

Natror (Grzegorz Murzynowski)

The gmverb Package*

March 4, 2010

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, 2009, 2010 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.lpl.html](http://www.ctan.org/tex--archive/help/Catalogue/licenses.lpl.html) for the details of that license.

LPPL status: "author-maintained".

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

```
79 \NeedsTeXFormat {LaTeX2e}
```

```
80 \ProvidesPackage {gmverb}
```

```
81 [2010/03/04 v0.93 After shortvrb (FM) but my way (GM)]
```

Contents

Intro, usage	1	Almost-Knuthian \ttverbatim	6
The package options	4	The core: from <code>shortvrb</code>	6
Installation	4	doc- and <code>shortvrb</code> -compatibility	13
Contents of the <code>gmverb.zip</code> archive	4	Grey visible spaces	13
Compiling of the documentation	4	Verbatim specials—C _S es in <code>verbatim</code> s	14
The code	4	Partial <code>\verb</code> in arguments	16
Preliminaries	4	Change History	17
The breakables	5	Index	18

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 v0.93 dated 2010/03/04.

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`.

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

`\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_EXbook’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 `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., `$2|o$` to get `2|o` and `+ \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 `\Old|MakeShortVerb` instead, what I do when I like to display short verbatims in `display-math`.

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

In August 2008 Will Robertson suggested grey 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. The colour of the visible spaces is named `visspacesgrey` and you can redefine

`\VisSpacesGrey`

`visspacesgrey`

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.
{#6} b optional in curly braces, additional stuff (commands) to be executed in a 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 )
```

To get a ‘verbatim special’ verbatim, precede it with the ‘verbatim escape’: `/ «» [¿] []` in the example above.

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

`\sub`s

`\(\)`

`alltt`

`semiverbatim`

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

The `\verbatimspecials` declaration goes a step further than \TeX ’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`

```
\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?

The package options

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 \TeX 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 $\Xe_{\text{L}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 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, \TeX will find it) twice, then `MakeIndex` on the `gmverb.idx` file, and then $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 `gmdoc.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 \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

```
331 \RequirePackage{gmutils}[2008/10/08]
```

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 '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` 345 `\let \verbhyphen \xiipercen`
Provide a declaration for easy changing it. Its argument should be of `\langle char \rangle` form (a `\langle char \rangle_{12}` is also allowed).

```
\VerbHyphen 351 \def \VerbHyphen#1 {%
352   {\escapechar \m@ne
353    \@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 362 \def \breaklbrace {%
363   \discretionary {\type@lbrace \verbhyphen} {} {\type@lbrace}%
364   \yeshy}

367 \foone {\catcode `\[ =1 \catcode ` \ { =\active \catcode ` \ ] =2 }%
368 [%

\dobreaklbrace 369   \def \dobreaklbrace [ \catcode ` \ { =\active
370   \def {%
\breaklbrace 371     [ \breaklbrace \gm@lbracehook ] ]%
372 ]
```

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

```
376 \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.

```
381 \foone {\catcode ` \ ! =0 \catcode ` \ @makeother \ }%
382 {%

\bslash 383   ! def !bslash { \ }%
384   }% of \foone.

\breakbslash 387 \def \breakbslash {%
388   \discretionary {\verbhyphen}%
389   {\type@bslash} {\type@bslash} \yeshy% it seems that we allow hyphen-
        ation after backslash but hyphenation will be allowed iff \hyphenchar
        % \font is nonnegative.
392 }% 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 801. For the others let's give the user a counter-crank:

```
\nbreakbslash 398 \pdef \nbreakbslash { \def \breakbslash {\type@bslash \yeshy} }% to use
\breakbslash   due to the common scoping rules. But for the special case of a backslash open-
              ing a verbatim scope, we deal specially in the line 801.
```

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

```
\nbreaklbrace 405 \pdef \nbreaklbrace { \def \breaklbrace {\type@lbrace \yeshy} }
\breaklbrace 410 \foone {\catcode ` \ ! =0 \catcode ` \ \ =\active}%
412 {%

\dobreakbslash 413   ! def !dobreakbslash { ! catcode ` \ \ = ! active \catcode ` \ ! def { \%
```

```
\breakbslash      !breakbslash}}%
414 }
```

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

```
\breakablevispace 423 \def\breakablevispace{\discretionary{\visiblepace}{}{\%
\visiblepace}}
```

The `\visiblepace` macro is `\let` in `gmutils` to `\xiispace` or to `\xxt@visiblepace` of `xltxtra` if available.

```
427 \foone\obeyspaces% it's just re\catcode'ing.
428 {%
```

```
\dobreakvisiblespace 429 \newcommand*\dobreakvisiblespace{\def\breakablevispace{\%
\breakablevispace\obeyspaces}% \defing it caused a stack overflow disaster with
gmdoc.
```

```
\dobreakblankspace 431 \newcommand*\dobreakblankspace{\let\space\obeyspaces}%
432 }
```

```
435 \foone{\@makeother\|}{%
\xiiclub 436 \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 445 \pdef\ttverbatim{%
446 \let\do=\do@noligs\verbatim@nolig@list
447 \let\do=\@makeother\dospecials
448 \dobreaklbrace\dobreakbslash
449 \dobreakspace
450 \makeatletter
454 \ifhmode
455 \setspaceskip
456 \fi
457 \tt
458 \edef\gmv@storedhyphenchar{\the\hyphenchar\font}%
459 \hyphenchar\font=\gmv@hyphenchar
460 \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.

