

Natror (Grzegorz Murzynowski)

The gmverb Package*

April 26, 2007

This is (a documentation of) file `gmverb.sty`, intended to be used with $\text{\LaTeX} 2_{\epsilon}$ as a package for a slight redefinition of the `\verb` macro and `verbatim` environment and for short verb marking such as `|\mymacro|`.

Written by Natror (Grzegorz Murzynowski),
natror at o2 dot pl

©2005, 2006 by Natror (Grzegorz Murzynowski).

This program is subject to the \LaTeX Project Public License.

See <http://www.ctan.org/tex-archive/help/Catalogue/licenses.lpl.html> for the details of that license.

LPPL status: "author-maintained".

Many thanks to my \TeX Guru Marcin Woliński for his \TeX nical support.

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{gmverb}
3 [2007/04/24_v0.82_After_shortvrb_(FM)_but_my_way_(GM)]
```

Contents

Intro, Usage	1	The Breakables	4
Installation	2	Almost-Knuthian <code>\ttverbatim</code>	5
Contents of the <code>gmverb.zip</code> Archive	2	The Core: From <code>shortvrb</code>	5
Compiling the Documentation	3	doc- And <code>shortvrb</code> -Compatibility	10
The Code	3	Change History	10
Preliminaries	3	Index	11

Intro, Usage

This package redefines the `\verb` command and the `verbatim` environment so that the verbatim text can break into lines, with % (or another character chosen to be the comment char) as a ‘hyphen’. Moreover, it allows the user to define his own `verbatim`-like environments provided their contents would be not *horribly* long (as long as a macro’s argument may be at most).

This package also allows the user to declare a chosen char(s) as a ‘short verb’ e.g., to write `|\a\verbatim\example|` instead of `\verb|\a\verbatim\example|`.

The `gmverb` package redefines the `\verb` command and the `verbatim` environment in such a way that `_`, `{` and `\` are breakable, the first with no ‘hyphen’ and the other two with the comment char as a hyphen. I.e. `{\subsequent text}` breaks into `{%`

* This file has version number v0.82 dated 2007/04/24.

$\langle subsequent\ text \rangle$ and $\langle text \rangle \backslash \text{mymacro}$ breaks into $\langle text \rangle \% \backslash \text{mymacro}$.

$\backslash \text{fixbslash}$ (If you don't like linebreaking at backslash, there's the $\backslash \text{fixbslash}$ declaration (observing the common scoping rules, hence OCSR) and an analogous declaration for the left brace: $\backslash \text{fixlbrace}$.)

$\backslash \text{VerbHyphen}$ The default 'hyphen' is % since it's the default comment char. If you wish another char to appear at the linebreak, use the $\backslash \text{VerbHyphen}$ declaration that takes $\backslash \langle char \rangle$ as the only argument. This declaration is always global.

$\backslash \text{verbeolOK}$ Another difference is the $\backslash \text{verbeolOK}$ declaration (OCSR). Within its scope, $\backslash \text{verb}$ allows an end of a line in its argument and typesets it just as a space.

As in the standard version(s), the plain $\backslash \text{verb}$ typesets the spaces blank and $\backslash \text{verb*}$ makes them visible.

$\backslash \text{MakeShortVerb}$ Moreover, gmverb provides the $\backslash \text{MakeShortVerb}$ macro that takes a one-char control sequence as the only argument and turns the char used into a short verbatim delimiter, e.g., after $\backslash \text{MakeShortVerb*} \backslash |$ (as you guess, the declaration has its starred version, which is for visible spaces, and the non-starred for the spaces blank) you may type $| \backslash \text{mymacro}$ to get $\backslash \text{mymacro}$ instead of typing $\backslash \text{verb} \backslash \text{mymacro}$. Because the char used in this example is my favourite and used just this way by DEK in the The T_EXbook's format, gmverb provides a macro $\backslash \text{dekclubs}$ as a shorthand for $\backslash \text{MakeShortVerb*} \backslash |$.

$\backslash \text{dekclubs}$ Be careful because such active chars may interfere with other things, e.g., the | with the vertical marker in tables and with the tikz package. If this happens, you can declare e.g., $\backslash \text{DeleteShortVerb} \backslash |$ and the previous meaning of the char used shall be restored.

One more difference between gmverb and shortverb is that the chars $\backslash \text{activeated}$ by $\backslash \text{MakeShortVerb}$ in the math mode behave as if they were 'other', so you may type e.g., $\$ | \$$ to get | and $+ \backslash \text{activeated}$ this way is in the math mode typeset properly etc.

$\backslash \text{OldMakeShortVerb}$ However, if you don't like such a conditional behaviour, you may use $\backslash \text{OldMakeShortVerb}$ instead, what I do when I like to display short verbatims in displaymath .

$\backslash \text{dekclubs}$ There's one more declaration provided by gmverb : $\backslash \text{dekclubs}$, which is a shorthand for $\backslash \text{MakeShortVerb*} \backslash |$ and $\backslash \text{dekclubs*}$ for $\backslash \text{OldMakeShortVerb*} \backslash |$.

So that, after the latter declaration, you can write

$\backslash [| \langle verbatim\ stuff \rangle | \backslash]$

instead of

$\backslash [\backslash \text{hbox} \{ | \langle the\ stuff \rangle | \} \backslash]$

to get a displayed shortverb.

Both versions of $\backslash \text{dekclubs}$ OCSR.

As many good packages, this also does not support any options.

Installation

Just put the gmverb.sty somewhere in the $\text{texmf}/\text{tex}/\text{latex}$ branch. Creating a $\text{texmf}/\text{tex}/\text{latex}/\text{gm}$ directory may be advisable if you consider using other packages written by me.

Then you should refresh your T_EX distribution's files' database most probably.

Contents of the gmverb.zip Archive

The distribution of the gmverb package consists of the following four files.

gmverb.sty
 README
 gmverbDoc.tex
 gmverbDoc.pdf

Compiling the Documentation

The last of the above files (the .pdf, i.e., *this file*) is a documentation compiled from the .sty file by running L^AT_EX on the gmverbDoc.tex file twice, then MakeIndex on the gmverb.idx file, and then L^AT_EX on gmverbDoc.tex once more.

MakeIndex shell command:

