

Grzegorz Murzynowski

The gmverb Package^{*}

Copyright © 2005, 2006, 2007, 2008, 2009, 2010, 2011

by Grzegorz ‘Natrór’ Murzynowski

natror (at) gmail (dot) com

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

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

LPL status: "author-maintained".

For documentation please refer to the file(s)

`gmverb.{sty,pdf}`.

```
49 \NeedsTeXFormat{LaTeX2e}
50 \ProvidesPackage{gmverb}
51 [2011/10/14 v0.98 After shortvrb (FM) but my way (GM)]
54 <*master>
    (A handful of meta-settings skipped)
83 </master>
```

Contents

Intro, usage	1	Almost-Knuthian <code>\ttverbatim</code>	6
The package options	4	The core: from <code>shortvrb</code>	7
Installation	4	doc- and shortvrb -compatibility	14
Contents of the gmverb.zip archive	4	Grey visible spaces	14
Compiling of the documentation	4	Verbatim specials—CSes in verbatims	15
The code	5	Partial <code>\verb</code> in arguments	17
Preliminaries	5	Change History	18
The breakables	5	Index	20

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 their 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|`.

^{*} This file has version number dated .

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 `{%`

`<subsequent text>}` and `<text>\mymacro` breaks into `<text>%`
`\mymacro`.

`\nbreakbslash` (If you don’t like line breaking at backslash, there’s the `\nbreakbslash` declaration (observing the common scoping rules, hence OCSR) and an analogous declaration for the left brace: `\nbreaklbrace`.)

`\VerbHyphen` The default ‘hyphen’ is `%` since it’s the default comment char. If you wish another char to appear at the line break, use the `\VerbHyphen` declaration that takes `<char>` as the only argument. This declaration is always global.

`\verbeolOK` 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.

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 `|<mymacro|` to get `\mymacro` instead of typing `\verb+<mymacro+>`. Because the char used in this example is my favourite and used just this way by DEK in the *The T_EX book*’s format, **gmverb** provides a macro `\dekclubs` as a shorthand for `\MakeShortVerb(*)|`.

`\DeleteShortVerb` Be careful because such active chars may interfere with other things, e.g. `|` with the **tikz** package. If this happens, you can declare `\DeleteShortVerb|` and the previous meaning of the char used shall be restored.

One more difference between **gmverb** and **shortvrb** is that the chars `\active`ated by `\MakeShortVerb` in the math mode behave as if they were ‘other’, so you may type e.g., `$2|0$` to get `2|0` and `+ \active`ated this way is in the math mode typeset properly etc.

However, if you don’t like such a conditional behaviour, you may specify some `$$$$`’s (any nonempty sequence) mixed with star(s) if you wish as an optional argument to keep the short-verbatim char behave as short-verbatim also in math mode.

`\OldMakeShortVerb` For compatibility with **gmdoc** and for convenience there is a shorthand for that provided, `\OldMakeShortVerb`.

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

`\edverbs` There’s one more declaration, `\edverbs` that makes `|` checks if the next token is an active char and opens an `\hbox` if so. That is done so that you can write (in `\edverbs`’ and `\dekclubs`’ scope)

`\[<verbatim stuff>|]`

instead of

`\[\hbox{|<verbatim stuff>|}]`

to get a displayed shortverb.

Both versions of `\dekclubs` OCSR.

The `verbatim` environment inserts `\topsep` before and after itself, just as in standard version (as if it was a `list`).

`\VisSpacesGrey` In August 2008 Will Robertson suggested visible spaces for **gmdoc**. I added a respective option to **gmdoc** but I find them so nice that I want to make them available for all verbatim environments so I bring here the declaration `\VisSpacesGrey`. It redefines only the visible spaces so affects `\verb*` and `verbatim*` and not the unstarred versions.
`visspacesgrey` The colour of the visible spaces is named `visspacesgrey` and you can redefine it `xcolor`

way.

`\verbatimspecials`

We also provide the `\verbatimspecials` declaration that takes six arguments:

- #1 `m` a char for verbatim escape char (for catcode 0), has to be unbraced¹,
- #2 `m` a char for group starter (for catcode 1), has to be unbraced,
- #3 `m` a char for group ender (for catcode 2), has to be unbraced,
- [#4] (optional) a char for verbatim math shift (for catcode 3); it has to be in square brackets if present. If absent, nothing is set for the verbatim math shift,
- [#5] (optional) a char for the shorthand for `\metachar`; it has to be in square brackets if present. If provided, e.g., `>` as I suggest in `\GMverbatimspecials`, then it itself becomes an active char let-equal to `\meta`, and a CS made of it, `\>` in this example, becomes `\string`.
- {#6} `b` optional in curly braces, additional stuff (commands) to be executed in a verbatim. All the specials defined this way, except the meta char, if preceded with the escape char, will be typeset verbatim.

For example, after telling \TeX

```
\verbatimspecials < « » [ ¿ ] [ > ] {\def\|{\$ \vert$}}
```

(the slash is Unicode Fractional Slash, spaces are ignored) you can write

```
|\macro<arg«arg. ¿n+1¿»\>[No]Value>(T|F)>|
```

to get

```
\macro{<arg. n + 1>}\[No]Value(T|F)
```

Note also that `<|` is a control sequence so it doesn't delimit the short verbatim `|`'s argument.

The `\verbatimspecials` declaration OCSR. Subsequent uses of it override the previous settings. If you specified the optionals at first and then specify `\verbatimspecials` without optionals, the previous optional settings are forgotten.

`\noverbatimspecials`

To turn the 'verbatim specials' off write `\noverbatimspecials`, which OCSR too.

Note that although we don't provide a 'verbatim superscript' nor 'verbatim subscript', you have the `\sup`s and `\sub`s CS'es defined by `gmutils`.

`\sup`s The 4th argument for the math shift is optional because you can use \LaTeX 's `\(` and `\)`.

`\sub`s The `\verbatimspecials` declaration goes a step further than \LaTeX 's `alltt` and Til Tantau's `beamer`'s `semiverbatim` environments. To get their effect, declare

```
\verbatimspecials\{}
```

`\scanverb`

There is something for verbatims in arguments of commands: `\scanverb[*]{<text>}`. However there are some limitations: if `%` is the comment char (which is usual situation), then you cannot use `%` in `<text>`, or rather, `%` will act as comment char anyway. Moreover, spaces are ignored. This last limitation may be worked around if you declare `\verbatimspecials`, say `<` (fraction slash) as the escape char. Then you can use `<` to put a space which will be typeset blank in the unstarred version and visible with star.

Not so long ago I started to use the 'broken bar' (U+00A6, `|`) character as a hyphen in hyperlinks, because it seems not to occur in hyperlinks at all unlike hyphen. I suggest the same char for verbatims, but if you don't like it, there's the `\verbDiscretionaryHyphen` declaration that takes two arguments. Broken bar is declared as

```
\verbDiscretionaryHyphen{"A6}{|}
```

¹ To be precise, the arguments cannot be wrapped in curly braces because those are recatcoded to 'other'. But if you make some other pair of chars category 1 and 2 that are not on the `\dospecials` list, then you can wrap the arguments in those chars. But what for?

Since version 0.95 (August 2010) this package also provides the `\VerbatimPitch` declaration that modifies the `verbatim` environments and derivatives so that the environment contents (recatcoded, i.e. “sanitised”) are wrapped in the macro `\VerbatimContents` and therefore available after environment’s end (after `\endverbatim` to be precise, so also in the end-def of a derivative `verbatim`).