```
468 \@emptify\ttverbatim@hook
```

```
\VerbT1 471 \def\VerbT1{\def\ttverbatim@hook{\fontencoding{T1}%
\VerbT \selectfont}}
```

We wish the visible spaces to be the default.

```
475 \let\dobreakspace=\dobreakvisiblespace
```

The core: from `shortvrb`

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

```

\MakeShortVerb 484 \def\MakeShortVerb{%
485   \gm@ifstar
\@shortvrbrdef 486   {\def\@shortvrbrdef{\verb*}\@MakeShortVerb}%
\@shortvrbrdef 487   {\def\@shortvrbrdef{\verb}\@MakeShortVerb}}

\@MakeShortVerb 490 \def\@MakeShortVerb#1{%
491   \@xa@ifx\csname_#1\endcsname\relax
492   \@shortvrbrinfo{Made_#1}\@shortvrbrdef
493   \add@special{#1}%
494   \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
495   \@xa
496   \xdef\csname_#1\endcsname{\the\catcode`#1}%
497   \begingroup
498   \catcode`\~\active_\lccode`\~=#1%
499   \lowercase{%
500     \global\@xa\let
501     \csname_#1\endcsname~%
502     \@xa\gdef\@xa~\@xa{%
503       \@xa\ifmmode\@xa\string\@xa~%
504       \@xa\else\@xa\afterfi{\@shortvrbrdef~}\fi}}% This terrible num-
505       ber of \expandafteers is to make the shortverb char just other in the
506       math mode (my addition).
507   \endgroup
508   \global\catcode`#1\active
509   \else
510   \@shortvrbrinfo\@empty{#1_already}{\@empty\verb(*)}%
511   \fi}

\DeleteShortVerb 515 \def\DeleteShortVerb#1{%
516   \@xa@ifx\csname_#1\endcsname\relax
517   \@shortvrbrinfo\@empty{#1_not}{\@empty\verb(*)}%
518   \else
519   \@shortvrbrinfo{Deleted_#1_as}{\@empty\verb(*)}%
520   \rem@special{#1}%
521   \global\catcode`#1\csname_#1\endcsname
522   \global_\@xa\let_\csname_#1\endcsname_\relax
523   \ifnum\catcode`#1=\active
524   \begingroup
525   \catcode`\~\active_\lccode`\~=#1%
526   \lowercase{%
527     \global\@xa\let\@xa~%
528     \csname_#1\endcsname}%
529   \endgroup_\fi_\fi}

My little addition

533 \@ifpackageloaded{gmdoc}{%
\gmv@packname 534   \def\gmv@packname{gmdoc}}{%
\gmv@packname 535   \def\gmv@packname{gmverb}}

\@shortvrbrinfo 538 \def\@shortvrbrinfo#1#2#3{%
539   \PackageInfo{\gmv@packname}{%
540     ^^J\@empty_#1\@xa\@gobble\string#2_a_short_reference
541     for_\@xa\string#3}}

\add@special 544 \def\add@special#1{%

```

```

545 \rem@special{#1}%
546 \@xa\gdef\@xa\dospecials\@xa
547 {\dospecials_\do_#1}%
548 \@xa\gdef\@xa\@sanitize\@xa
549 {\@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 560 \def\rem@special#1{%
561   \def\do##1{%
562     \ifnum`#1=`##1_\else_\@nx\do\@nx##1\fi}%
563   \xdef\dospecials{\dospecials}%
564   \begingroup
565   \def\@makeother##1{%
566     \ifnum`#1=`##1_\else_\@nx\@makeother\@nx##1\fi}%
567   \xdef\@sanitize{\@sanitize}%
568   \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 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 581 \def\verbatim{%
\verbatim 582   \edef\gmv@hyphenpe{\the\hyphenpenalty}%
583   \edef\gmv@exhyphenpe{\the\exhyphenpenalty}%
584   \@beginparpenalty_\predisplayspenalty_\@verbatim
585   \frenchspacing_\gmvobeyspaces_\@xverbatim
586   \hyphenpenalty=\gmv@hyphenpe\relax
587   \exhyphenpenalty=\gmv@exhyphenpe
588   \hyphenchar\font=\m@ne
589 }% in the LATEX version there's \vobeyspaces instead of \gmvobeyspaces.
verbatim* 594 \@namedef{verbatim*}{\@beginparpenalty_\predisplayspenalty_\@
\verbatim
595   \@sxverbatim_\% it's the same as \@xverbatim and defines the verbatim end
(a macro delimited with \end{<curr.envir.>}).
597 }
\endverbatim 599 \def\endverbatim{\@@par
600   \hyphenchar\font=\gmv@storedhyphenchar_\% hyphenchar assignments
are always global.
602   \ifdim\lastskip_\>\z@
603     \@tempskipa\lastskip_\vskip_\- \lastskip
604     \advance\@tempskipa\parskip_\advance\@tempskipa_\-%
\@outerparskip
605     \vskip\@tempskipa

```