```
makeindex -r gmverbDoc
```

The -r switch is to forbid MakeIndex to make implicit ranges since the (code line) numbers will be hyperlinks.

Compiling the documentation requires the packages: gmdoc (gmdoc.sty and gmdoc.cls), gmverb.sty, gmutils.sty and also some standard packages: hyperref.sty, color.sty, geometry.sty, multicols.sty, lmodern.sty and fontenc.sty that should be installed on your computer by default.

If you had not installed the mwcls classes (available on CTAN and present in T_EX Live e.g.), the result of your compilation might differ a bit from the .pdf provided in this .zip archive in formatting: If you had not installed mwcls, the standard article.cls class would be used.

The Code

Preliminaries

```
4 \RequirePackage{gmutils}[2007/04/24]
```

For \firstofone, \afterfi, \gmoobeyspaces, \@ifnextcat and \noexpand's and \expandafter's shorthands \@nx and \@xa resp.

```
5 \bgroup
6 \@makeother\%
7 \firstofone{\egroup
\twelvepercent 8 \def\twelvepercent{}}
```

Someone may want to use another char for comment, but we assume here 'orthodoxy'. Other assumptions in gmdoc are made. The 'knowledge' what char is the comment char is used to put proper 'hyphen' when a verbatim line is broken.

```
\verbhyphen 9 \let\verbhyphen\twelvepercent
```

Provide a declaration for easy changing it. Its argument should be of \langle char \rangle form (of course, a \langle char \rangle₁₂ is also allowed).

