

Natror (Grzegorz Murzynowski)

The gmverb Package^{*}

November 19, 2007

This is (a documentation of) file gmverb.sty, intended to be used with L^AT_EX 2_ε 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, 2007 by Natror (Grzegorz Murzynowski).

This program is subject to the L^AT_EX Project Public License.

See <http://www.ctan.org/tex-archive/help/Catalogue/licenses.lpp1.html> for the details of that license.

LPPL status: "author-maintained".

Many thanks to my T_EX Guru Marcin Woliński for his T_EXnical support.

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{gmverb}
3 [2007/11/13 v0.84 After shortvrb (FM) but my way (GM)]
```

Contents

Intro, Usage	1	The Breakables	3
Installation	2	Almost-Knuthian <code>\ttverbatim</code>	5
Contents of the gmverb.zip Archive	2	The Core: From shortvrb	5
Compiling the Documentation	3	doc- And shortvrb-Compatibility	9
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 v0.84 dated 2007/11/13.

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

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

`\fixlbrace`

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

`\VerbHyphen`

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

`\verbeolOK`

As in the standard version(s), the plain `\verb` typesets the spaces blank and `\verb*` makes them visible.

`\MakeShortVerb` Moreover, `gmverb` provides the `\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 `\MakeShortVerb*|` (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 $\% \backslash mymacro$ to get $\backslash mymacro$ instead of typing $\verb+ \backslash mymacro +$. 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 `\dekclubs` as a shorthand for `\MakeShortVerb*|`.

`\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., `\DeleteShortVerb|` and the previous meaning of the char used shall be restored.

`\DeleteShortVerb`

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

`\OldMakeShortVerb`

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

`\dekclubs`

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

`\dekclubs*`

So that, after the latter declaration, you can write

$\llbracket \langle verbatim\ stuff \rangle \rrbracket$

instead of

$\llbracket \langle the\ stuff \rangle \rrbracket$

to get a displayed `shortverb`.

Both versions of `\dekclubs` OCSR.

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

Installation

Unpack the `gmverb-tds.zip` archive (this is an archive conforming the `tds` standard, see CTAN/`tds/tds.pdf`) in a `texmf` directory or put the `gmverb.sty` somewhere in the `texmf/tex/latex` branch on your own. Creating a `texmf/tex/latex/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 and a `tds`-compliant archive.

gmverb.sty
README
gmverbDoc.tex
gmverbDoc.pdf
gmverb.tds.zip

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 gm-docc.cls), gmverb.sty, gmutils.sty and also some standard packages: hyperref.sty, color.sty, geometry.sty, multicol.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, \gmobeyspaces, \@ifnextcat, \foone and \noexpand's and \expandafter's shorthands \@nx and \@xa resp.

```
\xiipercen 5 \foone{\@makeother\%}  
6 {\def\xiipercen{}}
```

Someone may want to use another char for comment, but we assume here 'ortho-doxy'. 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 7 \let\verbhyphen\xiipercen
```

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 8 \def\VerbHyphen#1{%  
9   {\escapechar\m@ne  
10    \@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'.

```
11 \foone{\catcode`\[=1 \@makeother\{ \catcode`\]=2 }%  
12 [%
```

```

\breaklbrace 13 \def\breaklbrace[\discretionary[{%
14 \verbhyphen] [] [{%
15 }] %
\Xiilbrace 16 \def\Xiilbrace[{%
17 ] %
18 ] % of \firstofone
19 \foone{\catcode`\[=1 \catcode`\{=\active \catcode`\]=2 }%
20 [%
\dobreaklbrace 21 \def\dobreaklbrace[\catcode`\{=\active
22 \def{%
\breaklbrace 23 [\breaklbrace\gm@lbracehook]] %
24 ]

```

Now we only initialize the hook. Real use of it will be made in gmdoc.

```
25 \relaxen\gm@lbracehook
```

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

```

26 \foone{\catcode`\!=0 \@makeother\}%
27 {%
\bslash 28 !def!bslash{\}%
\breakbslash 29 !def!breakbslash{!discretionary{!verbhyphen}{\}{\}}%
30 }

```

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 186. For the others let’s give the user a countercrank:

```

\fixbslash 31 \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 186.