```

606 \fi
607 \addvspace \@topsepadd
608 \@endparenv}

611 \n@melet {endverbatim*} {endverbatim}

614 \begingroup \catcode \! = 0 %
615 \catcode \[ = 1 \catcode \] = 2 %
616 \catcode \{ = \active
617 \@makeother \}%
618 \catcode \\\ = \active%
\@xverbatim 619 !gdef! \@xverbatim [%
620   [! \endlinechar! m@ne \! \everyeof [! \@nx] %
621     ! \edef! \verbatim@currenvir [%
622       ! \@xa! \scantokens! \@xa [! \@currenvir] %
623     ] % of \verbatim@currenvir. This macro is defined as the meaning of
624       % \@currenvir rescanned. It's done specially for the active star in my
625       % verbatims. \@currenvir is fully expanded but my active star is \pro;
626       % tected.
627
628       ! \@xa] % and here a little trick with groups:
629       ! \@xa! \def! \@xa! \verbatim@currenvir
630       ! \@xa [! \verbatim@currenvir] %
631       ! \edef! \verbatim@edef [%
632       ! \def! \@nx! \verbatim@end %
633       #####! \noexpand \end! \@nx { %
634       ! \@xa! \unexpanded! \@xa [! \verbatim@currenvir] %
635     } [%
636     #####! \@nx! \end [! \@currenvir] ] ] %
637     ! \verbatim@edef
638     ! \verbatim@end] %
639 ! \endgroup
\@sxverbatim 646 \let \@sxverbatim = \@xverbatim

F. Mittelbach says the below is copied almost verbatim from LATEX source, modulo
\check@percent.

\@verbatim 651 \def \@verbatim { %

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

657 \parsep \parskip

From \@trivlist:

659 \if@noskipsec \leavevmode \fi
660 \@topsepadd \topsep
661 \ifvmode
662   \advance \@topsepadd \partopsep
663 \else
664   \unskip \par
665 \fi
666 \@topsep \topsepadd
667 \advance \@topsep \parskip
668 \@outerparskip \parskip

```

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

```

670 \@@par\advspace\@topsep
671 \if@minipage\else\vskip\parskip\fi
673 \advance\@totalleftmargin\verbatimleftskip
674 \raggedright
675 \leftskip\@totalleftmargin% so many assignments to preserve the list
    thinking 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[*].
681 \@@par% most probably redundant.
682 \@tempswafalse
683 \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
688 \ifvmode\if@tempswa\advspace\stanzaskip\@tempswafalse%
    \fi\fi
689 \@@par
690 \penalty\interlinepenalty_\check@percent}%
691 \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.
694 \obeylines
695 \ttverbatim
696 \verbatim@specials
697 }

```

```

\stanzaskip 699 \@ifundefined{stanzaskip}{\newlength\stanzaskip}{}
700 \stanzaskip=\medskipamount

```

```

\verbatimleftskip 703 \newskip\verbatimleftskip
705 \verbatimleftskip\leftmargini

```

```

\verbatimhangindent 707 \newskip\verbatimhangindent
709 \verbatimhangindent=3em

```

```

\check@percent 713 \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 720 \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* 729 \def\verb{%
\verb 730 \relax\ifmmode\hbox\else\leavevmode\null\fi
731 \bgroup
732 \ttverbatim
733 \verbatim@specials
734 \gm@verb@eol

```

