

The fancylabel package*

Dominik Siegel

dominik.siegel@yahoo.de

2017/12/30

Abstract

For L^AT_EX referencing it is necessary to have unique labels. But sometimes you have similar objects (symbols, chemical structures, etc.) that appear more often than once in your document. When you write your document you cannot know from the beginning on in which position your object appears for the first time. Hence you don't know whether to place a `\label` or a `\ref`.

This package basically provides the two new commands `\fancylabel` and `\fancyref`. With the `\fancylabel` command you don't have to know if it is time to set a `\label` or a `\ref`, `\fancylabel` will decide this for you.

Additionally this package allows you various different numerations including numbering order, numbering level and numbering families.

Contents

1	Introduction	2
2	Package Commands	3
2.1	Label commands	3
2.2	Sublabel commands	3
2.3	Ref and subref commands	4
3	Options	4
3.1	The families option	4
3.2	The IgnoreFamilies option	5
3.3	The numbering option	5
3.4	The level option	6
3.5	The sort option	6
3.6	The compress option	7
3.7	The CompressLink option	7
3.8	The CompressTwo option	7
3.9	The ListLastSublabelOnly option	7

*This document corresponds to fancylabel v1.1, dated 2017/12/30.

3.10	The ListSeparator option	7
3.11	The ListLastLabelSeparator option	7
3.12	The ListLastRefSeparator option	8
3.13	The DisableFamilyCheck option	8
4	Customization	8
4.1	Reset counters	8
4.2	Label und sublabel format	8
4.3	NewLabelEvent	8
4.4	ShowLabelEvent	9
4.5	ShowRefEvent	9
4.6	Lists of labels	9
5	Change history	9

1 Introduction

The idea of referencing in L^AT_EX is to associate an object with a `\label`. Whenever you want to refer to it you just use `\ref`. This works as long as you have an unique object like a figure, a table, a section etc. But if you want the refer to repeating objects which are expected to have the same number like symbols, (chemical-) structures etc. referencing becomes more complex.

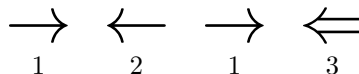


Figure 1: Some repeating objects. Here we refer to symbol 2.

In our example same arrows are supposed to get the same number. We define a counter and a new label command. This was actually quite simple. But the big disadvantage of this solution is, that you always have to take care where your object occurs for the first time. At the first occurrence you have to label it, at any other occurrence you have to refer. Every time you change the order you have to adjust the labeling by hand.

```

\newcounter{mylabelcounter}
\newcommand{\mylabel}[1]{\refstepcounter{mylabelcounter}\themylabelcounter\label{#1}}
\begin{figure}[h]
...figure code...
\begin{tabular}{cccc}
\Huge$\rightarrow$ & & \Huge$\leftarrow$ & & \\
\Huge$\rightarrow$ & & \Huge$\Leftarrow$ & & \\
\mylabel{arrow1} & & \mylabel{arrow2} & & \\
\ref{arrow1} & & & & \mylabel{arrow3}
\end{tabular}
...figure code...
\end{figure}

```

Beside other additional functions and options this is exactly what the `fancylabel` package does for you. Whenever you insert an object you can use `\fancylabel`. The package will recognize if the label already exists and therefore decides whether to label or to refer.

```
\begin{figure}[h]
...figure code...
  \begin{tabular}{ccc}
    \Huge$\rightarrow$ & \Huge$\leftarrow$ & \\
    \Huge$\rightarrow$ & \Huge$\Leftarrow$ & \\
    \fancylabel{arrow1} & \fancylabel{arrow2} & \\
    \fancylabel{arrow1} & \fancylabel{arrow3} & \\
  \end{tabular}
...figure code...
\end{figure}
```

2 Package Commands

2.1 Label commands

```
\fancylabel \fancylabel[<family>]{<marker>}
\fancylabel* \fancylabel*[<family>]{<marker>}
```

The `\fancylabel` command always checks if a label for `<marker>` already exists and creates it if not. Then it prints the reference as `\fancyref` does. Additionally a `<family>` can be defined (For description see section 3.1). The starred versions create a label without printing the reference.

The argument `<marker>` might also be a comma-separated list of indentifiers. In this case `\fancylabel` creates a label for all so far undefined indentifiers and prints a list of references. This list can be sorted and compressed automatically (see section 3.5 and 3.6).

