

Grzegorz Murzynowski

# The gmutils Package<sup>\*</sup>

Written by Grzegorz Murzynowski,  
natror at o2 dot pl

© 2005, 2006, 2007, 2008 by Grzegorz Murzynowski.

This program is subject to the L<sup>A</sup>T<sub>E</sub>X Project Public License.

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

LPPL status: "author-maintained".

Many thanks to my T<sub>E</sub>X Guru Marcin Woliński for his T<sub>E</sub>Xnical support.

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{gmutils}
3 [2008/08/06_vo.91_some_rather_TeXnical_macros,_some_of_them_
   tricky_(GM)]
```

## Contents

<b>Intro</b> . . . . .	2	Negative \addvspace . . . . .	21
Installation . . . . .	2	My heading setup for mwcls . . . . .	23
Contents of the gmutils.zip Archive . . . . .	2	<b>Compatibilising Standard and mwcls</b>	
Compiling of the Documentation . . . . .	2	<b>Sectionings</b> . . . . .	24
<b>A couple of abbreviations</b> . . . . .	3	<b>enumerate* and itemize*</b> . . . . .	25
\@ifnextcat, \@ifnextac . . . . .	5	<b>The Logos</b> . . . . .	26
\afterfi and Pals . . . . .	6	<b>Expanding turning stuff all into ‘other’</b> . . . . .	28
<b>Almost an Environment or</b>		<b>Brave New World of X<sub>3</sub>T<sub>E</sub>X</b> . . . . .	29
Redefinition of \begin . . . . .	7	Fractions . . . . .	29
<b>Improvement of \end</b> . . . . .	8	\resizegraphics . . . . .	30
<b>From relsize</b> . . . . .	8	<b>Varia</b> . . . . .	31
\firstofone and the Queer \catcodes . . . . .	9	\@ifempty . . . . .	36
<b>Some ‘other’ stuff</b> . . . . .	10	\include not only .tex’s . . . . .	36
<b>Metasymbols</b> . . . . .	11	Faked small caps . . . . .	37
<b>Macros for Printing Macros and</b>		See above/see below . . . . .	38
<b>Filenames</b> . . . . .	11	luzniej and napa-	
<b>Storing and Restoring the Meanings</b>		pierki—environments	
<b>of CSs</b> . . . . .	13	used in page breaking for money . . . . .	38
<b>Not only preamble!</b> . . . . .	16	<b>Settings for mathematics in main font</b> . . . . .	41
<b>Third Person Pronouns</b> . . . . .	17	<b>Typesetting dates in my memoirs</b> . . . . .	44
<b>To Save Precious Count Registers</b> . . . . .	17	Minion and Garamond Premier	
<b>Improvements to mwcls Sectioning</b>		kerning and ligature fixes . . . . .	48
<b>Commands</b> . . . . .	18	<b>Change History</b> . . . . .	48
An improvement of MW’s		<b>Index</b> . . . . .	49
\SetSectionFormatting . . . . .	20		

<sup>\*</sup> This file has version number vo.91 dated 2008/08/06.

## Intro

The `gmutils.sty` package provides some macros that are analogous to the standard L<sup>A</sup>T<sub>E</sub>X ones but extend their functionality, such as `\@ifnextcat`, `\addtomacro` or `\begin(*)`. The others are just conveniences I like to use in all my TeX works, such as `\afterfi`, `\pk` or `\cs`.

I wouldn't say they are only for the package writers but I assume some nonzero (L<sup>A</sup>)T<sub>E</sub>X-awareness of the user.

For details just read the code part.

## Installation

Unpack the `gmutils-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 T<sub>E</sub>X distribution's files' database most probably.

## Contents of the gmutils.zip Archive

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

```
gmutils.sty
README
gmutilsDoc.tex
gmutilsDoc.pdf
gmutils.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 L<sup>A</sup>T<sub>E</sub>X on the `gmutilsDoc.tex` file twice (`xelatex gmutils.sty` in the directory you wish the documentation to be in, you don't have copy the `.sty` file there, T<sub>E</sub>X will find it), then `MakeIndex` on the `gmutils.idx` file, and then L<sup>A</sup>T<sub>E</sub>X on `gmutilsDoc.tex` once more.

`MakeIndex` shell command:

```
makeindex -r gmutilsDoc
```

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`, `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.

If you had not installed the `mwcls` classes (available on CTAN and present in T<sub>E</sub>X 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.

```
4 \ifx\XeTeXversion\relax
5 \let\XeTeXversion\@undefined% If someone earlier used the \@ifundefined{%
    XeTeXversion} to test whether the engine is XETEX, then \XeTeXversion is
    defined in the sense of  $\varepsilon$ -TEX tests. In that case we \let it to something really
    undefined. Well, we might keep sticking to \@ifundefined, but it's a macro
```

and it eats its arguments, freezing their catcodes, which is not what we want in line 1129

```
6 \fi
7 \ifdefined\XeTeXversion
8 \XeTeXinputencoding=utf-8% we use Unicode dashes later in this file.
9 \fi% and if we are not in XeTeX, we skip them thanks to XeTeX-test.
```

## A couple of abbreviations

```
\@xa 10 \let\@xa\expandafter
\@nx 11 \let\@nx\noexpand
```

The `\newgif` declaration's effect is used even in the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> source by redefining some particular user defined ifs (UD-ifs henceforth) step by step. The goal is to make the UD-if's assignment global. I needed it at least twice during gmdoc writing so I make it a macro. It's an almost verbatim copy of L<sup>A</sup>T<sub>E</sub>X's `\newif` modulo the letter *g* and the `\global` prefix. (File d: ltdefns.dtx Date: 2004/02/20 Version v1.3g, lines 139–150)

```
\newgif 12 \def\newgif#1{%
13   {\escapechar\m@ne
14     \global\let#1\iffalse
15     \@gif#1\iftrue
16     \@gif#1\iffalse
17   }}
```

'Almost' is also in the detail that in this case, which deals with `\global` assignments, we don't have to bother with storing and restoring the value of `\escapechar`: we can do all the work inside a group.

```
\@gif 18 \def\@gif#1#2{%
19   \@xa\gdef\csname\@xa\@gobbletwo\string#1%
20     g% the letter g for '\global'.
21   \@xa\@gobbletwo\string#2\endcsname
22   {\global\let#1#2}}
```

After `\newgif\ifffoo` you may type `{\foogtrue}` and the `\ifffoo` switch becomes globally equal `\iftrue`. Simili modo `\foogfalse`. Note the letter *g* added to underline globalness of the assignment.

If for any reason, no matter how queer ;-) may it be, you need *both* global and local switchers of your `\if . . .`, declare it both with `\newif` and `\newgif`.

Note that it's just a shorthand. `\global\if<switch>true/false` *does* work as expected.

There's a trouble with `\refstepcounter`: defining `\@currentlabel` is local. So let's `\def` a `\global` version of `\refstepcounter`.

Warning. I use it because of very special reasons in gmdoc and in general it is probably not a good idea to make `\refstepcounter` global since it is contrary to the original L<sup>A</sup>T<sub>E</sub>X approach.

```
\grefstepcounter 23 \newcommand*\grefstepcounter[1]{%
24   {\let\protected@edef=\protected@xdef\refstepcounter{#1}}}
```

Naïve first try `\globaldefs=\tw@` raised an error unknown command `\reserved@e`. The matter was to globalize `\protected@edef` of `\@currentlabel`.

Thanks to using the true `\refstepcounter` inside, it observes the change made to `\refstepcounter` by `hyperref`.

Another shorthand. It may decrease a number of `\expandafte`s e.g.

```
\glet 25 \def\glet{\global\let}
```

L<sup>A</sup>T<sub>E</sub>X provides a very useful `\g@addto@macro` macro that adds its second argument to the current definition of its first argument (works iff the first argument is a no argument macro). But I needed it some times in a document, where `@` is not a letter. So:

```
\gaddtomacro 26 \let\gaddtomacro=\g@addto@macro
```

The redefining of the first argument of the above macro(s) is `\global`. What if we want it local? Here we are:

```
\addto@macro 27 \long\def\addto@macro#1#2{%
28   \toks@{\@xa{#1#2}}%
29   \edef#1{\the\toks@}%
30 }% (\toks@ is a scratch register, namely \tokso.)
```

And for use in the very document,

```
\addtomacro 31 \let\addtomacro=\addto@macro
\addtotoks 32 \long\def\addtotoks#1#2{%
33   #1=\@xa{\the#1#2}}
\@emptyify 34 \newcommand*\@emptyify[1]{\let#1=\@empty}
\emptyify 35 \@ifdefinable\emptyify{\let\emptyify\@emptyify}
```

Note the two following commands are in fact one-argument.

```
\g@emptyify 36 \newcommand*\g@emptyify{\global\@emptyify}
\gemptyify 37 \@ifdefinable\gemptyify{\let\gemptyify\g@emptyify}
\@relaxen 38 \newcommand*\@relaxen[1]{\let#1=\relax}
\relaxen 39 \@ifdefinable\relaxen{\let\relaxen\@relaxen}
```

Note the two following commands are in fact one-argument.

```
\g@relaxen 40 \newcommand*\g@relaxen{\global\@relaxen}
\grelaxen 41 \@ifdefinable\grelaxen{\let\grelaxen\g@relaxen}
```

For the heavy debugs I was doing while preparing `gmdoc`, as a last resort I used `\showlists`. But this command alone was usually too little: usually it needed setting `\showboxdepth` and `\showboxbreadth` to some positive values. So,

```
\gmshowlists 42 \def\gmshowlists{\showboxdepth=1000_\showboxbreadth=1000_%
\showlists}
```

```
\nameshow 43 \newcommand*\nameshow[1]{\@xa\show\csname#1\endcsname}
```

Standard `\string` command returns a string of ‘other’ chars except for the space, for which it returns `_10`. In `gmdoc` I needed the spaces in macros’ and environments’ names to be always `_12`, so I define

```
\xiistring 44 \def\xiistring#1{%
45   \if\@nx#1\xiispace
46     \xiispace
47   \else
48     \string#1%
49   \fi}
```

`\@ifnextcat, \@ifnextac`

As you guess, we `\def \@ifnextcat` à la `\@ifnextchar`, see L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  source dated 2003/12/01, file `d`, lines 253–271. The difference is in the kind of test used: while `\@ifnextchar` does `\ifx`, `\@ifnextcat` does `\ifcat` which means it looks not at the meaning of a token(s) but at their `\catcode`(s). As you (should) remember from *The T<sub>E</sub>Xbook*, the former test doesn't expand macros while the latter does. But in `\@ifnextcat` the peeked token is protected against expanding by `\noexpand`. Note that the first parameter is not protected and therefore it shall be expanded if it's a macro. Because an assignment is involved, you can't test whether the next token is an active char.

```
\@ifnextcat 50 \long\def\@ifnextcat#1#2#3{%
51   \def\reserved@d{#1}%
52   \def\reserved@a{#2}%
53   \def\reserved@b{#3}%
54   \futurelet\@let@token\@ifncat}

\@ifncat 55 \def\@ifncat{%
56   \ifx\@let@token\@sptoken
57     \let\reserved@c\@xifncat
58   \else
59     \ifcat\reserved@d\@nx\@let@token
60       \let\reserved@c\reserved@a
61     \else
62       \let\reserved@c\reserved@b
63     \fi
64   \fi
65   \reserved@c}

66 {\def\:\{\let\@sptoken= }\:}% this makes \@sptoken a space token.
67 \def\:\{\@xifncat}\@xa\gdef\:\{\futurelet\@let@token\@ifncat}}
```

Note the trick to get a macro with no parameter and requiring a space after it. We do it inside a group not to spoil the general meaning of `\:` (which we extend later).

The next command provides the real `\if` test for the next token. It should be called `\@ifnextchar` but that name is assigned for the future `\ifx` text, as we know. Therefore we call it `\@ifnextif`.

```
\@ifnextif 68 \long\def\@ifnextif#1#2#3{%
69   \def\reserved@d{#1}%
70   \def\reserved@a{#2}%
71   \def\reserved@b{#3}%
72   \futurelet\@let@token\@ifnif}

\@ifnif 73 \def\@ifnif{%
74   \ifx\@let@token\@sptoken
75     \let\reserved@c\@xifnif
76   \else
77     \if\reserved@d\@nx\@let@token
78       \let\reserved@c\reserved@a
79     \else
80       \let\reserved@c\reserved@b
81     \fi
82   \fi
83   \reserved@c}
```

```

84 {\def\:{\let\@sptoken=\}\:\:\_this\_makes\_|\@sptoken|_a\_space\_
    token.
85 \def\:{\@xifnif}\@xa\gdef\:{\futurelet\@let@token\@ifnif}}

```

But how to peek at the next token to check whether it's an active char? First, we look with `\@ifnextcat` whether there stands a group opener. We do that to avoid taking a whole `{...}` as the argument of the next macro, that doesn't use `\futurelet` but takes the next token as an argument, tests it and puts back intact.

```

\@ifnextcat 86 \long\def\@ifnextac#1#2{%
            87   \@ifnextcat\bgroup{#2}{\gm@ifnac{#1}{#2}}}
\gm@ifnac   88 \long\def\gm@ifnac#1#2#3{%
            89   \ifcat\@nx~\@nx#3\afterfi{#1#3}\else\afterfi{#2#3}\fi}

```

Yes, it won't work for an active char `\let` to `{_}`, but it *will* work for an active char `\let` to a char of catcode  $\neq 1$ . (Is there anybody on Earth who'd make an active char working as `\bgroup`?)

Now, define a test that checks whether the next token is a genuine space, <sup>10</sup> that is. First define a CS `\let` such a space. The assignment needs a little trick (*The T<sub>E</sub>Xbook* appendix D) since `\let`'s syntax includes one optional space after `=`.

```

90 \let\gmu@reserveda\*%
\*      91 \def\*{%
        92   \let\*\gmu@reserveda
        93   \let\gm@letspace=\}%
        94   \*\%
\@ifnextspace 95 \def\@ifnextspace#1#2{%
            96   \let\gmu@reserveda\*%
\*      97   \def\*{%
        98     \let\*\gmu@reserveda
        99     \ifx\@let@token\gm@letspace\afterfi{#1}%
       100     \else\afterfi{#2}%
       101     \fi}%
       102   \futurelet\@let@token\*}

```

First use of this macro is for an active `-` that expands to `---` if followed by a space. Another to make dot checking whether is followed by `~` without gobbling the space if it occurs instead.

## `\afterfi` and Pals

It happens from time to time that you have some sequence of macros in an `\if...` and you would like to expand `\fi` before expanding them (e.g., when the macros should take some tokens next to `\fi...` as their arguments. If you know how many macros are there, you may type a couple of `\expandafters` and not to care how terrible it looks. But if you don't know how many tokens will there be, you seem to be in a real trouble. There's the Knuthian trick with `\next`. And here another, revealed to me by my T<sub>E</sub>X Guru.

I think the situations when the Knuthian (the former) trick is not available are rather seldom, but they are imaginable at least: the `\next` trick involves an assignment so it won't work e.g. in `\edef`. But in general it's only a matter of taste which one to use.

One warning: those macros peel the braces off, i.e.,

```
\if...\afterfi{\@makeother\^~M}\fi
```

causes a leakage of  $\sim M_{12}$ . To avoid pollution write

```
\if.. \afterfi{\bgroup\@makeother\~M\egroup}\fi.
\afterfi 103 \long\def\afterfi#1#2\fi{\fi#1}
```

And two more of that family:

```
\afterfifi 104 \long\def\afterfifi#1#2\fi#3\fi{\fi\fi#1}
\afteriffifi 105 \long\def\afteriffifi#1#2\if#3\fi#4\fi{\fi#1}
```

Notice the refined elegance of those macros, that cover both ‘then’ and ‘else’ cases thanks to #2 that is discarded.

```
\afteriffiffifi 106 \long\def\afteriffiffiffifi#1#2\fi#3\fi#4\fi{\fi#1}
\afteriffiffifi 107 \long\def\afteriffiffiffifi#1#2\fi#3\fi#4\fi{\fi\fi#1}
\afteriffiffifi 108 \long\def\afteriffiffiffifi#1#2\fi#3\fi#4\fi{\fi\fi\fi#1}
```

## Almost an Environment or Redefinition of `\begin`

We’ll extend the functionality of `\begin`: the non-starred instances shall act as usual and we’ll add the starred version. The difference of the latter will be that it won’t check whether the ‘environment’ has been defined so any name will be allowed.

This is intended to structure the source with named groups that don’t have to be especially defined and probably don’t take any particular action except the scoping.

(If the `\begin*`’s argument is a (defined) environment’s name, `\begin*` will act just like `\begin`.)

Original L<sup>A</sup>T<sub>E</sub>X’s `\begin`:

```
\def\begin#1{%
  \@ifundefined{#1}%
    {\def\reserved@a{\@latex@error{Environment #1
      undefined}\@eha}}%
    {\def\reserved@a{\def\@currenvir{#1}%
      \edef\@currenvline{\on@line}%
      \csname #1\endcsname}}%
  \@ignorefalse
  \begingroup\@endpefalse\reserved@a}
\@begnamedgroup 109 \@ifdefinable\@begnamedgroup{\relax}
\@begnamedgroup 110 \def\@begnamedgroup#1{%
  111 \@ignorefalse% not to ignore blanks after group
  112 \begingroup\@endpefalse
  113 \def\@currenvir{#1}%
  114 \edef\@currenvline{\on@line}%
  115 \csname_#1\endcsname}% if the argument is a command’s name (an environ-
    ment’s e.g.), this command will now be executed. (If the corresponding
    control sequence hasn’t been known to TEX, this line will act as \relax.)
```

For back compatibility with my earlier works

```
\bnamegroup 116 \let\bnamegroup\@begnamedgroup
And for the ending
```

```
\enamegroup 117 \def\enamegroup#1{\end{#1}}
```

And we make it the starred version of `\begin`.

```
\old@begin 118 \let\old@begin\begin
\begin 119 \def\begin{\@ifstar{\@begnamedgroup}{\old@begin}}
\begin*
\begin
```

## Improvement of \end

It's very clever and useful that \end checks whether its argument is ifx-equivalent @currenvir. However, it works not quite as I would expect: Since the idea of environment is to open a group and launch the cs named in the \begin's argument. That last thing is done with \csname... \endcsname so the char catcodes are equivalent. Thus should be also in the \end's test and therefore we ensure the compared texts are both expanded and made all 'other'.

```
\@checkend 120 \def\@checkend#1{%
121   \edef\reserved@a{\@xa\string\csname#1\endcsname}%
122   \edef\exii@currenvir{\@xa\string\csname@currenvir\endcsname}%
123   \ifx\reserved@a\exii@currenvir\else\@badend{#1}\fi}
```

Thanks to it you may write \begin{macrocode\*} with \*<sub>12</sub> and end it with \end{macrocode\*} with \*<sub>11</sub> (that was the problem that led me to this solution). The error messages looked really funny:

! LaTeX Error: \begin{macrocode\*} on input line 1844 ended by \end{macrocode\*}.

