

Natror (Grzegorz Murzynowski)

The gmverb Package*

August 30, 2008

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, 2008 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.lppl.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.

```
71 \NeedsTeXFormat{LaTeX2e}
72 \ProvidesPackage{gmverb}
73 [2008/08/30_vo.89_After_shortvrb_(FM)_but_my_way_(GM)]
```

Contents

Intro, usage	1	Almost-Knuthian <code>\ttverbatim</code>	5
Installation	3	The core: from <code>shortvrb</code>	5
Contents of the gmverb.zip archive	3	doc- and shortvrb-compatibility	10
Compiling of the documentation	3	Grey visible spaces	10
The code	3	Change History	11
Preliminaries	3	Index	12
The breakables	4		

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 vo.89 dated 2008/08/30.

	<code><subsequent text>}</code> and <code><text>\mymacro</code> breaks into <code><text>%\mymacro</code> .
<code>\fixbslash</code>	(If you don't like linebreaking at backslash, there's the <code>\fixbslash</code> declaration (observing the common scoping rules, hence <code>ocsr</code>) and an analogous declaration for the left brace: <code>\fixlbrace</code> .)
<code>\VerbHyphen</code>	The default 'hyphen' is % since it's the default comment char. If you wish another char to appear at the linebreak, use the <code>\VerbHyphen</code> declaration that takes <code>\<char></code> as the only argument. This declaration is always global.
<code>\verbeolOK</code>	Another difference is the <code>\verbeolOK</code> declaration (<code>ocsr</code>). Within its scope, <code>\verb</code> allows an end of a line in its argument and typesets it just as a space.
	As in the standard version(s), the plain <code>\verb</code> typesets the spaces blank and <code>\verb*</code> makes them visible.
<code>\MakeShortVerb</code>	Moreover, <code>gmverb</code> provides the <code>\MakeShortVerb</code> 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 <code>\MakeShortVerb* </code> (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 <code> %\mymacro </code> to get <code>\mymacro</code> instead of typing <code>\verb+\mymacro+</code> . Because the char used in this example is my favourite and used just this way by DEK in the <i>The T_EXbook</i> 's format, <code>gmverb</code> provides a macro <code>\dekclubs</code> as a shorthand for <code>\MakeShortVerb(*)%\ </code> .
<code>\DeleteShortVerb</code>	Be careful because such active chars may interfere with other things, e.g., the <code> </code> with the vertical marker in tables and with the <code>tikz</code> package. If this happens, you can declare e.g., <code>\DeleteShortVerb\ </code> and the previous meaning of the char used shall be restored.
	One more difference between <code>gmverb</code> and <code>shortverb</code> is that the chars <code>\active</code> ated by <code>\MakeShortVerb</code> in the math mode behave as if they were 'other', so you may type e.g., <code>\$\$ </code> to get <code> </code> and <code>+\active</code> ated this way is in the math mode typeset properly etc.
<code>\OldMakeShortVerb</code>	However, if you don't like such a conditional behaviour, you may use <code>\OldMakeShortVerb</code> instead, what I do when I like to display short verbatims in <code>display-math</code> .
<code>\dekclubs</code>	There's one more declaration provided by <code>gmverb</code> : <code>\dekclubs</code> , which is a shorthand for <code>\MakeShortVerb\ </code> , <code>\dekclubs*</code> for <code>\MakeShortVerb* </code> and <code>\olddekclubs</code> for <code>\OldMakeShortVerb\ </code> .
<code>\edverbs</code>	There's one more declaration, <code>\edverbs</code> that makes <code>\[</code> checks if the next token is an active char and opens an <code>\hbox</code> if so. That is done so that you can write (in <code>\edverbs</code> ' and <code>\dekclubs</code> ' scope)
	<code>\[<verbatim stuff>]</code>
	instead of
	<code>\[\hbox{<verbatim stuff> }]</code>
	to get a displayed shortverb.
	Both versions of <code>\dekclubs</code> <code>ocsr</code> .
	The verbatim environment inserts <code>\topsep</code> before and after itself, just as in standard version (as if it was a list).
<code>\VisSpacesGrey</code>	In August 2008 Will Robertson suggested grey visible spaces for <code>gmdoc</code> . I added a respective option to <code>gmdoc</code> but I find them so nice that I want to make them available for all verbatim environments so I bring here the declaration <code>\VisSpacesGrey</code> . It redefines only the visible spaces so affects <code>\verb*</code> and <code>verbatim*</code> and not the unstarred versions. The colour of the visible spaces is named <code>visspacesgrey</code> and you can redefine it <code>xcolor</code> way.
	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 three files and a tds-compliant archive.