This may be useful for \TeX nical examples: you can rescan the contents of a `verbatim` with `\scantokens` and execute/typeset it. Such a thing is done in the `gmdoc` package, see the `verbatim@p` environment.

The package options

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

Installation

Unpack the `\jobname-tds.zip` archive (this is an archive that conforms the TDS standard, see [CTAN/tds/tds.pdf](#)) in some `texmf` directory or just put the `gmutils.sty` somewhere in the `texmf/\:tex/\:latex` branch. Creating a `texmf/\:tex/\:latex/\:gm` directory may be advisable if you consider using other packages written by me.

Then you should refresh your \TeX distribution’s files’ database most probably.

Contents of the `gmverb.zip` archive

The distribution of the `gmutils` package consists of the following three files and a TDS-compliant archive.

```
gmverb.sty
README
gmverb.pdf
gmverb.tds.zip
```

Compiling of the documentation

The last of the above files (the `.pdf`, i.e., *this file*) is a documentation compiled from the `.sty` file by running \LaTeX on the `gmverb.sty` file twice (`xelatex gmverb.sty` in the directory you wish the documentation to be in, you don’t have copy the `.\gmdExt` file there, \TeX will find it), then `MakeIndex` on the `\jobname.idx` file, and then \LaTeX on `\jobname.\gmdExt` once more.

`MakeIndex` shell commands:

```
makeindex -r gmverb
makeindex -r -s gmglo.ist -o gmverb.gls gmverb.glo
```

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`, the `gmutils` bundle, `gmiflink.sty` and also some standard packages: `hyperref.sty`, `color.sty`, `geometry.sty`, `multicol.sty`, `lmodern.sty`, `fontenc.sty` that should be installed on your computer by default.

Moreover, you should put the `gmglo.ist` file, a `MakeIndex` style for the changes’ history, into some `texmf/makeindex` (sub)directory.

Then you should refresh your \TeX distribution’s files’ database most probably.

If you had not installed the `mwcls` classes (available on CTAN and present in \TeX 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

```
466 \RequirePackage{gmcommand}[2011/10/12]
```

For `\firstofone`, `\afterfi`, `\gmoveyspaces`, `\@ifnextcat`, `\foone` and `\noexpand`'s and `\expandafter`'s shorthands `\@nx` and `\@xa` resp. and `\DeclareCommand`.

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 480 \let\verbhyphen\xiipercen
```

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

```
\VerbHyphen 486 \def\VerbHyphen#1{%
487   {\escapechar\m@ne
488    \@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'.

```
\breaklbrace 496 \def\breaklbrace{%
497   \discretionary{\type@lbrace\verbhyphen}{}{\type@lbrace}%
498   \yesy}
500 \foone{\catcode`\[=1 \catcode`\{=\active \catcode`\]=2 }%
501 [%
\dobreaklbrace 502   \def\dobreaklbrace[\catcode`\{=\active
503   \def{%
\breaklbrace 504     [\breaklbrace\gm@lbracehook]]%
505 ]
```

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

```
509 \relaxen\gm@lbracehook
```

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

```
514 \foone{\catcode`\!=0 \@makeother\}%
515 {%
\bslash 516   !def!bslash{\}%
517   }% of \foone.
\breakbslash 520 \def\breakbslash{%
521   \discretionary{\verbhyphen}%
522   {\type@bslash}{\type@bslash}\yesy% it seems that we allow hyphenation after
           backslash but hyphenation will be allowed iff \hyphenchar\font is nonnega-
           tive.
525 }% of \breakbslash.
```

Sometimes line breaking at a backslash may be unwelcome. The basic case, when the first CS in a `verbatim` breaks at the line end leaving there `%`, is covered by line 1006. For the others let's give the user a counter-crank:

```
\nbreakbslash 531 \pdef\nbreakbslash{\def\breakbslash{\type@bslash\yeshy}}% to use due to
\breakbslash    the common scoping rules. But for the special case of a backslash opening
                a verbatim scope, we deal specially in the line 1006.
```

Analogously, let's provide a possibility of 'nobreaking' the left brace:

```
\nbreaklbrace 538 \pdef\nbreaklbrace{\def\breaklbrace{\type@lbrace\yeshy}}
\breaklbrace
541 \foone{\catcode`\!=0 \catcode`\=\active}%
542 {%
\dobreakbslash 543 !def!dobreakbslash{!catcode`\!=!active !def\{!breakbslash}}%
\breakbslash    544 }
```

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

```
\breakablevisspace 551 \def\breakablevisspace{\discretionary{\visiblespace}{\visiblespace}}
                    The \visiblepace macro is \let in gutils to \xiispace or to \xxt@visiblepace
                    of xltxttra if available.
```

```
555 \foone\obeyspaces% it's just re\catcode'ing.
556 {%
```

```
\dobreakvisiblepace 557 \newcommand*\dobreakvisiblepace{\def {%
\breakablevisspace    \breakablevisspace}\obeyspaces}% \defing it caused a stack overflow
\breakablevisspace    disaster with gmdoc.
```

```
\dobreakblankspace 559 \newcommand*\dobreakblankspace{\let =\space\obeyspaces}%
560 }
```

```
563 \foone{\@makeother\|}{%
```

```
\xiiclub 564 \def\xiiclub{||}}
```

Almost-Knuthian `\ttverbatim`

`\ttverbatim` comes from *The T_EX book* 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 573 \pdef\ttverbatim{%
574 \let\do=\do@noligs \verbatim@nolig@list
575 \let\do=\@makeother \dospecials
576 \dobreaklbrace\dobreakbslash
577 \dobreakspace
578 \makeatletter
582 \ifhmode
583 \setspaceskip
584 \fi
585 \verbatimfont
586 \xdef\gmv@storedhyphenchar{\the\hyphenchar\font}%
```

Assignment of the hyphenchar is always global so let the above edefinition be also such.

```
589 \hyphenchar\font=\gmv@hyphenchar
590 \ttverbatim@hook
591 }% of \ttverbatim
```

(2010/08/14, v0.993:) rigid `\tt` in `\ttverbatim` changed to redefinable `\verbatimfont` due to absurd problems with bad fontifying of `gmdoc`

```
\verbatimfont 597 \def\verbatimfont{\tt}
```

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.

```

604 \@emptyify\ttverbatim@hook
\VerbT1 607 \def\VerbT1{\def\ttverbatim@hook{\fontencoding{T1}\selectfont}}
\VerbT  \VerbT
\ttverbatim@hook We wish the visible spaces to be the default.
611 \let\dobreakspace=\dobreakvisiblespace

```

The core: from shortvrb

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

```

\MakeShortVerb 620 \DeclareCommand \MakeShortVerb{
621 #1 Q {*$} %
622 #2 m % the char to be made shortverb. Has to be provided as \langle char \rangle (back-
        slashed).
623 }{%
624 \gmu@ifstrany {*}{#1}
\@shortvrbdef 625 {\def\@shortvrbdef{\verb*}}%
\@shortvrbdef 626 {\def\@shortvrbdef{\verb}}
628 \gmu@ifstrany {$}{#1}%

```

Depending on whether \$ was specified as optional argument to `\MakeShortVerb`, we define the CS `\gmV@mm\langle the char \rangle` as either sth. special or sth. usual in the math mode:

```

635 {\gmV@MakeShortVerbAlsoInMath #2 }% extracted to a macro because we reassign
        it while typesetting LATEX 2ε Sources.
637 {\gmV@MakeShortVerbNotInMath #2}%
639 \@MakeShortVerb #2%
640 }% of \MakeShortVerb.
\gmV@MakeShortVerbAlsoInMath 642 \pdef\gmV@MakeShortVerbAlsoInMath #1{% If some dollars were specified (paid ;-))
        ) in #1, then we direct the shortverb char to behave in the math mode the same
        as in normal text.
646 \Name\glet {gmV@mm\string#1}\@firstofone
647 }%
\gmV@MakeShortVerbNotInMath 649 \pdef\gmV@MakeShortVerbNotInMath #1{% If no dollar was paid ;-)) in #1, we get
        vicious and define shortverb to act in math mode as normal (other) char.
652 \edef\gmV@tempA {\strip@bslash {#1}}%
654 \@XA {%
655 \Name\gdef {gmV@mm\string#1}}%
656 \@xa{\@xa % these two expandafters serve to get “other” version of (unbackslashed)
        #1.
658 \ifmmode \gmV@tempA

```

Two subsequent expandafters become part of the definition and serve to gobble or execute the short-verbatim macro.