Of course, you might write also \end{macrocode\star} where \star is defined as 'other' star or letter star.

## From relsize

As file relsize.sty, v3.1 dated July 4, 2003 states, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> version of these macros was written by Donald Arseneau [asnd@triumf.ca](mailto:asnd@triumf.ca) and Matt Swift [swift@bu.edu](mailto:swift@bu.edu) after the L<sup>A</sup>T<sub>E</sub>X 2.09 smaller.sty style file written by Bernie Cosell [cosell@WILMA.BBN.COM](mailto:cosell@WILMA.BBN.COM).

I take only the basic, non-math mode commands with the assumption that there are the predefined font sizes.

```
\relsize You declare the font size with \relsize{<n>} where <n> gives the number of steps
("mag-step" = factor of 1.2) to change the size by. E.g., n = 3 changes from \normalsize
\smaller to \LARGE size. Negative n selects smaller fonts. \smaller == \relsize{-1};
\larger \larger == \relsize{1}. \smallerr(my addition) == \relsize{-2}; \largerr
\smallerr guess yourself.
\largerr (Since \DeclareRobustCommand doesn't issue an error if its argument has been de-
fined and it only informs about redefining, loading relsize remains allowed.)
```

```
\relsize 124 \DeclareRobustCommand*\relsize[1]{%
125   \ifmmode_\@nomath\relsize\else
126     \begingroup
127     \@tempcnta_% assign number representing current font size
128     \ifx\@currsz\normalsize_4\else_%% funny order is to have most
...
129     \ifx\@currsz\small_3\else_%% ...likely sizes checked first
130     \ifx\@currsz\footnotesize_2\else
131     \ifx\@currsz\large_5\else
132     \ifx\@currsz\Large_6\else
133     \ifx\@currsz\LARGE_7\else
134     \ifx\@currsz\scriptsize_1\else
135     \ifx\@currsz\tiny_0\else
136     \ifx\@currsz\huge_8\else
137     \ifx\@currsz\Huge_9\else
138     4\rs@unknown@warning_%% unknown state: \normalsize as
starting point
```



```

139      \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
Change the number by the given increment:
140      \advance\@tempcnta#1\relax
watch out for size underflow:
141      \ifnum\@tempcnta<\z@\rs@size@warning{small}\string\tiny}%
        \@tempcnta\z@\fi
142      \@xa\endgroup
143      \ifcase\@tempcnta_\% set new size based on altered number
144      \tiny_\or_\scriptsize_\or_\footnotesize_\or_\small_\or_\%
        \normalsize_\or
145      \large_\or_\Large_\or_\LARGE_\or_\huge_\or_\Huge_\else
146      \rs@size@warning{large}\string\Huge\Huge
147 \fi\fi}% end of \relsize.
\rs@size@warning 148 \providecommand*\rs@size@warning[2]{\PackageWarning{gmutils_
        (relsize)}{\\%
149 Size requested is too #1.\MessageBreak Using #2 instead}}
\rs@unknown@warning 150 \providecommand*\rs@unknown@warning{\PackageWarning{gmutils_
        (relsize)}{Current font size
151 is unknown! (Why?!?)\MessageBreak Assuming \string\normalsize}}
And a handful of shorthands:
\larger 152 \DeclareRobustCommand*\larger[1][\@one]{\relsize{+#1}}
\smaller 153 \DeclareRobustCommand*\smaller[1][\@one]{\relsize{-#1}}
\textlarger 154 \DeclareRobustCommand*\textlarger[2][\@one]{\relsize{+#1}#2}
\textsmaller 155 \DeclareRobustCommand*\textsmaller[2][\@one]{\relsize{-#1}#2}
\largerr 156 \DeclareRobustCommand*\largerr{\relsize{+2}}
\smallerr 157 \DeclareRobustCommand*\smallerr{\relsize{-2}}

```

## **\firstofone and the Queer \catcodes**

Remember that once a macro's argument has been read, its `\catcodes` are assigned forever and ever. That's what is `\firstofone` for. It allows you to change the `\catcodes` locally for a definition *outside* the changed `\catcodes` group. Just see the below usage of this macro 'with T<sub>E</sub>X's eyes', as my T<sub>E</sub>X Guru taught me.

```

158 \long\def\firstofone#1{#1}

```

The next command, `\foone`, is intended as two-argument for shortening of the `\bgroup... \firstofone{ \egroup... }` hack.

```

\foone 159 \long\def\foone#1{\bgroup#1\egroupfirstofone}
160 \long\def\egroupfirstofone#1{\egroup#1}
\foeatletter 161 \long\def\foeatletter{\foone\makeatletter}

```

And this one is defined, I know, but it's not `\long` with the standard definition.

```

\gobble 162 \long\def\gobble#1{}
\gobbletwo 163 \let\gobbletwo\@gobbletwo

```

## Some ‘other’ stuff

Here I define a couple of macros expanding to special chars made ‘other’. It’s important the cs are expandable and therefore they can occur e.g. inside `\csname . . . \endcsname` unlike e.g. `cs’es \chardefed`.

```

164 \foone{\catcode`\_ =8_}%
\subs 165 {\let\subs=_}

166 \foone{\@makeother\_}%
\xiiunder 167 {\def\xiiunder{_{}}%

168 \ifdefined\XeTeXversion
\xiiunder 169 \def\xiiunder{\char"005F_}%
170 \let\_ \xiiunder
171 \fi

172 \foone{\catcode`\ [=1_ \@makeother\{
173 \catcode`\ ] =2_ \@makeother\}}%
174 [%
\xiilbrace 175 \def\xiilbrace[{]%
\xiirbrace 176 \def\xiirbrace[}%
177 ]% of \firstofone

```

Note that L<sup>A</sup>T<sub>E</sub>X’s `\@charlb` and `\@charrb` are of catcode 11 (‘letter’), cf. The L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> Source file k, lines 129–130.

Now, let’s define such a smart `_` (underscore) which will be usual `_8` in the math mode and `_12` (‘other’) outside math.

```

178 \foone{\catcode`\_ =\active}
179 {%
\smartunder 180 \newcommand*\smartunder{%
181 \catcode`\_ =\active
182 \def_{\ifmmode\subs\else\_ \fi}}}% We define it as \_ not just as \xiiunder
because some font encodings don’t have _ at the \char`\_ position.

183 \foone{\catcode`\! =o
184 \@makeother\!}
\xiibackslash 185 {\!newcommand*\xiibackslash{\!}}
\bslash 186 \let\bslash=\xiibackslash

187 \foone{\@makeother\}%
\xiipercent 188 {\def\xiipercent{}}%

189 \foone{\@makeother\&}%
\xiiand 190 {\def\xiiand{&}}%

191 \foone{\@makeother\ }%
\xiispace 192 {\def\xiispace{ }}%

```

We introduce `\visible`space from Will Robertson’s `xltextra` if available. It’s not sufficient `\@ifpackageloaded{xltextra}` since `\xxt@visible`space is defined only unless `no-verb` option is set. 2008/08/06 I recognized the difference between `\xiispace` which has to be plain ‘other’ char (used in `\xiistring`) and something visible to be printed in any font.

```

193 \AtBeginDocument{%
194 \ifdefined\xxt@visible
195 \let\visible\xxt@visible
196 \else

```

```

197 \let\visiblespace\xiispace
198 \fi}

```

## Metasymbols

I fancy also another Knuthian trick for typesetting *metasymbols* in *The T<sub>E</sub>Xbook*. So I repeat it here. The inner `\meta` macro is copied verbatim from doc’s v2.1b documentation dated 2004/02/09 because it’s so beautifully crafted I couldn’t resist. I only don’t make it `\long`.

“The new implementation fixes this problem by defining `\meta` in a radically different way: we prevent hyphenation by defining a `\language` which has no patterns associated with it and use this to typeset the words within the angle brackets.”

```

\meta 199 \DeclareRobustCommand*\meta[1]{%

```

“Since the old implementation of `\meta` could be used in math we better ensure that this is possible with the new one as well. So we use `\ensuremath` around `\langle` and `\rangle`. However this is not enough: if `\meta@font@select` below expands to `\itshape` it will fail if used in math mode. For this reason we hide the whole thing inside an `\nfss@text` box in that case.”

```

200 \ensuremath\langle
201 \ifmmode\@xa\nfss@text\fi
202 {%
203 \meta@font@select

```

Need to keep track of what we changed just in case the user changes font inside the argument so we store the font explicitly.

```

204 #1\/%
205 }\ensuremath\rangle
206 }

```

But I define `\meta@font@select` as the brutal and explicit `\it` instead of the original `\itshape` to make it usable e.g. in the `gmdoc`’s `\cs` macro’s argument.

```

\meta@font@select 207 \def\meta@font@select{\it}

```

The below `\meta`’s drag<sup>1</sup> is a version of *The T<sub>E</sub>Xbook*’s one.

```

\<...> 208 \def\<#1>{\meta{#1}}

```

## Macros for Printing Macros and Filenames

First let’s define three auxiliary macros analogous to `\dywiz` from `polski.sty`: a short-hands for `\discretionary` that’ll stick to the word not spoiling its hyphenability and that’ll won’t allow a linebreak just before nor just after themselves. The `\discretionary` T<sub>E</sub>X primitive has three arguments: #1 ‘before break’, #2 ‘after break’, #3 ‘without break’, remember?

```

\discre 209 \def\discre#1#2#3{\kernosp\discretionary{#1}{#2}{#3}%
        \penalty10000\hskiposp\relax}
\discret 210 \def\discret#1{\kernosp\discretionary{#1}{#1}{#1}\penalty10000%
        \hskiposp\relax}

```

<sup>1</sup> Think of the drags that transform a very nice but rather standard ‘auntie’ (‘Tante’ in Deutsch) into a most adorable Queen ;-).

A tiny little macro that acts like \- outside the math mode and has its original meaning inside math.

```
211 \def\:{\ifmmode\afterfi{\mskip\medmuskip}\else\afterfi{\discret{%
    }}\fi}
\vs 212 \newcommand*\{vs}{\discre{\visiblespace}}{\visiblespace}}
```

Then we define a macro that makes the spaces visible even if used in an argument (i.e., in a situation where re\catcodeing has no effect).

```
\printspaces 213 \def\printspaces#1{\let~=\vs\let\ =\vs\gm@pswords#1\@@nil}}
\gm@pswords 214 \def\gm@pswords#1\#2\@@nil{%
215 \ifx\relax#1\relax\else#1\fi
216 \ifx\relax#2\relax\else\vs\penalty\hyphenpenalty\gm@pswords#2\@@nil%
    \fi}% note that in the recursive call of \gm@pswords the argument string is
    not extended with a guardian space: it has been already by \printspaces.
```

```
\sfname 217 \DeclareRobustCommand*\sfname[1]{\textsf{\printspaces{#1}}}
```

```
\gm@discretionaryslash 218 \def\gm@discretionaryslash{\discre{/}{\hbox{}}{}}}% the second pseudo-
    argument nonempty to get \hyphenpenalty not \exhyphenpenalty.
```

```
\file 219 \DeclareRobustCommand*\file[1]{\gm@printslashes#1/%
    \gm@printslashes}
```

```
\gm@printslashes 220 \def\gm@printslashes#1/#2\gm@printslashes{%
221 \sfname{#1}%
222 \ifx\gm@printslashes#2\gm@printslashes
223 \else
224 \textsf{\gm@discretionaryslash}%
225 \afterfi{\gm@printslashes#2\gm@printslashes}\fi}
```

it allows the spaces in the filenames (and prints them as `\`).

The below macro I use to format the packages' names.

```
\pk 226 \DeclareRobustCommand*\{pk}[1]{\textsf{\textup{#1}}}
```

Some (if not all) of the below macros are copied from doc and/or ltxdoc.

A macro for printing control sequences in arguments of a macro. Robust to avoid writing an explicit \ into a file. It calls \ttfamily not \tt to be usable in headings which are boldface sometimes.

```
\cs 227 \DeclareRobustCommand*\{cs}[2][\backslash]{\%
\ - 228 \def\{-{\discretionary{\rmfamily-}}{}{}}}%
229 \def\{\{\char`\}\}\def\{\{\char`\}\}\ttfamily\#1#2}}
```

```
\env 230 \DeclareRobustCommand*\{env}[1]{\cs[]{#1}}
```

And for the special sequences like `^A`:

```
231 \foone{\@makeother\^}
\hathat 232 {\DeclareRobustCommand*\hathat[1]{\cs[^^]{#1}}}
```

And one for encouraging linebreaks e.g., before long verbatim words.

```
\possfil 233 \newcommand*\possfil{\hfil\penalty1000\hfilneg}
```

The five macros below are taken from the ltxdoc.dtx.

"\cmd{\foo} Prints \foo verbatim. It may be used inside moving arguments. \cs{foo} also prints \foo, for those who prefer that syntax. (This second form may even be used when \foo is \outer)."

```
\cmd 234 \def\cmd#1{\cs{\@xa\cmd@to@cs\string#1}}
```

```

\cmd@to@cs 235 \def\cmd@to@cs#1#2{\char\number`#2\relax}
           \marg{text} prints {\text}, ‘mandatory argument’.

\marg 236 \def\marg#1{\tfamily\char`\{\meta{#1}\tfamily\char`\}}
           \oarg{text} prints [{text}], ‘optional argument’. Also \oarg[text] does that.

\oarg 237 \def\oarg{\@ifnextchar[\@oargsq\@oarg}
\@oarg 238 \def\@oarg#1{\tfamily\meta{#1}\tfamily}}
\@oargsq 239 \def\@oargsq[#1]{\@oarg{#1}}
           \parg{te,xt} prints (te,xt), ‘picture mode argument’.

\parg 240 \def\parg{\@ifnextchar(\@pargp\@parg}
\@parg 241 \def\@parg#1{\tfamily\meta{#1}\tfamily}}
\@pargp 242 \def\@pargp(#1){\@parg{#1}}

           But we can have all three in one command.

243 \AtBeginDocument{%
\arg 244 \let\math@arg\arg
\arg 245 \def\arg{\ifmmode\math@arg\else\afterfi{%
246 \ifnextchar[%
247 \@oargsq{\@ifnextchar(
248 \@pargp\marg}}\fi}%
249 }

```

## Storing and Restoring the Meanings of CSs

First a Boolean switch of globalness of assignments and its verifier.

```

\ifgmu@SMglobal 250 \newif\ifgmu@SMglobal
\SMglobal 251 \def\SMglobal{\gmu@SMglobaltrue}

```

The subsequent commands are defined in such a way that you can ‘prefix’ them with `\SMglobal` to get global (re)storing.

A command to store the current meaning of a CS in another macro to temporarily redefine the CS and be able to set its original meaning back (when grouping is not recommended):

```

\StoreMacro 252 \def\StoreMacro{%
253 \bgroup\makeatletter\@ifstar\egStore@MacroSt\egStore@Macro}

```

The unstarred version takes a cs and the starred version a text, which is intended for special control sequences. For storing environments there is a special command in line 316.

```

\egStore@Macro 254 \long\def\egStore@Macro#1{\egroup\Store@Macro{#1}}
\egStore@MacroSt 255 \long\def\egStore@MacroSt#1{\egroup\Store@MacroSt{#1}}
\Store@Macro 256 \long\def\Store@Macro#1{%
257 \escapechar92
258 \ifgmu@SMglobal\afterfi\global\fi
259 \@xa\let\csname_/gmu/store/string#1\endcsname#1%
260 \global\gmu@SMglobalfalse}
\Store@MacroSt 261 \long\def\Store@MacroSt#1{%
262 \edef\gmu@smttempa{%
263 \ifgmu@SMglobal\global\fi

```

```

264 \@nx\let\@xa\@nx\csname/gmu/store\bslash#1\endcsname% we add back-
      slash because to ensure compatibility between \(\Re)StoreMacro and
      \(\Re)StoreMacro*, that is. to allow writing e.g. \StoreMacro\kitten
      and then \RestoreMacro*{kitten} to restore the meaning of \kitten.
265 \@xa\@nx\csname#1\endcsname}
266 \gmu@smtempa
267 \global\gmu@SMglobalfalse}% we wish the globality to be just once.

```

We make the `\StoreMacro` command a three-step to allow usage of the most inner macro also in the next command.

The starred version, `\StoreMacro*` works with csnames (without the backslash). It's first used to store the meanings of robust commands, when you may need to store not only `\foo`, but also `\csname foo \endcsname`.

The next command iterates over a list of CSs and stores each of them. The CS may be separated with commas but they don't have to.

```

\StoreMacros 268 \long\def\StoreMacros{\bgroup\makeatletter\Store@Macros}
\Store@Macros 269 \long\def\Store@Macros#1{\egroup
270 \gmu@setsetSMglobal
271 \let\gml@storeCS\Store@Macro
272 \gml@storemacros#1.}
\gmu@setsetSMglobal 273 \def\gmu@setsetSMglobal{%
274 \ifgmu@SMglobal
275 \let\gmu@setSMglobal\gmu@SMglobaltrue
276 \else
277 \let\gmu@setSMglobal\gmu@SMglobalfalse
278 \fi}

```

And the inner iterating macro:

```

\gml@storemacros 279 \long\def\gml@storemacros#1{%
\gmu@reserveda 280 \def\gmu@reserveda{\@nx#1}% My TEX Guru's trick to deal with \fi and such,
      i.e., to hide #1 from TEX when it is processing a test's branch without expand-
      ing.
281 \if\gmu@reserveda.% a dot finishes storing.
282 \global\gmu@SMglobalfalse
283 \else
284 \if\gmu@reserveda,% The list this macro is put before may contain commas
      and that's O.K., we just continue the work.
285 \afterfifi\gml@storemacros
286 \else% what is else this shall be stored.
287 \gml@storeCS{#1}% we use a particular CS to may \let it both to the storing
      macro as above and to the restoring one as below.
288 \afterfifi{\gmu@setSMglobal\gml@storemacros}%
289 \fi
290 \fi}

```

And for the restoring

```

\RestoreMacro 291 \def\RestoreMacro{%
292 \bgroup\makeatletter\@ifstar\egRestore@MacroSt\egRestore@Macro}
\egRestore@Macro 293 \long\def\egRestore@Macro#1{\egroup\Restore@Macro{#1}}
\egRestore@MacroSt 294 \long\def\egRestore@MacroSt#1{\egroup\Restore@MacroSt{#1}}
\Restore@Macro 295 \long\def\Restore@Macro#1{%
296 \escapecharg2

```

```

297 \ifgmu@SMglobal\afterfi\global\fi
298 \@xa\let\@xa#1\csname_/gmu/store/string#1\endcsname
299 \global\gmu@SMglobalfalse}
\Restore@MacroSt 300 \long\def\Restore@MacroSt#1{%
301 \edef\gmu@smtempa{%
302 \ifgmu@SMglobal\global\fi
303 \@nx\let\@xa\@nx\csname#1\endcsname
304 \@xa\@nx\csname/gmu/store/bslash#1\endcsname}% cf. the commentary
in line 264.
305 \gmu@smtempa
306 \global\gmu@SMglobalfalse}
\RestoreMacros 307 \long\def\RestoreMacros{\bgroup\makeatletter\Restore@Macros}
\Restore@Macros 308 \long\def\Restore@Macros#1{\egroup
309 \gmu@setsetSMglobal
310 \let\gml@storeCS\Restore@Macro% we direct the core CS towards restoring
and call the same iterating macro as in line 272.
311 \gml@storemacros#1.}

```

As you see, the `\RestoreMacros` command uses the same iterating macro inside, it only changes the meaning of the core macro.

