The newverbs package

Martin Scharrer
martin@scharrer-online.de

v1.2 from 2011/02/16

Abstract

This package allows the definition of \verb variants which add TEX code before and after the verbatim text. When used together with the shortvrb package it allows the definition of short verbatim characters which use this variants instead of the normal \verb.

1 Usage

1.1 Defining new variants of \verb

```
\newverbcommand{⟨\macro⟩}{⟨code before⟩}{⟨code after⟩}
\renewverbcommand{⟨\macro⟩}{⟨code before⟩}{⟨code after⟩}
\provideverbcommand{⟨\macro⟩}{⟨code before⟩}{⟨code after⟩}
```

This macros allow the definition of \verb variants. The verbatim content is processed using \verb, but the variants can add TEX code before and after it. The three definition macros use \newcommand*, \renewcommand* and \providecommand* internally to define ⟨*macro*⟩, respectively. Afterwards ⟨*macro*⟩ can be used like \verb. The star version of ⟨*macro*⟩ will use \verb*.

See the implementation of \qverb in section 3.2 for an example.

1.2 Provided \verb variants

Two \verb variants are provided (i.e. with \provideverbcommand) by default.

```
\qverb⟨char⟩⟨verbatim material⟩⟨char⟩
```

This macro adds quote characters around the verbatim material. Two macros are used to insert the quotes: \qverbbeginquote (‘) and \qverbendquote (’). They can be redefined by the user if required. If the csquotes package was loaded beforehand the above macros use its macros \openinnerquote and \closeinnerquote to take advantage of the language dependent quotation marks. See the manual of csquotes for more details.

Using `\qverb`⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩ is equal to `\qverbbeginquote\verb`⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩`\qverbendquote`, or '`\verb`⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩' when the default definition of the quote macros is used.

`\fverb`⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩

This macro adds a frame (`\fbox{}`) around the verbatim text (`\fverb+$&^_%$+` → `$&^_$`). A TEX box is used to store the content first, then the box is framed. The user can define similar command using the following code:

```
      \newverbcommand{\myverb}{\begin{lrbox}{\verbbox}}
                      {\end{lrbox}\mycommand{\usebox{\verbbox}}}
```

The temporary box `\verbbox` is only provided inside a `\...verbcommand`.

1.3 Using `\verb` variants with short verbatim character

`\MakeSpecialShortVerb*{`⟨`\`*macro*⟩`}{\`⟨*char*⟩`}`

This package also defines a special version of the `\MakeShortVerb` macro from the `shortvrb` package. The orignal command `\MakeShortVerb*{\`⟨*char*⟩`}` changes the meaning of ⟨*char*⟩ so that ⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩ is a shorter alternative to `\verb*`⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩.

The new macro `\MakeSpecialShortVerb*{\`⟨*verb variant*⟩`}{\`⟨*char*⟩`}` does the same, but instead of `\verb*` it uses a `\`⟨*verb variant*⟩`*` which needs to be defined using `\newverbcommand`. The package `shortvrb` must be loaded by the user in order to make this macro work. It is not loaded automatically by `newverbs`.

The special meaning of ⟨*char*⟩ can be removed using `shortvrb`'s `\DeleteShortVerb`, i.e. the same way as for characters defined with the normal `\MakeShortVerb`. If a character was already made a short verbatim character it must be "deleted" before it can be redefined by `\MakeShortVerb` or `\MakeSpecialShortVerb`.

Examples:
`\MakeSpecialShortVerb{\qverb}{\"}` will make '`"`' a short, quoting verbatim character: `"$^&$"` → '`$^&$`'.
`\DeleteShortVerb{\"}\MakeSpecialShortVerb{\fverb}{\"}` will change it definition to use `\fverb`: `$^&$`.

2 Compatibility with other verbatim packages

The compatibility with other verbatim packages is not tested yet. This package relies on the normal internal definition of `\verb` and `\MakeShortVerb`. Any package which changes these might break this package. Users which encounter incompatibilities should not hesitate to contact the package author (with details!).

Since v1.2 from 2011/02/16 the new verbatim macros and their short versions can be used inside `tabularx` environments. This package patches an internal macro of `tabularx` to achieve this compatibility.