```
\VerbHyphen 10 \def\VerbHyphen#1{%
11   {\escapechar\m@ne
12    \@xa\gdef\@xa\verbhyphen\@xa{\string#1}}}
```

As you see, it's always global.

The Breakables

Let’s define a `\discretionary` left brace such that if it breaks, it turns `{%` at the end of line. We’ll use it in almost Knuthian `\ttverbatim`—it’s part of this ‘almost’.

```

13 \bgroup\catcode'\<=1\@makeother\{\catcode'\>=2%
14 \firstofone<\egroup
\breaklbrace 15 \def\breaklbrace<\discretionary<{\verbhyphen}<><{}>%
\twelvelbrace 16 \def\twelvelbrace<{}>%
17 >% of \firstofone
18 \bgroup\catcode'\<=1\catcode'\{=\active\catcode'\>=2
19 \firstofone<\egroup
20 \def\dobreaklbrace<\catcode'\{=\active\def{<\breaklbrace>>%
21 >% end of \firstofone.
```

The `\bslash` macro defined below I use also in more ‘normal’ \TeX ing, e.g., to `\typeout` some `\outer` macro’s name.

```

22 {\catcode'\!=0\@makeother\}%
\bslash 23 !gdef!bslash{\}%
\breakbslash 24 !gdef!breakbslash{!discretionary{!verbhyphen}{\}{\}}%
25 }
```

Sometimes linebreaking at a backslash may be unwelcome. The basic case, when the first CS in a verbatim breaks at the lineend leaving there `%`, is covered by line 182. For the others let’s give the user a countercrank:

```

\fixbslash 26 \newcommand*\fixbslash{\let\breakbslash=\bslash}% to use due to the common
scoping rules. But for the special case of a backslash opening a verbatim scope,
we deal specially in the line 182.
```

Analogously, let’s provide a possibility of ‘fixing’ the left brace:

```

\fixlbrace 27 \newcommand*\fixlbrace{\let\breaklbrace=\twelvelbrace}
28 {\catcode'\!=0%
29 !catcode'\!=\active
30 !gdef!dobreakbslash{!catcode'\!=\active\def{!breakbslash}}%
31 }
```

The macros defined below, `\visiblebreakspaces` and `\twelveclub` we’ll use in the almost Knuthian macro making verbatim. This ‘almost’ makes a difference.

```

32 \bgroup\catcode'\_ =12%
33 \firstofone{\egroup%
\twelvespace 34 \def\twelvespace{\_}%
35 \def\breakabletwelvespace{\discretionary{\_}{\}{\_}}
36 \bgroup\obeyspaces% it’s just re\catcode’ing.
37 \firstofone{\egroup%
\activespace 38 \newcommand*\activespace{\_}%
39 \newcommand*\dobreakvisibleospace{\let\_ =\breakabletwelvespace\obeyspaces}%
%\defining it caused a stack overflow disaster with gmdoc.
40 \newcommand*\dobreakblankospace{\let\_ =\space\obeyspaces}%
41 }
42 \bgroup\@makeother\|
\twelveclub 43 \firstofone{\egroup\def\twelveclub{||}}
```

Almost-Knuthian `\ttverbatim`

`\ttverbatim` comes from The $\text{T}_{\text{E}}\text{X}$ book too, but I add into it a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ macro changing the `\catcodes` and make spaces visible and breakable and left braces too.

```
\ttverbatim 44 \newcommand*\ttverbatim{%
45   \let\do=\do@noligs_\verbatim@nolig@list
46   \let\do=\@makeother_\dospecials
47   \dobreaklbrace\dobreakbslash
48   \dobreakspace
49   \tt
50   \ttverbatim@hook}
```

While typesetting stuff in the QX fontencoding I noticed there were no spaces in verbatims. That was because the QX encoding doesn't have any reasonable char at position 32. So we provide a hook in the very core of the verbatim making macros to set proper fontencoding for instance.