```

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

```

\fixlbrace 32 \newcommand*\fixlbrace{\let\breaklbrace=\Xiilbrace}
33 \foone{\catcode`\!=0 \catcode`\{=\active}%
34 {%
\dobreakbslash 35 !def!dobreakbslash{!catcode`\!=!active !def\{!breakbslash}}%
\breakbslash 36 }

```

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

```

37 \foone{\catcode`\ =12 }% note this space is 10 and is gobbled by parsing the num-
ber.
\Xiispace 38 {\def\Xiispace{ }%
\breakablexiispace 39 \def\breakablexiispace{\discretionary{ }{ }{ }}
40 \foone\obeyspaces% it’s just re\catcode’ing.
41 {%
\activespace 42 \newcommand*\activespace{ }%
\dobreakvisiblespace 43 \newcommand*\dobreakvisiblespace{\def {\breakablexiispace}\obeyspaces}%
\breakablexiispace \defing it caused a stack overflow disaster with gmdoc.
\dobreakblankspace 44 \newcommand*\dobreakblankspace{\let =\space\obeyspaces}%
45 }
46 \bgroup\@makeother\|
\Xiiclub 47 \firstofone{\egroup\def\Xiiclub{||}}

```

Almost-Knuthian `\ttverbatim`

`\ttverbatim` comes from *The T_EXbook* too, but I add into it a L^AT_EX macro changing the `\catcodes` and make spaces visible and breakable and left braces too.

```
\ttverbatim 48 \newcommand*\ttverbatim{%
49   \let\do=\do@noligs \verbatim@nolig@list
50   \let\do=\@makeother \dospecials
51   \dobreaklbrace\dobreakbslash
52   \dobreakspace
53   \tt
54   \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.

```
55 \@emptify\ttverbatim@hook
\VerbT1 56 \def\VerbT1{\def\ttverbatim@hook{\fontencoding{T1}\selectfont}}
\VerbT   \VerbT
\ttverbatim@hook We wish the visible spaces to be the default.
57 \let\dobreakspace=\dobreakvisiblespace
```

The Core: From `shortvrb`

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

```
\MakeShortVerb* 58 \def\MakeShortVerb{%
\MakeShortVerb 59   \@ifstar
\@shortvrbdef 60   {\def\@shortvrbdef{\verb*}\@MakeShortVerb}%
\@shortvrbdef 61   {\def\@shortvrbdef{\verb}\@MakeShortVerb}}
\@MakeShortVerb 62 \def\@MakeShortVerb#1{%
63   \@xa@ifx\csname cc\string#1\endcsname\relax
64   \@shortvrbinfo{Made }{#1}\@shortvrbdef
65   \add@special{#1}%
66   \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
67   \@xa
68   \xdef\csname cc\string#1\endcsname{\the\catcode`#1}%
69   \begingroup
70   \catcode`\~\active \lccode`\~`#1%
71   \lowercase{%
72     \global\@xa\let
73     \csname ac\string#1\endcsname~%
74     \@xa\gdef\@xa~\@xa{%
75       \@xa@ifmmode\@xa\string\@xa~%
76       \@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).
77   \endgroup
78   \global\catcode`#1\active
79   \else
80   \@shortvrbinfo\@empty{#1 already}{\@empty\verb(*)}%
81   \fi}
\DeleteShortVerb 82 \def\DeleteShortVerb#1{%
83   \@xa@ifx\csname cc\string#1\endcsname\relax
```

```

84 \shortvrbinfinfo\@empty{#1 not}{\@empty\verb(*)}%
85 \else
86 \shortvrbinfinfo{Deleted }{#1 as}{\@empty\verb(*)}%
87 \rem@special{#1}%
88 \global\catcode`#1\csname cc\string#1\endcsname
89 \global \@xa\let \csname cc\string#1\endcsname \relax
90 \ifnum\catcode`#1=\active
91 \begingroup
92 \catcode`\~\active \lccode`\~`#1%
93 \lowercase{%
94   \global\@xa\let\@xa~%
95   \csname ac\string#1\endcsname}%
96 \endgroup \fi \fi}

```

My little addition

```

\gm\@packname 97 \ifpackageloaded{gmdoc}{%
\gm\@packname 98 \def\gm\@packname{gmdoc}}{%
\gm\@packname 99 \def\gm\@packname{gmverb}}
\shortvrbinfinfo 100 \def\@shortvrbinfinfo#1#2#3{%
101   \PackageInfo{\gm\@packname}{%
102     ^^J\@empty #1\@xa\@gobble\string#2 a short reference
103     for \@xa\string#3}}
\add@special 104 \def\add@special#1{%
105   \rem@special{#1}%
106   \@xa\gdef\@xa\dospecials\@xa
107   {\dospecials \do #1}%
108   \@xa\gdef\@xa\@sanitize\@xa
109   {\@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 $\langle do's argument \rangle$ otherwise. With \do defined this way the entire list is just globally expanded itself. Analogous hack is done to the \@sanitize list.

```

\rem@special 110 \def\rem@special#1{%
111   \def\do##1{%
112     \ifnum`#1=##1 \else \@nx\do\@nx##1\fi}%
113   \xdef\dospecials{\dospecials}%
114   \begingroup
115   \def\@makeoother##1{%
116     \ifnum`#1=##1 \else \@nx\@makeoother\@nx##1\fi}%
117   \xdef\@sanitize{\@sanitize}%
118   \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., \@currenvir'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 119 \def\verbatim{\@beginparpenalty \predisplayskippenalty \@verbatim
\verbatim