```

735 \gm@ifstar
736 {\verb@lasthook \@sverb@chbsl}%
737 {\gmobeyspaces \frenchspacing \verb@lasthook \@sverb@chbsl}}%
      in the LATEX version there's \@vobeyspaces instead of \gmobeyspaces.
741 \emptify\verb@lasthook
\@sverb@chbsl 743 \def \@sverb@chbsl#1 {\@sverb#1 \check@bslash}
\@def@breakbslash 746 \def \@def@breakbslash {\breakbslash}% because \ is \defined as \break ;
      bslash 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 753 \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.
760 \@namedef {gmu@#2wd@name} {#2 \wd\of
761 \@xa \meaning \the \font \space \at \detoken@xa \f@size \pt}
763 \@namedef {gmu@measure#2} {%
764 \unless \ifcstype \csname \gmu@#2wd@name \endcsname \endcsname
765 \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.
768 \@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 expanded value of width of the char measured among copies of it.
772 \typeout {@@@ \package \gmverb:
773 \@xa \string
774 \csname \csname \gmu@#2wd@name \endcsname \endcsname
775 \space \is
776 \csname \csname \gmu@#2wd@name \endcsname \endcsname}%
777 \fi
778 }% of \gmu@measure<char-name>.
780 \@nameedef {type@#2} {%
781 \nx \leavevmode
782 \@xa \nx \csname \gmu@measure#2 \endcsname
783 \hbox \to \nx \csname
784 \@xa \nx \csname \gmu@#2wd@name \endcsname \nx \endcsname
785 {\IfValueTF {#3} {\nx#3} {\xa \nx \csname #2 \endcsname}%
786 \nx \hss}%
787 }% of \type@<char-name>,
788 }% of \DefineTypeChar.
790 \DefineTypeChar \{\bslash}% this defines \type@bslash and its aides \gmu@measurebsl
      and \gmu@bslashwd@name.
793 \DefineTypeChar \{\lbrace [\xiilbrace]% this defines \type@lbrace and
      its auxilia analogous to the above.
\check@bslash 796 \def \check@bslash {%
797 \@ifnextchar \@def@breakbslash
798 {\type@bslash \yesy \@gobble}% note we allow hyphenation but actually
      this will have effect if \hyphenchar \font allows hyphenation (when it's
      not > 0).

```

```

801   {}
805 \let \verb@balance@group \@empty
\verb@egroup 808 \def \verb@egroup {\global \let \verb@balance@group \@empty
809   \hyphenchar \font = \gmv@storedhyphenchar
810   \egroup}

```

```

\gm@verb@eol 814 \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.

```

826 \begingroup
827 \obeylines \obeyspaces%
828 \gdef \verb@eolOK {\obeylines%
\check@percent 829 \def ^^M {\check@percent}%
830 }%
831 \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 836 \def \verbeolOK {\let \gm@verb@eol \verb@eolOK}

```

And back to the main matter,

```

839 \def \@sverb#1 {%
840   \catcode `#1 \active \lccode `~ = `#1%
841   \gdef \verb@balance@group {\verb@egroup
842     \@latex@error {Illegal use of \backslash verb command}%
      \@ehc}%
843   \aftergroup \verb@balance@group
844   \lowercase {\let ~ \verb@egroup}% here we make the delimiter to be the
      macro closing the verbatim group.
846 }

```

```

\verbatim@nolig@list 848 \def \verbatim@nolig@list {\do ` \do < \do > \do , \do \ ' \do \-}

```

```

\do@noligs 850 \def \do@noligs#1 {%
851   \catcode `#1 \active
852   \begingroup
853   \lccode `~ = `#1 \relax
854   \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 859 \def \dekclubs {\gm@ifstar {\MakeShortVerb* \ | } {\MakeShortVerb%
\ | }}
\olddekclubs 860 \def \olddekclubs {\OldMakeShortVerb \ | }

```

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

```

\edverbs 868 \newcommand* \edverbs {%

```

```

869 \let \gmv@dismath \[%
870 \let \gmv@edismath \]%
871 \def \[ {%
872   \@ifnextac \gmv@disverb \gmv@dismath}%
873 \relaxen \edverbs}%

\gmv@disverb 875 \def \gmv@disverb {%
876   \gmv@dismath
878   \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 890 \def \old@MakeShortVerb#1 {%
891   \@xa \ifx \csname _cc \string#1 \endcsname \relax
892   \@shortvrbinfo {Made_} {#1} \@shortvrbdef
893   \add@special {#1}%
894   \AddtoPrivateOthers#1% a macro to be really defined in gmdoc.
896   \@xa
897   \xdef \csname _cc \string#1 \endcsname { \the \catcode `#1 }%
898   \begingroup
899   \catcode `~ \active_ \lccode `~ `#1%
900   \lowercase {%
901     \global \@xa \let \csname _ac \string#1 \endcsname ~%
902     \@xa \gdef \@xa ~ \@xa {%
903       \@shortvrbdef ~ } }%
904   \endgroup
905   \global \catcode `#1 \active
906   \else
907   \@shortvrbinfo \@empty {#1_ already} { \@empty \verb (*) }%
908   \fi}

\OldMakeShortVerb 911 \def \OldMakeShortVerb { \begingroup
912   \let \@MakeShortVerb = \old@MakeShortVerb
913   \gm@ifstar { \eg@MakeShortVerbStar } { \eg@MakeShortVerb } }

\eg@MakeShortVerbStar 916 \def \eg@MakeShortVerbStar#1 { \MakeShortVerb*#1 \endgroup}
\eg@MakeShortVerb 917 \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 \visibleSPACE till \begin{document} to recognise possible redefinition of it when xltextra is loaded.