```
51 \@emptify\ttverbatim@hook
11
\VerbT 52 \def\VerbT1{\def\ttverbatim@hook{\fontencoding{T1}\selectfont}}
We wish the visible spaces to be the default.
53 \let\dobreakspace=\dobreakvisiblespace
```

The Core: From `shortvrb`

The below is copied verbatim ;-) from doc.pdf and then is added my slight changes.

```
\MakeShortVerb* 54 \def\MakeShortVerb{%
\MakeShortVerb 55   \@ifstar
56   {\def\@shortvrbdef{\verb*}\@MakeShortVerb}%
57   {\def\@shortvrbdef{\verb}\@MakeShortVerb}}
\@MakeShortVerb 58 \def\@MakeShortVerb#1{%
59   \@xa\ifx\csname_cc\string#1\endcsname\relax
60   \@shortvrbinfo{Made_}{#1}\@shortvrbdef
61   \add@special{#1}%
62   \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
63   \@xa
64   \xdef\csname_cc\string#1\endcsname{\the\catcode'#1}%
65   \begingroup
66   \catcode'\~\active_\lccode'\~'#1%
67   \lowercase{%
68     \global\@xa\let
69     \csname_ac\string#1\endcsname~%
70     \@xa\gdef\@xa~\@xa{%
71       \@xa\ifmmode\@xa\string\@xa~%
72       \@xa\else\@xa\afterfi{\@shortvrbdef~}\fi}}% This terrible number of
        \expandafters is to make the shortverb char just other in the math
        mode (my addition).
73   \endgroup
74   \global\catcode'#1\active
75   \else
```

```

76 \shortvrbinf@empty{#1_already}{\empty\verb(*)}%
77 \fi}
\DeleteShortVerb 78 \def\DeleteShortVerb#1{%
79 \xa\ifx\csname_cc\string#1\endcsname\relax
80 \shortvrbinf@empty{#1_not}{\empty\verb(*)}%
81 \else
82 \shortvrbinf@Deleted_{#1_as}{\empty\verb(*)}%
83 \rem@special{#1}%
84 \global\catcode'#1\csname_cc\string#1\endcsname
85 \global_\xa\let_\csname_cc\string#1\endcsname_\relax
86 \ifnum\catcode'#1=\active
87 \begingroup
88 \catcode'\~\active_\lccode'\~'#1%
89 \lowercase{%
90 \global_\xa\let_\xa~%
91 \csname_ac\string#1\endcsname}%
92 \endgroup_\fi_\fi}

My little addition

93 \@ifpackageloaded{gmdoc}{%
94 \def\gmvp@packname{gmdoc}}{%
95 \def\gmvp@packname{gmverb}}

\shortvrbinf 96 \def\shortvrbinf#1#2#3{%
97 \PackageInfo{\gmvp@packname}{%
98 ~J\empty_#1\@xa@gobble\string#2_a_short_reference
99 for_\xa\string#3}}

\add@special 100 \def\add@special#1{%
101 \rem@special{#1}%
102 \xa\gdef\x@dospecials\x@
103 {\dospecials_\do_#1}%
104 \xa\gdef\x@sanitize\x@
105 {\@sanitize_\@makeoother_#1}}

```

For the commentary on the below macro see the doc package's documentation. Here let's only say it's just amazing: so tricky and wicked use of `\do`. The internal macro `\rem@special` defines `\do` to expand to nothing if the `\do`'s argument is the one to be removed and to unexpandable CSs `\do` and `\do's argument` otherwise. With `\do` defined this way the entire list is just globally expanded itself. Analogous hack is done to the `\@sanitize` list.