```

```

120 \frenchspacing \gmoveyspaces \@xverbatim}% in the LATEX version there's %%
    \vmoveyspaces instead of \gmoveyspaces.
verbatim* 121 \@namedef{verbatim*}{\@beginparpenalty \predisplaypenalty \@verbatim
122 \sxverbatim}
\endverbatim 123 \def\endverbatim{\@@par
124 \ifdim\lastskip >\z@
125 \@tempskipa\lastskip \vskip -\lastskip
126 \advance\@tempskipa\parskip \advance\@tempskipa -\@outerparskip
127 \vskip\@tempskipa
128 \fi
129 \addvspace\@topsepadd
130 \@endparenv}
131 \n@melet{endverbatim*}{endverbatim}
132 \begingroup \catcode `!=0 %
133 \catcode `[= 1 \catcode`=2 %
134 \catcode`\{=\active
135 \@makeother\}%
136 \catcode`\=\active%
\@xverbatim 137 !gdef!\@xverbatim[%
138 !edef!verbatim@edef[%
139 !def!noexpand!verbatim@end%
140 #####1!noexpand!end!noexpand{!@currentvir}{[%
141 #####1!noexpand!end[!@currentvir]]}%
142 !verbatim@edef
143 !verbatim@end]%
144 !endgroup
\sxverbatim 145 \let\sxverbatim=\@xverbatim
F. Mittelbach says the below is copied almost verbatim from LATEX source, modulo
\check@percent.
\@verbatim 146 \def\@verbatim{%
Originally here was just \trivlist \item[], but it worked badly in my document(s),
so let's take just highlights of it.
147 \parsep\parskip
From \@trivlist:
148 \if@noskipsec \leavevmode \fi
149 \@topsepadd \topsep
150 \ifvmode
151 \advance\@topsepadd \partopsep
152 \else
153 \unskip \par
154 \fi
155 \@topsep \@topsepadd
156 \advance\@topsep \parskip
157 \@outerparskip \parskip
(End of \trivlistlist and \@trivlist highlights.)
158 \@@par\addvspace\@topsep
159 \if@minipage\else\vskip\parskip\fi
160 \leftmargin\parindent% please notify me if it's a bad idea.

```

```

161 \advance\@totalleftmargin\leftmargin
162 \raggedright
163 \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(*).
164 \@@par% most probably redundant.
165 \@tempswafalse
166 \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
167 \ifvmode\if@tempswa\addvspace\stanzaskip\@tempswafalse\fi\fi
168 \@@par
169 \penalty\interlinepenalty \check@percent}%
170 \everypar{\@tempswatrue\hangindent\verbatimhangindent\hangafter\@ne}% since
    several chars are breakable, there's a possibility of breaking some lines. We
    wish them to be hanging indented.
171 \obeylines
172 \ttverbatim}

\stanzaskip 173 \ifundefined{stanzaskip}{\newlength\stanzaskip}{\fi}
174 \stanzaskip=\medskipamount

\verbatimhangindent 175 \newlength\verbatimhangindent
176 \verbatimhangindent=3em

\check@percent 177 \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.

\AddtoPrivateOthers 178 \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* 179 \def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
\verb 180 \bgroup
181 \ttverbatim
182 \gm@verb@eol
183 \@ifstar{\@sverb@chbsl}{\gmobeyspaces\frenchspacing\@sverb@chbsl}}% in
    the LATEX version there's \vobeyspaces instead of \gmobeyspaces.
\@sverb@chbsl 184 \def\@sverb@chbsl#1{\@sverb#1\check@bslash}
\@def@breakbslash 185 \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 186 \def\check@bslash{\@ifnextchar{\@def@breakbslash}{\bslash\@gobble}{}}
187 \let\verb@balance@group\@empty

\verb@egroup 188 \def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}

\gm@verb@eol 189 \let\gm@verb@eol\verb@eol@error