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

This package requires another package of mine, gmutils, also available on CTAN.

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 X_YL^AT_EX on the gmverb.sty file (xelatex gmverb.sty in the directory you wish the documentation to be in, you don't have copy the .sty file there, T_EX will find it) twice, then MakeIndex on the gmverb.idx file, and then L^AT_EX on gmverb.tex once more.

MakeIndex shell command:

```
makeindex -r gmverb
```

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 gmdocc.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

```
249 \RequirePackage{gmutils}[2008/08/06]
```

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

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 261 \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 267 \def\VerbHyphen#1{%
268   {\escapechar\m@ne
269    \@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 278 \def\breaklbrace{%
279   \discretionary{\xiilbrace\verbhyphen}{\xiilbrace}%
282   \foone{\catcode`\[=1\catcode`\{=\active\catcode`\]=2}%
283   [%
\dobreaklbrace 284   \def\dobreaklbrace[\catcode`\{=\active
285   \def{%
\breaklbrace 286   [\breaklbrace\gm@lbracehook]]%
287   ]

```

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

```
291 \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.

```

296 \foone{\catcode`\!=0\@makeother\}%
297 {%
\bslash 298   !def!bslash{\}%
\breakbslash 299   !def!breakbslash{!discretionary{!verbhyphen}{\}\{\}}%
300   }

```

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 614. For the others let's give the user a countercrank:

```

\fixbslash 307 \newcommand*\fixbslash{\let\breakbslash=\bslash}% to use due to the com-
               mon scoping rules. But for the special case of a backslash opening a verbatim
               scope, we deal specially in the line 614.

```

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

```

\fixlbrace 313 \newcommand*\fixlbrace{\let\breaklbrace=\xiilbrace}
316 \foone{\catcode`\!=0\catcode`\[=\active}%
318 {%
\dobreakbslash 319   !def!dobreakbslash{!catcode`\!=!active\def\{!breakbslash}}%
\breakbslash 320   }

```

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

```

326 \foone{\catcode`\_ =12}% note this space is 10 and is gobbled by parsing the
               number. \visiblespace is \let in gutils to \xiispace or \xxt@visible space
               of xltextra if available.
\breakablevispace 330 \def\breakablevispace{\discretionary{\visiblespace}{\}%
               \visiblespace}
333 \foone\obeyspaces% it's just re\catcode'ing.
334 {%

```

```

\activespace 335 \newcommand*\activespace{\_}%
\dobreakvisiblespace 336 \newcommand*\dobreakvisiblespace{\def\_{\breakablevisisspace}\obeyspaces}%%
\breakablevisisspace \defing it caused a stack overflow disaster with gmdoc.
\dobreakblankspace 338 \newcommand*\dobreakblankspace{\let\_=\space\obeyspaces}%
339 }
342 \bgroup\@makeoother\|
\Xiicclub 343 \firstofone{\egroup\def\Xiicclub{||}}

```

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 352 \newcommand*\ttverbatim{%
353   \let\do=\do@noligs\_verbatim@nolig@list
354   \let\do=\@makeoother\_dospecials
355   \dobreaklbrace\dobreakbslash
356   \dobreakspace
357   \tt
358   \ttverbatim@hook}
365 \@emptyify\ttverbatim@hook
\VerbT1 368 \def\VerbT1{\def\ttverbatim@hook{\fontencoding{T1}\selectfont}}
\VerbT
\ttverbatim@hook We wish the visible spaces to be the default.
372 \let\dobreakspace=\dobreakvisiblespace

```

The core: from shortvrb

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

