to use @.
\usepackage[sharp]easylist to use #.
\usepackage[ampersand]easylist to use &.
```

The selected character is made active between `\begin{eList}` and `\end{eList}` and then returns to its initial value. So when using # for instance, just make sure to define new commands outside the `eList`, or use the `\Activate` and `\Deactivate` commands in your list (see below), and everything will be right. (Ending a list and then beginning a new one will not be noticeable in the final product, so you can interrupt your list to create a new command.) In what follows, I will be using §, but everything applies to other symbols too.¹

2 Usage

The example above says it all, so let's repeat it for clarity's sake:

```
\begin{eList}
§ First proposition.
§§ Interesting comment.
§§§ A note on the comment.
§§§ Another note.
§§§§ By the way...
§§§§§ This is a subsub...-proposition.
§ Let's start something new...
\end{eList}
```

Your list must be enclosed between `\begin{eList}` and `\end{eList}`, which is welcome since a character is made active and active characters are notoriously dangerous. As you might have guessed, one § creates a proposition of the first level, two §'s make a proposition of the second level, and so on down to the fifth level. If you concatenate more than five §'s, you'll get an error and the problematic line will begin with `!!!!` instead of numbers. Every sequence of §'s must terminate with a space, otherwise numbers won't be printed.

Two things are worth mentioning. First, note that § automatically creates a new line, so typing your propositions in a row will not affect the result. Hence

```
\begin{eList} § First proposition.  §§ Interesting comment.  \end{eList}
```

still yields

```
1. First proposition.
1.1. Interesting comment.
```

Second, when skipping one or more level(s), that(those) level(s) will be numbered 0 (except otherwise specified—see below), as in:

```
\begin{eList}
§ First proposition.
§§§ A sub-comment to the first proposition.
\end{eList}
```

which yields:

```
1. First proposition.
1.0.1. A sub-comment to the first proposition.
```

That's it. The following sections deal with labelling and referring and layout options, but for basic purpose you don't need more than that.

¹You might not be happy with the symbols and maybe you'd like to use another one, or simply have your favorite symbol as default to avoid remembering such a cumbersome name as 'pilcrow'. Here's a simple hack that does the job: select the entire code of the package, and replace all occurrences of § with your symbol. Make sure you won't use it in the list for other purposes, though.

3 Referring to an item

It may be useful to refer to a previous item, so `easylist` includes a labelling device (note that it's not possible to refer to a *subsequent* item—maybe in a later version?). The command is `\Llabel{key}` (with two l's on purpose to avoid possible conflicts with packages also allowing labelling) and works exactly like L^AT_EX's `\label` command. It may be issued anywhere in the proposition, but if you type it between the sequence of §'s and your text, make sure to leave a space after the §'s. `\Lref{key}` and `\Pref{key}` are the corresponding referring commands, and they work just like `\ref` and `\pageref`, i.e. the former yields the number of the proposition and the latter the page where it appears.²

4 Options

Several parameters are available to achieve fine control over the final output. The general command is `\ListProperties(key=value,key=value...)`, where `key` is a parameter. Depending on where the command is called, it will have different scopes. If you call it inside an `eList`, it will be active for that list only, and in that list for items that follow it. Outside the `eList`, the parameters will affect all subsequent lists, unless specified otherwise (the parameter `Start`—see below—always works in this way, since it affects numbering). Let's review them in turn and indicate the default values.

Start1
Start2
Start3
Start4
Start5

This parameter sets the starting value of one of the five levels. Consecutive items (even in different lists) are normally consecutively numbered, but you may skip a value for whatever reason, for instance creating this document with all examples starting at 1, or avoiding the 0 mentioned above. Note that, for any given level, this command will work if and only if it is issued just before an item containing this level, as exemplified by:

```
\begin{eList}
\ListProperties(Start1=1,Start2=17)
§ First proposition.
§ Second proposition.
§§ Numbering doesn't work.
\ListProperties(Start2=17)
§§ This is better.
\end{eList}
```

which yields:

```
1. First proposition.
2. Second proposition.
2.1. Numbering doesn't work.
2.17. This is better.
```

This makes sense since every counter is set back to 1 when the one above it is incremented. The default value is of course 1.

Mark1
Mark2
Mark3
Mark4
FinalMark
Mark

This parameter controls the punctuation mark that follows the first, second... final number respectively. There is no `Mark5`, since the fifth number is always followed by the final mark (which is the mark that follows the final number, not simply the *fifth* number; the final mark always supersedes the first or second... mark in case they are the same). `Mark` sets all marks together. Note that the general rule for this parameter and the following ones is 'Last wins', i.e. `\ListProperties(Mark=.,FinalMark=)` yields a stop after each number except the final one, which will be followed by nothing, while on the other hand `\ListProperties(FinalMark=,Mark=.)` will produce a stop for all numbers including the final one, since `FinalMark` has been superseded by `Mark` (by the way notice that a missing value, i.e. the '=' sign followed by a comma or the closing parenthesis, is acceptable and understood as a null value).

²More precisely, `\Pref` refers to the page where the proposition starts, even when the corresponding `\Llabel` is at the end of the proposition and possibly on the following page in the final output. So if you want to refer to some part of your proposition, use `\label` and `\pageref` instead.

If you want one of the mark to be a comma, you have to enclose it between curly braces as in `\ListProperties(Mark1={,})`, since commas are also used to delimit parameters in this command. `FinalMark` may also be used to add some space before the text, as in `\ListProperties(FinalMark=. \hspace{1ex})`.

The default value for these parameters is a full stop.

This parameter sets the way numbers are printed. It takes the following values:

Numbers1
Numbers2
Numbers3
Numbers4
Numbers5
Numbers

`r` for lower case roman numerals;
`R` for upper case roman numerals;
`l` for lower case letters;
`L` for upper case letters;
anything else for arabic numbers, which are the default value.

Note that if you choose letters for a given counter, it should not exceed 26, or \TeX will complain—and you’ll have no number at all. Since I think that using letters with more than, say, 10 items, is ill-advised, I didn’t overcome that limitation.

This is used to set the style of the propositions of the first, second... levels, with `Style` specifying all the styles at once. Here you may use values such as `\large`, `\bfseries`... and so on. If you add a star to the name of the parameter, as in `Style3*` or `Style*`, only the numbers will be affected, as in:

Style1
Style2
Style3
Style4
Style5
Style