```

929 \let \gmd@preambleABD \AtBeginDocument
930 \AtBeginDocument { \let \gmd@preambleABD \firstofone}

932 \RequirePackage {xcolor}% for \providecolor

\VisSpacesGrey 934 \def \VisSpacesGrey {%

```

```

936 \providecolor {visspacesgrey} {gray} {0.5}%
937 \gmd@preambleABD{%
938   \edef\visiblespace{%
939     \hbox{\@nx\textcolor{visspacesgrey}%
940       {\@xa\unexpanded\@xa{\visiblespace}}}%
941   }}

```

Verbatim specials—CSes in verbatims

```

\verbatim@specials 946 \pdef\verbatim@specials{% This declaration only defines a bearer of the ‘ver-
                    batim 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.
957   \@bsphack
958   \begingroup
959   \let\do\@makeoother
960   \dospecials
961   \catcode`\_ =10
962   \verbatim@specials@iii}

\verbatim@specials@iii 964 \pdef\verbatim@specials@iii#1#2#3{% as you see, we take only first three
                        arguments in a despecialized group. It’s to avoid \futurelet of the op-
                        tionals’ 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 back-
                        slash would be somewhat difficult.
970   \endgroup
\verbatim@specials@list 971   \def\verbatim@specials@list{#1#2#3}%
972   \@ifnextchar[%
973   {\begingroup\let\do\@makeoother\dospecials
974     \catcode`\_ =9
975     \verbatim@specials@iv}%
976   {\addtomacro\verbatim@specials@list{\NoValue\NoValue}%
977     \verbatim@specials@vi}%
978 }% of \verbatim@specials@iii.

\verbatim@specials@iv 980 \pdef\verbatim@specials@iv[#1]{%
981   \endgroup
982   \addtomacro\verbatim@specials@list{#1}%
983   \@ifnextchar[%
984   {\begingroup\let\do\@makeoother\dospecials
985     \catcode`\_ =9
986     \verbatim@specials@v}%
987   {\addtomacro\verbatim@specials@list{\NoValue}%
988     \verbatim@specials@vi}%
989 }% of \verbatim@specials@iv.

\verbatim@specials@v 992 \pdef\verbatim@specials@v[#1]{%
993   \endgroup
994   \addtomacro\verbatim@specials@list{#1}%
995   \verbatim@specials@vi

```

996 }% of \verbatim@specials@v.

```
\verbatim@specials@vi 998 \DeclareCommand\verbatim@specials@vi\long{b}
999 {\addtomacro\verbatim@specials@list{{#1}}}%
1000 \@esphack}
```

```
\verbatim@specials 1002 \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.
1005 \ifdefined\verbatim@specials@list
1006 \@xa\verbatim@specials@\verbatim@specials@list
1007 \fi
1008 }% of \verbatim@specials.
```

```
\verbatim@specials@ 1010 \long\def\verbatim@specials@#1#2#3#4#5#6{%
1012 \catcode`#1=0
1013 \protected\@namedef{#1}{#1}%
1014 \catcode`#2=1
1015 \protected\@namedef{#2}{#2}%
1016 \catcode`#3=2
1017 \protected\@namedef{#3}{#3}%
1018 \edef\gmu@tempa{\the\endlinechar}%
1019 \endlinechar\m@ne␣% we have to suppress adding of a line end by \scant |
tokens since it would turn into an active char ^^M and raise an error (which
actually did happen).
1022 \scantokens{%
1023 #1let#1bgroup=#2%
1024 #1let#1egroup=#3%
1025 #1catcode#1backquote#1h=6#1relax%
1026 #1pdef#1<h1>#2#1meta#2h1#3#3%
1027 #1catcode#1backquote#1h=11#1relax%
1028 }%
1029 \endlinechar\gmu@tempa\relax
1030 \IfValueT{#4}{%
1031 \catcode`#4=3
1032 \protected\@namedef{#4}{#4}}%
1033 \IfValueT{#5}{%
1034 \begingroup
1035 \lccode`~=#5\lowercase{\endgroup\let~\metachar}%
1036 \protected\@namedef{#5}{#5}%
1037 \catcode`#5=\active
1038 }% of if value #5.
1039 \PutIfValue{#6}%
1040 }
```

```
\noverbatimspecials 1042 \pdef\noverbatimspecials{\let\verbatim@specials@list%
\@undefined}
```