```

\MakeShortVerb 381 \def\MakeShortVerb{%
382   \@ifstar
\@shortvrbdef 383   {\def\@shortvrbdef{\verb*}\@MakeShortVerb}%
\@shortvrbdef 384   {\def\@shortvrbdef{\verb}\@MakeShortVerb}}
\@MakeShortVerb 387 \def\@MakeShortVerb#1{%
388   \@xa@ifx\csname\_cc\string#1\endcsname\relax
389   \@shortvrbinfo{Made\_}{#1}\@shortvrbdef
390   \add@special{#1}%
391   \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
393   \@xa
394   \xdef\csname\_cc\string#1\endcsname{\the\catcode`#1}%
395   \begingroup
396   \catcode`\~\active\_lccode`\~`#1%
397   \lowercase{%
398     \global\@xa\let
399     \csname\_ac\string#1\endcsname~%
400     \@xa\gdef\@xa~\@xa{%
401       \@xa@ifmmode\@xa\string\@xa~%

```

```

402      \xa\else\xa\afterfi{\@shortvrbbdef~}\fi}}% This terrible number
      of \expandafters is to make the shortverb char just other in the math
      mode (my addition).
405      \endgroup
406      \global\catcode`#1\active
407      \else
408      \@shortvrbbinfo\@empty{#1\already}{\@empty\verb(*)}}%
409      \fi}

\DeleteShortVerb 412 \def\DeleteShortVerb#1{%
413      \xa\ifx\csname_cc\string#1\endcsname\relax
414      \@shortvrbbinfo\@empty{#1\not}{\@empty\verb(*)}}%
415      \else
416      \@shortvrbbinfo{Deleted_}{#1_as}{\@empty\verb(*)}}%
417      \rem@special{#1}%
418      \global\catcode`#1\csname_cc\string#1\endcsname
419      \global\@xa\let_\csname_cc\string#1\endcsname_\relax
420      \ifnum\catcode`#1=\active
421      \begingroup
422      \catcode`\~\active_\lccode`\~`#1%
423      \lowercase{%
424          \global\@xa\let\@xa~%
425          \csname_ac\string#1\endcsname}%
426      \endgroup_\fi_\fi}

My little addition

430 \@ifpackageloaded{gmdoc}{%
\gmvp@packname 431 \def\gmvp@packname{gmdoc}}{%
\gmvp@packname 432 \def\gmvp@packname{gmverb}}

\@shortvrbbinfo 435 \def\@shortvrbbinfo#1#2#3{%
436      \PackageInfo{\gmvp@packname}{%
437          ^^J\@empty_#1\@xa\@gobble\string#2_a_short_reference
438          for_\@xa\string#3}}

\add@special 441 \def\add@special#1{%
442      \rem@special{#1}%
443      \xa\gdef\xa\dospecials\xa
444      {\dospecials_\do_#1}%
445      \xa\gdef\xa\@sanitize\xa
446      {\@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 457 \def\rem@special#1{%
458      \def\do##1{%
459          \ifnum`#1=`##1_\else_\@nx\do\@nx##1\fi}%
460      \xdef\dospecials{\dospecials}%
461      \begingroup
462      \def\@makeother##1{%
463          \ifnum`#1=`##1_\else_\@nx\@makeother\@nx##1\fi}%

```

```

464 \xdef\@sanitize{\@sanitize}%
465 \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 478 \def\verbatim{%
\verbatim 479 \edef\gmv@hyphenpe{\the\hyphenpenalty}%
480 \edef\gmv@exhyphenpe{\the\exhyphenpenalty}%
481 \@beginparpenalty_\predisplaypenalty_\@verbatim
482 \frenchspacing_\gmoveyspaces_\@xverbatim
483 \hyphenpenalty=\gmv@hyphenpe\relax
484 \exhyphenpenalty=\gmv@exhyphenpe
485 \hyphenchar\font=\m@ne}% in the LATEX version there's \@vobeyspaces in-
instead of \gmoveyspaces.
verbatim* 490 \@namedef{verbatim*}{\@beginparpenalty_\predisplaypenalty_\%
\@verbatim
491 \@sxverbatim}
\endverbatim 493 \def\endverbatim{\@@par
494 \ifdim\lastskip_\>\z@
495 \@tempskipa\lastskip_\vskip_\-\lastskip
496 \advance\@tempskipa\parskip_\advance\@tempskipa_\-%
\@outerparskip
497 \vskip\@tempskipa
498 \fi
499 \addvspace\@topsepadd
500 \@endparenv}
503 \n@melet{endverbatim*}{endverbatim}
506 \begingroup_\catcode_\!=o_\%
507 \catcode_\`=[_1_\catcode_\`=2_\%
508 \catcode_\{=\active
509 \@makeother\}%
510 \catcode_\=\active%
\@xverbatim 511 !gdef!\@xverbatim[%
512 !edef!verbatim@edef [%
513 !def!noexpand!verbatim@end%
514 #####1!noexpand\end!noexpand{!\@currenvir}[%
515 #####1!noexpand!end[!\@currenvir]]]%
516 !verbatim@edef
517 !verbatim@end]%
518 !endgroup
\@sxverbatim 522 \let\@sxverbatim=\@xverbatim

```

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