And to restore *and* use immediately:

```

\StoredMacro 312 \def\StoredMacro{\bgroup\makeatletter\Stored@Macro}
\Stored@Macro 313 \long\def\Stored@Macro#1{\egroup\Restore@Macro#1#1}

```

To be able to call a stored cs without restoring it.

```

\storedcsname 314 \def\storedcsname#1{%
315 \csname_/gmu/store/bslash#1\endcsname}
2008/08/03 we need to store also an environment.

```

```

\StoreEnvironment 316 \def\StoreEnvironment#1{%
317 \StoreMacro*{#1}\StoreMacro*{end#1}}
\RestoreEnvironment 318 \def\RestoreEnvironment#1{%
319 \RestoreMacro*{#1}\RestoreMacro*{end#1}}

```

It happened (see the definition of `\@docinclude` in `gmdoc.sty`) that I needed to `\relax` a bunch of macros and restore them after some time. Because the macros were rather numerous and I wanted the code more readable, I wanted to `\do` them. After a proper defining of `\do` of course. So here is this proper definition of `\do`, provided as a macro (a declaration).

```

\StoringAndRelaxingDo 320 \long\def\StoringAndRelaxingDo{%
321 \gmu@SMdo@setscope
322 \long\def\do##1{%
323 \gmu@SMdo@scope
324 \@xa\let\csname_/gmu/store/string##1\endcsname##1%
325 \gmu@SMdo@scope\let##1\relax}}
\gmu@SMdo@setscope 326 \def\gmu@SMdo@setscope{%
327 \ifgmu@SMglobal\let\gmu@SMdo@scope\global
328 \else\let\gmu@SMdo@scope\relax
329 \fi
330 \global\gmu@SMglobalfalse}

```

And here is the counter-definition for restore.

```

\RestoringDo 331 \long\def\RestoringDo{%
332   \gmu@SMdo@setscope
333   \long\def\do##1{%
334     \gmu@SMdo@scope
335     \@xa\let\@xa##1\csname_/gmu/store/string##1\endcsname}}

```

Note that both `\StoringAndRelaxingDo` and `\RestoringDo` are sensitive to the `\SMglobal` ‘prefix’.

And to store a cs as explicitly named cs, i.e. to `\let` one csname another (`\n@melet` not `\@namelet` because the latter is defined in Till Tantau’s beamer class another way) (both arguments should be text):

```

\n@melet 336 \def\n@melet#1#2{%
337   \edef\gmu@nl@reserveda{%
338     \let\@xa\@nx\csname#1\endcsname
339     \@xa\@nx\csname#2\endcsname}%
340   \gmu@nl@reserveda}

```

The `\global` prefix doesn’t work with `\n@melet` so we define the alternative.

```

\gn@melet 341 \def\gn@melet#1#2{%
342   \edef\gmu@nl@reserveda{%
343     \global\let\@xa\@nx\csname#1\endcsname
344     \@xa\@nx\csname#2\endcsname}%
345   \gmu@nl@reserveda}

```

## Not only preamble!

Let’s remove some commands from the list to erase at begin document! Primarily that list was intended to save memory not to forbid anything. Nowadays, when memory is cheap, the list of only-preamble commands should be rethought imo.

```

\not@onlypreamble 346 \newcommand\not@onlypreamble[1]{%
347   \def\do##1{\ifx##1\else\@nx\do\@nx##1\fi}%
348   \xdef\@preamblecmds{\@preamblecmds}}
349 \not@onlypreamble\@preamblecmds
350 \not@onlypreamble\ifpackageloaded
351 \not@onlypreamble\ifclassloaded
352 \not@onlypreamble\ifl@aded
353 \not@onlypreamble\@pkgextension

```

And let’s make the message of only preamble command’s forbidden use informative a bit:

```

\gm@notprerr 354 \def\gm@notprerr{can be used only in preamble(\on@line)}
355 \AtBeginDocument{%
356   \def\do#1{\@nx\do\@nx#1}%
357   \edef\@preamblecmds{%
\@nx 358     \def\@nx\do##1{%
\@nx 359       \def##1{\@nx\PackageError{gmutils/LaTeX}%
360         {\@nx\string##1\@nx\gm@notprerr}\@nx\@eha}}%
361     \@preamblecmds}}

```

A subtle error raises: the L<sup>A</sup>T<sub>E</sub>X standard `\@onlypreamble` and what `\document` does with `\@preamblecmds` makes any two of ‘only preamble’ cs’s `\ifx`-identical inside document. And my change makes any two cs’s `\ifx`-different. The first it causes



a problem is `\nocite` that checks `\ifx\@onlypreamble\document`. So hoping this is a rare problem, we circumvent in with

```
\nocite 362 \def\nocite#1{%
363   \@bsphack{\setboxo=\hbox{\cite{#1}}}\@esphack}
```

### Third Person Pronouns

Is a reader of my documentations ‘she’ or ‘he’ and does it make a difference?

Not to favour any gender in the personal pronouns, define commands that’ll print alternately masculine and feminine pronoun of third person. By ‘any’ I mean not only typically masculine and typically feminine but the entire amazingly rich variety of people’s genders, *including* those who do not describe themselves as ‘man’ or ‘woman’.

One may say two pronouns is far too little to cover this variety but I could point Ursula’s K. LeGuin’s *The Left Hand Of Darkness* as another acceptable answer. In that moody and moderate SF novel the androgynous persons are usually referred to as ‘mister’, ‘sir’ or ‘he’: the meaning of reference is extended. Such an extension also my automatic pronouns do suggest. It’s *not* political correctness, it’s just respect to people’s diversity.

```
gm@PronounGender 364 \newcounter{gm@PronounGender}
\gm@atppron 365 \newcommand*\gm@atppron[2]{%
366   \stepcounter{gm@PronounGender}% remember \stepcounter is global.
367   \ifodd\value{gm@PronounGender}#1\else#2\fi}
\heshe 368 \newcommand*\heshe{\gm@atppron{he}{she}}
\hisher 369 \newcommand*\hisher{\gm@atppron{his}{her}}
\himher 370 \newcommand*\himher{\gm@atppron{him}{her}}
\hishers 371 \newcommand*\hishers{\gm@atppron{his}{hers}}
\HeShe 372 \newcommand*\HeShe{\gm@atppron{He}{She}}
\HisHer 373 \newcommand*\HisHer{\gm@atppron{His}{Her}}
\HimHer 374 \newcommand*\HimHer{\gm@atppron{Him}{Her}}
\HisHers 375 \newcommand*\HisHers{\gm@atppron{His}{Hers}}
```

### To Save Precious Count Registers

It’s a contribution to T<sub>E</sub>X’s ecology ;-). You can use as many CSs as you wish and you may use only 256 count registers (although in  $\epsilon$ -T<sub>E</sub>X there are 2<sup>16</sup> count registers, which makes the following a bit obsolete).

```
\nummacro 376 \newcommand*\nummacro[1]{\gdef#1{o}}
\stepnummacro 377 \newcommand*\stepnummacro[1]{%
378   \@tempcnta=#1\relax
379   \advance\@tempcnta by 1\relax
380   \xdef#1{\the\@tempcnta}}% Because of some mysterious reasons explicit \counto
interfered with page numbering when used in \gmd@evpaddonce in gm-
doc.
\addtonummacro 381 \newcommand*\addtonummacro[2]{%
382   \counto=#1\relax
383   \advance\countoby#2\relax
384   \xdef#1{\the\count\z@}}
```

Need an explanation? The `\nummacro` declaration defines its argument (that should be a CS) as `{o}` which is analogous to `\newcount` declaration but doesn’t use up any count register.

Then you may use this numeric macro as something between TeX's count CS and L<sup>A</sup>T<sub>E</sub>X's counter. The macros `\stepnummacro` and `\addtonummacro` are analogous to L<sup>A</sup>T<sub>E</sub>X's `\stepcounter` and `\addtocounter` respectively: `\stepnummacro` advances the number stored in its argument by 1 and `\addtonummacro` advances it by the second argument. As the L<sup>A</sup>T<sub>E</sub>X's analogoi, they have the global effect (the effect of global warming ;-)).

So far I've used only `\nummacro` and `\stepnummacro`. Notify me if you use them and whether you need sth. more, `\multiplynummacro` e.g.

## Improvements to mwcls Sectioning Commands

That is, 'Expe-ri-mente'<sup>2</sup> mit MW sectioning & `\refstepcounter` to improve mwcls's cooperation with hyperref. They shouldn't make any harm if another class (non-mwcls) is loaded.

We `\refstep` sectioning counters even if the sectionings are not numbered, because otherwise

1. pdfTeX cried of multiply defined `\labels`,
2. e.g. in a table of contents the hyperlink `<rozdzia\l\ Kwiaty polskie>` linked not to the chapter's heading but to the last-before-it change of `\ref`.

385 `\AtBeginDocument{% because we don't know when exactly hyperref is loaded and maybe after this package.`

NoNumSecs 386 `\@ifpackageloaded{hyperref}{\newcounter{NoNumSecs}%`

387 `\setcounter{NoNumSecs}{617}% to make \refing to an unnumbered section visible (and funny?).`

388 `\def\gm@hyperrefstepcounter{\refstepcounter{NoNumSecs}}%`

389 `\DeclareRobustCommand*\gm@targetheading[1]{%`

390 `\hypertarget{#1}{#1}}}% end of then`

391 `{\def\gm@hyperrefstepcounter{}}%`

392 `\def\gm@targetheading#1{#1}}}% end of else`

393 `}% of \AtBeginDocument`

Auxiliary macros for the kernel sectioning macro:

394 `\def\gm@dontnumbersectionsoutofmainmatter{%`

395 `\if@mainmatter\else\HeadingNumberedfalse\fi}`

396 `\def\gm@clearpagesduetoopenright{%`

397 `\if@openright\cleardoublepage\else\clearpage\fi}`

To avoid `\defing` of `\mw@sectionxx` if it's undefined, we redefine `\def` to gobble the definition and restore the original meaning of itself.

Why shouldn't we change the ontological status of `\mw@sectionxx` (not define if undefined)? Because some macros (in gmdocc e.g.) check it to learn whether they are in an mwcls or not.

But let's make a shorthand for this test since we'll use it three times in this package and maybe also somewhere else.

\@ifnotmw 398 `\long\def\@ifnotmw#1#2{\@ifundefined{mw@sectionxx}{#1}{#2}}`

399 `\let\gmu@def\def`

\@ifnotmw 400 `\@ifnotmw{%`

\gmu@def 401 `\StoreMacro\gmu@def\def\gmu@def#1#2{\RestoreMacro\gmu@def}}{}`

I know it may be of bad taste (to write such a way *here*) but I feel so lonely and am in an alien state of mind after 3 hour sleep last night and, worst of all, listening to sir Edward Elgar's flamboyant Symphonies d'Art Nouveau.

<sup>2</sup> A. Berg, *Wozzeck*.

A *decent* person would just wrap the following definition in `\@ifundefined's Else`. But look, the definition is so long and I feel so lonely etc. So, I define `\def` (for some people there's nothing sacred) to be a macro with two parameters, first of which is delimited by digit 4 (the last token of `\mw@sectionxx's` parameter string) and the latter is undelimited which means it'll be the body of the definition. Such defined `\def` does nothing else but restores its primitive meaning by the way sending its arguments to the Gobbled Tokens' Paradise. Luckily, `\RestoreMacro` contains `\let not \def`.

The kernel of MW's sectioning commands:

```
402 \gmu@def\mw@sectionxx#1#2[#3]#4{%
403   \edef\mw@HeadingLevel{\csname_#1@level\endcsname
404     \space}% space delimits level number!
405   \ifHeadingNumbered
406     \ifnum_\mw@HeadingLevel>\c@secnumdepth_
         \HeadingNumberedfalse_\fi
```

line below is in `ifundefined` to make it work in classes other than `mwbk`

```
407   \@ifundefined{if@mainmatter}{\}%
         \gm@dontnumbersectionsoutofmainmatter}
408 \fi
%   \ifHeadingNumbered
%     \refstepcounter{#1}%
%     \protected@edef\HeadingNumber{\csname
%       the#1\endcsname\relax}%
%   \else
%     \let\HeadingNumber\@empty
%   \fi
\HeadingRHeadText 409 \def\HeadingRHeadText{#2}%
\HeadingTOCText    410 \def\HeadingTOCText{#3}%
\HeadingText       411 \def\HeadingText{#4}%
\mw@HeadingType    412 \def\mw@HeadingType{#1}%
413 \if\mw@HeadingBreakBefore
414   \if@specialpage\else\thispagestyle{closing}\fi
415   \@ifundefined{if@openright}{\}\gm@clearpagesduetoopenright}%
416   \if\mw@HeadingBreakAfter
417     \thispagestyle{blank}\else
418     \thispagestyle{opening}\fi
419   \global\@topnum\z@
420 \fi% of \if\mw@HeadingBreakBefore
```

placement of `\refstep` suggested by me (GM)

```
421   \ifHeadingNumbered
422     \refstepcounter{#1}%
423     \protected@edef\HeadingNumber{\csname_#1\endcsname\relax}%
424   \else
425     \let\HeadingNumber\@empty
426     \gm@hyperrefstepcounter
427   \fi% of \ifHeadingNumbered
428 \if\mw@HeadingRunIn
429   \mw@runinheading
430 \else
431   \if\mw@HeadingWholeWidth
```

```

432     \if@twocolumn
433     \if\mw@HeadingBreakAfter
434     \onecolumn
435     \mw@normalheading
436     \pagebreak\relax
437     \if@twoside
438     \null
439     \thispagestyle{blank}%
440     \newpage
441     \fi% of \if@twoside
442     \twocolumn
443     \else
444     \@topnewpage[\mw@normalheading]%
445     \fi% of \if\mw@HeadingBreakAfter
446     \else
447     \mw@normalheading
448     \if\mw@HeadingBreakAfter\pagebreak\relax\fi
449     \fi% of \if@twocolumn
450     \else
451     \mw@normalheading
452     \if\mw@HeadingBreakAfter\pagebreak\relax\fi
453     \fi% of \if\mw@HeadingWholeWidth
454     \fi% of \if\mw@HeadingRunIn
455     }

```

### An improvement of MW's \SetSectionFormatting

A version of MW's \SetSectionFormatting that lets to leave some settings unchanged by leaving the respective argument empty ({} or []).

Notice: If we adjust this command for new version of MWCLS, we should name it \SetSectionFormatting and add issuing errors if the inner macros are undefined.

- #1 (optional) the flags, e.g. breakbefore, breakafter;
- #2 the sectioning name, e.g. chapter, part;
- #3 preskip;
- #4 heading type;
- #5 postskip

```

456 \relaxen\SetSectionFormatting
\SetSectionFormatting
457 \newcommand*\SetSectionFormatting[5][\empty]{%
458   \ifx\empty#1\relax\else% empty (not \empty!) #1 also launches \else.
459   \def\mw@HeadingRunIn{10}\def\mw@HeadingBreakBefore{10}%
460   \def\mw@HeadingBreakAfter{10}\def\mw@HeadingWholeWidth{10}%
461   \@ifempty{#1}{}{\mw@processflags#1,\relax}% If #1 is omitted, the flags
      are left unchanged. If #1 is given, even as [], the flags are first cleared and
      then processed again.
462   \fi
463   \@ifundefined{#2}{\@namedef{#2}{\mw@section{#2}}}{}%
464   \mw@secdef{#2}{@preskip}{#3}{2\oblig.}%
465   \mw@secdef{#2}{@head}{#4}{3\oblig.}%
466   \mw@secdef{#2}{@postskip}{#5}{4\oblig.}%
467   \ifx\empty#1\relax
468     \mw@secundef{#2@flags}{1\optional)}%
469   \else\mw@setflags{#2}%

```

```

470 \fi}
\mw@secdef 471 \def\mw@secdef#1#2#3#4{% #1 the heading name,
              % #2 the command distinctior,
              % #3 the meaning,
              % #4 the number of argument to error message.
472 \@ifempty{#3}
473   {\mw@secundef{#1#2}{#4}}
474   {\@namedef{#1#2}{#3}}}
\mw@secundef 475 \def\mw@secundef#1#2{%
476   \@ifundefined{#1}{%
477     \ClassError{mwcls/gm}{%
478       command_\bslash#1_undefined_\MessageBreak
479       after_\bslash_SetSectionFormatting!!!\MessageBreak}{%
480       Provide_the_#2_argument_of_\bslash_
         SetSectionFormatting.}}{}}

```

First argument is a sectioning command (wo. \) and second the stuff to be added at the beginning of the heading declarations.

```

\addtoheading 481 \def\addtoheading#1#2{%
482   \n@melet{gmu@reserveda}{#1@head}%
483   \toks\z@=\@xa{gmu@reserveda}%
484   \toks\tw@={#2}%
485   \edef\gmu@reserveda{\the\toks\tw@\the\toks\z@}%
486   \n@melet{#1@head}{gmu@reserveda}%
487 }

```

## Negative \addvspace

When two sectioning commands appear one after another (we may assume that this occurs only when a lower section appears immediately after higher), we prefer to put the *smaller* vertical space not the larger, that is, the preskip of the lower sectioning not the postskip of the higher.