```

\rem@special 106 \def\rem@special#1{%
107 \def\do##1{%
108 \ifnum'#1='##1_\else_\@nx\do\@nx##1\fi}%
109 \xdef\dospecials{\dospecials}%
110 \begingroup
111 \def\@makeoother##1{%
112 \ifnum'#1='##1_\else_\@nx\@makeoother\@nx##1\fi}%
113 \xdef\@sanitize{\@sanitize}%
114 \endgroup}

```

And now the definition of `verbatim` itself. As you'll see (I hope), the internal macros of it look for the name of the current environment (i.e., `\currenenvir`'s meaning) to set

their expectation of the environment's `\end` properly. This is done to allow the user to define his/her own environments with `\verbatim` inside them. I.e., as with the `verbatim` package, you may write `\verbatim` in the begdef of your environment and then necessarily `\endverbatim` in its enddef. Of course (or maybe surprisingly), the commands written in the begdef after `\verbatim` will also be executed at `\begin{environment}`.

```

verbatim 115 \def\verbatim{\@beginparpenalty\predisplaypenalty\@verbatim
\verbatim 116 \frenchspacing\gmoveyspaces\@xverbatim}% in the LATEX version there's %^
          \vobeyspaces instead of \gmoveyspaces.
verbatim* 117 \@namedef{verbatim*}{\@beginparpenalty\predisplaypenalty\@verbatim
          118 \@sxverbatim}
\endverbatim 119 \def\endverbatim{\@par
          120 \ifdim\lastskip>\z@
          121 \@tempskipa\lastskip\vskip-\lastskip
          122 \advance\@tempskipa\parskip\advance\@tempskipa-\@outerparskip
          123 \vskip\@tempskipa
          124 \fi
          125 \addvspace\@topsepadd
          126 \@endparenv}
          127 \n@melet{endverbatim*}{endverbatim}
          128 \begingroup\catcode'\!=0%
          129 \catcode'\[=1\catcode'\]=2%
          130 \catcode'\{=\active
          131 \@makeother\}%
          132 \catcode'\=\active%
\@xverbatim 133 !gdef!\@xverbatim[%
          134 !edef!\verbatim@edef[%
          135 !def!noexpand!\verbatim@end%
          136 #####1!noexpand\end!noexpand{!\currentenvir}%
          137 #####1!noexpand!end[!\currentenvir]]%
          138 !verbatim@edef
          139 !verbatim@end]%
          140 !endgroup
\@sxverbatim 141 \let\@sxverbatim=\@xverbatim

```

F. Mittelbach says the below is copied almost verbatim from L^AT_EX source, modulo `\checkpercent`.

```

\@verbatim 142 \def\@verbatim{%

```

Originally here was just `\trivlist\item[]`, but it worked badly in my document(s), so let's take just highlights of it.

```

          143 \parsep\parskip
          From \@trivlist:
          144 \if@noskipsec\leavevmode\fi
          145 \@topsepadd\topsep
          146 \ifvmode
          147 \advance\@topsepadd\partopsep
          148 \else
          149 \unskip\par
          150 \fi

```

```

151 \topsep_\topsepadd
152 \advance\topsep_\parskip
153 \outerparskip_\parskip
    (End of \trivlistlist and \@trivlist highlights.)
154 \@@par\addvspace\topsep
155 \if@minipage\else\vskip\parskip\fi
156 \leftmargin\parindent% please notify me if it's a bad idea.
157 \advance\totalleftmargin\leftmargin
158 \raggedright
159 \leftskip\totalleftmargin% so many assignments to preserve the list think-
    ing for possible future changes. However, we may be sure no internal list
    shall use \totalleftmargin as far as no inner environments are possible
    in verbatim(*).
160 \@@par% most probably redundant.
161 \tempswafalse
162 \def\par{% but I don't want the terribly ugly empty lines when a blank line is met.
    Let's make them gmdoc-like i.e., let a vertical space be added as in between
    stanzas of poetry. Originally \if@tempswa\hbox{}\fi, in my version will
    be
163 \ifvmode\if@tempswa\addvspace\stanzaskip\tempswafalse\fi\fi
164 \@@par
165 \penalty\interlinepenalty_\check@percent}%
166 \everypar{\@tempwattrue\hangindent\verbatimhangindent\hangafter\@ne}%
    since several chars are breakable, there's a possibility of breaking some lines.
    We wish them to be hanging indented.
167 \obeylines
168 \ttverbatim}

169 \@ifundefined{stanzaskip}{\newlength\stanzaskip}{}
170 \stanzaskip=\medskipamount
\verbatimhangindent 171 \newlength\verbatimhangindent
172 \verbatimhangindent=3em
173 \providecommand*\check@percent{}

    In the gmdoc package shall it be defined to check if the next line begins with a com-
    ment char.