2.2 Sublabel commands

```
\fancysublabel \fancysublabel[<family>]{<marker>}{<submarker>}
\fancysublabel- \fancysublabel-[<family>]{<marker>}{<submarker>}
\fancysublabel* \fancysublabel*[<family>]{<marker>}{<submarker>}
```

The `\fancysublabel` command is working in the same way as `\fancylabel` while providing a second level of referencing. If the corresponding label for `<marker>` was not previously defined, it will be created as well. The output of `\fancysublabel` is the complete reference consisting in label and sublabel (e.g. 1a). The output of `\fancysublabel-` is only the sublabel (e.g. a). The starred version `\fancysublabel*` creates the sublabel without any output.

While the argument `<marker>` has to be a single identifier, the `<submarker>` argument might be a comma-separated list. In this case `\fancysublabel` creates a sublabel for all so far undefined identifiers and prints a list of references. This list can be sorted and compressed automatically (see section 3.5 and 3.6).

2.3 Ref and subref commands

```

\fancyref    \fancyref[<family>]{<marker>}
\fancyref*  \fancyref* [<family>]{<marker>}
\fancysubref \fancysubref[<family>]{<marker>}{<submarker>}
\fancysubref- \fancysubref- [<family>]{<marker>}{<submarker>}
\fancysubref* \fancysubref* [<family>]{<marker>}{<submarker>}

```

The function of these commands is comparable to their corresponding label commands. `\fancyref` prints the reference of `<marker>` and remembers that this `<marker>` has been requested. `\fancyref*` remembers the request without any output.

`\fancysubref` prints the label and sublabel of `<marker>` and `<submarker>` and remembers the request. `\fancysubref-` prints only the sublabel and remembers the request. `\fancysubref*` does only remember the request.

3 Options

3.1 The families option

```
families= \usepackage[families={<comma-separated list>}]{fancylabel}
```

You can define different families for your labels to have different counters. This can be done with the `families` option. If you don't specify the family in label or ref commands `family=default` will be used. This is the only predefined family.

```

\begin{figure}[h]
...figure code...
  \begin{tabular}{ccc}
    \Huge$\rightarrow$ & & \Large\textcopyright & & \\
    \Huge$\oplus$ & & \Huge$\ominus$ & & \\
    \fancylabel{arrow1} & & \fancylabel[mark]{mark1} & & \\
    \fancylabel[sym]{symbol1} & & \fancylabel[sym]{symbol2} & & \\
  \end{tabular}
...figure code...
\end{figure}

```

The (sub-)labels and (sub-)refs will have `<family>-` as prefix. This Prefix can be changed by redefining `\fancylabelformat<family>` and `\fancysublabelformat<family>`. For example the definition of the default-prefix can be changed by:

```
\renewcommand{\fancylabelformatdefault}{\arabic{fancylabel}}
```

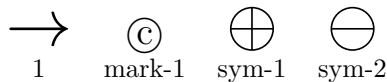


Figure 2: Objects of different families. For this example the families `mark` and `sym` have been defined using the family option.

3.2 The IgnoreFamilies option

`IgnoreFamilies=` `\usepackage[IgnoreFamilies=<boolean>]{fancylabel}`

If the option `IgnoreFamilies=true`, every label will be interpreted as it would be part of the default family.

3.3 The numbering option

`numbering=` `\usepackage[numbering=<value>]{fancylabel}`

The option `numbering` can have one of the following values: `continuous` (default), `ondemand` or `referenced`. We will now have a look on the following code compiled with different values for this option.

```
...figure code...
\begin{tabular}{cccc}
\Huge$\Rightarrow$ & & \Huge$\Leftrightarrow$ & & \\
\Huge$\Leftarrow$ & & \Huge$\rightleftharpoons$ & & \\
\fancylabel{arrow4} & & \fancylabel{arrow5} & & \\
\fancylabel{arrow6} & & \fancylabel{arrow7} & & \\
\end{tabular}
\caption{This is some random text where I refer to symbol%
\fancyref{arrow7} and symbol \fancyref{arrow5}.}
...figure code...
```

`numbering=continuous` The `continuous` numbering means, that every `\fancylabel` will get its reference. The referencing order is in the order `\fancylabel` asked for.

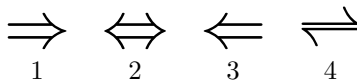


Figure 3: This is some random text where I refer to symbol 4 and symbol 2.