For that purpose we modify the very inner macros of MWCLS to introduce a check whether the previous vertical space equals the postskip of the section one level higher.

```

488 \@ifnotmw{}{% We proceed only in MWCLS

```

The information that we are just after a heading will be stored in the \gmu@prevsec macro: any heading will define it as the section name and \everypar (any normal text) will clear it.

```

\@afterheading 489 \def\@afterheading{%
490   \@nobreaktrue
491   \xdef\gmu@prevsec{\mw@HeadingType}% added now
492   \everypar{%
493     \grelaxen\gmu@prevsec% added now. All the rest is original LATEX.
494     \if@nobreak
495     \@nobreakfalse
496     \clubpenalty_\@M
497     \if@afterindent_\else
498     {\setbox\z@\lastbox}%
499     \fi
500     \else
501     \clubpenalty_\@clubpenalty

```

```

502 \everypar{}%
503 \fi}}

```

If we are (with the current heading) just after another heading (one level lower I suppose), then we add the less of the higher header's post-skip and the lower header pre-skip or, if defined, the two-header-skip. (We put the macro defined below just before `\addvspace` in MWCLS inner macros.)

```

\gmu@checkaftersec 504 \def\gmu@checkaftersec{%
505   \@ifundefined{gmu@prevsec}{-}{%
506     \ifgmu@postsec% an additional switch that is true by default but may be
                        turned into an \ifdim in special cases, see line 534.
507     {\@xa\mw@getflags\@xa{\gmu@prevsec}%
508       \glet\gmu@reserveda\mw@HeadingBreakAfter}%
\gmu@reserveda 509     \if\mw@HeadingBreakBefore\def\gmu@reserveda{11}\fi% if the current
                        heading inserts page break before itself, all the play with vskips is irrele-
                        vant.
510     \if\gmu@reserveda\else
511       \penalty10000\relax
512       \skip\z@=\csname\gmu@prevsec_\@postskip\endcsname\relax
513       \skip\tw@=\csname\mw@HeadingType_\@preskip\endcsname\relax
514       \@ifundefined{\mw@HeadingType_\@twoheadskip}{
515         \ifdim\skip\z@>\skip\tw@
516         \vskip-\skip\z@% we strip off the post-skip of previous header if it's bigger
                        than current pre-skip
517       \else
518         \vskip-\skip\tw@% we strip off the current pre-skip otherwise
519       \fi}{% But if the two-header-skip is defined, we put it
520       \penalty10000
521       \vskip-\skip\z@
522       \penalty10000
523       \vskip-\skip\tw@
524       \penalty10000
525       \vskip\csname\mw@HeadingType_\@twoheadskip\endcsname
526       \relax}%
527     \penalty10000
528     \hrule_\height\z@\relax% to hide the last (un)skip before subsequent \addvspaces.
529     \penalty10000
530     \fi
531     \fi
532   }% of \@ifundefined{gmu@prevsec} 'else'
533 }% of \def\gmu@checkaftersec

\ParanoidPostsec 534 \def\ParanoidPostsec{% this version of \ifgmu@postsec is intended for the spe-
                        cial case of sections may contain no normal text, as while gmdocing.

\ifgmu@postsec 535 \def\ifgmu@postsec{% note this macro expands to an open \if.
536   \skip\z@=\csname\gmu@prevsec_\@postskip\endcsname\relax
537   \ifdim\lastskip=\skip\z@\relax% we play with the vskips only if the last
                        skip is the previous heading's postskip (a counter-example I met while
                        gmdocing).
538   }}
539 \let\ifgmu@postsec\iftrue

\gmu@getaddvs 540 \def\gmu@getaddvs#1\addvspace#2\gmu@getaddvs{%
541   \toks\z@={#1}

```

```
542 \toks\tw@={#2}}
```

And the modification of the inner macros at last:

```
\gmu@setheading 543 \def\gmu@setheading#1{%
544 \xa\gmu@getaddvs#1\gmu@getaddvs
545 \edef#1{%
546 \the\toks\z@\@nx\gmu@checkaftersec
547 \@nx\addvspace\the\toks\tw@}}
548 \gmu@setheading\mw@normalheading
549 \gmu@setheading\mw@runinheading
\SetTwoheadSkip 550 \def\SetTwoheadSkip#1#2{\@namedef{#1@twoheadskip}{#2}}
551 }% of \@ifnotmw
```

### My heading setup for mwcls

The setup of heading skips was tested in ‘real’ typesetting, for money that is. The skips are designed for 11/13 pt leading and together with my version of mw11.clo option file for mwcls make the headings (except paragraph and subparagraph) consist of an integer number of lines. The name of the declaration comes from my employer, “Wiedza Powszechna” Editions.

```
552 \@ifnotmw{}{% We define this declaration only when in mwcls.
\WPheadings 553 \def\WPheadings{%
554 \SetSectionFormatting[breakbefore,wholewidth]
555 {part}{\z@\@plus1fill}{\z@\@plus3fill}%
556 \@ifundefined{chapter}{}%
557 \SetSectionFormatting[breakbefore,wholewidth]
558 {chapter}
559 {66\p@}% {67\p@} for Adventor/Schola 0,95.
560 {\FormatHangHeading{\LARGE}}
561 {27\p@\@plus0,2\p@\@minus1\p@}%
562 }%
563 \SetTwoheadSkip{section}{27\p@\@plus0,5\p@}%
564 \SetSectionFormatting{section}
565 {24\p@\@plus0,5\p@\@minus5\p@}%
566 {\FormatHangHeading{\Large}}
567 {10\p@\@plus0,5\p@}% ed. Krajewska of “Wiedza Powszechna”, as we un-
derstand her, wants the skip between a heading and text to be rigid.
568 \SetTwoheadSkip{subsection}{11\p@\@plus0,5\p@\@minus1\p@}%
569 \SetSectionFormatting{subsection}
570 {19\p@\@plus0,4\p@\@minus6\p@}
571 {\FormatHangHeading{\large}}% 12/14 pt
572 {6\p@\@plus0,3\p@}% after-skip 6 pt due to p.12, not to squeeze the before-
skip too much.
573 \SetTwoheadSkip{subsubsection}{10\p@\@plus1,75\p@\@minus1\p@}%
574 \SetSectionFormatting{subsubsection}
575 {10\p@\@plus0,2\p@\@minus1\p@}
576 {\FormatHangHeading{\normalsize}}
577 {3\p@\@plus0,1\p@}% those little skips should be smaller than you calcu-
late out of a geometric progression, because the interline skip enlarges
them.
578 \SetSectionFormatting[runin]{paragraph}
579 {7\p@\@plus0,15\p@\@minus1\p@}}
```

```

580      {\FormatRunInHeading{\normalsize}}
581      {2\p@}%
582  \SetSectionFormatting[runin]{subparagraph}
583      {4\p@\@plus1\p@\@minus0,5\p@}
584      {\FormatRunInHeading{\normalsize}}
585      {\z@}%
586  }% of \WPheadings
587  }% of \@ifnotmw

```

## Compatibilising Standard and mwcls Sectionings

If you use Marcin Woliński’s document classes (mwcls), you might have met their little queerness: the sectioning commands take two optional arguments instead of standard one. It’s reasonable since one may wish one text to be put into the running head, another to the toc and yet else to the page. But the order of optionalities causes an incompatibility with the standard classes: MW section’s first optional argument goes to the running head not to toc and if you’ve got a source file written with the standard classes in mind and use the first (and only) optional argument, the effect with mwcls would be different if not error.

Therefore I counter-assign the commands and arguments to reverse the order of optional arguments for sectioning commands when mwcls are in use and reverse, to make mwcls-like sectioning optionals usable in the standard classes.

With the following in force, you may both in the standard classes and in mwcls give a sectioning command one or two optional arguments (and mandatory the last, of course). If you give just one optional, it goes to the running head and to toc as in scls (which is unlike in mwcls). If you give two optionals, the first goes to the running head and the other to toc (like in mwcls and unlike in scls).

(In both cases the mandatory last argument goes only to the page.)

What more is unlike in scls, it’s that even with them the starred versions of sectioning commands allow optionals (but they still send them to the Gobbled Tokens’ Paradise).

(In mwcls, the only difference between starred and non-starred sec commands is (not) numbering the titles, both versions make a contents line and a mark and that’s not changed with my redefinitions.)

```

588 \@ifnotmw{% we are not in mwcls and want to handle mwcls-like sectionings i.e.,
      those written with two optionals.
\gm@secini 589   \def\gm@secini{gm@1a}%
\gm@secxx 590   \def\gm@secxx#1#2[#3]#4{%
591     \ifx\gm@secstar\@empty
592       \n@melet{gm@true@#1mark}{#1mark}% a little trick to allow a special ver-
      sion of the heading just to the running head.
593       \@namedef{#1mark}##1{% we redefine \<sec>mark to gobble its argument
      and to launch the stored true marking command on the appropriate
      argument.
594       \csname_\gm@true@#1mark\endcsname{#2}%
595       \n@melet{#1mark}{gm@true@#1mark}% after we’ve done what we wanted
      we restore original \#1mark.
596     }%
\gm@secstar 597   \def\gm@secstar{[#3]}% if \gm@secstar is empty, which means the sec-
      tioning command was written starless, we pass the ‘true’ sectioning
      command #3 as the optional argument. Otherwise the sectioning com-
      mand was written with star so the ‘true’ s.c. takes no optional.

```



```

598     \fi
599     \@xa\@xa\csize\gm@secini#1\endcsname
600     \gm@secstar{#4}}%
601 }{% we are in mwcls and want to reverse MW's optionals order i.e., if there's just one
    optional, it should go both to toc and to running head.
\gm@secini 602     \def\gm@secini{\gm@mw}%
603     \let\gm@secmarkh\@gobble% in mwcls there's no need to make tricks for special
    version to running headings.
\gm@secxx 604     \def\gm@secxx#1#2[#3]#4{%
605         \@xa\@xa\csize\gm@secini#1\endcsname
606         \gm@secstar[#2][#3]{#4}}%
607 }

\gm@sec 608 \def\gm@sec#1{\@dblarg{\gm@secx{#1}}}
\gm@secx 609 \def\gm@secx#1[#2]{%
610     \@ifnextchar[{\gm@secxx{#1}{#2}}{\gm@secxx{#1}{#2}[#2]}}% if there's
    only one optional, we double it not the mandatory argument.
\gm@straightensec 611 \def\gm@straightensec#1{% the parameter is for the command's name.
612     \@ifundefined{#1}{\% we don't change the ontological status of the command
    because someone may test it.
613     \n@melet{\gm@secini#1}{#1}%
614     \@namedef{#1}{%
\gm@secstar 615         \@ifstar{\def\gm@secstar{*}\gm@sec{#1}}{%
\gm@secstar 616         \def\gm@secstar{\gm@sec{#1}}}%
617     }%

618     \let\do\gm@straightensec
619     \do{part}\do{chapter}\do{section}\do{subsection}\do{%
    subsubsection}
620     \@ifnotmw{\do{paragraph}}% this 'straightening' of \paragraph with the stan-
    dard article caused the 'TEX capacity exceeded' error. Anyway, who on Earth
    wants paragraph titles in toc or running head?

```

## enumerate\* and itemize\*

We wish the starred version of enumerate to be just numbered paragraphs. But hyperref redefines \item so we should do it a smart way, to set the L<sup>A</sup>T<sub>E</sub>X's list parameters that is.

(Marcin Woliński in mwcls defines those environments slightly different: his item labels are indented, mine are not; his subsequent paragraphs of an item are not indented, mine are.)

```

enumerate* 621 \@namedef{enumerate*}{%
622     \ifnum\@enumdepth>\thr@@
623         \@toodeep
624     \else
625         \advance\@enumdepth\@ne
626         \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
627         \@xa\list\csize\label\@enumctr\endcsname{%
628             \partopsep\topsep\topsep\z@\leftmargin\z@
629             \itemindent\@parindent\advance\itemindent\labelsep
630             \labelwidth\@parindent
631             \advance\labelwidth-\labelsep

```

```

632     \listparindent\@parindent
633     \usecounter\@enumctr
634     \def\makelabel##1{##1\hfil}}%
635     \fi}
636 \@namedef{endenumerate*}{\endlist}

itemize* 637 \@namedef{itemize*}{%
638     \ifnum\@itemdepth>\thr@@
639         \@toodeep
640     \else
641         \advance\@itemdepth\@ne
642         \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
643         \@xa\list\csname\@itemitem\endcsname{%
644             \partopsep\topsep\@topsep\z@\@leftmargin\z@
645             \itemindent\@parindent
646             \labelwidth\@parindent
647             \advance\labelwidth-\labelsep
648             \listparindent\@parindent
649             \def\makelabel##1{##1\hfil\@}}%
650     \fi}
651 \@namedef{enditemize*}{\endlist}

```

## The Logos

We'll modify The L<sup>A</sup>T<sub>E</sub>X logo now to make it fit better to various fonts.

```

652 \let\oldLaTeX\LaTeX
653 \let\oldLaTeXe\LaTeXe

654 \def\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}

\DeclareLogo 655 \newcommand*\DeclareLogo[3][\relax]{%
    #1 is for non-LATEX spelling and will be used in the PD1 encoding (to make pdf book-
    marks);
    #2 is the command, its name will be the PD1 spelling by default,
    #3 is the definition for all the font encodings except PD1.

\gmu@reserveda 656 \ifx\relax#1\def\gmu@reserveda{\@xa\@gobble\string#2}%
657 \else
\gmu@reserveda 658 \def\gmu@reserveda{#1}%
659 \fi
660 \edef\gmu@reserveda{%
\@nx 661 \@nx\DeclareTextCommand\@nx#2{PD1}{\gmu@reserveda}}
662 \gmu@reserveda
663 \DeclareTextCommandDefault#2{#3}%
\DeclareRobustCommand* 664 \DeclareRobustCommand*#2{#3}}% added for XYLATEX
\DeclareLogo 665 \DeclareLogo\LaTeX{%
666     {%
667         L%
668         \setbox\z@\hbox{\check@mathfonts
669             \fontsize\sf@size\z@
670             \math@fontsfalse\selectfont
671             A}%
672         \kern-.57\wd\z@

```

```

673     \sbox\tw@_T%
674     \vbox_to\ht\tw@{\copy\z@_vss}%
675     \kern-.2\wd\z@}% originally -, 15 em for T.
676   {%
677     \ifdim\fontdimen1\font=\z@
678     \else
679       \count\z@=\fontdimen5\font
680       \multiply\count\z@_by_64\relax
681       \divide\count\z@_by\p@
682       \count\tw@=\fontdimen1\font
683       \multiply\count\tw@_by\count\z@
684       \divide\count\tw@_by_64\relax
685       \divide\count\tw@_by\tw@
686       \kern-\the\count\tw@_sp\relax
687     \fi}%
688   \TeX}

\LaTeXe 689 \DeclareLogo\LaTeXe{\mbox{\m@th_\if
690     b\expandafter\@car\f@series\@nil\boldmath\fi
691     \LaTeX\kern.15em2$_{\textstyle\varepsilon}$}}

692 \StoreMacro\LaTeX
693 \StoreMacro*\LaTeX_

‘(L)TeX’ in my opinion better describes what I work with/in than just ‘LATeX’.

\LaTeXpar 694 \DeclareLogo[(La)TeX]{\LaTeXpar}{%
695   {%
696     \setbox\z@\hbox{({}% )
697     \copy\z@
698     \kern-.2\wd\z@_L%
699     \setbox\z@\hbox{\check@mathfonts
700       \fontsize\sf@size\z@
701       \math@fontsfalse\selectfont
702       A}%
703     \kern-.57\wd\z@
704     \sbox\tw@_T%
705     \vbox_to\ht\tw@{\box\z@%
706       \vss}%
707   }%
708   \kern-.07em% originally -, 15 em for T.
709   {% (
710     \sbox\z@)%
711     \kern-.2\wd\z@\copy\z@
712     \kern-.2\wd\z@}\TeX
713 }

```

“Here are a few definitions which can usefully be employed when documenting package files: now we can readily refer to  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ ,  $\text{\BibTeX}$  and  $\text{\SLTeX}$ , as well as the usual  $\text{\TeX}$  and  $\text{\L}^{\text{\A}}\text{\TeX}$ . There’s even a  $\text{\PLAIN}\text{\TeX}$  and a  $\text{\WEB}$ .”

```

714 \@ifundefined{AmSTeX}
\AmSTeX 715 {\def\AmSTeX{\leavevmode\hbox{$\mathcal{A}\kern-.2em%
       \lower.376ex%
716       \hbox{$\mathcal{M}$}\kern-.2em\mathcal{S}$-\TeX}}{\}}

\BibTeX 717 \DeclareLogo\BibTeX{\rmfamily_B\kern-.05em%

```

```

718      \textsc{i{\kern-.025em}b}\kern-.08em% the kern is wrapped in braces
          for my \fakescaps' sake.
719      \TeX}}
\SLiTeX 720 \DeclareLogo\SLiTeX{\rmfamily_S\kern-.06emL\kern-.18em%
          \raise.32ex\hbox
721          {\scshape_i}\kern-.03em\TeX}}
\PlainTeX 722 \DeclareLogo\PlainTeX{\textsc{Plain}\kern2pt\TeX}
\Web 723 \DeclareLogo\Web{\textsc{Web}}

There's also the (L)TEX logo got with the \LaTeXpar macro provided by gmutils. And
here The TEXbook's logo:

\TeXbook 724 \DeclareLogo[The_TeX_book]\TeXbook{\textsl{The_\TeX_book}}
725 \let\TB\TeXbook% TUG Boat uses this.

\TeX 726 \DeclareLogo[e-TeX]\TeX{%
727     \ensuremath{\varepsilon}-\kern-.125em\TeX}% definition sent by Karl Berry
        from TUG Boat itself.

\pdfTeX 728 \DeclareLogo[pdfe-TeX]\pdfTeX{pdf\TeX}
\pdfTeX 729 \DeclareLogo\pdfTeX{pdf\TeX}

730 \@ifundefined{XeTeX}{%
\XeTeX 731 \DeclareLogo\XeTeX{X\kern-.125em\relax
732     \@ifundefined{reflectbox}{%
733         \lower.5ex\hbox{E}\kern-.1667em\relax}{%
734         \lower.5ex\hbox{\reflectbox{E}}\kern-.1667em\relax}%
735     \TeX}}{}

736 \@ifundefined{XeLaTeX}{%
\XeLaTeX 737 \DeclareLogo\XeLaTeX{X\kern-.125em\relax
738     \@ifundefined{reflectbox}{%
739         \lower.5ex\hbox{E}\kern-.1667em\relax}{%
740         \lower.5ex\hbox{\reflectbox{E}}\kern-.1667em\relax}%
741     \LaTeX}}

```

As you see, if T<sub>E</sub>X doesn't recognize \reflectbox (graphics isn't loaded), the first E will not be reversed. This version of the command is intended for non-X<sub>E</sub>T<sub>E</sub>X usage. With X<sub>E</sub>T<sub>E</sub>X, you can load the xltextra package (e.g. with the gmutils \XeTeXthree declaration) and then the reversed E you get as the Unicode Latin Letter Reversed E.

## Expanding turning stuff all into 'other'

While typesetting a unicode file contents with inputenc package I got a trouble with some Unicode sequences that expanded to unexpandable CSs: they could'nt be used within \csname...\endcsname. My T<sub>E</sub>XGuru advised to use \meaning to make all the name 'other'. So—here we are.