```

661 \@xa \@gobble
662 \else
663 \@xa \@firstofone
664 \fi
665 }%

```

666 }% of if to be normal in math mode

We store the original catcode of a shortverb char in a CS `\gmV@cc` (*the char*). Moreover, if the char is originally active (of catcode 13), then we create another CS to store its original meaning, `\gmV@ac` (*the char*).

```
\@MakeShortVerb 674 \def\@MakeShortVerb#1{%
675   \@xa\ifx\csname gmV@cc\string#1\endcsname\relax
676     \@shortvrbinf{Made }{#1}\@shortvrbdef
677     \add@special{#1}%
678     \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
680     \@xa
681     \xdef\csname gmV@cc\string#1\endcsname{\the\catcode`#1}%
682     \begingroup
683     \catcode`\~\active \lccode`\~=#1%
684     \lowercase{%
685       \Name\glet {ac\string#1}~%
686       \@xa\gdef\@xa~\@xa{%
687         \csname \strip@bslash{\gmV@mm}\string#1\endcsname
688         {\@shortvrbdef~}%
689       }% of active #1 def
690     }% of lowercase
691     \endgroup
692     \global\catcode`#1\active
693   \else
694     \@shortvrbinf\@empty{#1 already}{\@empty\verb(*)}%
695   \fi
696 }% of \@MakeShortVerb.

\DeleteShortVerb 699 \def\DeleteShortVerb#1{%
700   \@xa\ifx\csname gmV@cc\string#1\endcsname\relax
701     \@shortvrbinf\@empty{#1 not}{\@empty\verb(*)}%
702   \else
703     \@shortvrbinf{Deleted }{#1 as}{\@empty\verb(*)}%
704     \rem@special{#1}%
705     \global\catcode`#1\csname gmV@cc\string#1\endcsname
706     \global \@xa\let \csname gmV@cc\string#1\endcsname \relax
707     \ifnum\catcode`#1=\active
708       \begingroup
709       \catcode`\~\active \lccode`\~`#1%
710       \lowercase{%
711         \global\@xa\let\@xa~%
712         \csname gmV@ac\string#1\endcsname}%
713       \endgroup
714     \fi
715   \fi
716 }

My little addition

720 \ifpackageloaded{gmdoc}{%
\gmv@packname 721   \def\gmv@packname{gmdoc}}{%
\gmv@packname 722   \def\gmv@packname{gmverb}}

\@shortvrbinf 725 \def\@shortvrbinf#1#2#3{%
726   \PackageInfo{\gmv@packname}{%
```



```

727   ^^J\@empty #1\@xa\@gobble\string#2 a short reference
728   for \@xa\string#3}}

```

```

\add@special 731 \def\add@special#1{%
732   \rem@special{#1}%
733   \@xa\gdef\@xa\dospecials\@xa
734   {\dospecials \do #1}%
735   \@xa\gdef\@xa\@sanitize\@xa
736   {\@sanitize \@makeother #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 CSeS `\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 747 \def\rem@special#1{%
748   \def\do##1{%
749     \ifnum`#1=`##1 \else \@nx\do\@nx##1\fi
750   }%
751   \xdef\dospecials{\dospecials}%
752   \begingroup
753   \def\@makeother##1{%
754     \ifnum`#1=`##1 \else \@nx\@makeother\@nx##1\fi}%
755   \xdef\@sanitize{\@sanitize}%
756   \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., `\@currenenv`'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 begin definition of your environment and then necessarily `\endverbatim` in its end definition. Of course (or maybe *surprisingly*), the commands written in the begin definition after `\verbatim` will also be executed at `\begin{environment}`.

```

verbatim 769 \def\verbatim{%
\verbatim 770   \edef\gmv@hyphenpe{\the\hyphenpenalty}%
771   \edef\gmv@exhyphenpe{\the\exhyphenpenalty}%
772   \@beginparpenalty \predisplaypenalty \@verbatim
773   \frenchspacing \gmoveyspaces \@xverbatim
774   \hyphenpenalty=\gmv@hyphenpe\relax
775   \exhyphenpenalty=\gmv@exhyphenpe
776   \hyphenchar\font=\m@ne

```

The line below serves as the delimiter for `\verbatim@PitchContents`, to discard the stuff before it (see l. 850).

```

782   \@gobble\verbatim
783 }% in the LATEX version there's \vobeyspaces instead of \gmoveyspaces.
verbatim* 788 \@namedef{verbatim*}{\@beginparpenalty \predisplaypenalty \@verbatim
789   \@sxverbatim % it's the same as \@xverbatim and defines the verbatim end
              (a macro delimited with \end{curr.envir.})}.
791 }
\endverbatim 793 \def\endverbatim{\@@par

```

%% \hyphenchar\font=\gmv@storedhyphenchar % hyphenchar assignments are
 %% % always global. And for an entire paragraph works the one last in it so we hide
 it.

```
797 \ifdim\lastskip >\z@
798 \@tempkipa\lastskip \vskip -\lastskip
799 \advance\@tempkipa\parskip \advance\@tempkipa -\@outerparskip
800 \vskip\@tempkipa
801 \fi
802 \addvspace\@topsepadd
803 \@endparenv
804 }
```

*

```
807 \n@melet{endverbatim*}{endverbatim}
```

```
810 \begingroup \catcode `!=0 %
811 \catcode `[= 1 \catcode`=2 %
812 \catcode`\{=\active
813 \@makeother\}%
814 \catcode`\=\active%
```

\@xverbatim

```
815 !gdef!\@xverbatim[%
816 [!endlinechar!m@ne !everyeof[!@nx]%
817 !edef!verbatim@currenvir[%
818 !@xa!scantokens!@xa[!@currenvir]%
819 ]% of \verbatim@currenvir. This macro is defined as the meaning of \@cur|
renvir rescanned. It's done specially for the active star in my verbatims.
% \@currenvir is fully expanded but my active star is \protected.
823 !@xa]% and here a little trick with groups:
824 !@xa!def!@xa!verbatim@currenvir
825 !@xa[!verbatim@currenvir]%
826 !edef!verbatim@edef[%
827 !def!@nx!verbatim@end%
828 #####1!noexpand\end!@nx{%
829 !@xa!unexpanded!@xa[!verbatim@currenvir]%
830 }[%
831 !@nx!verbatim@PitchContents #####1%
832 !@nx!verbatim@PitchContents@Delim% added 2010/8/16
833 #####1!@nx!end[!@currenvir]]%
834 !verbatim@edef
835 !verbatim@end]%
836 !endgroup
```