```
\GMverbatim@specials 1045 \def\GMverbatim@specials{%
1046 \gmu@testdefined\XeTeXversion\iftrue
1047 \verbatim@specials␣<»[;][>%
1048 {\def\|\{\metachar{\$}\vert$}}%
1050 \makestarlow
1051 \relaxen\`% to provide a CS separator (space is not 10 in verbatims).
1052 }% of #6.
```

```

1053 \fi
1054 }% of \GMverbatim specials.

```

Partial \verb in arguments

Now command for partial verbatims in arguments of commands:

```

1060 \let \gmu@tempa \all@stars
1061 \@xa \addtomacro \@xa \gmu@tempa \@xa {\all@unders}

```

```

\gmv@hashhalfing
\xiihash

```

```

1063 \foone {\catcode `#=\active}
1064 {\def \gmv@hashhalfing {%
1066 \def# {\xiihash \@ifnextchar# \gobble {} }%
1067 \catcode `#\active}%
1068 }

```

```

1071 \foone {\@makeother \^^R} {%
1072 \@xa \DeclareCommand \@xa \scanverb \@xa {%
1074 \@xa_Q \@xa {\gmu@tempa} >Pm} {%
% #1 Q{*_}
% #2 m the stuff to be rescanned and typeset verbatim. Note that %
will be executed during first scan so at best will disappear.

```

Spaces are ignored (because of detokenizers that add a space after a CS) but if you declare some \verbatim specials, then you can use / /_ where / denotes the escape char in verbatim.

```

1083 \begingroup
1084 \gmu@septify
1085 \endlinechar=\m@ne
1086 \@xa \IfIntersect \@xa {\all@stars} {#1}%
1087 {\def \_ {\breakablevispace}}%
1088 {\let \_ =\space}%
1089 \@xa \IfIntersect \@xa {\all@unders} {#1}%
1090 {} {% We make spaces ignored only if there was no underscore in #1 and if #2
doesn't contain \_.
1092 \IfAmong \_ \among {#2}%
1093 {} {\addtomacro \verb@lasthook {\catcode `\_ =9\_}}}%
1094 \addtomacro \verb@lasthook {\gmv@hashhalfing\_}%
1095 \@makeother \^^R%
1096 \edef \gmu@tempa {%
1097 \@nx \scantokens {%
1098 \bslash\_verb%
1099 \^^R \detokenize {#2} \^^R% we delimit the \verb's argument with
'other' \^^R assuming this char to be used very seldom if at all.
1102 }% of \scantokens,
1103 } \gmu@tempa
1104 \endgroup
1105 }% of \scanverb,
1106 }% of \foone.

```

```

\verbDiscretionaryHyphen
\gmv@hyphenchar
\gmv@hyphen

```

```

1109 \def \verbDiscretionaryHyphen#1#2 {%
1112 \def \gmv@hyphenchar {\numexpr#1 \relax}%
1113 \def \gmv@hyphen {#2}%
1114 }
1116 \verbDiscretionaryHyphen {"A6} {}

```



```

\gmu@tempa 1118 \def\gmu@tempa {%
\verbLongDashes 1119   \DeclareCommand\verbLongDashes {
1120     >iS{-}% to memorise which dash we set
1121     B{1.41}□% expansion of en-dash
1122     >iS{-}□% as above
1123     B{2}□% expansion of em-dash
1124   }%
1125 }