Don't use them in \edefs, they would expand not quite.

The next macro is intended to be put in \edefs with a macro argument. The meaning of the macro will be made all 'other' and the words '(long) macro:->' gobbled.

```

\all@other 742 \def\all@other#1{\@xa@gm@gobmacro\meaning#1}

```

The \gm@gobmacro macro above is applied to gobble the \meaning's beginnig, long macro:-> all 'other' that is. Use of it:

```

743 \edef\gmu@reserveda{%

```

```

\@nx 744 \def\@nx\gm@gobmacro##1\@xa\@gobble\string\macro:->{}}
\gm@gobmacro 745 \gm@reserveda

```

In the next two macros' names, 'unex' stands both for not expanding the argument(s) and for disastrously partial unexpandability of the macros themselves.

```

\unex@namedef 746 \long\def\unex@namedef#1#2{%
747 \edef@other\gm@reserveda{#1}%
748 \@xa\long\@xa\def\csname\gm@reserveda\endcsname{#2}}
\unex@nameuse 749 \long\def\unex@nameuse#1{%
750 \edef@other\gm@reserveda{#1}%
751 \csname\gm@reserveda\endcsname}

```

## Brave New World of XeTeX

```

\@ifXeTeX 752 \newcommand\@ifXeTeX[2]{%
753 \ifdefined\XeTeXversion
754 \unless\ifx\XeTeXversion\relax\afterfifi{#1}\else\afterfifi{%
#2}\fi
755 \else\afterfi{#2}\fi}
XeTeXthree 756 \def\XeTeXthree{%
757 \@ifXeTeX{%
758 \@ifpackageloaded{gmverb}{\StoreMacro\verb}{}}%
759 \RequirePackage{xltextra}% since v 0.4 (2008/07/29) this package re-
defines \verb and verbatim*, and quite elegantly provides an option to
suppress the redefinitions, but unfortunately that option excludes also
a nice definition of \xxt@visible space which I fancy.
760 \@ifpackageloaded{gmverb}{\RestoreMacro\verb}{}}%
761 \AtBeginDocument{%
762 \RestoreMacro\LaTeX\RestoreMacro*{LaTeX_}}}% my version of the LATEX
logo has been stored just after defining, in line 693.
763 }{}}

```

The \udigits declaration causes the digits to be typeset uppercase. I provide it since by default I prefer the lowercase (nautical) digits.

```

764 \AtBeginDocument{%
765 \@ifpackageloaded{fontspec}{%
\udigits 766 \DeclareRobustCommand*\udigits{%
767 \addfontfeature{Numbers=Uppercase}}}%
768 }{%
769 \emptify\udigits}}

```

## Fractions

```

\Xedekfrac 770 \def\Xedekfrac{\@ifstar\gm@xedekfracstar\gm@xedekfracplain}

```

(plain) The starless version turns the font feature frac on. (\*) But nor Minion GM neither T<sub>E</sub>X Gyre Pagella doesn't feature the frac font feature properly so, with the starred version of the declaration we use the characters from the font where available (see the \@namedefs below) and the numr and dnom features with the fractional slash otherwise (via \gm@dekfrac). (\*\*) But Latin Modern Sans Serif Quotation doesn't support the numerator and denominator positions so we provide the double star version for it, which takes the char from font if it exist and typesets with lowers and kerns otherwise.

```

\gmu@xedekfracstar 771 \def\gmu@xedekfraccstar{%
\gmu@xfraccdef 772 \def\gmu@xfraccdef##1##2{%
773 \iffontchar\font_##2
774 \@namedef{gmu@xfracc##1}{\char##2_}%
775 \else
776 \n@melet{gmu@xfracc##1}{relax}%
777 \fi}%
\gmu@dekfracc 778 \def\gmu@dekfracc##1/##2{%
779 {\addfontfeature{VerticalPosition=Numerator}##1}%
\gmu@numeratorkern
780 \char"2044_\gmu@denominatorkern
781 {\addfontfeature{VerticalPosition=Denominator}##2}}%

We define the fractional macros. Since Adobe Minion Pro doesn't contain  $\frac{n}{5}$  nor  $\frac{n}{6}$ ,
we don't provide them here.

782 \gmu@xfraccdef{1/4}{"BC}%
783 \gmu@xfraccdef{1/2}{"BD}%
784 \gmu@xfraccdef{3/4}{"BE}%
785 \gmu@xfraccdef{1/3}{"2153}%
786 \gmu@xfraccdef{2/3}{"2154}%
787 \gmu@xfraccdef{1/8}{"215B}%
788 \gmu@xfraccdef{3/8}{"215C}%
789 \gmu@xfraccdef{5/8}{"215D}%
790 \gmu@xfraccdef{7/8}{"215E}%
\dekfracc 791 \def\dekfracc##1/##2{%
\gm@duppa 792 \def\gm@duppa{##1/##2}%
793 \@ifundefined{gmu@xfracc\all@other\gm@duppa}{%
794 \gmu@dekfracc{##1}/{##2}}{%
795 \csname_\gmu@xfracc\all@other\gm@duppa\endcsname}}%
796 \@ifstar{\let\gmu@dekfracc\gmu@dekfraccsimple}{}%
797 }

\gmu@xedekfraccplain 798 \def\gmu@xedekfraccplain{% 'else' of the main \@ifstar
\dekfracc 799 \def\dekfracc##1/##2{%
800 \addfontfeature{Fractions=On}%
801 ##1/##2}}%
802 }

\gmu@numeratorkern 803 \def\gmu@numeratorkern{\kern-.05em\relax}
804 \let\gmu@denominatorkern\gmu@numeratorkern

```

What have we just done? We defined two versions of the `\Xefractions` declaration. The starred version is intended to make use only of the built-in fractions such as  $\frac{1}{2}$  or  $\frac{7}{8}$ . To achieve that, a handful of macros is defined that expand to the Unicodes of built-in fractions and `\dekfracc` command is defined to use them.

The unstarred version makes use of the `Fraction` font feature and therefore is much simpler.

Note that in the first argument of `\@ifstar` we wrote 8 (eight) #s to get the correct definition and in the second argument 'only' 4. (The L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> Source claims that that is changed in the 'new implementation' of `\@ifstar` so maybe it's subject to change.)

A simpler version of `\dekfracc` is provided in line 950

```
\resizegraphics
```

```
805 \ifXeTeX{%
```

```

\resizegraphics 806 \def\resizegraphics#1#2#3{%
807     \setboxo=\hbox{\XeTeXpicfile_#3_}%
808     \ifx!#1\else
809         \dimeno=#1\relax
810         \count2=\wdo
811         \divide\count2_by1000\relax
812         \counto=\dimeno\relax
813         \divide\counto\count2
814     \fi
815     \ifx!#2\else
816         \dimeno=#1\relax
817         \count6=\hto
818         \divide\count6_by1000\relax
819         \count4=\dimeno\relax
820         \divide\count4\count6
821     \fi
822     \ifx!#1\counto=\count4\fi
823     \ifx!#2\count4=\counto\fi
824     \XeTeXpicfile_#3_xscaled_\counto_yscaled_\count4
825 }}{%
\resizegraphics 826 \def\resizegraphics#1#2#3{%
827     \resizebox{#1}{#2}{%
828         \includegraphics{#3}}}%

```

The [options] in the \XeTeXpicfile command use the following keywords:

width *<dimen>*

height *<dimen>*

scaled *<scalefactor>*

xscaled *<scalefactor>*

yscaled *<scalefactor>*

rotated *<degrees>*

```

\GMtextsuperscript 829 \def\GMtextsuperscript{%
830     \@ifXeTeX{%
\textsuperscript 831         \def\textsuperscript##1{%
832             \addfontfeature{VerticalPosition=Numerator}##1}}%
833     }{\truetextsuperscript}}
\truetextsuperscript 834 \def\truetextsuperscript{%
\textsuperscript 835     \DeclareRobustCommand*\textsuperscript[1]{%
836         \@textsuperscript{\selectfont##1}}%
\@textsuperscript 837     \def\@textsuperscript##1{%
838         {\m@th\ensuremath{\sim\mbox{\fontsize\sfontsize\z@##1}}}}}%

```

## Varia

A very neat macro provided by doc. I copy it ~verbatim.

```

\gmu@tilde 839 \def\gmu@tilde{%
840     \leavevmode\lower.8ex\hbox{$\,\widetilde{\mbox{ }}\,\$, $}}

```

Originally there was just `\` instead of `\mbox{ }` but some commands of ours do redefine `\`.

```

\* 841 \DeclareRobustCommand*\*{\gmu@tilde}

```

842 \AtBeginDocument{% to bypass redefinition of \~ as a text command with various encodings

\texttilde 843 \DeclareRobustCommand\*\texttilde{%  
844 \@ifnextchar/{\gmu@tilde\kern-0,1667em\relax}\gmu@tilde}}

We prepare the proper kerning for “~/”.

The standard \obeyspaces declaration just changes the space’s \catcode to 13 (‘active’). Usually it is fairly enough because no one ‘normal’ redefines the active space. But we are *not* normal and we do *not* do usual things and therefore we want a declaration that not only will \activate the space but also will (re)define it as the \\_ primitive. So define \gmobeyspaces that obeys this requirement.

(This definition is repeated in gmverb.)

845 \foone{\catcode\\_ \active}%  
\gmobeyspaces 846 {\def\gmobeyspaces{\catcode\\_ \active\let\\_ }}

While typesetting poetry, I was surprised that sth. didn’t work. The reason was that original \obeylines does \let not \def, so I give the latter possibility.

847 \foone{\catcode\\_ \^M\active}% the comment signs here are crucial.  
\defobeylines 848 {\def\defobeylines{\catcode\\_ \^M=13\\_ \def\\_M{\par}}}

Another thing I dislike in L<sup>A</sup>T<sub>E</sub>X yet is doing special things for \...skip’s, ‘cause I like the Knuthian simplicity. So I sort of restore Knuthian meanings:

\dekssmallskip 849 \def\dekssmallskip{\vskip\smallskipamount}  
\undeekssmallskip 850 \def\undeekssmallskip{\vskip-\smallskipamount}  
\dekmedskip 851 \def\dekmedskip{\vskip\medskipamount}  
\dekbigskip 852 \def\dekbigskip{\vskip\bigskipamount}  
\hfillneg 853 \def\hfillneg{\hskip\\_opt\\_plus\\_ifill\relax}

In some \if(cat?) test I needed to look only at the first token of a tokens’ string (first letter of a word usually) and to drop the rest of it. So I define a macro that expands to the first token (or {\text}) of its argument.

\@firstofmany 854 \long\def\@firstofmany#1#2\@@nil{#1}

A mark for the **TODO!**s:

\TODO 855 \newcommand\*\TODO[1][]{\{%  
856 \sffamily\bfseries\huge\\_TODO!\if\relax#1\relax\else\space%\fi#1}}

I like twocolumn tables of contents. First I tried to provide them by writing \begin{multicols}{2} and \end{multicols} outto the .toc file but it worked wrong in some cases. So I redefine the internal L<sup>A</sup>T<sub>E</sub>X macro instead.

\twocoltoc 857 \newcommand\*\twocoltoc{%  
858 \RequirePackage{multicol}%  
\@starttoc 859 \def\@starttoc##1{%  
860 \begin{multicols}{2}\makeatletter\@input\_{\jobname\\_##1}%  
861 \if@filesw\\_@xa\\_newwrite\\_csize\\_tf@##1\endcsize  
862 \immediate\\_openout\\_csize\\_tf@##1\endcsize\\_jobname\\_##1\relax  
863 \fi  
864 \@nobreakfalse\end{multicols}}}  
865 \@onlypreamble\twocoltoc

The macro given below is taken from the multicol package (where its name is \enough@room). I put it in this package since I needed it in two totally different works.



```

\enoughpage 866 \newcommand\enoughpage[1]{%
867   \par
868   \dimeno=\pagegoal
869   \advance\dimeno_\by-\pagetotal
870   \ifdim\dimeno<#1\relax\newpage\fi}

```

Two shorthands for debugging:

```

\tOnLine 871 \newcommand*\tOnLine{\typeout{\on@line}}
\OnAtLine 872 \let\OnAtLine\on@line

```

An equality sign properly spaced:

```

\equals 873 \newcommand*\equals{${}={}$}

```

And for the L<sup>A</sup>T<sub>E</sub>X's pseudo-code statements:

```

\eequals 874 \newcommand*\eequals{${}=={}$}

```

While typesetting a UTF-8 ls-R result I found a difficulty that follows: UTF-8 encoding is handled by the inputenc package. It's O.K. so far. The UTF-8 sequences are managed using active chars. That's O.K. so far. While writing such sequences to a file, the active chars expand. You feel the blues? When the result of expansion is read again, it sometimes is again an active char, but now it doesn't start a correct UTF-8 sequence.

Because of that I wanted to 'freeze' the active chars so that they would be \written to a file unexpanded. A very brutal operation is done: we look at all 256 chars' catcodes and if we find an active one, we \let it \relax. As the macro does lots and lots of assignments, it shouldn't be used in \edefs.

```

\freeze@actives 875 \def\freeze@actives{%
876   \count\z@\z@
877   \@whilenum\count\z@<\@cclvi\do{%
878     \ifnum\catcode\count\z@=\active
879       \uccode`~=\count\z@
880       \uppercase{\let~\relax}%
881     \fi
882     \advance\count\z@\@one}}

```

A macro that typesets all 256 chars of given font. It makes use of \@whilenum.

```

\ShowFont 883 \newcommand*\ShowFont[1][6]{%
884   \begin{multicols}{#1}[The\currentfont(the\fontencoding%
      \ encoding):]
885     \parindent\z@
886     \count\z@\m@ne
887     \@whilenum\count\z@<\@cclv\do{
888       \advance\count\z@\@one
889       \the\count\z@:\char\count\z@\par}
890   \end{multicols}}

```

A couple of macros for typesetting liturgic texts such as psalms of Liturgia Horarum. I wrap them into a declaration since they'll be needed not every time.

```

\liturgiques 891 \newcommand*\liturgiques[1][red]{% Requires the color package.
892   \gmu@RPif{color}{color}%
\czerwo 893   \newcommand*\czerwo{\small\color{#1}}% environment
\czer 894   \newcommand{\czer}[1]{\leavevmode\czerwo##1}% we leave vmode be-
      cause if we don't, then verse's \everypar would be executed in a group
      and thus its effect lost.
\* 895   \def\*\czer{${}$}

```

```

\+ 896 \def\+{\czer{$\dag$}}
\nieczer 897 \newcommand*\nieczer[1]{\textcolor{black}{##1}}

```

After the next definition you can write `\gmu@RP[options]{package}{csname}` to get the package #2 loaded with options #1 if the csname #3 is undefined.

```

\gmu@RPif 898 \newcommand*\gmu@RPif[3][]{%
899 \ifx\relax#1\relax
\gmu@resa 900 \else\def\gmu@resa{#1}%
901 \fi
902 \@xa\RequirePackage\gmu@resa{#2}}

```

Since inside document we cannot load a package, we'll redefine `\gmu@RPif` to issue a request before the error issued by undefined CS.

```

903 \AtBeginDocument{%
\gmu@RPif 904 \renewcommand*\gmu@RPif[3][]{%
905 \ifundefined{#3}{%
906 \ifpackageloaded{#2}{%
907 \typeout{^^J!\Package`#2' not loaded!!!(%
\on@line)^^J}}{}}

```

It's very strange to me but it seems that `c` is not defined in the basic math packages. It is missing at least in the *Symbols* book.

```

\continuum 908 \providecommand*\continuum{\gmu@RPif{eufrak}{mathfrak}{\mathfrak{c}}
c}}

```

And this macro I saw in the *ltugproc* document class nad I liked it.

```

\iteracro 909 \def\iteracro{%
\acro 910 \DeclareRobustCommand*\acro[1]{\gmu@acrospaces##1\%
\gmu@acrospaces}%
911 }
912 \iteracro
\gmu@acrospaces 913 \def\gmu@acrospaces#1#2\gmu@acrospaces{%
914 \gmu@acroinner#1\gmu@acroinner
915 \ifx\relax#2\relax\else
916 \space
917 \afterfi{\gmu@acrospaces#2\gmu@acrospaces}% when #2 is nonempty, it
is ended with a space. Adding one more space in this line resulted in an
infinite loop.
918 \fi}
\gmu@acroinner 919 \def\gmu@acroinner#1{%
920 \ifx\gmu@acroinner#1\relax\else
921 \ifcat\@nx#1\relax%
922 \ifnum`#1=\uccode`#1%
923 {\acrocore{#1}}%
924 \else{#1}% tu było \smallerr
925 \fi
926 \else#1%
927 \fi
928 \afterfi\gmu@acroinner
929 \fi}

```

We extract the very thing done to the letters to a macro because we need to redefine it in fonts that don't have small caps.

\acrocore 930 \def\acrocore{\scshape\lowercase}

Since the fonts I am currently using do not support required font feature, I skip the following definition.

\IMO 931 \newcommand\*\IMO{\acro{IMO}}

\AKA 932 \newcommand\*\AKA{\acro{AKA}}

\usc 933 \DeclareRobustCommand\*\usc[1]{\addfontfeature{%  
Letters=UppercaseSmallCaps}#1}}

\uscacro 934 \def\uscacro{\let\acro\usc}

\qxenc 935 \newcommand\*\qxenc{\fontencoding{QX}\selectfont}

The \copyright command is unavailable in T1 and U (unknown) encodings so provide

\qxcopyright 936 \newcommand\*\qxcopyright{\qxenc\copyright}

\qxcopyrights 937 \newcommand\*\qxcopyrights{%  
938 \let\gmu@copyright\copyright  
939 \def\copyright{\qxenc\gmu@copyright}}

\fixcopyright 940 \newcommand\*\fixcopyright{%  
941 \@ifXeTeX{\def\copyright{\char"00A9}}{\qxcopyrights}}

Probably the only use of it is loading gmdocc.cls ‘as second class’. This command takes first argument optional, options of the class, and second mandatory, the class name. I use it in an article about gmdoc.

\secondclass 942 \def\secondclass{%  
ifSecondClass 943 \newif\ifSecondClass  
944 \SecondClasstrue  
945 \@fileswithoptions\@clsextension}% [outeroff,gmeometric]{gmdocc}  
it's loading gmdocc.cls with all the bells and whistles except the error message.

Cf. *The T<sub>E</sub>Xbook* exc. 11.6.

A line from L<sup>A</sup>T<sub>E</sub>X:

% \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont

didn't work as I would wish: in a \footnotesize's scope it still was \scriptsize, so too large.