\@sxverbatim

```
840 \let\@sxverbatim=\@xverbatim
```

rbatim@PitchContents@Left
 \verbatim@PitchContents

```
842 \def\verbatim@PitchContents@Left{%
843 \long\def\verbatim@PitchContents
844 ##1\@gobble\verbatim ##2\verbatim@PitchContents@Delim
845 }
```

By default we make \verbatim@PitchContents a gobbler.

```
848 \verbatim@PitchContents@Left {}
```

\VerbatimPitch

```
850 \pdef\VerbatimPitch{%
```

But in this declaration scope we make `\@xverbatim` pitch the contents of `verbatim`s in a macro. We use that in `gmdoc` not to repeat examples' code.

```
855 \verbatim@PitchContents@Left {%
856   \gdef\VerbatimContents{##2}%
857 }%
859 }% of \VerbatimThrow.
```

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

```
\@verbatim 864 \def\@verbatim{%
```

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

```
870 \parsep\parskip
      From \@trivlist:
872 \if@noskipsec \leavevmode \fi
873 \@topsepadd \topsep
874 \ifvmode
875   \advance\@topsepadd \partopsep
876 \else
877   \unskip \par
878 \fi
879 \@topsep \@topsepadd
880 \advance\@topsep \parskip
881 \@outerparskip \parskip
```

(End of `\trivlistlist` and `\@trivlist` highlights.)

```
883 \@@par\advspace\@topsep
884 \if@minipage\else\vskip\parskip\fi%
885 \advance\@totalleftmargin\verbatimleftskip
886 \parskip\verbatimparskip % added 2010/6/2
887 \raggedright
888 \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[*].
892 \@@par% most probably redundant.
893 \@tempwafalse
894 \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
899   \ifvmode\if@tempswa\advspace\stanzaskip\@tempwafalse\fi\fi
900   \@@par
901   \penalty\interlinepenalty \check@percent}%
902 \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.
905 \obeylines
906 \ttverbatim
907 \verbatim@specials
```

```

908 }
\stanzaskip 910 \@ifundefined{stanzaskip}{\newlength\stanzaskip}{}
911 \stanzaskip=\medskipamount
\verbatimleftskip 913 \newskip\verbatimleftskip
915 \verbatimleftskip\leftmargini
\verbatimhangindent 917 \newskip\verbatimhangindent
919 \verbatimhangindent=3em
\verbatimparskip 921 \newskip\verbatimparskip
922 \verbatimparskip\z@skip
\check@percent 924 \providecommand*\check@percent{}

```

In the **gmdoc** package shall it be defined to check if the next line begins with a comment 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 931 \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* 940 \def\verb{%
\verb 941 \relax\ifmmode\hbox\else\leavevmode\null\fi
942 \bgroup
943 \ttverbatim
944 \verbatim@specials
945 \gm@verb@eol
946 \gmu@ifstar
947 {\verb@lasthook\@sverb@chbsl}%
948 {\gmobeyspaces\frenchspacing\verb@lasthook\@sverb@chbsl}}% in the
LATEX version there's \vobeyspaces instead of \gmobeyspaces.
952 \emptify\verb@lasthook

```

```
\@sverb@chbsl 954 \def\@sverb@chbsl#1{\@sverb#1\check@bslash}

```

```
\@def@breakbslash 957 \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.

```

\DefineTypeChar 963 \DeclareCommand\DefineTypeChar{mmo}{%
% #1 m the char as a CS,
% #2 m short name of the char.
% [#3] o the cs of the char in 'other' catcode.
970 \@namedef{gmu@#2wd@name}{#2 wd of
971 \@xa\meaning\the\font\space at \detoken@xa\f@size pt}
973 \@namedef{gmu@measure#2}{%
974 \unless\ifcsname\csname gmu@#2wd@name\endcsname\endcsname

```

```

975 \gmu@measurewd{#1}% \edefs \gmu@tempa as the width of the char and \gmu@tempb
    as the width of the char among 20 copies of itself.
978 \@xa\gn@melet\csname gmu@#2wd@name\endcsname{gmu@tempb}% here we let
    the CS with the name contained in \gmu@<char-name>wd@name to the ex-
    panded value of width of the char measured among copies of it.
982 \fi
983 }% of \gmu@measure<char-name>.
985 \@nameedef{type@#2}{%
986 \@nx\leavevmode
987 \@xanxcs{gmu@measure#2}%
988 \hbox to \@nx\csname
989 \@xanxcs{gmu@#2wd@name}\@nx\endcsname
990 {\gmuIfValueTF{#3}{\@nx#3}{\@xanxcs{#2}}}%
991 \@nx\hss}%
992 }% of \type@<char-name>,
993 }% of \DefineTypeChar.
995 \DefineTypeChar\{\bslash}% this defines \type@bslash and its aides \gmu@measurebslash
    and \gmu@bslashwd@name.
998 \DefineTypeChar\{{lbrace}\xiilbrace}% this defines \type@lbrace and its aux-
    ilia analogous to the above.
\check@bslash 1001 \def\check@bslash{%
1002 \@ifnextchar\@def@breakbslash
1003 {\type@bslash\yeshy\@gobble}% note we allow hyphenation but actually this will
    have effect if \hyphenchar\font allows hyphenation (when it's not > 0 ).
1006 {}
1010 \let\verb@balance@group\@empty
\verb@egroup 1013 \def\verb@egroup{\global\let\verb@balance@group\@empty
    %% \hyphenchar\font=\gmv@storedhyphenchar % \hyphenchar behaves like \hy|
    phenpenalty etc.: the last one in a paragraph is in charge for entire.
1017 \egroup
1018 }
\gm@verb@eol 1022 \let\gm@verb@eol\verb@eol@error
    The latter is a LATEX 2ε 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.
1034 \begingroup
1035 \obeylines\obeyspaces%
1036 \gdef\verb@eolOK{\obeylines%
\check@percent 1037 \def^M{ \check@percent}%
1038 }%
1039 \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 1044 \def\verbeolOK{\let\gm@verb@eol\verb@eolOK}
                And back to the main matter,
1047 \def\@sverb#1{%
1048   \catcode`#1\active \lccode`~=#1%
1049   \gdef\verb@balance@group{\verb@egroup
1050     \@latex@error{Illegal use of \backslash verb command}\@ehc}%
1051   \aftergroup\verb@balance@group
1052   \lowercase{\let~\verb@egroup}% here we make the delimiter to be the macro
                closing the verbatim group.
1054 }

```

```

\verbatim@nolig@list 1056 \def\verbatim@nolig@list{\do\` \do\<\do\>\do\, \do\' \do\ -}

```

```

\do@noligs 1058 \def\do@noligs#1{%
1059   \catcode`#1\active
1060   \begingroup
1061   \lccode`~=#1\relax
1062   \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 1067 \def\dekclubs{\gmu@ifstar{\MakeShortVerb*}\{\MakeShortVerb\}}
\olddekclubs 1068 \def\olddekclubs{\OldMakeShortVerb\}

```

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

```

\edverbs 1076 \newcommand*\edverbs{%
1077   \let\gmv@dismath\[
1078   \let\gmv@edismath\]
1079   \def\[{%
1080     \@ifnextac\gmv@disverb\gmv@dismath}%
1081   \relaxen\edverbs}%
\gmv@disverb 1083 \def\gmv@disverb{%
1084   \gmv@dismath%
1085   \hbox\bgroup\def\]{\egroup\gmv@edismath}%
1086 }