    Similarly, the next macro shall in gmdoc be defined to update a list useful to that
    package. For now let it just gobble its argument.
174 \providecommand*\AddtoPrivateOthers[1]{}

    Both of the above are \provided to allow the user to load gmverb after gmdoc (which
    would be redundant since gmdoc loads this package on its own, but anyway should be
    harmless).

    Let's define the 'short' verbatim command.
\verb* 175 \def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
\verb 176 \bgroup
177 \ttverbatim
178 \gm@verb@eol
179 \@ifstar{\@sverb@chbsl}{\gmobeyspaces\frenchspacing\@sverb@chbsl}}% in
    the LATEX version there's \vobeyspaces instead of \gmobeyspaces.
180 \def\@sverb@chbsl#1{\@sverb#1\check@bslash}

```



```
181 \def\@def@breakbslash{\breakbslash}% because \ is \defined as \breakbslash
    not \let.
```

For the special case of a backslash opening a (short) verbatim, in which it shouldn't be breakable, we define the checking macro.

```
\check@bslash 182 \def\check@bslash{\@ifnextchar{\@def@breakbslash}{\bslash@gobble}{}}
183 \let\verb@balance@group\@empty
\verb@egroup 184 \def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}
\gm@verb@eol 185 \let\gm@verb@eol\verb@eol@error
```

The latter is a L^AT_EX 2_ε kernel macro that `\active`ates line end and defines it to close the verb group and to issue an error message. We use a separate CS 'cause we are not quite positive to the forbidden line ends idea. (Although the allowed line ends with a forgotten closing shortverb char caused funny disasters at my work a few times.) Another reason is that `gmdoc` wishes to redefine it for its own queer purpose.

However, let's leave my former 'permissive' definition under the `\verb@eol` name.

```
186 \begingroup
187 \obeylines\obeyspaces%
\verb@eolOK 188 \gdef\verb@eolOK{\obeylines%
189 \def~M{\check@percent}%
190 }%
191 \endgroup
```

The `\check@percent` macro here is `\provided` to be `\@empty` but in `gmdoc` employed shall it be.

Let us leave (give?) a user freedom of choice:

```
\verbeolOK 192 \def\verbeolOK{\let\gm@verb@eol\verb@eolOK}
```

And back to the main matter,

```
193 \def\@sverb#1{%
194   \catcode'#1\active\lccode'\~'#1%
195   \gdef\verb@balance@group{\verb@egroup
196     \@latex@error{Illegal use of \bslash\verb\command}\@ehc}%
197   \aftergroup\verb@balance@group
198   \lowercase{\let~\verb@egroup}}
\verbatim@nolig@list 199 \def\verbatim@nolig@list{\do\` \do\< \do\> \do\, \do\' \do\~}
\do@noligs 200 \def\do@noligs#1{%
201   \catcode'#1\active
202   \begingroup
203   \lccode'\~='#1\relax
204   \lowercase{\endgroup\def~{\leavevmode\kern\z@\char'#1}}}
```

And finally, what I thought to be so smart and clever, now is just one of many possible uses of a general almost Rainer Schöpf's macro:

```
\dekclubs 205 \def\dekclubs{\@ifstar{\OldMakeShortVerb*}{\MakeShortVerb*}}
```

But even if a shortverb is unconditional, the spaces in the math mode are not printed. So,

```
\edverbs 206 \newcommand*\edverbs{%
207   \let\gmv@dismath\[%
208   \let\gmv@edismath\]
```