\gmu@dekfraccsimple 946 \def\gmu@dekfraccsimple#1/#2{\leavevmode\kern.1em  
947 \raise.5ex\hbox{\udigits\smaller[3]#1}\gmu@numerator\kern  
948 \dekfracslash\gmu@denominator\kern  
949 {\udigits\smaller[3]#2}}%

\dekfraccsimple 950 \def\dekfraccsimple{%  
951 \let\dekfrac\gmu@dekfraccsimple  
952 }

\dekfracslash 953 \@ifXeTeX{\def\dekfracslash{\char"2044}}%

\dekfracslash 954 \def\dekfracslash{/}% You can define it as the fraction slash, \char"2044  
955 \dekfraccsimple

A macro that acts like \, (thin and unbreakable space) except it allows hyphenation afterwards:

\ikern 956 \newcommand\*\ikern{\,\penalty10000\hskip0sp\relax}

And a macro to forbid hyphenation of the next word:

```

\nohy 957 \newcommand*\nohy{\kernosp\relax}
\yeshy 958 \newcommand*\yeshy{\penalty10000\hskiposp\relax}

```

In both of the above definitions ‘osp’ not ‘z@’ to allow their writing to and reading from files where ‘@’ is ‘other’.

**\@ifempty**

```

\@ifempty 959 \long\def\@ifempty#1#2#3{%
\gmu@reserveda 960 \def\gmu@reserveda{#1}%
961 \ifx\gmu@reserveda\@empty\afterfi{#2}%
962 \else\afterfi{#3}\fi
963 }

```

**\include not only .tex's**

\include modified by me below lets you to include files of any extension provided that extension in the argument.

If you want to \include a non-.tex file and deal with it with \includeonly, give the latter command full file name, with the extension that is.

```

\gmu@getext 964 \def\gmu@getext#1.#2\@nil{%
\gmu@filename 965 \def\gmu@filename{#1}%
\gmu@fileext 966 \def\gmu@fileext{#2}}

967 \def\include#1{\relax
968 \ifnum\@auxout=\@partaux
969 \@latex@error{\string\include\space cannot be nested}\@eha
970 \else\@include#1\fi}

\@include 971 \def\@include#1{%
972 \gmu@getext#1.\@nil
\gmu@fileext 973 \ifx\gmu@fileext\empty\def\gmu@fileext{tex}\fi
974 \clearpage
975 \if@filesw
976 \immediate\write\@mainaux{\string\@input{\gmu@filename.aux}}%
977 \fi
978 \@tempswatrue
979 \if@partsw
980 \@tempswafalse
981 \edef\reserved@b{#1}%
982 \@for\reserved@a:=\@partlist\do{%
983 \ifx\reserved@a\reserved@b\@tempswatrue\fi}%
984 \fi
985 \if@tempswa
986 \let\@auxout\@partaux
987 \if@filesw
988 \immediate\openout\@partaux\gmu@filename.aux
989 \immediate\write\@partaux{\relax}%
990 \fi
991 \@input@{\gmu@filename.\gmu@fileext}%
992 \inclasthook
993 \clearpage
994 \@writeckpt{\gmu@filename}%
995 \if@filesw
996 \immediate\closeout\@partaux

```

```

997     \fi
998   \else

```

If the file is not included, reset \@include \deadcycles, so that a long list of non-included files does not generate an ‘Output loop’ error.

```

999     \deadcycles\z@
1000     \@nameuse{cp@\gmu@filename}%
1001   \fi
1002   \let\@auxout\@mainaux}

```

```

\whenonly 1003 \newcommand\whenonly[3]{%
\gmu@whonly 1004   \def\gmu@whonly{#1,}%
1005   \ifx\gmu@whonly\@partlist\afterfi{#2}\else\afterfi{#3}\fi}

```

I assume one usually includes chapters or so so the last page style should be closing.

```

\inclasthook 1006 \def\inclasthook{\thispagestyle{closing}}

```

### Faked small caps

```

\gmu@scapLetters 1007 \def\gmu@scapLetters#1{%
1008   \ifx#1\relax\relax\else% two \relaxes to cover the case of empty #1.
1009     \ifcat_a#1\relax
1010       \ifnum\the\lccode`#1=`#1\relax
1011         {\fakescapsscore\MakeUppercase{#1}}}% not Plain \uppercase because
           that works bad with inputenc.
1012       \else#1%
1013       \fi
1014     \else#1%
1015     \fi%
1016     \@xa\gmu@scapLetters
1017   \fi}%
\gmu@scapSpaces 1018 \def\gmu@scapSpaces#1_#2\@@nil{%
1019   \ifx#1\relax\relax
1020   \else\gmu@scapLetters#1\relax
1021   \fi
1022   \ifx#2\relax\relax
1023   \else\afterfi{\ \gmu@scapSpaces#2\@@nil}%
1024   \fi}
\gmu@scapss 1025 \def\gmu@scapss#1\@@nil{{\def~{{\nobreakspace}}%
\nobreakspace 1026   \gmu@scapSpaces#1_\@@nil}}}% \def\\{{\newline}}\relax adding re-
           definition of \\ caused stack overflow Note it disallows hyphenation ex-
           cept at \-.
\fakescaps 1027 \DeclareRobustCommand\fakescaps[1]{%
1028   \gmu@scapss#1\@@nil}}
1029 \let\fakescapsscore\gmu@scalematchX
Experimente z akcentami patrz no3.tex.
\tinycapae 1030 \def\tinycapae{{\tiny\AE}}}% to use in \fakescaps[\tiny]{...}
1031 \RequirePackage{calc}
wg \zf@calc@scale pakietu fontspec.
1032 \@ifXeTeX{%
\gmu@scalar 1033   \def\gmu@scalar{1.0}%
\zf@scale 1034   \def\zf@scale{}%

```

```

\gmu@scalematchX 1035 \def\gmu@scalematchX{%
1036 \begingroup
\gmu@scalar 1037 \ifx\zf@scale\empty\def\gmu@scalar{1.0}%
1038 \else\let\gmu@scalar\zf@scale\fi
1039 \setlength\@tempdima{\fontdimen5\font}% 5—ex height
1040 \setlength\@tempdimb{\fontdimen8\font}% 8—XTEX synthesized up-
percase height.
1041 \divide\@tempdimb\by1000\relax
1042 \divide\@tempdima\by\@tempdimb
1043 \setlength{\@tempdima}{\@tempdima*\real{\gmu@scalar}}%
1044 \@ifundefined{fakesc@extrascale}{}{%
1045 \setlength{\@tempdima}{\@tempdima*\real{%
\fakec@extrascale}}}%
1046 \@tempcnta=\@tempdima
1047 \divide\@tempcnta\by1000\relax
1048 \@tempcntb=-1000\relax
1049 \multiply\@tempcntb\by\@tempcnta
1050 \advance\@tempcntb\by\@tempdima
1051 \xdef\gmu@scscale{\the\@tempcnta.%
1052 \ifnum\@tempcntb<1000\fi
1053 \ifnum\@tempcntb<100\fi
1054 \the\@tempcntb}%
1055 \endgroup
1056 \addfontfeature{Scale=\gmu@scscale}%
1057 }}{\let\gmu@scalematchX\smallerr}

\fakec@extrascale 1058 \def\fakec@extrascale#1{\def\fakec@extrascale{#1}}
\fakec@extrascale

```

### See above/see below

To generate a phrase as in the header depending of whether the respective label is before of after.

```

\wyzejnizej 1059 \newcommand*\wyzejnizej[1]{%
1060 \edef\gmu@tempa{\@ifundefined{r#1}{\arabic{page}}{%
1061 \@xa\@xa\@xa\@secondoftwo\csname_r#1\endcsname}}%
1062 \ifnum\gmu@tempa<\arabic{page}\relax\wy\zej\fi
1063 \ifnum\gmu@tempa>\arabic{page}\relax\ni\zej\fi
1064 \ifnum\gmu@tempa=\arabic{page}\relax\@xa\ignorespaces\fi
1065 }

```

### luzniej and napapierki—environments used in page breaking for money

The name of first of them comes from Polish typesetters' phrase "rozbijać [skład] na papierki"—'to broaden [leading] with paper scratches'.

```

\napapierkistretch 1066 \def\napapierkistretch{0,3pt}% It's quite much for 11/13pt typesetting
\napapierkicore 1067 \def\napapierkicore{\advance\baselineskip%
1068 \by\optplus\napapierkistretch\relax}

napapierki 1069 \newenvironment*{napapierki}{%
1070 \par\global\napapierkicore}%
1071 \par\dimen\z@=\baselineskip
1072 \global\baselineskip=\dimen\z@}% so that you can use \endnapapierki in
interlacing environments

```

```

\gmu@luzniej 1073 \newcount\gmu@luzniej
\luzniejcore 1074 \newcommand*\luzniejcore[1][1]{%
1075   \advance\gmu@luzniej\@ne% We use this count to check whether we open the
        environment or just set \looseness inside it again.
1076   \ifnum\gmu@luzniej=\@ne\multiply\tolerance\by2\fi
1077   \looseness=#1\relax}

```

After `\begin{luzniej}` we may put the optional argument of `\luzniejcore`

```

luzniej 1078 \newenvironment*{luzniej}{\par\luzniejcore}{\par}

```

The starred version does that `\everypar`, which has its advantages and disadvantages.

```

luzniej* 1079 \newenvironment*{luzniej*}[1][1]{%
1080   \multiply\tolerance\by2\relax
1081   \everypar{\looseness=#1\relax}}{\par}

```

```

\nawj 1082 \newcommand*\nawj{\kern.1em\relax}% to put between parentheses and let-
        ters with lower ... such as j or y in certain fonts.

```

The original `\pauza` of *polski* has the skips rigid (one is even a kern). It begins with `\ifhmode` to be usable also at the beginning of a line as the mark of a dialogue.

```

1083 \ifdefined\XeTeXversion
1084 \AtBeginDocument{% to be independent of moment of loading of polski.
\ 1085   \DeclareRobustCommand*\-{%
1086     \ifhmode
1087       \unskip\penalty10000
1088       \afterfi{%
1089         \@ifnextspace{\hskip.2em\plus.1em\relax
\pauzacore 1090         \pauzacore\hskip.2em\plus.1em\relax\ignorespaces}%
1091         {\pauzacore\penalty\hyphenpenalty\hskip\z@}}%
1092     \else

```

According to *Instrukcja technologiczna. Skład ręczny i maszynowy* the dialogue dash should be followed by a rigid `\hskip` of  $\frac{1}{2}$ em.

```

1093     \leavevmode\pauzacore\penalty10000\hskip.5em\ignorespaces
1094     \fi}%

```

The next command's name consists of letters and therefore it eats any spaces following it, so `\@ifnextspace` would always be false.

```

\pauza 1095 \DeclareRobustCommand*\pauza{%
1096   \ifhmode
1097     \unskip\penalty10000
1098     \hskip.2em\plus.1em\relax
1099     \pauzacore\hskip.2em\plus.1em\relax\ignorespaces%
1100   \else

```

According to *Instrukcja technologiczna. Skład ręczny i maszynowy* the dialogue dash should be followed by a rigid `\hskip` of  $\frac{1}{2}$ em.

```

1101     \leavevmode\pauzacore\penalty10000\hskip.5em\ignorespaces
1102     \fi}%

```

And a version with no space at the left, to begin a `\noindent` paragraph or a dialogue in quotation marks:

```

\lpauza 1103 \DeclareRobustCommand*\lpauza{%
1104   \pauzacore\hskip.2em\plus.1em\ignorespaces}%

```

We define `\ppauza` as an en dash surrounded with thin stretchable spaces and sticking to the upper line or bare but discretionary depending on the next token being `space_0`. Of course you'll never get such a space after a literal CS so an explicit `\ppauza` will always result with a bare discretionary en dash, but if we `\let-\ppauza...`

```

\DeclareRobustCommand*\-{%
  \ifvmode\PackageError{gmutils}{%
    command_\backslashppauza(en_dash)_not_intended_for_vmode.}{%
    Use_\backslashppauza(en_dash)_only_in_number_and_numerical_
      ranges.}%
  \else
    \afterfi{%
      \@ifnextspace{\unskip\penalty10000\hskipo.2em\pluso.1em%
        \relax
        -\hskip.2em\pluso.1em\ignorespaces}{\unskip%
          \discretionary{-}{-}{-}}}%
    \fi}%
\ppauza \DeclareRobustCommand*\ppauza{%
  \ifvmode\PackageError{gmutils}{%
    command_\backslashppauza(en_dash)_not_intended_for_vmode.}{%
    Use_\backslashppauza(en_dash)_only_in_number_and_numerical_
      ranges.}%
  \else
    \unskip\discretionary{-}{-}{-}%
  \fi}%
\emdash \def\emdash{\char`-}
}% of at begin document

\longpauza \def\longpauza{\def\pauzacore{-}}
\pauzacore \longpauza
\shortpauza \def\shortpauza{%
\pauzacore \def\pauzacore{-\kern,23em\relax\llap{-}}}%
\fi% of if XeTeX.

```

If you have all the three dashes on your keyboard (as I do), you may want to use them for short instead of `\pauza`, `\ppauza` and `\dywiz`. The shortest dash is defined to be smart in math mode and result with `-`.

```

\ifdefined\XeTeXversion
\foone{\catcode`\- \active \catcode`\- \active \catcode`\- \active}{%
\adashes \def\adashes{\AtBeginDocument\adashes}% because \pauza is defined at
  begin document.
\adashes \AtBeginDocument{\def\adashes{%
  \catcode`\- \active \let-\-
  \catcode`\- \active \let-\-
}}
\else
\relaxen\adashes
\fi

```

The hyphen shouldn't be active IMO because it's used in TeX control such as `\hskip-2pt`. Therefore we provide the `\ahyphen` declaration reluctantly, because sometimes we need it and always use it with caution. Note that my active hyphen in vertical and math modes expands to `-12`.

```

\gmu@dywiz \def\gmu@dywiz{\ifmmode-\else
\ifvmode-\else\afterfifi\dywiz\fi\fi}%

```



```

1140 \foone{\catcode`\- \active}{%
\ahyphen 1141 \def\ahyphen{\let-\gmudywiz\catcode`\- \active}}

To get current time. Works in  $\varepsilon$ -TeXs, including XeTeX.

\czas 1142 \newcommand*\czas[1][.]{%
1143 \the\numexpr(\time-30)/60\relax#1%
1144 \@tempcnta=\numexpr\time-(\time-30)/60*60\relax
1145 \ifnum\@tempcnta<10_o\fi\the\@tempcnta}

To push the stuff up to the header and have the after heading skip after the stuff

\przeniesvskip 1146 \long\def\przeniesvskip#1{%
1147 \edef\gmudLastSkip{\the\lastskip}%
1148 \vskip-\gmudLastSkip\relax
1149 \vspace*{osp}%
1150 #1\vskip\gmudLastSkip\relax}

\textbullet 1151 \@ifXeTeX{\chardef\textbullet="2022_{\def\textbullet{\$ \bullet \$}}
tytulowa 1152 \newenvironment*{tytulowa}{\newpage}{\par\thispagestyle{empty}%
\newpage}

Nazwisko na stronę redakcyjną

\nazwired 1153 \def\nazwired{\quad\textsc}
```

## Settings for mathematics in main font

I used this terrible macros while typesetting E. Szarzyński's *Letters* in 2008.

```
\gmath 1154 \def\gmath{%  
1155   \def\do##1{\edef##1{{\@nx\mathit{\@xa@gobble\string##1}}}%  
1156   \do\A_\do\a_\do\B_\do\b_\do\c_\do\C_\do\d_\do\D_\do\e_\do\E_\do\f  
1157   \do\F_\do\g_\do\G_\do\i_\do\I_\do\j_\do\J_\do\k_\do\K_\do\l_\do\L_  
       \do\m  
1158   \do\M_\do\n_\do\N_\do\P_\do\p_\do\q_\do\Q_\do\R_\do\r  
1159   \let\sectionsign\S_\do\S_\do\s_\do\T_\do\t_\do\u_\do\U_\do\v%  
       \do\V  
1160   \do\w_\do\W_\do\x_\do\X_\do\Y_\do\y_\do\z_\do\Z  
1161   \def\do##1{\edef##1{{\@nx\mathrm{\@xa@gobble\string##1}}}%  
1162   \do\o\do\1\do\2\do\3\do\4\do\5\do\6\do\7\do\8\do\9%  
1163   \relaxen\do  
1164   \newcommand*\do[4][\mathit]{\def##2{##3{##1{\char"##4}}}%  
1165   \do\alpha{}{03B1}%  
1166   \do[\mathrm]\Delta{}{0394}%  
1167   \do\varepsilon{}{03B5}%  
1168   \do\vartheta{}{03D1}%  
1169   \do\nu{}{03BD}%  
1170   \do\pi{}{03C0}%  
1171   \do\phi{}{03D5}%  
1172   \do[\mathrm]\Phi{}{0424}%  
1173   \do\sigma{}{03C3}%  
1174   \do\varsigma{}{03DA}%  
1175   \do\psi{}{03C8}%  
1176   \do\omega{}{03C9}%  
1177   \do\infty{}{221E}%  
1178   \do[\mathrm]\neg{\mathbin}{00AC}%
```

```

1179 \do[\mathrm]\neq{\mathrel}{2260}%
1180 \do\partial{}{2202}%
1181 \do[\mathrm]\pm{}{00B1}%
1182 \do[\mathrm]\pm{\mathbin}{00B1}%
1183 \do[\mathrm]\sim{\mathrel}{007E}%
1184 \def\do##1##2##3{\def##1{%
\mathop{\mathchoice{\hbox{%
\rm
\edef\gma@tempa{\the\fontdimen8\font}%
\larger[3]%
\lower\dimexpr(\fontdimen8\font-\gma@tempa)/2_{%
\hbox{##2}}}{\hbox{%
\rm
\edef\gma@tempa{\the\fontdimen8\font}%
\larger[2]%
\lower\dimexpr(\fontdimen8\font-\gma@tempa)/2_{%
\hbox{##2}}}{%
{\mathrm{##2}}{\mathrm{##2}}}{##3}}}%
\do\sum{\char"2211}{}%
\do\forall{\gma@quantifierhook\rotatebox[origin=c]{180}{A}%
\setboxo=\hbox{A}\setbox2=\hbox{\scriptsize_x}%
\kern\dimexpr\ht2/3*2_{-}\wdo/2\relax}{\nolimits}%
\do\exists{\rotatebox[origin=c]{180}{\gma@quantifierhook_E}}}%
\nolimits%
\def\do##1##2##3{\def##1{##3{%
\mathchoice{\hbox{\rm##2}}{\hbox{\rm##2}}}%
{\hbox{\rm\scriptsize##2}}{\hbox{\rm\tiny##2}}}}}%
\do\vee{\rotatebox[origin=c]{90}{<}}\mathbin
\do\wedge{\rotatebox[origin=c]{-90}{<}}\mathbin
\do\leftarrow{\char"2190}\mathrel
\do\rightarrow{\char"2192}\mathrel
\do\leftrightarrow{\char"2190\kern-0,1em\char"2192}\mathrel
\def\do##1##2##3{%
\catcode`##1=12\relax
\scantokens{\mathcode`##1="8000\relax
\foone{\catcode`##1=\active}{\def##1}{##3{%
\mathchoice{\hbox{\rm##2}}{\hbox{\rm##2}}}%
{\hbox{\rm\scriptsize##2}}{\hbox{\rm\tiny##2}}}}}%
\ignorespaces}}}% to eat the lineend (scantokens acts as \read including
line end.
\do..\mathpunct_{\do,,\mathpunct_{\do.....\mathpunct
\do(\mathopen
\@ifundefined{resetMathstrut@}{}{% an error occurred 'bad mathchar etc.'
because amsmath.sty doesn't take account of a possibility of ( ) being math-
active.
\resetMathstrut@ \def\resetMathstrut@{%
\setbox\z@\hbox{%
%% \mathchardef\@tempa\mathcode`\(\relax%% \def\@tempb##1"##2##3{%
\the\textfont"##3\char"}%% \expandafter\@tempb\meaning\@tempa \relax
}%
\ht\Mathstrutbox@\ht\z@\dp\Mathstrutbox@\dp\z@
}}%