```

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:

```

\OldMakeShortVerb 1099 \pdef\OldMakeShortVerb{\MakeShortVerb $$}%

```

Grey visible spaces

In August 2008 Will Robertson suggested grey spaces for `gmdoc`. I added a respective option to that package but I like the grey spaces so much that I want provide them for any verbatim environments, so I bring the definition here. The declaration, if put in the preamble, postpones redefinition of `\visiblespaces` till `\begin{document}` to recognise possible redefinition of it when `xltxtra` is loaded.

```

1111 \let\gmd@preambleABD\AtBeginDocument
1112 \AtBeginDocument{\let\gmd@preambleABD\firstofone}

1114 \RequirePackage{xcolor}% for \providecolor

\VisSpacesGrey 1116 \def\VisSpacesGrey{%
1118   \providecolor{visspacesgrey}{gray}{0.5}%
1119   \gmd@preambleABD{%
1120     \edef\visiblespace{%
1121       \hbox{\@nx\textcolor{visspacesgrey}%
1122         {\@xa\unexpanded\@xa{\visiblespace}}}%
1123     }%

```

Verbatim specials—CSes in verbatims

```

\verbatimsspecials 1127 \pdef\verbatimsspecials{% This declaration only defines a bearer of the ‘verbatim
specials’.
% #1 m char for verbatim escape char (for catcode 0), has to be unbraced,
% #2 m char for verbatim group begin (for catcode 1), has to be unbraced,
% #3 m char for verbatim group end (for catcode 2), has to be unbraced,
% [#4] o char for verbatim math shift (for catcode 3),
% [#5] o char for a shorthand for \metachar.
% {#6} b (optional braced) additional stuff (commands) to be executed at the
beginning of the verbatims.

1138 \@bsphack
1139 \begingroup
1140 \let\do\@makeother
1141 \dospecials
1142 \catcode\ =10
1143 \verbatim@specials@iii}

\verbatim@specials@iii 1145 \pdef\verbatim@specials@iii#1#2#3{% as you see, we take only first three argu-
ments in a despecialized group. It’s to avoid \futurelet of the optionals’
parser to touch (and thus spoil) subsequent token. Yes, we could handle the
case of a space or single line end but handling the case of a backslash would
be somewhat difficult.

1151 \endgroup
\verbatim@specials@list 1152 \def\verbatim@specials@list{#1#2#3}%
1153 \@ifnextchar[%
1154 {\begingroup\let\do\@makeother\dospecials
1155   \catcode\ =9
1156   \verbatim@specials@iv}%
1157 {\addtomacro\verbatim@specials@list{\NoValue\NoValue}%
1158   \verbatim@specials@vi}%
1159 }% of \verbatim@specials@iii.

\verbatim@specials@iv 1161 \pdef\verbatim@specials@iv[#1]{%
1162   \endgroup
1163   \addtomacro\verbatim@specials@list{#1}%
1164   \@ifnextchar[%
1165   {\begingroup\let\do\@makeother\dospecials
1166     \catcode\ =9
1167     \verbatim@specials@v}%
1168   {\addtomacro\verbatim@specials@list{\NoValue}%
1169   \verbatim@specials@vi}%

```

```

1170 }% of \verbatim@specials@iv.
\verbatim@specials@v 1172 \pdef\verbatim@specials@v[#1]{%
1173   \endgroup
1174   \addtomacro\verbatim@specials@list{#1}%
1175   \verbatim@specials@vi
1176 }% of \verbatim@specials@v.
\verbatim@specials@vi 1178 \DeclareCommand\verbatim@specials@vi\long{b}
1179 {\addtomacro\verbatim@specials@list{#{#1}}%
1180 \@esphack}
\verbatim@specials 1182 \def\verbatim@specials{% this is the macro that actually sets the chars given in
% \verbatim@specials@list as the escape char, group begin and group end.
1185   \ifdefined\verbatim@specials@list
1186   \@xa\verbatim@specials@\verbatim@specials@list
1187   \fi
1188 }% of \verbatim@specials.
\verbatim@specials@ 1190 \long\def\verbatim@specials@#1#2#3#4#5#6{%
1191   \catcode`#1=0
1192   \protected\@namedef{#1}{#1}%
1193   \catcode`#2=1
1194   \protected\@namedef{#2}{#2}%
1195   \catcode`#3=2
1196   \protected\@namedef{#3}{#3}%
1197   \edef\gmu@tempa{\the\endlinechar}%
1198   \endlinechar\m@ne % we have to suppress adding of a line end by \scantokens
% since it would turn into an active char ^^M and raise an error (which actually
% did happen).
1201   \scantokens{%
1202     #1let#1bgroup=#2%
1203     #1let#1egroup=#3%
1204     #1catcode#1backquote#1h=6#1relax%
1205     #1pdef#1<h1>#2#1meta#2h1#3#3%
1206     #1catcode#1backquote#1h=11#1relax%
1207   }%
1208   \endlinechar\gmu@tempa\relax
1209   \gmuIfValueT{#4}{%
1210     \catcode`#4=3
1211     \protected\@namedef{#4}{#4}}%
1212   \gmuIfValueT{#5}{%
1213     \begingroup
1214     \lccode`#5\lowercase{\endgroup\let~\metachar}%
1215     \protected\@namedef{#5}{#5}%
1216     \catcode`#5=\active
1217   }% of if value #5.
1218   \gmuPutIfValue{#6}%
1219 }
\verbatim@specials 1221 \pdef\verbatim@specials{\let\verbatim@specials@list\@undefined}
\GMverbatim@specials 1223 \def\GMverbatim@specials{%
1224   \gmu@ifCSdefined {\XeTeXversion}%
1226   {\verbatim@specials
1227     /% escape

```



```

1228     «»% \bgroup and \egroup
1229     [¿]% math shift
1230     [>]% meta-char
1231     {\def\|\{\metachar{\$\\vert$}}}%
1232     \makestarlow
1233     \relaxen\'% to provide a CS separator (space is not 10 in verbatims).
1234     \let\>\string
1235     }% of #6.
1236   }%
1237 }%
1238 {}%
1239 }% of \GMverbatim specials.

```

Partial \verb in arguments

Now command for partial verbatims in arguments of commands:

```

1245 \let\gmu@tempa\all@stars
1246 \@xa\addtomacro\@xa\gmu@tempa\@xa{\all@unders}
1247
1248 \foone{\catcode`#=\active}
\gmv@hashhalfing 1249 {\def\gmv@hashhalfing{%
\xiihash         1251   \def#\xiihash@ifnextchar#\gobble{}}}%
1252   \catcode`#\active}%
1253 }
1254
1255 \foone{\@makeother\^^R}{%
1256   \@xa\DeclareCommand\@xa\scanverb\@xa{%
1257     \@xa Q\@xa{\gmu@tempa}>Pm}{%
1258       % #1 Q{*_*}
1259       % #2 m the stuff to be rescanned and typeset verbatim. Note that % will
1260       % be executed during first scan so at best will disappear.
1261       Spaces are ignored (because of detokenizers that add a space after a CS) but if you
1262       declare some \verbatim specials, then you can use / where / denotes
1263       the escape char in verbatim.
1264     }%
1265   }%
1266   \begingroup
1267   \gmu@septify
1268   \endlinechar=\m@ne
1269   \@xa\IfIntersect\@xa{\all@stars}{#1}%
1270   {\def\ {\breakablevispace}}%
1271   {\let\ =\space}%
1272   \@xa\IfIntersect\@xa{\all@unders}{#1}%
1273   }{% We make spaces ignored only if there was no underscore in #1 and if #2
1274     doesn't contain \ .
1275   }%
1276   \gmu@ifxany\ {#2}%
1277   }{\addtomacro\verb@lasthook{\catcode`\ =9 } }%
1278   \addtomacro\verb@lasthook{\gmv@hashhalfing }%
1279   \@makeother\^^R%
1280   \edef\gmu@tempa{%
1281     \@nx\scantokens{%
1282       \bslash verb%
1283       ^^R\detokenize{#2}^^R% we delimit the \verb's argument with 'other' ^^R
1284       assuming this char to be used very seldom if at all.
1285     }% of \scantokens,
1286   }\gmu@tempa
1287 }%
1288 \endgroup

```

```

1289 }% of \scanverb,
1290 }% of \foone.