```

\@verbatim 527 \def\@verbatim{%

```

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

```

533 \parsep\parskip
From \@trivlist:
535 \if@noskipsec\leavevmode\fi
536 \@topsepadd\topsep
537 \ifvmode
538 \advance\@topsepadd\partopsep
539 \else
540 \unskip\par
541 \fi
542 \@topsep\@topsepadd
543 \advance\@topsep\parskip
544 \@outerparskip\parskip
(End of \trivlistlist and \@trivlist highlights.)
546 @@par\addvspace\@topsep
547 \if@minipage\else\vskip\parskip\fi
548 \leftmargin\parindent% please notify me if it's a bad idea.
549 \advance\@totalleftmargin\leftmargin
550 \raggedright
551 \leftskip\@totalleftmargin% so many assignments to preserve the list
    thinking for possible future changes. However, we may be sure no inter-
    nal list shall use \@totalleftmargin as far as no inner environments are
    possible in verbatim(*).
557 @@par% most probably redundant.
558 @tempwafalse
559 \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@tempwa\hbox{}\fi, in my version will
    be
564 \ifvmode\if@tempwa\addvspace\stanzaskip\@tempwafalse\fi\fi
565 @@par
566 \penalty\interlinepenalty\check@percent}%
567 \everypar{\@tempwatrue\hangindent\verbatimhangindent\hangafter%
    \@ne}% since several chars are breakable, there's a possibility of breaking
    some lines. We wish them to be hanging indented.
570 \obeylines
571 \ttverbatim}
\stanzaskip 573 \@ifundefined{stanzaskip}{\newlength\stanzaskip}{}
574 \stanzaskip=\medskipamount
\verbatimhangindent 578 \newlength\verbatimhangindent
579 \verbatimhangindent=3em
\check@percent 581 \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 588 \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* 597 \def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
\verb 598 \bgroup
599 \ttverbatim
600 \gm@verb@eol
601 \@ifstar{\@sverb@chbsl}{\gmoveyspaces\frenchspacing\@sverb@chbsl}}%in
the LATEX version there's \@vobeyspaces instead of \gmoveyspaces.
\@sverb@chbsl 605 \def\@sverb@chbsl#1{\@sverb#1\check@bslash}
\@def@breakbslash 608 \def\@def@breakbslash{\breakbslash}% because \ is \defined as \breakb-
slash 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 614 \def\check@bslash{\@ifnextchar{\@def@breakbslash}{\bslash%
\@gobble}}{}
618 \let\verb@balance@group\@empty
\verb@egroup 621 \def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}
\gm@verb@eol 625 \let\gm@verb@eol\verb@eol@error

```

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

```

637 \begingroup
638 \obeylines\obeyspaces%
639 \gdef\verb@eolOK{\obeylines%
\check@percent 640 \def~M{\check@percent}%
641 }%
642 \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 647 \def\verbeolOK{\let\gm@verb@eol\verb@eolOK}

```

And back to the main matter,

```

650 \def\@sverb#1{%
651 \catcode`#1\active\lccode~\~`#1%
652 \gdef\verb@balance@group{\verb@egroup
653 \@latex@error{Illegal use of \bslash\verb\command}\@ehc}%
654 \aftergroup\verb@balance@group
655 \lowercase{\let~\verb@egroup}}
\verbatim@nolig@list 657 \def\verbatim@nolig@list{\do~\do<\do>\do\,\do'\do-}
\do@noligs 659 \def\do@noligs#1{%
660 \catcode`#1\active
661 \begingroup
662 \lccode~\~`#1\relax
663 \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 668 \def\dekclubs{\@ifstar{\MakeShortVerb*}{\MakeShortVerb\|}}
\olddekclubs 669 \def\olddekclubs{\OldMakeShortVerb\|}
```

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

