The multienv Package

Martin Scharrer
martin@scharrer-scharrer.de

CTAN: http://www.ctan.org/pkg/multienv

VC: https://bitbucket.org/martin_scharrer/multienv

Version v1.0 – 2012/05/20

Abstract

This package provides a `multienv` environment which allows to easily add multiple environments using a key=value syntax. Macros to define environments using this syntax are also provided.

1 Introduction

It is sometimes required to add multiple environments around some text in order to achieve a specific format. This macro simplifies this task especially for larger number of environments by allowing the environments and their arguments to be specified using a key=value syntax. A special `add code` key is also provided to allow arbitrary code being added before and after the environment content. Furthermore macros are provided to allow the definition of new environments which contain multiple other environments and code.

2 Usage

```
\begin{multienv}{⟨env=arg⟩,⟨key=value⟩,...}
  ⟨content⟩
\end{multienv}
```

This environment takes a comma separated list of key=values where the keys are usually environment names and the values the arguments to this environment. The environments are added around the content in the given order, i.e. the first environment is the most outer one and further environments are placed inside it.

At the moment only the following key is defined. Any other key is taken as an environment name.

```
add code={⟨code before⟩}{⟨code after⟩}
```

This adds the given code before and after the content. Any code is permitted including macro argument tokens like '`#1`'.

⟨*environment*⟩
⟨*environment*⟩=⟨*arguments*⟩

Any unknown key is taken as environment name and the value as argument(s) to the environment. If the value does not start with a brace '{', bracket '[' or angle '<' it is taken as a single mandatory argument and is placed inside braces after the environment, otherwise it is placed unchanged after the environment:

⟨*env*⟩ ⇒ `\begin{`⟨*env*⟩`}` ... `\end{`⟨*env*⟩`}`
⟨*env*⟩= ⇒ `\begin{`⟨*env*⟩`}` ... `\end{`⟨*env*⟩`}`
⟨*env*⟩=⟨*arg*⟩ ⇒ `\begin{`⟨*env*⟩`}{`⟨*arg*⟩`}` ... `\end{`⟨*env*⟩`}`
⟨*env*⟩={⟨*arg*⟩} ⇒ `\begin{`⟨*env*⟩`}{`⟨*arg*⟩`}` ... `\end{`⟨*env*⟩`}`
⟨*env*⟩={⟨*arg1*⟩}{⟨*arg2*⟩} ⇒ `\begin{`⟨*env*⟩`}{`⟨*arg1*⟩`}{`⟨*arg2*⟩`}` ... `\end{`⟨*env*⟩`}`
⟨*env*⟩=[⟨*oarg*⟩] ⇒ `\begin{`⟨*env*⟩`}[`⟨*oarg*⟩`]` ... `\end{`⟨*env*⟩`}`
⟨*env*⟩=[⟨*oarg*⟩]{⟨*marg*⟩} ⇒ `\begin{`⟨*env*⟩`}[`⟨*oarg*⟩`]{`⟨*marg*⟩`}` ... `\end{`⟨*env*⟩`}`
⟨*env*⟩=<⟨*aarg*⟩> ⇒ `\begin{`⟨*env*⟩`}<`⟨*aarg*⟩`>` ... `\end{`⟨*env*⟩`}`
⟨*env*⟩=<⟨*aarg*⟩>{⟨*marg*⟩} ⇒ `\begin{`⟨*env*⟩`}<`⟨*aarg*⟩`>{`⟨*marg*⟩`}` ... `\end{`⟨*env*⟩`}`
etc.

```
\begin{multienv*}{⟨env=arg⟩,⟨key=value⟩,...}
  ⟨content⟩
\end{multienv*}
```

This environment is identical to `multienv` but will apply the keys in the reverse order, i.e. every environment or code is added around the existing environments or code.

`\newmultienvironment{`⟨*env name*⟩`}[`⟨*number of args*⟩`][`⟨*default value*⟩`]{`⟨*env=arg*⟩`,...}`

This macro defines a new environment like `\newenvironment` does but uses the `multienv` syntax. The resulting environment will contain all given environments and code. The environment can have arguments including one leading optional argument. The arguments can be used as part of the keys and values. It should be noted that it is not allowed to use arguments in the second argument of the `add code` key, because that code will be part of `\end{`⟨*env name*⟩`}`.

An error is raised if the environment already exists.

`\renewmultienvironment{`⟨*env name*⟩`}[`⟨*number of args*⟩`][`⟨*default value*⟩`]{`⟨*env=arg*⟩`,...}`

Like `\newmultienvironment` but can be used to redefine existing environments. It is is not relevant *how* the environment was defined previously. An error is raised if the environment does not already exist.

`\providemultienvironment{`⟨*env name*⟩`}[`⟨*number of args*⟩`][`⟨*default value*⟩`]{`⟨*env=arg*⟩`,...}`

Like `\newmultienvironment` but the environment is only defined if it is not already defined. The environment is also not defined if there a macro with the same name exists.

3 Examples

The following `multienv` environment usage:

```
\begin{multienv}{enva,envb=arg,add code={code before}{code after},
    envc=[oarg]{marg}}
  content
\end{multienv}
```

is basically identical to:

```
\begin{enva}
  \begin{envb}{arg}
    code before
    \begin{enc}[oarg]{marg}
       content
    \end{enc}
    code after
  \end{envb}
\end{enva}
```

Using `multienv*` the order the environments can be given in the opposite order:

```
\begin{multienv}{envc=[oarg]{marg},add code={code before}{code after},
    envb=arg,enva}
  content
\end{multienv}
```

The following code defines a `centeredminipage` environment which content is placed in a centered `minipage` which inner content is also centered.

```
\newmultienvironment{centeredminipage}[1]
  {center,minipage=#1,add code={\centering}{}}
```

This is basically identical to:

```
\newenvironment{centeredminipage}[1]
  {\begin{center}\begin{minipage}{#1}\centering}
  {\end{minipage}\end{center}}
```

However actually the "plain" form of the environment are used, because it is more efficient and will also produce a correct error message if an incorrect `\end` macro is found:

```
\newenvironment{centeredminipage}[1]
  {\begingroup\center\begingroup\minipage{#1}\centering}
  {\endminipage\endgroup\endcenter\endgroup}
```

The groups are added because some environments require to be placed in a group by themselves, which is normally done by `\begin` and `\end`.