```
\begin{eList}
\ListProperties(Style1*=\bfseries,Style2*=\itshape)
§ A fundamental proposition.
§§ An essential comment.
\end{eList}
```

which produces:

1. A fundamental proposition.
1.1. An essential comment.

With two stars, as in `Style2**` or `Style**`, only the text of the item will be affected. The default value for this parameter is `\normalsize`, `\upshape`, `\rmfamily` and `\mdseries`, and *not* the style you might have set at the beginning of your document. So for instance if you’re writing with sans-serif and include an `eList` in your document, it will be typeset in roman, unless you specify `\ListProperties(Style=\sffamily)`. This default style is accessible with `\NormalStyle`, so you may redefine the latter at the beginning of your document to match your needs.

This parameter sets the distance from the left margin at which the concerned propositions should start. Once again, `Margin` sets all the parameters at once. So you can have a clearly designed list, as in:

Margin1
Margin2
Margin3
Margin4
Margin5
Margin

```
\begin{eList}
\ListProperties(Margin2=3ex,Margin3=6ex,Margin4=9ex,Margin5=12ex)
§ First proposition.
§§ Interesting comment.
§§§ A note on the comment.
§§§§ Another note.\\ (Those backslashes are part of the code! See below.)
This is fascinating. We can start a new paragraph at any level.
§§§§§ By the way...
§§§§§§ This is a subsub...-proposition.
§ Let’s start something new...
\end{eList}
```

and we have:

1. First proposition.
 - 1.1. Interesting comment.
 - 1.1.1. A note on the comment.
 - 1.1.2. Another note.

This is fascinating. We can start a new paragraph at any level.

 - 1.1.2.1. By the way...
 - 1.1.2.1.1. This is a subsub...-proposition.
2. Let's start something new...

Your items may be made of paragraphs, but if you want them to be properly aligned, you should use `\\` or `\newline` instead of a blank line to create a new paragraph, or you'll get unwanted results.

The default value for this parameter is no distance at all from the margin.

This is used to control the indentation of your items from their left justification. It is meant to receive commands such as `\indent`, `\noindent`, `\hspace{5mm}`, etc, but not simply `5mm`, for instance, as with `Margin`. The default value is `\noindent`.

Indent1
Indent2
Indent3
Indent4
Indent5
Indent

5 Troubles with boxes

Active characters, as is well known, are nasty beings, and it's hard to keep them activated all the way down. That's why the `eList` environment in its present state won't work in boxes. That is

```
\fbox{%
\begin{eList}
§ First proposition.
\end{eList}}
```

will print:

```
§ First proposition.
```

where `§` is the normal value of `§`. Lists in boxes aren't that common (except for this document), but such a limitation is always annoying. So here are two solutions³ (in a later version I will hopefully overcome the problem). First, you can save your code in a `\lrbox` and then issue it as wanted:

```
\newsavebox{\mybox} (This can be done anywhere.)
\begin{lrbox}{\mybox}
\begin{eList}
§ First proposition.
\end{eList}
\end{lrbox}
\fbox{\usebox{\mybox}}
```

Once the content of your box has become useless you can reuse `\mybox` with a new content. The following technique is faster (but also more dangerous if you're not careful). What you have to do is to make active the character you've chosen to create items just before the box and then turn it back to its initial value. For that purpose, `easylist` has the two commands `\Activate` and `\Deactivate`. So here's what you'll do:

³The first one is due to Enrico Gregorio on the `comp.text.tex` user group.

```

\Activate
\fbbox{%
\begin{eList}
§ First proposition.
\end{eList}}
\Deactivate

```

and you'll get:

```

1. First proposition.

```

Unfortunately, this has to be issued for every box. That is, the following code won't work (even if you surround your command definition with `\Activate` and `\Deactivate`—I told you that active characters were nasty):

```

\newcommand{\myfbbox}[1]{\Activate\fbbox{#1}\Deactivate}
\myfbbox{%
\begin{eList}
§ First proposition.
\end{eList}}

```

The same impossibility arises with the previous solution, so choose your favorite or avoid boxes!