```
\edverbs 677 \newcommand*\edverbs{%
678   \let\gmv@dismath\[%
679   \let\gmv@edismath\]%
680   \def\[%
681     \@ifnextac\gmv@disverb\gmv@dismath}%
682   \relaxen\edverbs}%
\gmv@disverb 684 \def\gmv@disverb{%
685   \gmv@dismath
687   \hbox\bgroup\def\|\egroup\gmv@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 699 \def\old@MakeShortVerb#1{%
700   \@xa@ifx\csname_cc\string#1\endcsname\relax
701   \@shortvrbinfo{Made_}{#1}\@shortvrbdef
702   \add@special{#1}%
703   \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
704   \@xa
705   \xdef\csname_cc\string#1\endcsname{\the\catcode`#1}%
706   \begingroup
707   \catcode`\~\active\lccode`\~`#1%
708   \lowercase{%
709     \global\@xa\let\csname_ac\string#1\endcsname~%
710     \@xa\gdef\@xa~\@xa{%
711       \@shortvrbdef~}}%
712   \endgroup
713   \global\catcode`#1\active
714   \else
715   \@shortvrbinfo\@empty{#1_already}{\@empty\verb(*)}%
716   \fi}
\OldMakeShortVerb 720 \def\OldMakeShortVerb{\begingroup
721   \let\@MakeShortVerb=\old@MakeShortVerb
722   \@ifstar{\eg@MakeShortVerbStar}{\eg@MakeShortVerb}}
\eg@MakeShortVerbStar 725 \def\eg@MakeShortVerbStar#1{\MakeShortVerb*#1\endgroup}
\eg@MakeShortVerb 726 \def\eg@MakeShortVerb#1{\MakeShortVerb#1\endgroup}
```

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 `\visible` till `\begin{document}` to recognize possible redefinition of it when `xltxtra` is loaded.