\gmu@tempb 1127 \def\gmu@tempb{\catcode`-\active□\catcode`-\active}
1129 \foone{\catcode`-\active\catcode`-\active}
1130 {%
1131   \edef\gmu@tempa{\@xau\gmu@tempa
1132     {%
1133       \@nx\addtomacro\@nx\ttverbatim@hook{%
1134         \@xau\gmu@tempb
1135         \def\@nx-{\@nx\scalebox{##1}[1]{\string-}}%
1136         \def\@nx-{\@nx\scalebox{##2}[1]{\string-}}%
1137       }%
1138     }%
1139   }%
1140 }%
1141 \gmu@tempa
1144 \endinput% for the Tradition.

```

Change History

<p>v0.79 \edverbs: added, 860</p>	<p>v0.84 General: Checksum 658 , 0</p>
<p>v0.80 \edverbs: debugged, i.e. \hbox added back and redefinition of \[, 860 \xiiclub: \ttverbatim@hook added, 436</p>	<p>v0.85 General: added restoring of \hyphenpenalty and \exhyphenpenalty and setting \hyphenchar=-1, 1144 Checksum 673 , 0</p>
<p>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, 1144</p>	<p>v0.87 General: Checksum 661 , 0 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, 1144</p>
<p>v0.82 General: Checksum 663 , 0</p>	<p>v0.88 General: Checksum 682 , 0 \VisSpacesGrey: added, or rather moved here from gmdoc, 934</p>
<p>v0.83 General: added a hook in the active left brace definition intended for gmdoc automatic detection of definitions (in line 371), 1144 Checksum 666 , 0</p>	<p>v0.89 General:</p>

- \dekclubs, \dekclubs* and \olddekclubs made more consistent, shorthands for \MakeShortVerb\|, \MakeShortVerb*\| and \OldMakeShortVerb\| respectively., 1144
 - Checksum 686 , 0
- vo.90
 - General:
 - Checksum 684 , 0
 - some \(\b|e)group changed to \(\begin|end)group, 1144
- vo.91
 - General:
 - Checksum 686 , 0
 - put to CTAN on 2008/11/21, 0
 - \verbatimleftskip:
 - added, 703
- vo.92
 - General:
 - Checksum 979 because of \verbatimspecials, hyphenation
 - in verbatims, low star in verbatims, kerning of backslash in shrunk fonts, 0
- vo.93
 - General:
 - Checksum 1035 because of a bug fix in \scanverb (halving the hashes), 0
 - put to CTAN on 2010/03/04, 0
 - \gmv@hashhalving:
 - cut out as separate macro, 1064
 - \verbDiscretionaryHyphen:
 - added to synchronise hyphen chars in gmdoc's documentation, 1109
 - \xiihash:
 - mandatory argument made long (a bug fix), 1072

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.

- | | | |
|--|--|---|
| <ul style="list-style-type: none"> \-, 848 \@MakeShortVerb, 486, 487, <u>490</u>, 912 \@currenvir, 625, 639 \@def@breakbslash, 746, 797 \@emptify, 468 \@ifnextac, 872 \@nameedef, 780 \@shortvrbdef, <u>486</u>, <u>487</u>, 492, 505, <u>892</u>, 903 \@shortvrbinfo, 492, 511, 517, 519, <u>538</u>, 892, 907 \@sverb@chbsl, 736, 737, <u>743</u> \@topsep, 666, 667, 670 \@topsepadd, 607, 660, 662, 666 \@undefined, 1042 \@xau, 1131, 1134 \add@special, 493, <u>544</u>, 893 | <ul style="list-style-type: none"> \addtomacro, 976, 982, 987, 994, 999, 1061, 1093, 1094, 1133 \AddtoPrivateOthers, 494, <u>720</u>, 894 \afterfi, 505 \all@stars, 1060, 1086 \all@unders, 1061, 1089 alltt, p. 3 \among, 1092 \AtBeginDocument, 929, 930 \breakablevispace, <u>423</u>, <u>429</u>, 1087 \breakbslash, <u>387</u>, <u>398</u>, 413, 746 \breaklbrace, <u>362</u>, <u>371</u>, <u>405</u> \bslash, <u>383</u>, 842, 1098 \check@bslash, 743, <u>796</u> \check@percent, 690, <u>713</u>, <u>829</u> | <ul style="list-style-type: none"> \DeclareCommand, 753, 998, 1072, 1119 \DefineTypeChar, <u>753</u>, 790, 793 \dekclubs, p. 2, <u>859</u> \dekclubs*, p. 2 \DeleteShortVerb, p. 2, <u>515</u> \detoken@xa, 761 \detokenize, 1099 \do@noligs, 446, <u>850</u> \dobreakblankspace, <u>431</u> \dobreakbslash, <u>413</u>, 448 \dobreaklbrace, <u>369</u>, 448 \dobreakspace, 449, 475 \dobreakvisi blespace, <u>429</u>, 475 \edverbs, p. 2, <u>868</u>, 873 \eg@MakeShortVerb, 913, 917 \eg@MakeShortVerbStar, 913, 916 \emptify, <u>741</u> |