`numbering=ondemand` The `ondemand` numbering means, that a `\fancylabel` will only get its reference, if a `\fancyref` referred to it. The referencing order is in the order `\fancylabel` asked for.

`numbering=referenced` The `referenced` numbering means, that a `\fancylabel` will only get its reference, if a `\fancyref` referred to it. The referencing order is in the order `\fancyref` asked for.

3.6 The compress option

`compress=` `\usepackage[compress=<boolean>]{fancylabel}`

If `compress=true` the output of `fancylabels` und `fancyrefs` compressed. By default this option is `false`.

3.7 The CompressLink option

`CompressLink=` `\usepackage[CompressLink=<separator>]{fancylabel}`

With the `CompressLink` option you can define the separator between the compressed labels. The default value is `CompressLink={-}`.

3.8 The CompressTwo option

`CompressTwo=` `\usepackage[CompressTwo=<boolean>]{fancylabel}`

By this option you can define if compression takes place even if there are only two sequential labels. By default this option is `false`. Example: `CompressTwo=true` → 1-2 and 1-3 or `CompressTwo=false` → 1,2 but 1-3.

3.9 The ListLastSublabelOnly option

`ListLastSublabelOnly=` `\usepackage[ListLastSublabelOnly=<boolean>]{fancylabel}`

If `ListLastSublabelOnly=true` (default), the output of a list of `fancysublabels` of `fancysublabels` is e. g. 1a-d,e. If `ListLastSublabelOnly=false`, the output of a list of `fancysublabels` of `fancysublabels` is e. g. 1a-1d,1e.

3.10 The ListSeparator option

`ListSeparator=` `\usepackage[ListSeparator=<separator>]{fancylabel}`

This option defines the separator between two non-compressed values in a list. By default, `ListSeparator={, \hspace{-0.16667em} }` which is a small white space allowing linebreak.

3.11 The ListLastLabelSeparator option

`ListLastLabelSeparator=` `\usepackage[ListLastLabelSeparator=<separator>]{fancylabel}`

This option defines the separator between the last two values in a list (e.g. `ListLastLabelSeparator={~and~}` → 1a-c,d and f). By default, this option has the same value as the `ListSeparator` option. Note: This option will only be applied in label commands!

3.12 The ListLastRefSeparator option

`ListLastRefSeparator=` `\usepackage[ListLastRefSeparator=<separator>]{fancylabel}`

This option behaves like the `ListLastLabelSeparator` option but will only be applied in ref commands.

3.13 The DisableFamilyCheck option

`DisableFamilyCheck=` `\usepackage[DisableFamilyCheck=<boolean>]{fancylabel}`

When a new `\fancylabel` is defined the fancylabel package checks whether this `<marker>` has already been used in other families and prints a warning. The option `DisableFamilyCheck=true` disables this check. The default value is `DisableFamilyCheck=false`.

4 Customization

4.1 Reset counters

The label counters of each family can be reset by using the following command. Note that `<number>=0` means that the next label will get the value 1. Note: This might cause crazy lists if the options `sort` or `compress` are used!

```
\renewcommand{\fancylabel@familystorage@<family>}{<number>}
```

4.2 Label und sublabel format

The format of labels and sublabels of each family can be adjusted by redefining the corresponding `labelformat`. In the following example you have to replace `<family>` by `default` to adjust the format of the default family.

```
\renewcommand{\fancylabelformat<family>}{\arabic{fancylabel}}
\renewcommand{\fancysublabelformat<family>}{\arabic{fancylabel}\alph{fancysublabel}}
\renewcommand{\fancyonlysublabelformat<family>}{\alph{fancysublabel}}
```

4.3 NewLabelEvent

If you want to execute a macro everytime a new label has been defined you can modify the following command:

```
\fancylabelNewLabelEvent
```


4.4 ShowLabelEvent

If you want to execute a macro everytime a label is shown you can modify the following command:

```
\fancylabelShowLabelEvent
```

4.5 ShowRefEvent

If you want to execute a macro everytime a label is shown you can modify the following command:

```
\fancylabelShowRefEvent
```

4.6 Lists of labels

The fancylabel package draws up a comma-separated list of all labels that have been defined for each family. You can use these lists for further costumization.

```
\fancylabel@list@<family>
```

5 Change history

2014/07/15 Initial version

2017/12/30 Dependencies updated (xifthen instead of ifthen)