The newverbs Package

Martin Scharrer
martin.scharrer@web.de

Version v1.6a – 2022/10/25

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⟩}[⟨\verbmacro⟩]{⟨code before⟩}{⟨code after⟩}
\renewverbcommand{⟨\macro⟩}[⟨\verbmacro⟩]{⟨code before⟩}{⟨code after⟩}
\provideverbcommand{⟨\macro⟩}[⟨\verbmacro⟩]{⟨code before⟩}{⟨code after⟩}
```

This macros allow the definition of \verb variants. The verbatim content is first processed using ⟨\verbmacro⟩ which defaults to \verb, then the given TeX code is added before and afterwards. 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 ⟨\verbmacro⟩* (default: \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.

<div style="border:1px solid">

\qverb⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩

</div>

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 \openautoquote and \closeautoquote 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.

<div style="border:1px solid">

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

</div>

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

<div style="border:1px solid">

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

</div>

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 installed in order to make this macro work. It is 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: $^&$.

```
\collectverb{⟨code⟩}⟨char⟩⟨verbatim material⟩⟨char⟩
\collectverb*{⟨code⟩}⟨char⟩⟨verbatim material⟩⟨char⟩
\collectverb{⟨code⟩}{⟨verbatim material⟩}
\collectverb*{⟨code⟩}{⟨verbatim material⟩}
```

This macro is supposed to be used with its {⟨*code*⟩} argument at the end of user or package macro which want to typeset verbatim material. It will collect everything between the following ⟨*char*⟩ and its next occurrence as verbatim material. An exception is if the following ⟨*char*⟩ is '{', then '}' is taken as the end ⟨*char*⟩ to simulate a normal argument to increase user friendliness. Afterwards ⟨*code*⟩ is expanded with {⟨*verbatim material*⟩} direct behind it. The macro ensures proper font settings to typeset the verbatim material. For this, a group is opened before the material is collected and closed directly after the given code is processed. Therefore all changes done by the ⟨*code*⟩ are local and the material should be typeset directly. (In special cases when the group is disruptive, ⟨*code*⟩ can be a macro which reads both the verbatim material and the **\endgroup** as two arguments. However, then special care must be taken to use the correct font and some of the special characters may be active but have lost their definition.) The starred version will make spaces appear as '␣' instead of displaying them as normal spaces.

```
\Collectverb{⟨code⟩}⟨char⟩⟨verbatim material⟩⟨char⟩
\Collectverb*{⟨code⟩}⟨char⟩⟨verbatim material⟩⟨char⟩
\Collectverb{⟨code⟩}{⟨verbatim material⟩}
\Collectverb*{⟨code⟩}{⟨verbatim material⟩}
```

This macro is supposed to be used with its {⟨*code*⟩} argument at the end of user or package macro which want to collect plain verbatim material suitable to be written in auxiliary files or log messages. It will collect everything between the following ⟨*char*⟩ and its next occurrence as verbatim material without adjusting the font or defining any characters in a special way (besides being verbatim). The starred version will make spaces appear as '␣' when typeset but still be written to auxiliary files as normal spaces. An exception is if the following ⟨*char*⟩ is '{', then '}' is taken as the end ⟨*char*⟩ to simulate a normal argument to increase user friendliness. Afterwards ⟨*code*⟩ is expanded with {⟨*verbatim material*⟩} direct behind it. This macro does not add any group around the code. Should the material be typeset after all a proper font (e.g. **\ttfamily** or **\newverbsfont**) must be enabled manually.

```
\collectverbenv{⟨code⟩}
\collectverbenv*{⟨code⟩}
```

This macro is supposed to be used with its {⟨*code*⟩} argument at the end of the begin-code of an user or package environment definition. It then collects the content of the environment as verbatim material and feeds it as an argument to the provided ⟨*code*⟩ like **\collectverb** does (see there for further details which also apply here). This has the following limitations: When used the **\begin** of the environment must end with a line break, i.e. the source line must not include any other material afterwards. If the environment is defined with arguments, which is supported, the line break must be after the arguments. The **\end** of the macro must be at the beginning of an own source code line. If this conditions are not met incorrect results or an error

may occur. Currently trailing material on the `\begin` line is simply ignored, but this behaviour might change in future versions.

The starred version will make the spaces inside the environment appear as '␣'. Example usage:

```
\newenvironment{myenv}{\maybesomeothercode\collectverbenv{\mycmd}}{\someendcode}
```

`\Collectverbenv{`⟨*code*⟩`}`
`\Collectverbenv*{`⟨*code*⟩`}`

This macro works like `\collectverbenv` but collects the environment content as plain verbatim material suitable to be written in auxiliary files or log messages. After collecting the environment the ⟨*code*⟩ is expanded with {⟨*verbatim material*⟩} direct behind it. This macro does not add any group around the code. Should the material be typeset after all a proper font (e.g. `\ttfamily` or `\newverbsfont`) must be enabled manually.

The starred version will make spaces appear as '␣' when typeset but still be written to auxiliary files as normal spaces.

`\newverbsfont`

Macro which activates the font used by the newverbs package for the verbatim text. This macro can be used manually if verbatim material collected with `\Collectverb` or `\Collectverbenv` should be typeset afterall.

`\verbdef`⟨*macro*⟩⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩
`\verbdef*`⟨*macro*⟩⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩
`\verbdef`⟨*macro*⟩`{`⟨*verbatim material*⟩`}`
`\verbdef*`⟨*macro*⟩`{`⟨*verbatim material*⟩`}`

This macro defines the ⟨*\macro*⟩ as a robust macro which typesets the ⟨*verbatim material*⟩ in the usual verbatim font. For this the material is placed in a brace group with `\newverbsfont`. If a different font is wanted, this macro can be redefined locally.

If the ⟨*\macro*⟩ existed before it will be overwritten silently. If an error should be raced instead use `\newcommand{\macro}{}` just before the `\verbdef`.

Note that this macro is also provided by the verbdef package. If that package is loaded as well it definition of this macro is used, independent on the order of loading the two packages.

`\Verbdef`⟨*macro*⟩⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩
`\Verbdef*`⟨*macro*⟩⟨*char*⟩⟨*verbatim material*⟩⟨*char*⟩
`\Verbdef`⟨*macro*⟩`{`⟨*verbatim material*⟩`}`
`\Verbdef*`⟨*macro*⟩`{`⟨*verbatim material*⟩`}`

This macro uses `\Collectverb` internally to define ⟨*\macro*⟩ as the plain ⟨*verbatim material*⟩. This can be used to define macros for special characters, so these can be used in error or warning messages or be written into auxiliary files.

If the ⟨*\macro*⟩ existed before it will be overwritten silently. If an error should be raced instead use `\newcommand{\macro}{}` just before the `\Verbdef`.

Note that for maximum flexibility the such defined macros are not defined as robust macros. Therefore using them inside sectioning commands they should be protected using `\protect` to avoid syntax issues in the `.aux` file due to verbatim characters.

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

```
1  %<!COPYRIGHT>
2  \ProvidesPackage{newverbs}[%
3  %<!DATE>
4  %<!VERSION>
5  %<*DRIVER>
6      2099/01/01 develop
7  %</DRIVER>
8      Define new 'verb' commands and short verb. characters]
```

3.1   Verb Definition Commands

> **\newverbcommand**

> **\renewverbcommand**

> **\provideverbcommand**

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

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

> **\new@verbcommand**

#1: underlying definition macro
#2: macro to be defined
Checks for optional argument and calls \new@@verbcommand accordingly.

```
12  \def\new@verbcommand#1#2{%
13    \@ifnextchar[%
14      {\new@@verbcommand{#1}{#2}}%
15      {\new@@verbcommand{#1}{#2}[\verb]}%
16  }
```

> **\new@verbcommand**

#1: underlying definition macro
#2: macro to define
#3: verb macro to be used
#4: code before
#5: code after
The trailing code is inserted by patching \verb@egroup which is called by \verb after the verbatim content.

5

```
17  \let\newverbs@end\@empty
18  \def\new@@verbcommand#1#2[#3]#4#5{%
19    #1*#2{%
20      \relax\ifmmode\hbox\else\leavevmode\null\fi
21      \bgroup
22      \newverbcommand@settings
23      \ifx\newverbs@end\@empty
24      \expandafter\def\expandafter\verb@egroup\expandafter{\
          verb@egroup\newverbs@end}%
25      \fi
26      \begingroup\def\@tempa{#5}%
27      \expandafter\expandafter\expandafter\endgroup
28      \expandafter\expandafter\expandafter\def
29      \expandafter\expandafter\expandafter\newverbs@end
30      \expandafter\expandafter\expandafter{\expandafter\@tempa\
          newverbs@end\egroup}%
31      \def\newverbs@txend{#5\egroup}%
32      \verbatim@font\let\verbatim@font\relax
33      #4#3%
34    }%
35  }
```

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