3 Implementation

3.1 Verb Definition Commands

`\newverbcommand`

`\renewverbcommand`

`\provideverbcommand`

This macro calls the real macro with the to be used definition macro.

```
\newcommand*\newverbcommand{\new@verbcommand\newcommand}
\newcommand*\renewverbcommand{\new@verbcommand\renewcommand}
\newcommand*\provideverbcommand{\new@verbcommand\providecommand}
```

`\new@verbcommand`

`#1`: underlying definition macro
`#2`: macro to define
`#3`: code before
`#4`: code after

The trailing code is inserted by patching `\verb@egroup` which is called by `\verb` after the verbatim content.

```
\def\new@verbcommand#1#2#3#4{%
  #1*#2{%
    \begingroup
    \newverbcommand@settings
    \def\verb@egroup{\verb@orig@egroup#4\endgroup}%
    \def\newverbs@txend{#4\endgroup}
    #3\verb
  }%
}
```

`\newverbs@tabularxsupport`

Enables support for the new verbatim macros inside `tabularx` environments. This environment defines its own almost-verbatim form of `\verb`which lacks the end-macro we patch above. The following code inserts such an end-macro.

```
\def\newverbs@tabularxsupport{%
    \begingroup
    \def\orig@TX@vb##1{\def\@tempa####1##1{\toks@↙
        {####1}\edef\@tempa{\the\toks@}%
        \expandafter\TX@v\meaning\@tempa\\ \\\ifnum↙
            0=‘{\fi}}\@tempa!}
    \ifx\TX@vb\orig@TX@vb
        \endgroup
        \PackageInfo{newverbs}{Patching ’TX@vb’ macro↙
             of the ’tabularx’ package.}%
        \def\TX@vb##1{\def\@tempa####1##1{\toks@↙
            {####1}\edef\@tempa{\the\toks@}%
             \expandafter\TX@v\meaning\@tempa\\ \\\↙
                ifnum0=‘{\fi}\newverbs@txend}\@tempa!}
    \else
        \endgroup
        \PackageWarning{newverbs}{Couldn’t patch ’↙
            TX@vb’ macro of the ’tabularx’ package. ↙
            Definition unknown.}%
    \fi
    \let\newverbs@tabularxsupport\relax
}
```

The end-macro is initially empty and is set for every call of a new verb macro.

```
\def\newverbs@txend{}
```

The support is activated either now or at the begin of the document if the `tabularx` is loaded.

```
\@ifpackageloaded{tabularx}{%
    \newverbs@tabularxsupport
}{%
    \AtBeginDocument{\@ifpackageloaded{tabularx}{\↙
        newverbs@tabularxsupport}{}}%
}
```

`\newverbcommand@settings`

Some settings required for all new `\verb`-like commands. The original end group macro from `\verb` is saved away. Also the 'temp box a' is provided with a user friendly name.

```
\def\newverbcommand@settings{%
  \let\verb@orig@egroup\verb@egroup
  \let\verbbox\@tempboxa
}
```

3.2 Provided New Verb Commands

`\qverb`

Quoting version of `\verb`. Places a quote character before and after the verbatim content: '`verb`'.

```
\provideverbcommand{\qverb}{\qverbbeginquote}{\qverbendquote}
```

`\qverbbeginquote`

`\qverbendquote`

This macros insert the actual quotes. They can be redefined by the user to contain the required quotes. If available the quoting macros of `csquotes` are used.

```
\@ifundefined{openinnerquote}{%
  \def\qverbbeginquote{`}%
  \def\qverbendquote{'}%
}{%
  \def\qverbbeginquote{\openinnerquote}%
  \def\qverbendquote{\closeinnerquote}%
}
```

`\fverb`

A framed version of `\verb`. Stores the verbatim content first into a box. Then the box content is framed.

```
\newverbcommand{\fverb}
  {\setbox\verbbox\hbox\bgroup\color@setgroup}
  {\color@endgroup\egroup\fbox{\box\verbbox}}
```

3.3 Make Special Short Verbatim Characters

\MakeSpecialShortVerb

#1: verbatim macro
Uses the definition of \MakeShortVerb from shortvrb except with \verb replaced with the first argument. The second argument is then read by \@MakeShortVerb.

```
\newcommand*\MakeSpecialShortVerb[1]{%
  \@ifstar
    {\def\@shortvrbdef{#1*}\@MakeShortVerb}%
    {\def\@shortvrbdef{#1}\@MakeShortVerb}%
}
```