```

209 \def\[%
210 \ifnextac\gmvd@isverb\gmvd@ismath}%
211 \let\edverbs\relax}%
212 \def\gmvd@isverb{%
213 \gmvd@ismath
214 \hbox\bgroup\def\[\{\egroup\gmvd@edismath}}

```

doc- And shortvrb-Compatibility

One of minor errors while T_EXing doc.dtx was caused by my understanding of a ‘shortverb’ char: at my settings, in the math mode an active ‘shortverb’ char expands to itself’s ‘other’ version thanks to \string. doc/shortvrb’s concept is different, there a ‘shortverb’ char should work as usual in the math mode. So let it may be as they wish:

```

\old@MakeShortVerb 215 \def\old@MakeShortVerb#1{%
216 \ifx\csname_cc\string#1\endcsname\relax
217 \shortvrbinfo{Made_}{#1}\shortvrbdef
218 \add@special{#1}%
219 \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
220 \ifx
221 \xdef\csname_cc\string#1\endcsname{\the\catcode'#1}%
222 \begingroup
223 \catcode'\~\active_\lccode'\~'#1%
224 \lowercase{%
225 \global\ifx\let\csname_ac\string#1\endcsname~%
226 \ifx\gdef\csname~\csname{%
227 \shortvrbdef~}}%
228 \endgroup
229 \global\catcode'#1\active
230 \else
231 \shortvrbinfo\empty{#1_already}{\empty\verb(*)}%
232 \fi}

\OldMakeShortVerb 233 \def\OldMakeShortVerb{\begingroup
234 \let\@MakeShortVerb=\old@MakeShortVerb
235 \ifstar{\eg@MakeShortVerbStar}{\eg@MakeShortVerb}}

\eg@MakeShortVerbStar 236 \def\eg@MakeShortVerbStar#1{\MakeShortVerb*#1\endgroup}
\eg@MakeShortVerb 237 \def\eg@MakeShortVerb#1{\MakeShortVerb#1\endgroup}