```
36  \def\newverbs@tabularxsupport{%
37    \begingroup
38    \def\origa@TX@vb##1{\def\@tempa####1##1{\toks@{####1}\edef\
        @tempa{\the\toks@}%
39      \expandafter\TX@v\meaning\@tempa\\ \\\ifnum0=`{\fi}}\
          @tempa!}
40    \def\origb@TX@vb##1{\def\@tempa####1##1{\toks@{####1}\edef\
        @tempa{\the\toks@}%
41      \expandafter\TX@v\meaning\@tempa\\\\\\ifnum0=`{\fi}\@tempa
          !}%
42    \ifcase0%
43      \ifx\TX@vb\origa@TX@vb 1\else
44      \ifx\TX@vb\origb@TX@vb 1\fi\fi
45    \relax
46      \endgroup
47      \PackageWarning{newverbs}{Couldn't patch 'TX@vb' macro of
          the 'tabularx' package. Definition unknown.}%
48    \else
49      \endgroup
50      \PackageInfo{newverbs}{Patching 'TX@vb' macro of the '
          tabularx' package.}%
51      \def\TX@vb##1{\def\@tempa####1##1{\toks@{####1}\edef\
          @tempa{\the\toks@}%
52        \expandafter\TX@v\meaning\@tempa\\\\\\ifnum0=`{\fi}\
            newverbs@txend}\@tempa!}%
53    \fi
54    \let\newverbs@tabularxsupport\relax
55  }
```

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

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

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

```
57  \@ifpackageloaded{tabularx}{%
58      \newverbs@tabularxsupport
59  }{%
60      \AtBeginDocument{\@ifpackageloaded{tabularx}{\/
            newverbs@tabularxsupport}{}}%
61  }
```

\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.

```
62  \def\newverbcommand@settings{%
63      \let\verb@orig@egroup\verb@egroup
64      \let\verbbox\@tempboxa
65  }
```

3.2 Provided New Verb Commands

\qverb

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

```
66  \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.

```
67  \@ifundefined{openinnerquote}{%
68      \def\qverbbeginquote{``}%
69      \def\qverbendquote{''}%
70  }{%
71      \def\qverbbeginquote{\openautoquote}%
72      \def\qverbendquote{\closeautoquote}%
73  }
```

\fverb

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

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

3.3 Make Special Short Verbatim Characters

```
77  \RequirePackage{shortvrb}
```

```
      ┌─────────────────────┐
      │ \MakeShortVerb      │
      └─────────────────────┘

78  \def\MakeShortVerb{%
79    \@ifstar
80      {\newverbs@MakeShortVerb*}%
81      {\newverbs@MakeShortVerb{}}%
82  }


      ┌──────────────────────────────┐
      │ \newverbs@MakeShortVerb      │
      └──────────────────────────────┘

          #1: star or empty

83  \def\newverbs@MakeShortVerb#1{%
84    \@ifnextchar[%
85      {\newverbs@@MakeShortVerb{#1}}%
86      {\@MakeSpecialShortVerb{#1}{\verb}}%
87  }


      ┌───────────────────────────────┐
      │ \newverbs@@MakeShortVerb       │
      └───────────────────────────────┘

          #1: star or empty
          #2: verbatim macro

88  \def\newverbs@@MakeShortVerb#1[#2]{%
89    \@MakeSpecialShortVerb{#1}{#2}%
90  }


      ┌──────────────────────────┐
      │ \@MakeSpecialShortVerb   │
      └──────────────────────────┘

          #1: star or empty
          #2: verbatim macro
          #3: escaped short verbatim character
```

Uses the definition of \MakeShortVerb from shortvrb except with \verb replaced with the first argument. The last argument is then read by \@MakeShortVerb.

```
91  \def\@MakeSpecialShortVerb#1#2#3{%
92    %\expandafter\ifx\csname cc\string#3\endcsname\relax
93    %\else
94    %  \DeleteShortVerb{#3}%
95    %\fi
96    \def\@shortvrbdef{#2#1}%
97    \@MakeShortVerb{#3}%
98  }


      ┌────────────────────────┐
      │ \MakeSpecialShortVerb  │
      └────────────────────────┘
```