|--|--|---|

`\endlinechar`, 620, 1018, 1019, 1029, 1085
`\endverbatim`, [599](#)
`\everyeof`, 620
`\everypar`, 691
`\exhyphenpenalty`, 583, 587

`\f@size`, 761
`\fontencoding`, 471
`\foone`, 367, 381, 410, 427, 435, 1063, 1071, 1129

`\gm@ifstar`, 485, 735, 859, 913
`\gm@lbracehook`, 371, 376
`\gm@testdefined`, 1046
`\gm@verb@eol`, 734, 814, 836
`\gmd@preambleABD`, 929, 930, 937
`\gmobeyspaces`, 585, 737
`\gmu@measurewd`, 765
`\gmu@septify`, 1084
`\gmu@tempa`, 1018, 1029, 1060, 1061, 1074, 1096, 1103, [1118](#), 1131, 1141
`\gmu@tempb`, [1127](#), 1134
`\gmv@dismath`, 869, 872, 876
`\gmv@disverb`, 872, [875](#)
`\gmv@edismath`, 870, [878](#)
`\gmv@exhyphenpe`, 583, 587
`\gmv@hashhalving`, 1064, 1094
`\gmv@hyphen`, [1113](#)
`\gmv@hyphenchar`, 459, [1112](#)
`\gmv@hyphenpe`, 582, 586
`\gmv@packname`, [534](#), [535](#), 539
`\gmv@storedhyphenchar`, 458, 600, 809
`\GMverbatimspecials`, [1045](#)
`\gn@melet`, 768
`\gobble`, 1066

`\hyphenpenalty`, 582, 586

`\if@noskipsec`, 659
`\IfAmong`, 1092
`\ifcsname`, 764
`\ifdefined`, 1005
`\IfIntersect`, 1086, 1089
`\IfValueT`, 1030, 1033
`\IfValueTF`, 785
`\interlinepenalty`, 690

`\leftmargini`, 705

`\MakeShortVerb`, *p.* 2, [484](#), 859, 916, 917
`\makestarlow`, 1050
`\metachar`, 1035, 1048

`\n@melet`, 611
`\newlength`, 699
`\newskip`, 703, 707
`\nobreakbslash`, *p.* 2, [398](#)
`\nobreaklbrace`, *p.* 2, [405](#)
`\NoValue`, 976, 987
`\noververbatimspecials`, *p.* 3, [1042](#)
`\numexpr`, [1112](#)

`\obeyspaces`, 427, 429, 431, 827
`\old@MakeShortVerb`, [890](#), 912
`\olddekclubs`, *p.* 2, [860](#)
`\OldMakeShortVerb`, *p.* 2, 860, [911](#)

`\PackageInfo`, 539
`\parsep`, 657
`\partopsep`, 662
`\pdf`, 398, 405, 445, 946, 964, 980, 992, 1042
`\predisplaypenalty`, 584, 594
`\protected`, 1013, 1015, 1017, 1032, 1036
`\providecolor`, 936
`\PutIfValue`, 1039

`\relaxen`, 376, 873, 1051
`\rem@special`, 520, 545, [560](#)
`\RequirePackage`, 331, 932

`\scalebox`, 1135, 1136
`\scantokens`, 625, 1022, 1097
`\scanverb`, *p.* 3, 1072
`semiverbatim`, *p.* 3
`\setspaceskip`, 455
`\stanzaskip`, 688, [699](#), 700
`\subs`, *p.* 3
`\sup`, *p.* 3

`\textcolor`, 939
`\ttverbatim`, [445](#), 695, 732
`\ttverbatim@hook`, 460, 468, [471](#), 1133
`\type@bslash`, 389, 398, 798
`\type@lbrace`, 363, 405

`\unexpanded`, 637, 940
`\unless`, 764

`\verb`, 486, 487, 511, 517, 519, [729](#), 907
`\verb*`, [729](#)
`\verb@balance@group`, 805, 808, 841, 843
`\verb@egroup`, [808](#), 841, 844
`\verb@eol@error`, 814
`\verb@eolOK`, 828, 836
`\verb@lasthook`, 736, 737, 741, 1093, 1094
`\verbatim`, [581](#)
`verbatim`, [581](#)
`verbatim*`, [594](#)
`\verbatim@currentenv`, 621, 632, 633, 637
`\verbatim@edef`, 634, 640
`\verbatim@end`, 635, 641
`\verba|`
`tim@nolig@list`, 446, 848
`\verbatim@specials`, 696, 733, [1002](#)
`\verbatim@specials@`, 1006, [1010](#)
`\verba|`
`tim@specials@iii`, 962, [964](#)
`\verba|`
`tim@specials@iv`, 975, [980](#)
`\verba|`
`tim@specials@list`, [971](#), 976, 982, 987, 994, 999, 1005, 1006, 1042
`\verba|`
`tim@specials@v`, 986, [992](#)
`\verba|`
`tim@specials@vi`, 977, 988, 995, 998
`\verbatim@hangindent`, 691, [707](#), 709
`\verbatim@leftskip`, 673, [703](#), 705
`\verbatimspecials`, *p.* 3, 946, 1047
`\verbDiscretionary|`
`Hyphen`, *p.* 3, [1109](#), 1116
`\verbeolOK`, *p.* 2, [836](#)
`\VerbHyphen`, *p.* 2, [351](#)
`\verbhyphen`, 345, 353, 363, 388
`\verbLongDashes`, [1119](#)
`\VerbT`, [471](#)
`\VerbT1`, [471](#)

`\vert`, 1048
`\visiblespace`, 423, 938,
940
`\VisSpacesGrey`, p. 2, 934
`visspacesgrey`, p. 2

`\XeTeXversion`, 1046
`\xiiclub`, 436
`\xiihash`, 1066
`\xiilbrace`, 793
`\xiipercen`t, 345

`\yesy`, 364, 389, 398, 405, 798
`\'`, 1051