

# The newverbs package

Martin Scharrer  
martin@scharrer-online.de

v1.1 from 2010/04/13

## 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  $\text{\TeX}$  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 original 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!).

## 3 Implementation

### 3.1 Verb Definition Commands

`\newverbcommand`

`\renewverbcommand`

`\provideverbcommand`

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

```
1 \newcommand*\newverbcommand{\new@verbcommand\✓  
    newcommand}  
2 \newcommand*\renewverbcommand{\new@verbcommand\✓  
    renewcommand}  
3 \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.

```
4 \def\new@verbcommand#1#2#3#4{%  
5   #1*#2{%  
6     \begingroup  
7     \newverbcommand@settings  
8     \def\verb@egroup{\verb@orig@egroup#4\endgroup}%  
9     #3\verb  
10  }%  
11 }
```

`\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.

```
12 \def\newverbcommand@settings{%
13   \let\verb@orig@egroup\verb@egroup
14   \let\verbbox\@tempboxa
15 }
```

## 3.2 Provided New Verb Commands

`\qverb`

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

```
16 \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.

```
17 \@ifundefined{openinnerquote}{%
18   \def\qverbbeginquote{'}%
19   \def\qverbendquote{'}%
20 }{%
21   \def\qverbbeginquote{\openinnerquote}%
22   \def\qverbendquote{\closeinnerquote}%
23 }
```

`\fverb`

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

```

24 \newverbcommand{\fverb}
25   {\setbox\verbbox\hbox\bgroup\color@setgroup}
26   {\color@endgroup\egroup\fbbox{\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`.

```

27 \newcommand*\MakeSpecialShortVerb[1]{%
28   \@ifstar
29     {\def\@shortvrbdef{#1*}\@MakeShortVerb}%
30     {\def\@shortvrbdef{#1}\@MakeShortVerb}%
31 }

```