Checks for the starred version and calls \@MakeSpecialShortVerb appropriately. The star needs to be added again as \@ifstar removes it.

```
99   \newcommand*\MakeSpecialShortVerb{%
100    \@ifstar
101      {\@MakeSpecialShortVerb{*}}%
102      {\@MakeSpecialShortVerb{}}%
103  }
```

3.4   Collect verbatim argument

`\collectverb`

Collects verbatim text which can be typeset. Checks for an existing star.

```
104  \newcommand*\collectverb{%
105      \begingroup
106      \verbatim@font
107      \@ifstar
108          \@scollectverb
109          \@collectverb
110  }
```

`\@collectverb`

#1: <code>
Changes catcodes and ensures that spaces are displayed normally.

```
111  \def\@collectverb#1{%
112      \verb@eol@error
113      \let\do\@makeother
114      \dospecials
115      \@vobeyspaces
116      \frenchspacing
117      \@noligs
118      \@@collectverb{#1}%
119  }
```

`\@scollectverb`

#1: <code>
Changes catcodes.

```
120  \def\@scollectverb#1{%
121      \verb@eol@error
122      \let\do\@makeother
123      \dospecials
124      \@noligs
125      \@@collectverb{#1}%
126  }
```

`\@@collectverb`

#1: <code>
#2: <char>
Defines `\@@@collectverb` to read everything to the next occurrence of ⟨*char*⟩ and feed it to the given ⟨*code*⟩.

```
127  \def\@@collectverb#1#2{%
128      \ifnum`#2=`\{%
129          \catcode`\}\active
130      \else
131          \catcode`#2\active
132      \fi
133      \begingroup
134      \ifnum`#2=`\{%
135          \lccode`\~`\}%
136      \else
137          \lccode`\~`#2%
```

```
138        \fi
139        \lowercase{\endgroup
140            \def\@@@collectverb##1~}{#1{##1}\endgroup}%
141        \@@@collectverb
142    }
```

\collectverbenv

Collects verbatim text which can be typeset. Checks for an existing star.

```
143    \newcommand*\collectverbenv{%
144        \begingroup
145        \verbatim@font
146        \@ifstar
147            \@scollectverbenv
148            \@collectverbenv
149    }
```

\@collectverbenv

    #1: <code>
Changes catcodes and ensures that spaces are displayed normally.

```
150    \def\@collectverbenv#1{%
151        \newverb@catcodes
152        \@vobeyspaces
153        \frenchspacing
154        \@noligs
155        \expandafter\@@collectverbenv\expandafter{\@currenvir}{#1}%
156    }
```

\@scollectverbenv

    #1: <code>
Changes catcodes.

```
157    \def\@scollectverbenv#1{%
158        \newverb@catcodes
159        \@noligs
160        \expandafter\@@collectverbenv\expandafter{\@currenvir}{#1}%
161    }
```

\@@collectverbenv

    #1: <envname>
    #2: <code>

```
162    \begingroup
163    \catcode'\|=0
164    \catcode'\(=1
165    \catcode'\)=2
166    \@makeother\{
167    \@makeother\}
168    \@makeother\\
169    |catcode'|^^M=|active%
170    |gdef|@@collectverbenv#1#2(%
```

```
171        |long|def|@@@collectverb##1^^M##2^^M\end{#1}(#2(##2)|endgroup|/
              end(#1))%
172        |@@@collectverb%
173    )%
174    |endgroup%
```

\Collectverb

Collects argument as plain verbatim and feeds it to the given code. The text is suitable to be printed to
auxiliary files.

```
175    \newcommand*\Collectverb{%
176        \begingroup
177        \@ifstar
178            \@sCollectverb
179            \@Collectverb
180    }
```

\@Collectverb

> #1: <code to be executed afterwards>

```
181    \def\@Collectverb#1{%
182        \verb@eol@error
183        \let\do\@makeother
184        \dospecials
185        \obeyspaces
186        \@@Collectverb{#1}%
187    }
```

\@sCollectverb

> #1: <code to be executed afterwards>

```
188    \def\@sCollectverb#1{%
189        \verb@eol@error
190        \let\do\@makeother
191        \dospecials
192        \@@Collectverb{#1}%
193    }
```