```

738 \let\gmd@preambleABD\AtBeginDocument
739 \AtBeginDocument{\let\gmd@preambleABD\firstofone}
741 \RequirePackage{xcolor}% for \providecolor
\VisSpacesGrey 743 \def\VisSpacesGrey{%
745   \providecolor{visspacesgrey}{gray}{0.5}%
746   \gmd@preambleABD{%
747     \edef\visible{%
748       \hbox{\@nx\textcolor{visspacesgrey}%
749         {\@xa\unexpanded\@xa{\visible}}}%
750     }%
756 \endinput% for the Tradition.

```

Change History

vo.79	General:	
\edverbs:	added restoring of <code>\hyphenpenalty</code>	
added, 669	and <code>\exhyphenpenalty</code> and setting	
vo.80	<code>\hyphenchar=-1</code> , 756	
\edverbs:	Checksum 673, 0	
debugged, i.e. <code>\hbox</code> added back and		vo.87
redefinition of <code>\[</code> , 669	General:	
\ttverbatim:	Checksum 661, 0	
\ttverbatim@hook added, 343	visible space tidied and taken from	
vo.81	<code>xltxtra</code> if available. <code>gmutils</code> required.	
General:	The <code>\xii . . . cses</code> moved to <code>gmutils</code> .	
\afterfi made two-argument (first	The documentation driver moved	
undelimited, the stuff to be put after	into the <code>.sty</code> file, 756	
\fi, and the other, delimited with		vo.88
\fi, to be discarded, 756	General:	
vo.82	Checksum 682, 0	
General:	\VisSpacesGrey:	
Checksum 663, 0	added, or rather moved here from	
vo.83	<code>gmdoc</code> , 743	
General:		vo.89
added a hook in the active left brace	General:	
definition intended for <code>gmdoc</code>	\dekclubs, \dekclubs* and	
automatic detection of definitions (in	\olddekclubs made more	
line 286), 756	consistent, shorthands for	
Checksum 666, 0	\MakeShortVerb\ ,	
vo.84	\MakeShortVerb* and	
General:	\OldMakeShortVerb\	
Checksum 658, 0	respectively., 756	
vo.85	Checksum 686, 0	

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.

- \-, 657
- \@MakeShortVerb, 383, 384, 387, 721
- \@currenvir, 514, 515
- \@def@breakbslash, 608, 614
- \@emptify, 365
- \@ifnextac, 681
- \@shortvrbrdef, 383, 384, 389, 402, 701, 712
- \@shortvrbrinfo, 389, 408, 414, 416, 435, 701, 716
- \@sverb@chbsl, 601, 605
- \@topsep, 542, 543, 546
- \@topsepadd, 499, 536, 538, 542
- \activespace, 335
- \add@special, 390, 441, 702
- \AddtoPrivateOthers, 391, 588, 703
- \afterfi, 402
- \AtBeginDocument, 738, 739
- \breakablevisspace, 330, 336
- \breakbslash, 299, 307, 319, 608
- \breaklbrace, 278, 286, 313
- \bslash, 298, 307, 614, 653
- \check@bslash, 605, 614
- \check@percent, 566, 581, 640
- \dekclubs, *p. 2*, 668
- \dekclubs*, *p. 2*
- \DeleteShortVerb, *p. 2*, 412
- \do@noligs, 353, 659
- \dobreakblankspace, 338
- \dobreakbslash, 319, 355
- \dobreaklbrace, 284, 355
- \dobreakspace, 356, 372
- \dobreakvisiblespace, 336, 372
- \edverbs, *p. 2*, 677, 682
- \eg@MakeShortVerb, 722, 726
- \eg@MakeShortVerbStar, 722, 725
- \endverbatim, 493
- \everypar, 567
- \exhyphenpenalty, 480, 484
- \fixbslash, *p. 2*, 307
- \fixlbrace, *p. 2*, 313
- \fontencoding, 368
- \foone, 282, 296, 316, 326, 333
- \gm@lbracehook, 286, 291
- \gm@verb@eol, 600, 625, 647
- \gmd@preambleABD, 738, 739, 746
- \gmobeyspaces, 482, 601
- \gmvd@ismath, 678, 681, 685
- \gmvd@isverb, 681, 684
- \gmvd@edismath, 679, 687
- \gmvd@exhyphenpe, 480, 484
- \gmvd@hyphenpe, 479, 483
- \gmvd@packname, 431, 432, 436
- \hyphenpenalty, 479, 483
- \if@noskipsec, 535
- \interlinepenalty, 566
- \leftmargin, 548, 549
- \MakeShortVerb, *p. 2*, 381, 668, 726
- \MakeShortVerb*, 668, 725
- \n@melet, 503
- \newlength, 573, 578
- \obeyspaces, 333, 336, 338, 638
- \old@MakeShortVerb, 699, 721
- \olddekclubs, *p. 2*, 669
- \OldMakeShortVerb, *p. 2*, 669, 720
- \PackageInfo, 436
- \parsep, 533
- \partopsep, 538
- \predisplaypenalty, 481, 490
- \providecolor, 745
- \relaxen, 291, 682
- \rem@special, 417, 442, 457
- \RequirePackage, 249, 741
- \stanzaskip, 564, 573, 574
- \textcolor, 748
- \ttverbatim, 352, 571, 599
- \ttverbatim@hook, 358, 365, 368
- \unexpanded, 749
- \verb, 384, 408, 414, 416, 597, 716
- \verb*, 383, 597
- \verb@balance@group, 618, 621, 652, 654
- \verb@egroup, 621, 652, 655
- \verb@eol@error, 625
- \verb@eolOK, 639, 647
- \verbatim, 478
- verbatim, 478
- verbatim*, 490
- \verbatim@edef, 512, 516
- \verbatim@end, 513, 517
- \verbatim@nolig@list, 353, 657
- \verbatimhangindent, 567, 578, 579
- \verbeolOK, *p. 2*, 647
- \VerbHyphen, *p. 2*, 267
- \verbhyphen, 261, 269, 279, 299
- \VerbT, 368
- \VerbT1, 368
- \visiblespace, 330, 747, 749
- \VisSpacesGrey, *p. 2*, 743
- \xiiclub, 343
- \xiilbrace, 279, 313
- \xiipercenat, 261