```

```

1225 \do))\mathclose
1226 \do[[\mathopen\do]]\mathclose
1227 \do-{ \char"2212}\mathbin_\do++\mathbin_\do==\mathrel_\do\times%
      \mathbin
1228 \do:.\mathbin_\do..\mathbin_\do//\mathbin_\do<<\mathrel
1229 \do>>\mathrel
1230 \def\do##1##2##3{\def##1####1{##2{\hbox{%
1231     \rm
1232     \setboxo=\hbox{####1}%
1233     \edef\gma@tempa{\the\hto}%
1234     \edef\gma@tempb{\the\dpo}%
1235     ##3%
1236     \setboxo=\hbox{####1}%
1237     \lower\dimexpr(\hto_+_dpo)/2-\dpo_-(\gma@tempa+%
      \gma@tempb)/2-\gma@tempb)__%
1238     \boxo}}}%
1239 \do\bigl\mathopen\larger
1240 \do\bigr\mathclose\larger
1241 \do\Bigl\mathopen\largerr
1242 \do\Bigr\mathclose\largerr
1243 \do\biggl\mathopen{\larger[3]}%
1244 \do\biggr\mathclose{\larger[3]}%
1245 \do\Biggl\mathopen{\larger[4]}%
1246 \do\Biggr\mathclose{\larger[4]}%
1247 \def\do##1##2{\def##1{\ifmmode##2{\mathchoice
1248     {\hbox{\rm\char`##1}}{\hbox{\rm\char`##1}}%
1249     {\hbox{\rm\scriptsize\char`##1}}{\hbox{\rm\tiny%
      \char`##1}}}%
1250     \else\char`##1\fi}}%
1251 \StoreMacros{\{\}}%
1252 \do{\mathopen
1253 \do{\mathclose
1254 \def\={\mathbin{=}}%
1255 \def\neqb{\mathbin{\neq}}%
1256 \def\do##1{\edef\gma@tempa{%
1257     \def\@xa\@nx\csname_\@xa\gobble\string##1r\endcsname{%
1258     \@nx\mathrel{\@nx##1}}}%
1259     \gma@tempa}%
1260 \do\vee_\do\wedge_\do\neg
1261 \def\fakern{\mkern-3mu}%
1262 \thickmuskip=8mu_plus_4mu\relax
1263 \gma@gmathhook
1264 }% of def gmath
1265 \emptify\gma@quantifierhook
1266 \def\quantifierhook#1{%
1267     \def\gma@quantifierhook{#1}}
1268 \emptify\gma@gmathhook
1269 \def\gma@gmathhook#1{\addtomacro\gma@gmathhook{#1}}
1270 \def\gma@dollar$#1${{\gmath$#1$}}%
1271 \def\gma@bare#1{\gma@dollar$#1$}%
1272 \def\gma@checkbracket{\@ifnextchar\[%

```

```

1273 \gma@bracket\gma@bare}
\gma@bracket 1274 \def\gma@bracket\[#1\]{\gmath\[#1\]}@ifnextchar\par{}{%
\noindent}}
\gma 1275 \def\gma{\@ifnextchar$%
1276 \gma@dollar\gma@checkbracket}
\garamath 1277 \def\garamath{%
1278 \quantifierhook{\addfontfeature{OpticalSize=800}}}%
\gma@arrowdash 1279 \def\gma@arrowdash{%
1280 \setboxo=\hbox{\char"2192}\copyo\kern-o,6\wdo
1281 \bgcolor\rule[-\dpo]{o,6\wdo}{\dimexpr\hto+\dpo}\kern-o,6%
\wdo}}%
\gma@gmathhook 1282 \def\gma@gmathhook{%
1283 \def\do####1####2####3{\def####1{####3}%
\mathchoice 1284 \mathchoice{\hbox{\rm####2}}{\hbox{\rm####2}}%
{\hbox{\rm\scriptsize####2}}{\hbox{\rm\tiny####2}}}%
1285 \do\mapsto{\rule[o,4ex]{o,1ex}{o,4ex}\kern-o,05em%
\gma@arrowdash\kern-o,05em\char"2192}\mathrel
1286 \do\cup{\scshape\mathbin
1287 \do\varnothing{\setboxo=\hbox{\gma@quantifierhook%
\addfontfeature{Scale=1.272727}o}%
1288 \setbox2=\hbox{\char"2044}%
1289 \copyo\kern-o,5\wdo\kern-o,5\wd2\lowero,125\wdo\copy2
1290 \kerno,5\wdo\kern-o,5\wd2}{}%
1291 \do\leftarrow{\char"2190\kern-o,05em\gma@arrowdash}\mathrel
1292 \do\rightarrow{\gma@arrowdash\kern-o,05em\char"2192}\mathrel
1293 \do\in{\gma@quantifierhook\char"0454}\mathbin
1294 \do\in{\gma@quantifierhook\char"0454}\mathbin
1295 }}
1296 }}

```

## Typesetting dates in my memoirs

A date in the YYYY-MM-DD format we'll transform into DD mmmm YYYY format or we'll just typeset next two tokens/{...} if the arguments' string begins with --. The latter option is provided to preserve compatibility with already used macros and to avoid a starred version of \thedata and the same time to be able to turn \datef off in some cases (for SevSev04.tex).

```

\polskadata 1297 \newcommand*\polskadata{%
\datef 1298 \def\datef##1-##2-##3##4{%
1299 \if\relax##2\relax##3##4%
1300 \else
1301 \ifnum##3##4=0\relax
1302 \else
1303 \ifnum##3=0\relax
1304 \else##3%
1305 \fi##4%
1306 \fi
1307 \ifcase##2\relax\or\ stycznia\or\ lutego%
1308 \or\ marca\or\ kwietnia\or\ maja\or\ czerwca\or\ lipca\or%
\ sierpnia%
1309 \or\ wrzeźnia\or\ października\or\ listopada\or\ grudnia%
\else
1310 {}%

```

```

1311     \fi
1312     \if\relax##1\relax\else\ \fi_##1%
1313     \fi}%
\datefsl 1314 \def\datefsl##1/##2/##3##4{%
1315     \if\relax##2\relax##3##4%
1316     \else
1317         \ifnum##3##4=0\relax
1318         \else
1319             \ifnum##3=0\relax
1320             \else##3%
1321             \fi##4%
1322         \fi
1323         \ifcase##2\relax\or\ stycznia\or\ lutego%
1324             \or\ marca\or\ kwietnia\or\ maja\or\ czerwca\or\ lipca\or%
                \ sierpnia%
1325             \or\ wrzesnia\or\ pazdziernika\or\ listopada\or\ grudnia%
                \else
1326             {}%
1327         \fi
1328         \if\relax##1\relax\else\ \fi_##1%
1329     \fi}%
1330 }% of \polskadata
1331 \polskadata
    For documentation in English:
\englishdate 1332 \newcommand*\englishdate{%
    \datef 1333 \def\datef##1-##2-##3##4{%
1334     \if\relax##2\relax##3##4%
1335     \else
1336         \ifcase##2\relax\or_January\or_February%
1337             \or_March\or_April\or_May\or_June\or_July\or_August%
1338             \or_September\or_October\or_November\or_December\else
1339             {}%
1340         \fi
1341         \ifnum##3##4=0\relax
1342         \else
1343             \ %
1344             \ifnum##3=0\relax
1345             \else##3%
1346             \fi##4%
1347         \ifcase##3##4\relax\or_st\or_nd\or_rd\else_th\fi
1348         \fi
1349         \if\relax##1\relax\else,\ \fi_##1%
1350     \fi
1351 }%
\datefsl 1352 \def\datefsl##1/##2/##3##4{%
1353     \if\relax##2\relax##3##4%
1354     \else
1355         \ifcase##2\relax\or_January\or_February%
1356             \or_March\or_April\or_May\or_June\or_July\or_August%
1357             \or_September\or_October\or_November\or_December\else
1358             {}%
1359         \fi

```

```

1360         \ifnum##3##4=0\relax
1361         \else
1362             \ %
1363             \ifnum##3=0\relax
1364             \else##3%
1365             \fi##4%
1366             \ifcase##3##4\relax\or_st\or_nd\or_rd\else_th\fi
1367         \fi
1368         \if\relax##1\relax\else,\ \fi_##1%
1369     \fi
1370 }%
1371 }

\ifgmu@dash 1372 \newif\ifgmu@dash
\gmu@ifnodash 1373 \def\gmu@ifnodash#1-#2\@nil{%
1374     \def\@tempa{#2}%
1375     \ifx\@tempa\@empty}

\gmu@testdash 1376 \def\gmu@testdash#1\ifgmu@dash{%
1377     \gmu@ifnodash#1-\@nil
1378     \gmu@dashfalse
1379     \else
1380     \gmu@dashtrue
1381     \fi
1382     \ifgmu@dash}

```

A word of explanation to the above pair of macros. `\gmu@testdash` sets `\iftrue` the `\ifgmu@dash` switch if the argument contains an explicit `-`. To learn it, an auxiliary `\gmu@ifdash` macro is used that expands to an open (unfied) `\ifx` that tests whether the dash put by us is the only one in the argument string. This is done by matching the parameter string that contains a dash: if the investigated sequence contains (another) dash, #2 of `\gmu@ifdash` becomes the rest of it and the ‘guardian’ dash put by us so then it’s nonempty. Then #2 is took as the definiens of `\@tempa` so if it was empty, `\@tempa` becomes equal `\@empty`, otherwise it is not.

Why don’t we use just `\gmu@ifdash`? Because we want to put this test into another `\if...`. A macro that doesn’t *mean* `\if...` wouldn’t match its `\else` nor its `\fi` while  $\TeX$  would skip the falsified branch of the external `\if...` and that would result in the ‘extra `\else`’ or ‘extra `\fi`’ error.

Therefore we wrap the very test in a macro that according to its result sets an explicit Boolean switch and write this switch right after the testing macro. (Delimiting `\gmu@testdash`’s parameter with this switch is intended to bind the two which are not one because of  $\TeX$ ’s reasons only.

Warning: this pair of macros may result in ‘extra `\else`/extra `\fi`’ errors however, if `\gmu@testdash` was `\expandaftered`.

Dates for memoirs to be able to typeset them also as diaries.

```

\ifdate 1383 \newif\ifdate
           \% \newcounter{dateinsection}[section]

\data 1384 \newcommand*{\data}[1]{%
1385     \ifdate\gmu@testdash#1\ifgmu@dash\datef#1\else\datefsl#1\fi\fi}

\linedate 1386 \newcommand*{\linedate}[1]{\par\ifdate\addvspace{\dateskip}%
1387     \date@line{\footnotesize\itshape\date@biway{#1}}%
1388     \nopagebreak\else%%\ifnum\arabic{dateinsection}>0\dekbigskip\fi}

```

```

1389 \addvspace{\bigskipamount}%
1390 \fi}% end of \linedate.

1391 \let\dateskip\medskipamount

\date@biway 1392 \def\date@biway#1{%
1393 \gmu@testdash#1\ifgmu@dash\datef#1\else\datefsl#1\fi}

\rdate 1394 \newcommand*\rdate[1]{\let\date@line\rightline_\linedate{#1}}
\ldate 1395 \newcommand*\ldate[1]{\let\date@line\leftline_\linedate{#1}}
\runindate 1396 \newcommand*\runindate[1]{%
1397 \paragraph{\footnotesize\itshape_\datef#1@@nil}\stepcounter{%
dateinsection}}

I'm not quite positive which side I want the date to be put to so let's let for now and
we'll be able to change it in the very documents.

1398 \let\thedata\ldate

\zwrobcy 1399 \DeclareRobustCommand*\zwrobcy[1]{\emph{#1}}_\% ostinato, allegro con moto,
garden party etc., także komplement

\tytul 1400 \DeclareRobustCommand*\tytul[1]{\emph{#1}}

Maszynopis w świecie justowanym zrobi delikatną chorągiewkę.

maszynopis 1401 \newenvironment{maszynopis}[1][\ttfamily
1402 \hyphenchar\font=45\relax% to przypisanie jest globalne do fontu.
1403 \@tempskipa=\glueexpr\rightskip+\leftskip\relax
1404 \ifdim\gluestretch\@tempskipa=\z@
1405 \tolerance900

sprawdziło się przy tolerancji 900

1406 \advance\rightskip_\by\z@_pluso,5em\relax\fi
1407 \fontdimen3\font=\z@% zabraniamy rozciągania odstępów, ale% \fontdimen4%
\font=\z@ dopuszczamy ich skurczenie
1408 \hyphenpenaltyo_\% żeby nie stresować TEXa: w maszynopisie ten wspaniały al-
gorytm dzielenia akapitu powinien być wyłączony, a każdy wiersz łamany
na ostatnim dopuszczalnym miejscu przełamania.

1409 \StoreMacro\pauzacore
\pauzacore 1410 \def\pauzacore{-\rlap{\kern-o,3em-}-}%
1411 }\{\par}

\justified 1412 \newcommand*\justified{%
1413 \leftskip=1\leftskip% to preserve the natural length and discard stretch and
shrink.
1414 \rightskip=1\rightskip
1415 \parfillskip=1\parfillskip
1416 \advance\parfillskip_\by_osp_plus_\ifil\relax
1417 \let\\\@normalcr}

For dati under poems.

\wherncore 1418 \newcommand\wherncore[1]{%
1419 \rightline{%
1420 \parbox{o,7666\textwidth}{
1421 \leftskiposp_plus_\textwidth
1422 \parfillskiposp\relax
1423 \let\\\linebreak
1424 \footnotesize_\#1}}}

```

```

\whern 1425 \newcommand\whern[1]{%
1426 \vskip\whernskip
1427 \wherncore{#1}}

\whernskip 1428 \newskip\whernskip
1429 \whernskip2\baselineskip\minus2\baselineskip\relax

\whernup 1430 \newcommand\whernup[1]{\par\wherncore{#1}}

```

## Minion and Garamond Premier kerning and ligature fixes

„Ws” nie będzie robiło długiego „s”, bo źle wygląda przy „W”

```

\Ws 1431 \DeclareRobustCommand*\Ws{W\kern-0,08em\penalty10000\hskiposp%
\relax
1432 s\penalty10000\hskiposp\relax}

\Wz 1433 \DeclareRobustCommand*\Wz{W\kern-0,05em\penalty10000\hskiposp%
\relax_z}

1434 \endinput

```

## Change History

- |  |   |
|--|---|
| <p>vo.74<br/>General:<br/>Added macros to make sectioning commands of mwcls and standard classes compatible. Now my sectionings allow two optionals in both worlds and with mwcls if there's only one optional, it's the title to toc and running head not just to the latter, <a href="#">48</a></p> <p>\begin:<br/>The catcodes of \begin and \end argument(s) don't have to agree strictly anymore: an environment is properly closed if the \begin's and \end's arguments result in the same \cename, <a href="#">8</a></p> <p>vo.75<br/>\@ifnextac:<br/>added, <a href="#">6</a><br/>\@ifnextcat:<br/>\let for #1 changed to \def to allow things like \noexpand~ , <a href="#">5</a><br/>\@ifnextif:<br/>\let for #1 changed to \def to allow things like \noexpand~ , <a href="#">5</a></p> <p>vo.76<br/>General:<br/>A 'fixing' of \dots was rolled back since it came out they were O.K. and that was the QX encoding that prints them very tight, <a href="#">48</a></p> | <p>\freeze@actives:<br/>added, <a href="#">33</a></p> <p>vo.77<br/>General:<br/>\afterfi &amp; pals made two-argument as the Marcin Woliński's analogoi are. At this occasion some redundant macros of that family are deleted, <a href="#">48</a></p> <p>vo.78<br/>General:<br/>\@namelet renamed to \n@melet to solve a conflict with the beamer class. The package contents regrouped, <a href="#">48</a></p> <p>vo.79<br/>\not@onlypreamble:<br/>All the actions are done in a group and therefore \xdef used instead of \edef because this command has to use \do (which is contained in the \@preamblecmds list) and \not@onlypreamble itself should be able to be let to \do, <a href="#">16</a></p> <p>vo.80<br/>General:<br/>Checksum 1689, <a href="#">1</a><br/>\hfillneg:<br/>added, <a href="#">32</a></p> <p>vo.81<br/>\dekfracslash:<br/>moved here from pmlectionis.cls, <a href="#">35</a><br/>\ifSecondClass:</p> |
|--|---|



- moved here from pmlectionis.cls, [35](#)
- vo.82
  - `\ikern`:
  - added, [35](#)
- vo.83
  - `\~`:
  - postponed to `\begin{document}` to
  - avoid overwriting by a text command
  - and made sensible to a subsequent `/`, [31](#)
- vo.84
  - General:
  - Checksum 2684, [1](#)
- vo.85
  - General:
  - Checksum 2795, [1](#)
  - fixed behaviour of too clever headings
  - with gmdoc by adding an `\ifdim` test, [48](#)
- vo.86
  - `\texttilde`:
  - renamed from
- texttilde since the latter is one of L<sup>A</sup>T<sub>E</sub>X
- accents, [32](#)
- vo.87
  - General:
  - Checksum 4027, [1](#)
  - the package goes  $\varepsilon$ -T<sub>E</sub>X even more,
  - making use of `\ifdefined` and the
  - code using UTF-8 chars is wrapped in
  - a X<sub>Y</sub>T<sub>E</sub>X-condition, [48](#)
- vo.88
  - General:
  - Checksum 4040, [1](#)
  - `\RestoreEnvironment`:
  - added, [15](#)
  - `\storedcsname`:
  - added, [15](#)
  - `\StoreEnvironment`:
  - added, [15](#)
- vo.89
  - General:
  - removed obsolete adjustment of pgf for
  - X<sub>Y</sub>T<sub>E</sub>X, [48](#)
- vo.90
  - General:
  - Checksum 4035, [1](#)
  - `\XeTeXthree`:
  - adjusted to the redefinition of `\verb` in
  - xlxtra 2008/07/29, [29](#)
- vo.91
  - General:
  - Checksum 4055, [1](#)
  - removed `\jobnamewoe` since
  - `\jobname` is always without
  - extension. `\xiispace` forked to
  - `\visiblepace \let` to
  - `\xxt@visiblepace` of xlxtra if
  - available. The documentation driver
  - integrated with the .sty file, [48](#)