\@@Collectverb

> #1: <code to be executed afterwards>
> #2: <delimiter character>

```
194    \def\@@Collectverb#1#2{%
195        \ifnum'#2='\{%
196            \catcode'\}\active
197        \else
198            \catcode'#2\active
199        \fi
200        \begingroup
201        \ifnum'#2='\{%
202            \lccode'\~'\}%
203        \else
204            \lccode'\~'#2%
```

11

```
205        \fi
206        \lowercase{\endgroup
207            \def\@@@Collectverb##1~}{\endgroup#1{##1}}%
208        \@@@Collectverb
209    }
```

Collects environment content as plain verbatim and feeds it to the given code. The text is suitable to be printed to auxiliary files.

```
210    \newcommand*\Collectverbenv{%
211        \begingroup
212        \@ifstar
213            \@sCollectverbenv
214            \@Collectverbenv
215    }
```

   #1: <code to be executed afterwards>

```
216    \def\@Collectverbenv#1{%
217        \newverb@catcodes
218        \obeyspaces
219        \expandafter\@@Collectverbenv\expandafter{\@currenvir}{#1}%
220    }
```

```
221    \begingroup
222    \catcode'\^^I=\active
223    \gdef\newverb@catcodes{%
224        \let\do\@makeother
225        \dospecials
226        \obeylines
227        \endlinechar=13
228        \catcode'\^^I=\active
229        \def^^I{\newverb@tab}%
230    }
231    \gdef^^I{\newverb@tab}%
232    \endgroup
```

```
233    \edef\newverb@tab{\space}%\space\space\space}
```

   #1: <code to be executed afterwards>

```
234    \def\@sCollectverbenv#1{%
235        \newverb@catcodes
236        \expandafter\@@Collectverbenv\expandafter{\@currenvir}{#1}%
237    }
```

<div style="border:1px solid #3366cc; display:inline-block; padding:4px 8px;">**\@@Collectverbenv**</div>

```
238  \begingroup
239  \catcode`\|=0
240  \catcode`\(=1
241  \catcode`\)=2
242  \@makeother\{
243  \@makeother\}
244  \@makeother\\
245  |catcode`|^^M=|active%
246  |gdef|@@Collectverbenv#1#2(%
247      |long|def|@@@Collectverb##1^^M##2^^M\end{#1}(|endgroup#2(##2)|↙
              end(#1))%
248      |@@@Collectverb%
249  )%
250  |gdef|misj(|def^^M(^^J))%
251  %|gdef|misj(|def^^M##1(|ifx##1|endmarker|else|noexpand^^M|↙
          expandafter##1|fi))%
252  |endgroup%
```

<div style="border:1px solid #3366cc; display:inline-block; padding:4px 8px;">**\newverbsfont**</div>

```
253  \newcommand\newverbsfont{%
254      \verbatim@font
255      \frenchspacing
256  }
```

<div style="border:1px solid #3366cc; display:inline-block; padding:4px 8px;">**\Verbdef**</div>

```
257  \newcommand*\Verbdef{%
258      \@ifstar
259          {\@Verbdef*}%
260          {\@Verbdef{}}%
261  }
```

<div style="border:1px solid #3366cc; display:inline-block; padding:4px 8px;">**\@Verbdef**</div>

> #1: <star or empty>
> #2: <macro to be defined>

```
262  \def\@Verbdef#1#2{%
263      \Collectverb#1{\DeclareRobustCommand#2}%
264  }
```

<div style="border:1px solid #3366cc; display:inline-block; padding:4px 8px;">**\verbdef**</div>

Provides an own definition of **\verbdef** which is also defined by the verbdef package.

```
265  \providecommand*\verbdef{%
266      \@ifstar
267          {\newverbs@verbdef*}%
268          {\newverbs@verbdef{}}%
269  }
```

\@Verbdef

#1: <star or empty>
#2: <macro to be defined>

```
270  \def\newverbs@verbdef#1#2{%
271      \Collectverb#1{\newverbs@@verbdef{#2}}%
272  }
```

\@Verbdef

#1: <star or empty>
#2: <macro to be defined>

```
273  \def\newverbs@@verbdef#1#2{%
274      \DeclareRobustCommand{#1}{{\newverbsfont#2}}%
275  }
```