```


The latter is a L^AT_EX_{2_ε} kernel macro that \activeates 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.

```

190 \begingroup
191 \obeylines\obeyspaces%
192 \gdef\verb@eolOK{\obeylines%
\check@percent 193 \def~M{ \check@percent}%
194 }%
195 \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 196 \def\verbeolOK{\let\gm@verb@eol\verb@eolOK}

```

And back to the main matter,

```

197 \def\@sverb#1{%
198   \catcode`#1\active \lccode`~`#1%
199   \gdef\verb@balance@group{\verb@egroup
200     \@latex@error{Illegal use of \bslash verb command}\@ehc}%
201   \aftergroup\verb@balance@group
202   \lowercase{\let~\verb@egroup}}
\verbatim@nolig@list 203 \def\verbatim@nolig@list{\do~\do<\do>\do,\do'\do-}
\do@noligs 204 \def\do@noligs#1{%
205   \catcode`#1\active
206   \begingroup
207   \lccode`~`#1\relax
208   \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 209 \def\dekclubs{\@ifstar{\OldMakeShortVerb*}{\MakeShortVerb*}}

```

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

```

\edverbs 210 \newcommand*\edverbs{%
211   \let\gmvdismath\[%
212   \let\gmvedismath\]%
213   \def\[%
214     \@ifnextac\gmvdisverb\gmvdismath}%
215   \let\edverbs\relax}%
\gmvdisverb 216 \def\gmvdisverb{%
217   \gmvdismath
218   \hbox\bgroup\def\]{\egroup\gmvedismath}}

```

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/shortverb`’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 219 \def\old@MakeShortVerb#1{%
220   \@xa\ifx\csname cc\string#1\endcsname\relax
221   \@shortvrbinf{Made }{#1}\@shortvrbdef
222   \add@special{#1}%
223   \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
224   \@xa
225   \xdef\csname cc\string#1\endcsname{\the\catcode`#1}%
226   \begingroup
227   \catcode`\~\active \lccode`\~`#1%
228   \lowercase{%
229     \global\@xa\let\csname ac\string#1\endcsname~%
230     \@xa\gdef\@xa~\@xa{%
231       \@shortvrbdef~}}%
232   \endgroup
233   \global\catcode`#1\active
234   \else
235   \@shortvrbinf{\@empty{#1 already}}{\@empty\verb(*)}%
236   \fi}

\OldMakeShortVerb 237 \def\OldMakeShortVerb{\begingroup
238   \let\@MakeShortVerb=\old@MakeShortVerb
239   \@ifstar{\eg@MakeShortVerbStar}{\eg@MakeShortVerb}}

\eg@MakeShortVerbStar 240 \def\eg@MakeShortVerbStar#1{\MakeShortVerb*#1\endgroup}
\eg@MakeShortVerb 241 \def\eg@MakeShortVerb#1{\MakeShortVerb#1\endgroup}