```

```

\verbDiscretionaryHyphen 1292 \def\verbDiscretionaryHyphen#1#2{%
\gmv@hyphenchar 1295 \def\gmv@hyphenchar{\numexpr#1\relax}%
\gmv@hyphen 1296 \def\gmv@hyphen{#2}%
1297 }
1299 \verbDiscretionaryHyphen{"A6}{|}
1301 \ifgmuXeTeX

```

(2010/06/28, v0.94:) due to Will Robertson’s remark that recatcoding long (no-ASCII) dashes works only under X_ƒTEX and LuaTEX, I embrace them in a X_ƒTEX conditional

```

\gmu@tempa 1306 \def\gmu@tempa{%
\verbLongDashes 1307 \DeclareCommand\verbLongDashes{
1308 >iT{—}% to memorise which dash we set
1309 B{1.41} % expansion of en-dash
1310 >iT{—} % as above
1311 B{2} % expansion of em-dash
1312 }%
1313 }
\gmu@tempb 1315 \def\gmu@tempb{\catcode`-\active \catcode`—\active}
1317 \foone{\catcode`-\active\catcode`—\active}
1318 {%
1319 \edef\gmu@tempa{\@xau\gmu@tempa
1320 {%
1321 \@nx\addtomacro\@nx\ttverbatim@hook{%
1322 \@xau\gmu@tempb
1323 \def\@nx-{\@nx\scalebox{##1}[1]{\string-}}%
1324 \def\@nx—{\@nx\scalebox{##2}[1]{\string—}}%
1325 }%
1326 }%
1327 }%
1328 }%
1329 \gmu@tempa
1331 \fi % of if XƒTEX.

```

Note that we have two “hyphens”: one for places where a line may be broken with a comment char and another, provided as `\hyphenchar`, for discretionary hyphens at points where correct TEX code cannot be broken, such as CS names.

```

1369 \endinput

```

End of file ‘gmverb.sty’.

□

Change History

gmverb

General:

Checksum 1040 , 83

Checksum 997 , 83

gmverb v0.79

\edverbs:

- added, 1068
- gmverb v0.80**
 - \edverbs:**
 - debugged, i.e. `\hbox` added back and redefinition of `\[`, 1068
 - \xiiclub:**
 - `\ttverbatim@hook` added, 564
- gmverb v0.81**
 - General:
 - `\afterfi` made two-argument (first undelimited, the stuff to be put after `\fi`, and the other, delimited with `\fi`, to be discarded), 1331
- gmverb v0.82**
 - General:
 - Checksum 663 , 83
- gmverb v0.83**
 - General:
 - added a hook in the active left brace definition intended for **gmdoc** automatic detection of definitions (in line 504), 1331
 - Checksum 666 , 83
- gmverb v0.84**
 - General:
 - Checksum 658 , 83
- gmverb v0.85**
 - General:
 - added restoring of `\hyphenpenalty` and `\exhyphenpenalty` and setting `\hyphenchar=-1`, 1331
 - Checksum 673 , 83
- gmverb v0.87**
 - General:
 - Checksum 661 , 83
 - visible space tidied and taken from `xltxtra` if available. **gmutils** required. The `\xii...` CSes moved to **gmutils**. The documentation driver moved into the `.sty` file, 1331
- gmverb v0.88**
 - General:
 - Checksum 682 , 83
 - \VisSpacesGrey:**
 - added, or rather moved here from **gmdoc**, 1116
- gmverb v0.89**
 - General:
 - `\dekclubs`, `\dekclubs*` and `\olddekclubs` made more consistent, shorthands for `\MakeShortVerb\|`, `\MakeShortVerb*\|` and `\OldMakeShortVerb\|` respectively., 1331
 - Checksum 686 , 83
- gmverb v0.90**
 - General:
 - Checksum 684 , 83
 - some `\(b|e)group` changed to `\(begin|end)group`, 1331
- gmverb v0.91**
 - General:
 - Checksum 686 , 83
 - put to CTAN on 2008/11/21, 83
 - \verbatimleftskip:**
 - added, 913
- gmverb v0.92**
 - General:
 - Checksum 979 because of `\verbatimspecials`, hyphenation in verbatims, low star in verbatims, kerning of backslash in shrunk fonts, 83
 - \breakbslash:**
 - renamed from `\fixbslash`, 531
 - \breaklbrace:**
 - renamed from `\fixlbrace`, 538
 - \ttverbatim:**
 - added `\makeatletter` to sound with the ‘verbatim specials’, namely to allow control sequences containing `@`, 578
- gmverb v0.93**
 - General:
 - Checksum 1035 because of a bug fix in `\scanverb`(halving the hashes), 83
 - put to CTAN on 2010/03/04, 83
 - \gmv@hashhalving:**
 - cut out as separate macro, 1249
 - \verbDiscretionaryHyphen:**
 - added to synchronise hyphen chars in **gmdoc**’s documentation, 1292
 - \xiihash:**
 - mandatory argument made long (a bug fix), 1256
- gmverb v0.94**
 - General:
 - Checksum 979 because of wrapping the UTF-8 dashes’ setting in a `XƒTeX`’s conditional, 83
 - put to CTAN on 2010/07/07, 83
 - \gmv@hyphen:**
 - due to Will Robertson’s remark that recatcoding long (no-ASCII) dashes works only under `XƒTeX` and `LuaTeX`, I embrace them in a `XƒTeX` conditional, 1301
- gmverb v0.95**
 - General:
 - put to CTAN on 2011/02/02, 83
- gmverb v0.98**
 - General:
 - put to CTAN on 2011/10/14, 83
- gmverb v0.993**
 - \ttverbatim:**
 - rigid `\tt` in `\ttverbatim` changed to redefinable `\verbatimfont` due to absurd problems with bad fontifying of **gmdoc**, 591

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.