238 \endinput% for the Tradition.

```

Change History

v0.79	\ttverbatim@hook added, 5
\edverbs:	v0.81
added, 9	General:
v0.80	\afterfi made two-argument, 10
\edverbs:	v0.82
debugged, i.e. \hbox added back and	General:
redefinition of \[, 9	Checksum 663, 1
\ttverbatim:	

Index

Numbers written in *italic* refer to the code lines where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used. The numbers preceded with ‘p.’ are page numbers. All the numbers are hyperlinks.

<code>\-</code> , 199	<code>\do@noligs</code> , 45, <u>200</u>	<code>\OldMakeShortVerb*</code> , 205
<code>\@MakeShortVerb</code> , 56, 57, <u>58</u> , 234	<code>\dobreakblankspace</code> , <u>40</u>	<code>\PackageInfo</code> , 97
<code>\@currentenv</code> , 136, 137	<code>\dobreakbslash</code> , <u>30</u> , 47	<code>\parsep</code> , 143
<code>\@def@breakbslash</code> , <u>181</u> , 182	<code>\dobreaklbrace</code> , <u>20</u> , 47	<code>\partopsep</code> , 147
<code>\@emptify</code> , 51	<code>\dobreakspace</code> , 48, <u>53</u>	<code>\predisplayspace</code> , 115, 117
<code>\@ifnextac</code> , 210	<code>\dobreakvisiblespace</code> , <u>39</u> , 53	<code>\rem@special</code> , 83, 101, <u>106</u>
<code>\@nx</code> , 108, 112	<code>\edverbs</code> , <u>206</u> , 211	<code>\RequirePackage</code> , 4
<code>\@shortvrbdef</code> , 56, 57, 60, 72, 217, 227	<code>\eg@MakeShortVerb</code> , 235, <u>237</u>	<code>\stanzaskip</code> , 163, 169, 170
<code>\@shortvrbinfo</code> , 60, 76, 80, 82, <u>96</u> , 217, 231	<code>\eg@MakeShortVerbStar</code> , 235, <u>236</u>	<code>\ttverbatim</code> , <u>44</u> , 168, 177
<code>\@sverb@chbsl</code> , 179, <u>180</u>	<code>\endverbatim</code> , <u>119</u>	<code>\ttverbatim@hook</code> , 50, 51, 52
<code>\@topsep</code> , 151, 152, 154	<code>\everypar</code> , 166	<code>\twelveclub</code> , <u>43</u>
<code>\@topsepadd</code> , 125, 145, 147, 151	<code>\fixbslash</code> , <i>p. 2</i> , <u>26</u>	<code>\twelvebrace</code> , <u>16</u> , 27
<code>\@xa</code> , 12, 59, 63, 68, 70, 71, 72, 79, 85, 90, 98, 99, 102, 104, 216, 220, 225, 226	<code>\fixlbrace</code> , <i>p. 2</i> , <u>27</u>	<code>\twelvepercent</code> , <u>8</u> , 9
<code>\activespace</code> , <u>38</u>	<code>\fontencoding</code> , 52	<code>\twelvespace</code> , <u>34</u>
<code>\add@special</code> , 61, <u>100</u> , 218	<code>\gm@verb@eol</code> , 178, <u>185</u> , 192	<code>\verb</code> , 57, 76, 80, 82, <u>175</u> , 231
<code>\AddtoPrivateOthers</code> , 62, 174, 219	<code>\gmobeyspaces</code> , 116, 179	<code>\verb*</code> , 56, <u>175</u>
<code>\afterfi</code> , 72	<code>\gmvdismath</code> , 207, 210, 213	<code>\verb@balance@group</code> , 183, 184, <u>195</u> , 197
<code>\breakabletwelvespace</code> , <u>35</u> , 39	<code>\gmvdisverb</code> , 210, 212	<code>\verb@egroup</code> , <u>184</u> , 195, 198
<code>\breakbslash</code> , <u>24</u> , 26, 30, 181	<code>\gmvedismath</code> , 208, 214	<code>\verb@eol@error</code> , 185
<code>\breaklbrace</code> , <u>15</u> , 20, 27	<code>\gmvpackname</code> , <u>94</u> , 95, 97	<code>\verb@eolOK</code> , <u>188</u> , 192
<code>\bslash</code> , <u>23</u> , 26, 182, 196	<code>\if@noskipsec</code> , 144	<code>\verbatim</code> , 115
<code>\check@bslash</code> , 180, <u>182</u>	<code>\interlinepenalty</code> , 165	<code>verbatim</code> , <u>115</u>
<code>\check@percent</code> , 165, 173, 189	<code>\leftmargin</code> , 156, 157	<code>verbatim*</code> , <u>117</u>
<code>\dekclubs</code> , <i>p. 2</i> , <u>205</u>	<code>\MakeShortVerb</code> , <i>p. 2</i> , <u>54</u> , 237	<code>\verbatim@edef</code> , 134, 138
<code>\dekclubs*</code> , <i>p. 2</i>	<code>\MakeShortVerb*</code> , <u>54</u> , 205, 236	<code>\verbatim@end</code> , 135, 139
<code>\DeleteShortVerb</code> , <i>p. 2</i> , <u>78</u>	<code>\n@melet</code> , 127	<code>\verbatim@nolig@list</code> , 45, <u>199</u>
	<code>\newlength</code> , 169, 171	<code>\verbatimhangindent</code> , 166, <u>171</u> , 172
	<code>\obeyspaces</code> , 36, 39, 40, 187	<code>\verbeolOK</code> , <i>p. 2</i> , <u>192</u>
	<code>\old@MakeShortVerb</code> , <u>215</u> , 234	<code>\VerbHyphen</code> , <i>p. 2</i> , <u>10</u>
	<code>\OldMakeShortVerb</code> , <i>p. 2</i> , <u>233</u>	<code>\verbhyphen</code> , <u>9</u> , 12, 15, 24
		<code>\VerbT</code> , <u>52</u>