242 \endinput% for the Tradition.

```

Change History

v0.79	v0.82
\edverbs:	General:
added, 9	Checksum 663, 1
v0.80	v0.83
\edverbs:	General:
debugged, i.e. \hbox added back and	added a hook in the active left brace
redefinition of \[, 9	definition intended for gmdoc
\ttverbatim:	automatic detection of definitions (in
\ttverbatim@hook added, 5	line 23), 10
v0.81	Checksum 666, 1
General:	v0.84
\afterfi made two-argument (first	General:
undelimited, the stuff to be put after	Checksum 658, 1
\fi, and the other, delimited with	
\fi, to be discarded, 10	

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> , 203	<code>\dobreaklbrace</code> , <u>21</u> , 51	<code>\parsep</code> , 147
<code>\@MakeShortVerb</code> , 60, 61, <u>62</u> , 238	<code>\dobreakspace</code> , 52, <u>57</u>	<code>\partopsep</code> , 151
<code>\@currenvir</code> , 140, 141	<code>\dobreakvisiblespace</code> , <u>43</u> , 57	<code>\predisplayspace</code> , 119, 121
<code>\@def@breakbslash</code> , <u>185</u> , 186	<code>\edverbs</code> , <u>210</u> , 215	<code>\relaxen</code> , 25
<code>\@emptyify</code> , 55	<code>\eg@MakeShortVerb</code> , 239, <u>241</u>	<code>\rem@special</code> , 87, 105, <u>110</u>
<code>\@ifnextac</code> , 214	<code>\eg@MakeShortVerbStar</code> , 239, <u>240</u>	<code>\RequirePackage</code> , 4
<code>\@nx</code> , 112, 116	<code>\endverbatim</code> , <u>123</u>	<code>\stanzaskip</code> , 167, <u>173</u> , 174
<code>\@shortvrbrdef</code> , 60, <u>61</u> , 64, 76, 221, 231	<code>\everypar</code> , 170	<code>\ttverbatim</code> , <u>48</u> , 172, 181
<code>\@shortvrbrinfo</code> , 64, 80, 84, 86, <u>100</u> , 221, 235	<code>\fixbslash</code> , <i>p.</i> 2, <u>31</u>	<code>\ttverbatim@hook</code> , 54, 55, <u>56</u>
<code>\@sverb@chbsl</code> , 183, <u>184</u>	<code>\fixlbrace</code> , <i>p.</i> 2, <u>32</u>	<code>\verb</code> , 61, 80, 84, 86, <u>179</u> , 235
<code>\@topsep</code> , 155, 156, 158	<code>\fontencoding</code> , 56	<code>\verb*</code> , 60, <u>179</u>
<code>\@topsepadd</code> , 129, 149, 151, 155	<code>\foone</code> , 5, 11, 19, 26, 33, 37, 40	<code>\verb@balance@group</code> , 187, 188, 199, 201
<code>\@xa</code> , 10, 63, 67, 72, 74, 75, 76, 83, 89, 94, 102, 103, 106, 108, 220, 224, 229, 230	<code>\gm@lbracehook</code> , 23, 25	<code>\verb@egroup</code> , <u>188</u> , 199, 202
<code>\activespace</code> , <u>42</u>	<code>\gm@verb@eol</code> , 182, <u>189</u> , 196	<code>\verb@eol@error</code> , 189
<code>\add@special</code> , 65, <u>104</u> , 222	<code>\gmobeyspaces</code> , 120, 183	<code>\verb@eol@OK</code> , 192, 196
<code>\AddtoPrivateOthers</code> , 66, <u>178</u> , 223	<code>\gmvdismath</code> , 211, 214, 217	<code>\verbatim</code> , <u>119</u>
<code>\afterfi</code> , 76	<code>\gmvdisverb</code> , 214, <u>216</u>	<code>\verbatim*</code> , <u>121</u>
<code>\breakablexiispace</code> , 39, <u>43</u>	<code>\gmvedismath</code> , 212, 218	<code>\verbatim@edef</code> , 138, 142
<code>\breakbslash</code> , <u>29</u> , 31, <u>35</u> , 185	<code>\gmvpackname</code> , <u>98</u> , <u>99</u> , 101	<code>\verbatim@end</code> , 139, 143
<code>\breaklbrace</code> , <u>13</u> , <u>23</u> , 32	<code>\if@noskipsec</code> , 148	<code>\verbatim@nolig@list</code> , 49, <u>203</u>
<code>\bslash</code> , <u>28</u> , 31, 186, 200	<code>\interlinepenalty</code> , 169	<code>\verbatimhangindent</code> , 170, <u>175</u> , 176
<code>\check@bslash</code> , 184, <u>186</u>	<code>\leftmargin</code> , 160, 161	<code>\verbeolOK</code> , <i>p.</i> 2, <u>196</u>
<code>\check@percent</code> , 169, <u>177</u> , <u>193</u>	<code>\MakeShortVerb</code> , <i>p.</i> 2, <u>58</u> , 241	<code>\VerbHyphen</code> , <i>p.</i> 2, <u>8</u>
<code>\dekclubs</code> , <i>p.</i> 2, <u>209</u>	<code>\MakeShortVerb*</code> , <u>58</u> , 209, 240	<code>\verbhyphen</code> , <u>7</u> , 10, 14, 29
<code>\dekclubs*</code> , <i>p.</i> 2	<code>\n@melet</code> , 131	<code>\VerbT</code> , <u>56</u>
<code>\DeleteShortVerb</code> , <i>p.</i> 2, <u>82</u>	<code>\newlength</code> , 173, 175	<code>\VerbT1</code> , <u>56</u>
<code>\do@noligs</code> , 49, <u>204</u>	<code>\obeyspaces</code> , 40, 43, 44, 191	<code>\xiiclub</code> , <u>47</u>
<code>\dobreakblankspace</code> , <u>44</u>	<code>\old@MakeShortVerb</code> , <u>219</u> , 238	<code>\xiilbrace</code> , <u>16</u> , 32
<code>\dobreakbslash</code> , <u>35</u> , 51	<code>\OldMakeShortVerb</code> , <i>p.</i> 2, <u>237</u>	<code>\xiipercents</code> , <u>6</u> , 7
	<code>\OldMakeShortVerb*</code> , 209	<code>\xiispace</code> , <u>38</u>
	<code>\PackageInfo</code> , 101	