`\!`, 514, 541
`\'`, 1056
`\,`, 1056
`\-`, 1056
`\<`, 1056
`\>`, 1056
`\@par`, 793, 883, 892, 900
`\@MakeShortVerb`, 639, 674
`\@XA`, 654
`\@beginparpenalty`, 772, 788
`\@bsphack`, 1138
`\@currenvir`, 818, 833
`\@def@breakslash`, 957, 1002
`\@ehc`, 1050
`\@emptify`, 604
`\@empty`, 694, 701, 703, 727, 1010, 1013
`\@endparenv`, 803
`\@esphack`, 1180
`\@firstofone`, 646, 663
`\@gobble`, 661, 727, 782, 844, 1003
`\@ifnextac`, 1080
`\@ifnextchar`, 1002, 1153, 1164, 1251
`\@ifpackageloaded`, 720
`\@ifundefined`, 910
`\@latex@error`, 1050
`\@makeother`, 514, 563, 575, 736, 753, 754, 813, 1140, 1154, 1165, 1255, 1279
`\@namedef`, 788, 970, 973, 1192, 1194, 1196, 1211, 1215
`\@nameedef`, 985
`\@ne`, 902
`\@nx`, 749, 754, 816, 827, 828, 831, 832, 833, 986, 988, 989, 990, 991, 1121, 1281, 1321, 1323, 1324
`\@outerparskip`, 799, 881
`\@sanitize`, 735, 736, 755
`\@shortvrbdef`, 625, 626, 676, 688
`\@shortvrbinfo`, 676, 694, 701, 703, 725
`\@sverb`, 954, 1047
`\@sverb@chbs1`, 947, 948, 954
`\@sxverbatim`, 789, 840
`\@tempskipa`, 798, 799, 800
`\@tempwafalse`, 893, 899
`\@tempwatrue`, 902
`\@topsep`, 879, 880, 883
`\@topsepadd`, 802, 873, 875, 879
`\@totalleftmargin`, 885, 888
`\@undefined`, 1221
`\@verbatim`, 772, 788, 864
`\@xa`, 488, 656, 661, 663, 675, 680, 686, 700, 706, 711, 727, 728, 733, 735, 818, 823, 824, 825, 829, 971, 978, 1122, 1186, 1246, 1256, 1258, 1270, 1273
`\@xanxcs`, 987, 989, 990
`\@xau`, 1319, 1322
`\@xverbatim`, 773, 815, 840
`\[`, 500, 1077, 1079
`\]`, 514, 541, 543, 814, 995
`\]`, 500, 1078, 1085
`\^`, 1255, 1279
`\``, 1056
`\|`, 563, 1067, 1068, 1231
`\~`, 683, 709, 1048, 1061, 1214
`\`, 1142, 1155, 1166, 1271, 1272, 1276, 1277
`\active`, 500, 502, 541, 543, 683, 692, 707, 709, 812, 814, 1048, 1059, 1216, 1248, 1252, 1315, 1317
`\add@special`, 677, 731
`\addtomacro`, 1157, 1163, 1168, 1174, 1179, 1246, 1277, 1278, 1321
`\AddtoPrivateOthers`, 678, 931
`\advspace`, 802, 883, 899
`\advance`, 799, 875, 880, 885
`\aftergroup`, 1051
`\all@stars`, 1245, 1270
`\all@unders`, 1246, 1273
`alltt`, p. 3
`\AtBeginDocument`, 1111, 1112
`\begingroup`, 682, 708, 752, 810, 1034, 1060, 1139, 1154, 1165, 1213, 1267
`\bgroup`, 942, 1085
`\breakablevissspace`, 551, 557, 1271
`\breakslash`, 520, 531, 543, 957
`\breaklbrace`, 496, 504, 538
`\bslash`, 516, 1050, 1282
`\catcode`, 500, 502, 514, 541, 543, 681, 683, 692, 705, 707, 709, 810, 811, 812, 814, 1048, 1059, 1142, 1155, 1166, 1191, 1193, 1195, 1210, 1216, 1248, 1252, 1277, 1315, 1317
`\char`, 1062
`\check@bslash`, 954, 1001
`\check@percent`, 901, 924, 1037
`\csname`, 675, 681, 687, 700, 705, 706, 712, 974, 978, 988
`\DeclareCommand`, 620, 963, 1178, 1256, 1307
`\def`, 486, 496, 502, 503, 516, 520, 531, 538, 543, 551, 557, 564, 597, 607, 625, 626, 674, 699, 721, 722, 725, 731, 747, 748, 753, 769, 793, 824, 827, 842, 843, 864, 894, 940, 954, 957, 1001, 1013, 1037, 1044, 1047, 1056, 1058, 1062, 1067, 1068, 1079, 1083, 1085, 1116, 1152, 1182, 1190, 1223, 1231, 1249, 1251, 1271, 1292, 1295, 1296, 1306, 1315, 1323, 1324

`\DefineTypeChar`, 963, 995, 998
`\dekclubs`, p. 2, 1067
`\dekclubs*`, p. 2
`\DeleteShortVerb`, p. 2, 699
`\detoken@xa`, 971
`\detokenize`, 1283
`\discretionary`, 497, 521, 551
`\do`, 574, 575, 734, 748, 749, 1056, 1140, 1154, 1165
`\do@noligs`, 574, 1058
`\dobreakblankspace`, 559
`\dobreakbslash`, 543, 576
`\dobreaklbrace`, 502, 576
`\dobreakspace`, 577, 611
`\dobreakvisiblespace`, 557, 611
`\dospecials`, 575, 733, 734, 751, 1141, 1154, 1165

`\edef`, 652, 770, 771, 817, 826, 1120, 1197, 1280, 1319
`\edverbs`, p. 2, 1076, 1081
`\egroup`, 1017, 1085
`\else`, 662, 693, 702, 749, 754, 876, 884, 941
`\emptify`, 952
`\end`, 833
`\endcsname`, 675, 681, 687, 700, 705, 706, 712, 974, 978, 989
`\endgroup`, 691, 713, 756, 836, 1039, 1062, 1151, 1162, 1173, 1214, 1288
`\endinput`, 1369
`\endlinechar`, 816, 1197, 1198, 1208, 1269
`\endverbatim`, 793
`\escapechar`, 487
`\everyeof`, 816
`\everypar`, 902
`\exhyphenpenalty`, 771, 775

`\f@size`, 971
`\fi`, 584, 664, 695, 714, 715, 749, 754, 801, 872, 878, 884, 899, 941, 982, 1187, 1331
`\firstofone`, 1112
`\font`, 586, 589, 776, 971
`\fontencoding`, 607
`\foone`, 500, 514, 541, 555, 563, 1248, 1255, 1317
`\frenchspacing`, 773, 948

`\gdef`, 488, 655, 686, 733, 735, 815, 856, 1036, 1049
`\glet`, 646, 685
`\global`, 692, 705, 706, 711, 1013
`\gm@lbracehook`, 504, 509
`\gm@verb@eol`, 945, 1022, 1044
`\gmd@preambleABD`, 1111, 1112, 1119
`\gmobeyspaces`, 773, 948
`\gmu@ifCSdefined`, 1224
`\gmu@ifstar`, 946, 1067
`\gmu@ifstrany`, 624, 628
`\gmu@ifxany`, 1276
`\gmu@measurewd`, 975
`\gmu@septify`, 1268
`\gmu@tempa`, 1197, 1208, 1245, 1246, 1258, 1280, 1287, 1306, 1319, 1329
`\gmu@tempb`, 1315, 1322
`\gmuIfValueT`, 1209, 1212
`\gmuIfValueTF`, 990
`\gmuPutIfValue`, 1218
`\gmv@dismath`, 1077, 1080, 1084
`\gmv@disverb`, 1080, 1083
`\gmv@edismath`, 1078, 1085
`\gmv@exhyphenpe`, 771, 775
`\gmv@hashhalving`, 1249, 1278
`\gmv@hyphen`, 1296
`\gmv@hyphenchar`, 589, 1295
`\gmv@hyphenpe`, 770, 774
`\gmV@MakeShortVerbAlsoInMath`, 635, 642
`\gmV@MakeShortVerbNotInMath`, 637, 649
`\gmV@mm`, 687
`\gmv@packname`, 721, 722, 726
`\gmv@storedhyphenchar`, 586
`\gmV@tempA`, 652, 658
`\GMverbatim specials`, 1223
`\gn@melet`, 978
`\gobble`, 1251

`\hangafter`, 902
`\hangindent`, 902
`\hbox`, 941, 988, 1085, 1121
`\hss`, 991
`\hyphenchar`, 586, 589, 776
`\hyphenpenalty`, 770, 774

`\if@minipage`, 884
`\if@noskipsec`, 872
`\if@tempwa`, 899
`\ifcsname`, 974
`\ifdefined`, 1185

`\ifdim`, 797
`\ifgmuXeTeX`, 1301
`\ifhmode`, 582
`\IfIntersect`, 1270, 1273
`\ifmmode`, 658, 941
`\ifnum`, 707, 749, 754
`\ifvmode`, 874, 899
`\ifx`, 675, 700
`\interlinepenalty`, 901

`\kern`, 1062

`\lastskip`, 797, 798
`\lccode`, 683, 709, 1048, 1061, 1214
`\leavevmode`, 872, 941, 986, 1062
`\leftmargini`, 915
`\leftskip`, 888
`\let`, 480, 559, 574, 575, 611, 706, 711, 840, 1010, 1013, 1022, 1044, 1052, 1077, 1078, 1111, 1112, 1140, 1154, 1165, 1214, 1221, 1236, 1245, 1272
`\long`, 843, 1178, 1190
`\lowercase`, 684, 710, 1052, 1062, 1214

`\m@ne`, 487, 776, 816, 1198, 1269
`\makeatletter`, 578
`\MakeShortVerb`, p. 2, 620, 1067, 1099
`\makestarlow`, 1233
`\meaning`, 971
`\medskipamount`, 911
`\metachar`, 1214, 1231

`\n@melet`, 807
`\Name`, 646, 655, 685
`\NeedsTeXFormat`, 49
`\newcommand`, 557, 559, 1076
`\newlength`, 910
`\newskip`, 913, 917, 921
`\nobreakbslash`, p. 2, 531
`\nobreaklbrace`, p. 2, 538
`\noexpand`, 828
`\NoValue`, 1157, 1168
`\noverbatimspecials`, p. 3, 1221
`\null`, 941
`\numexpr`, 1295

`\obeylines`, 905, 1035, 1036

<code>\obeyspaces</code> , 555, 557, 559, 1035	<code>\tt</code> , 597	1168, 1174, 1179, 1185, 1186, 1221
<code>\olddekclubs</code> , p. 2, 1068	<code>\ttverbatim</code> , 573, 906, 943	
<code>\OldMakeShortVerb</code> , p. 2, 1068, 1099	<code>\ttverbatim@hook</code> , 604, 607, 1321	<code>\verbatim@specialsv</code> , 1167, 1172
<code>\PackageInfo</code> , 726	<code>\type@bslash</code> , 522, 531, 1003	<code>\verbatim@specialsvi</code> , 1158, 1169, 1175, 1178
<code>\par</code> , 877, 894	<code>\type@lbrace</code> , 497, 538	<code>\VerbatimContents</code> , 856
<code>\parsep</code> , 870	<code>\unexpanded</code> , 829, 1122	<code>\verbatimfont</code> , 585, 597
<code>\parskip</code> , 799, 870, 880, 881, 884, 886	<code>\unless</code> , 974	<code>\verbatimhangindent</code> , 902, 917, 919
<code>\partopsep</code> , 875	<code>\unskip</code> , 877	<code>\verbatimleftskip</code> , 885, 913, 915
<code>\pdef</code> , 531, 538, 573, 642, 649, 850, 1099, 1127, 1145, 1161, 1172, 1221	<code>\verb</code> , 625, 626, 694, 701, 703, 940	<code>\verbatimparskip</code> , 886, 921, 922
<code>\penalty</code> , 901	<code>\verb*</code> , 940	<code>\VerbatimPitch</code> , 850
<code>\predisplaypenalty</code> , 772, 788	<code>\verb@balance@group</code> , 1010, 1013, 1049, 1051	<code>\verbatimspecials</code> , p. 3, 1127, 1226
<code>\protected</code> , 1192, 1194, 1196, 1211, 1215	<code>\verb@egroup</code> , 1013, 1049, 1052	<code>\verbDiscretionaryHyphen</code> , p. 3, 1292, 1299
<code>\providecolor</code> , 1118	<code>\verb@eol@error</code> , 1022	<code>\verbeolOK</code> , p. 2, 1044
<code>\providecommand</code> , 924, 931	<code>\verb@eolOK</code> , 1036, 1044	<code>\VerbHyphen</code> , p. 2, 486
<code>\ProvidesPackage</code> , 50	<code>\verb@lasthook</code> , 947, 948, 952, 1277, 1278	<code>\verbhyphen</code> , 480, 488, 497, 521
<code>\raggedright</code> , 887	<code>\verbatim</code> , 769, 782, 844	<code>\verbLongDashes</code> , 1307
<code>\relax</code> , 675, 700, 706, 774, 941, 1061, 1208, 1295	<code>verbatim</code> , 769	<code>\VerbT</code> , 607
<code>\relaxen</code> , 509, 1081, 1234	<code>verbatim*</code> , 788	<code>\VerbT1</code> , 607
<code>\rem@special</code> , 704, 732, 747	<code>\verbatim@currentenv</code> , 817, 824, 825, 829	<code>\vert</code> , 1231
<code>\RequirePackage</code> , 466, 1114	<code>\verbatim@edef</code> , 826, 834	<code>\visibleSpace</code> , 551, 1120, 1122
<code>\scalebox</code> , 1323, 1324	<code>\verbatim@end</code> , 827, 835	<code>\VisSpacesGrey</code> , p. 2, 1116
<code>\scantokens</code> , 818, 1201, 1281	<code>\verbatim@nolig@list</code> , 574, 1056	<code>visspacesgrey</code> , p. 2
<code>\scanverb</code> , p. 3, 1256	<code>\verbatim@PitchContents</code> , 831, 843	<code>\vskip</code> , 798, 800, 884
<code>\selectfont</code> , 607	<code>\verba </code>	<code>\xdef</code> , 586, 681, 751, 755
<code>semiverbatim</code> , p. 3	<code>tim@PitchContents@Delim</code> , 832, 844	<code>\XeTeXversion</code> , 1224
<code>\setspace</code> , 583	<code>\verba </code>	<code>\xiiclub</code> , 564
<code>\space</code> , 559, 971, 1272	<code>tim@PitchContents@Left</code> , 842, 848, 855	<code>\xiihash</code> , 1251
<code>\stanzaskip</code> , 899, 910, 911	<code>\verbatim@specials</code> , 907, 944, 1182	<code>\xiilbrace</code> , 998
<code>\string</code> , 488, 646, 655, 675, 681, 685, 687, 700, 705, 706, 712, 727, 728, 1236, 1323, 1324	<code>\verbatim@specials@</code> , 1186, 1190	<code>\xiipercents</code> , 480
<code>\strip@bslash</code> , 652, 687	<code>\verbatim@specials@iii</code> , 1143, 1145	<code>\yeshy</code> , 498, 522, 531, 538, 1003
<code>\subs</code> , p. 3	<code>\verbatim@specials@iv</code> , 1156, 1161	<code>\z@</code> , 797, 1062
<code>\sup</code> , p. 3	<code>\verbatim@specials@list</code> , 1152, 1157, 1163,	<code>\z@skip</code> , 922
<code>\textcolor</code> , 1121		<code>\'</code> , 1234
<code>\the</code> , 586, 681, 770, 771, 971, 1197		<code>\></code> , 1236
<code>\topsep</code> , 873		