## Index

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

- |  |   |  |
|--|---|--|
| <code>\*</code> , <a href="#">90</a> , <a href="#">91</a> , <a href="#">92</a> , <a href="#">94</a> , <a href="#">96</a> , <a href="#">97</a> , <a href="#">98</a> ,<br><a href="#">102</a> , <a href="#">841</a> , <a href="#">895</a>  | <code>\@clsextension</code> , <a href="#">945</a>   | <code>\@ifnextac</code> , <a href="#">86</a>   |
| <code>\+</code> , <a href="#">896</a>  | <code>\@clubpenalty</code> , <a href="#">501</a>  | <code>\@ifnextcat</code> , <a href="#">50</a> , <a href="#">87</a>   |
| <code>\-</code> , <a href="#">228</a> , <a href="#">1141</a>   | <code>\@currenvir</code> , <a href="#">113</a> , <a href="#">122</a>  | <code>\@ifnextif</code> , <a href="#">68</a>   |
| <code>\&lt;...&gt;</code> , <a href="#">208</a>  | <code>\@currenvline</code> , <a href="#">114</a>  | <code>\@ifnextspace</code> , <a href="#">95</a> , <a href="#">1089</a> , <a href="#">1111</a>  |
| <code>\@@nil</code> , <a href="#">213</a> , <a href="#">214</a> , <a href="#">216</a> , <a href="#">854</a> ,<br><a href="#">964</a> , <a href="#">972</a> , <a href="#">1018</a> , <a href="#">1023</a> ,<br><a href="#">1025</a> , <a href="#">1026</a> , <a href="#">1028</a> ,<br><a href="#">1373</a> , <a href="#">1377</a> , <a href="#">1397</a> | <code>\@currsize</code> , <a href="#">128</a> , <a href="#">129</a> , <a href="#">130</a> ,<br><a href="#">131</a> , <a href="#">132</a> , <a href="#">133</a> , <a href="#">134</a> ,<br><a href="#">135</a> , <a href="#">136</a> , <a href="#">137</a> | <code>\@ifnif</code> , <a href="#">72</a> , <a href="#">73</a>   |
| <code>\@M</code> , <a href="#">496</a>   | <code>\@emptify</code> , <a href="#">34</a> , <a href="#">35</a> , <a href="#">36</a>   | <code>\@ifnotmw</code> , <a href="#">398</a> , <a href="#">400</a> , <a href="#">488</a> ,<br><a href="#">552</a> , <a href="#">588</a> , <a href="#">620</a>  |
| <code>\@afterheading</code> , <a href="#">489</a>  | <code>\@enumctr</code> , <a href="#">626</a> , <a href="#">627</a> , <a href="#">633</a>  | <code>\@include</code> , <a href="#">970</a> , <a href="#">971</a>   |
| <code>\@badend</code> , <a href="#">123</a>  | <code>\@enumdepth</code> , <a href="#">622</a> , <a href="#">625</a> , <a href="#">626</a>  | <code>\@itemdepth</code> , <a href="#">638</a> , <a href="#">641</a> , <a href="#">642</a>   |
| <code>\@begnamedgroup</code> , <a href="#">109</a> ,<br><a href="#">110</a> , <a href="#">116</a> , <a href="#">119</a>  | <code>\@fileswithoptions</code> , <a href="#">945</a>   | <code>\@itemitem</code> , <a href="#">642</a> , <a href="#">643</a>  |
| <code>\@car</code> , <a href="#">690</a>   | <code>\@firstofmany</code> , <a href="#">854</a>  | <code>\@minus</code> , <a href="#">561</a> , <a href="#">565</a> , <a href="#">568</a> , <a href="#">570</a> ,<br><a href="#">573</a> , <a href="#">575</a> , <a href="#">579</a> , <a href="#">583</a>  |
| <code>\@cclv</code> , <a href="#">887</a>  | <code>\@gif</code> , <a href="#">15</a> , <a href="#">16</a> , <a href="#">18</a>   | <code>\@nbreakfalse</code> , <a href="#">495</a> , <a href="#">864</a>   |
| <code>\@cclvi</code> , <a href="#">877</a>   | <code>\@ifXeTeX</code> , <a href="#">752</a> , <a href="#">757</a> , <a href="#">805</a> ,<br><a href="#">830</a> , <a href="#">941</a> , <a href="#">953</a> , <a href="#">1032</a> , <a href="#">1151</a>   | <code>\@nbreaktrue</code> , <a href="#">490</a>  |
| <code>\@checkend</code> , <a href="#">120</a>  | <code>\@ifempty</code> , <a href="#">461</a> , <a href="#">472</a> , <a href="#">959</a>  | <code>\@normalcr</code> , <a href="#">1417</a>   |
|  | <code>\@ifl@aded</code> , <a href="#">352</a>   | <code>\@nx</code> , <a href="#">11</a> , <a href="#">45</a> , <a href="#">59</a> , <a href="#">77</a> , <a href="#">89</a> , <a href="#">264</a> ,<br><a href="#">265</a> , <a href="#">280</a> , <a href="#">303</a> , <a href="#">304</a> ,<br><a href="#">338</a> , <a href="#">339</a> , <a href="#">343</a> , <a href="#">344</a> , |
|  | <code>\@ifncat</code> , <a href="#">54</a> , <a href="#">55</a> , <a href="#">67</a>  |  |

347, 356, 358, 359,  
 360, 546, 547, 661,  
 661, 744, 921, 1155,  
 1161, 1257, 1258  
 \@oarg, 237, 238, 239  
 \@oargsq, 237, 239, 247  
 \@onlypreamble, 865  
 \@parg, 240, 241, 242  
 \@pargp, 240, 242, 248  
 \@parindent, 629, 630,  
 632, 645, 646, 648  
 \@pkgextension, 353  
 \@preamblecmds, 348,  
 349, 357, 361  
 \@relaxen, 38, 39, 40  
 \@starttoc, 859  
 \@tempdima, 1039, 1042,  
 1043, 1045, 1046, 1050  
 \@tempdima\*, 1043, 1045  
 \@tempdimb, 1040, 1041, 1042  
 \@textsuperscript, 836, 837  
 \@toodeep, 623, 639  
 \@topnewpage, 444  
 \@undefined, 5  
 \@whilenum, 877, 887  
 \@xa, 10, 19, 21, 28, 33, 43,  
 67, 121, 122, 142, 201,  
 234, 259, 264, 265,  
 298, 303, 304, 324,  
 335, 338, 339, 343,  
 344, 483, 507, 544,  
 599, 605, 627, 643,  
 656, 742, 744, 748,  
 861, 902, 1016, 1061,  
 1064, 1155, 1161,  
 1257, 1257  
 \@xifncat, 57, 67  
 \@xifnif, 75  
 \acro, 910, 931, 932, 934  
 \acrocore, 923, 930  
 \adashes, 1130, 1130, 1131,  
 1136  
 \addfontfeature, 767,  
 779, 781, 800, 832,  
 933, 1056, 1278, 1289  
 \addto@macro, 27, 31  
 \addtoheading, 481  
 \addtomacro, 31, 1269  
 \addtonummacro, 381  
 \addtotoks, 32  
 \AE, 1030  
 \afterfi, 89, 99, 100, 103,  
 211, 225, 245, 258,  
 297, 755, 917, 928,  
 961, 962, 1005, 1023,  
 1088, 1110  
 \afterfifi, 104, 285, 288,  
 754, 1139  
 \afterfififi, 108  
 \afteriffifi, 105  
 \afteriffififi, 107  
 \afterififfififi, 106  
 \ahyphen, 1141  
 \AKA, 932  
 \all@other, 742, 793, 795  
 \alpha, 1165  
 \AmSTeX, 715  
 \arg, 244, 245  
 \AtBeginDocument, 193,  
 243, 355, 385, 761,  
 764, 842, 903, 1084,  
 1130, 1131  
 \begin, 118, 119  
 \begin\*, 119  
 \bgcolor, 1281  
 \BibTeX, 717  
 \Biggl, 1245  
 \biggl, 1243  
 \Biggr, 1246  
 \biggr, 1244  
 \Bigl, 1241  
 \bigl, 1239  
 \Bigr, 1242  
 \bigr, 1240  
 \bigskipamount, 852, 1389  
 \bnamegroup, 116  
 \boldmath, 690  
 \box, 705, 1238  
 \bslash, 186, 227, 264,  
 304, 315, 478, 479,  
 480, 1107, 1108, 1116,  
 1117  
 \bullet, 1151  
 \c@gm@PronounGender, 364  
 \c@NoNumSecs, 386  
 \c@secnumdepth, 406  
 \chardef, 1151  
 \cite, 363  
 \ClassError, 477  
 \cleardoublepage, 397  
 \clubpenalty, 496, 501  
 \cmd, 234  
 \cmd@to@cs, 234, 235  
 \color, 893  
 \continuum, 908  
 \copy, 674, 697, 711, 1280, 1291  
 \count, 382, 383, 384, 679,  
 680, 681, 682, 683,  
 684, 685, 686, 810,  
 811, 812, 813, 817,  
 818, 819, 820, 822,  
 823, 824, 876, 877,  
 878, 879, 882, 886,  
 887, 888, 889  
 \cs, 227, 230, 232, 234  
 \cup, 1288  
 \czas, 1142  
 \czer, 894, 895, 896  
 \czerwo, 893, 894  
 \dag, 896  
 \data, 1384  
 \date@biway, 1387, 1392  
 \date@line, 1387, 1394, 1395  
 \datef, 1298, 1333, 1385,  
 1393, 1397  
 \datefsl, 1314, 1352, 1385,  
 1393  
 \dateskip, 1386, 1391  
 \deadcycles, 999  
 \DeclareLogo, 655, 665,  
 689, 694, 717, 720,  
 722, 723, 724, 726,  
 728, 729, 731, 737  
 \DeclareRobustCommand,  
 1027  
 \DeclareRobustCommand\*,  
 124, 152, 153, 154,  
 155, 156, 157, 199,  
 217, 219, 226, 227,  
 230, 232, 389, 664,  
 766, 835, 841, 843,  
 910, 933, 1085, 1095,  
 1103, 1105, 1114,  
 1399, 1400, 1431, 1433  
 \DeclareTextCommand, 661  
 \DeclareTextCommandDefault,  
 663  
 \defobeylines, 848  
 \dekbigskip, 852  
 \dekfracc, 791, 799, 951  
 \dekfraccsimple, 950, 955  
 \dekfraccslash, 948,  
 953, 954  
 \dekmedskip, 851  
 \deksmallskip, 849  
 \Delta, 1166  
 \dimen, 809, 812, 816, 819,  
 868, 869, 870, 1071, 1072  
 \dimexpr, 1189, 1194, 1200,  
 1237, 1281

<code>\discre</code> , <a href="#">209</a> , <a href="#">212</a> , <a href="#">218</a>	<code>\g@empty</code> , <a href="#">36</a> , <a href="#">37</a>	<code>\gml@storeCS</code> , <a href="#">271</a> , <a href="#">287</a> , <a href="#">310</a>
<code>\discret</code> , <a href="#">210</a> , <a href="#">211</a>	<code>\g@relaxen</code> , <a href="#">40</a> , <a href="#">41</a>	<code>\gml@storemacros</code> , <a href="#">272</a> ,
<code>\divide</code> , <a href="#">681</a> , <a href="#">684</a> , <a href="#">685</a> , <a href="#">811</a> ,	<code>\gaddtomacro</code> , <a href="#">26</a>	<a href="#">279</a> , <a href="#">285</a> , <a href="#">288</a> , <a href="#">311</a>
<a href="#">813</a> , <a href="#">818</a> , <a href="#">820</a> , <a href="#">1041</a> ,	<code>\garamath</code> , <a href="#">1277</a>	<code>\gmobeyspaces</code> , <a href="#">846</a>
<a href="#">1042</a> , <a href="#">1047</a>	<code>\gempty</code> , <a href="#">37</a> , <a href="#">37</a>	<code>\gmshowlists</code> , <a href="#">42</a>
<code>\dp</code> , <a href="#">1223</a> , <a href="#">1234</a> , <a href="#">1237</a> , <a href="#">1281</a>	<code>\glet</code> , <a href="#">25</a> , <a href="#">508</a>	<code>\GMtextsuperscript</code> , <a href="#">829</a>
<code>\dywiz</code> , <a href="#">1139</a>	<code>\glueexpr</code> , <a href="#">1403</a>	<code>\gmu@acroinner</code> , <a href="#">914</a> , <a href="#">919</a> ,
<code>\edef@other</code> , <a href="#">747</a> , <a href="#">750</a>	<code>\gluestretch</code> , <a href="#">1404</a>	<a href="#">920</a> , <a href="#">928</a>
<code>\eequals</code> , <a href="#">874</a>	<code>\gm@atppron</code> , <a href="#">365</a> , <a href="#">368</a> ,	<code>\gmu@acrospace</code> , <a href="#">910</a> ,
<code>\egRestore@Macro</code> , <a href="#">292</a> , <a href="#">293</a>	<a href="#">369</a> , <a href="#">370</a> , <a href="#">371</a> , <a href="#">372</a> ,	<a href="#">913</a> , <a href="#">913</a> , <a href="#">917</a>
<code>\egRestore@MacroSt</code> ,	<a href="#">373</a> , <a href="#">374</a> , <a href="#">375</a>	<code>\gmu@checkaftersec</code> ,
<a href="#">292</a> , <a href="#">294</a>	<code>\gm@clearpagesduetoopenright</code> ,	<a href="#">504</a> , <a href="#">546</a>
<code>\egroupfirstofone</code> , <a href="#">159</a>	<a href="#">396</a> , <a href="#">415</a>	<code>\gmu@copyright</code> , <a href="#">938</a> , <a href="#">939</a>
<code>\egStore@Macro</code> , <a href="#">253</a> , <a href="#">254</a>	<code>\gm@dontnumbersectionsoutofrange</code> , <a href="#">394</a> , <a href="#">407</a>	<code>\gm@dashfalse</code> , <a href="#">1378</a>
<code>\egStore@MacroSt</code> , <a href="#">253</a> , <a href="#">255</a>	<code>\gm@duppa</code> , <a href="#">792</a> , <a href="#">793</a> , <a href="#">795</a>	<code>\gm@dashtrue</code> , <a href="#">1380</a>
<code>\emdash</code> , <a href="#">1121</a>	<code>\gm@gobmacro</code> , <a href="#">742</a> , <a href="#">744</a>	<code>\gm@def</code> , <a href="#">399</a> , <a href="#">401</a> , <a href="#">401</a> , <a href="#">402</a>
<code>\empty</code> , <a href="#">35</a> , <a href="#">35</a> , <a href="#">769</a> ,	<code>\gm@hyperrefstepcounter</code> ,	<code>\gm@dekfracc</code> , <a href="#">778</a> , <a href="#">794</a> , <a href="#">796</a>
<a href="#">1265</a> , <a href="#">1268</a>	<a href="#">388</a> , <a href="#">391</a> , <a href="#">426</a>	<code>\gm@dekfraccsimple</code> ,
<code>\enamegroup</code> , <a href="#">117</a>	<code>\gm@ifnac</code> , <a href="#">87</a> , <a href="#">88</a>	<a href="#">796</a> , <a href="#">946</a> , <a href="#">951</a>
<code>\endlist</code> , <a href="#">636</a> , <a href="#">651</a>	<code>\gm@letspace</code> , <a href="#">93</a> , <a href="#">99</a>	<code>\gm@denominatorkern</code> ,
<code>\englishdate</code> , <a href="#">1332</a>	<code>\gm@notprerr</code> , <a href="#">354</a> , <a href="#">360</a>	<a href="#">780</a> , <a href="#">804</a> , <a href="#">948</a>
<code>\enoughpage</code> , <a href="#">866</a>	<code>\gm@PronounGender</code> , <a href="#">364</a>	<code>\gm@discretionaryslash</code> ,
<code>\ensuremath</code> , <a href="#">200</a> , <a href="#">205</a> ,	<code>\gm@pswords</code> , <a href="#">213</a> , <a href="#">214</a> , <a href="#">216</a>	<a href="#">218</a> , <a href="#">224</a>
<a href="#">727</a> , <a href="#">838</a>	<code>\gm@sec</code> , <a href="#">608</a> , <a href="#">615</a> , <a href="#">616</a>	<code>\gm@dywiz</code> , <a href="#">1138</a> , <a href="#">1141</a>
<code>enumerate*</code> , <a href="#">621</a>	<code>\gm@secini</code> , <a href="#">589</a> , <a href="#">599</a> , <a href="#">602</a> ,	<code>\gm@fileext</code> , <a href="#">966</a> , <a href="#">973</a> ,
<code>\env</code> , <a href="#">230</a>	<a href="#">605</a> , <a href="#">613</a>	<a href="#">973</a> , <a href="#">991</a>
<code>\equals</code> , <a href="#">873</a>	<code>\gm@secmarkh</code> , <a href="#">603</a>	<code>\gm@filename</code> , <a href="#">965</a> , <a href="#">976</a> ,
<code>\eTeX</code> , <a href="#">726</a> , <a href="#">728</a>	<code>\gm@secstar</code> , <a href="#">591</a> , <a href="#">597</a> ,	<a href="#">988</a> , <a href="#">991</a> , <a href="#">994</a> , <a href="#">1000</a>
<code>\everypar</code> , <a href="#">492</a> , <a href="#">502</a> , <a href="#">1081</a>	<a href="#">600</a> , <a href="#">606</a> , <a href="#">615</a> , <a href="#">616</a>	<code>\gm@getaddvs</code> , <a href="#">540</a> , <a href="#">540</a> , <a href="#">544</a>
<code>\exii@currenvir</code> , <a href="#">122</a> , <a href="#">123</a>	<code>\gm@secx</code> , <a href="#">608</a> , <a href="#">609</a>	<code>\gm@getext</code> , <a href="#">964</a> , <a href="#">972</a>
<code>\exists</code> , <a href="#">1201</a>	<code>\gm@secxx</code> , <a href="#">590</a> , <a href="#">604</a> , <a href="#">610</a>	<code>\gm@ifnodash</code> , <a href="#">1373</a> , <a href="#">1377</a>
<code>\f@encoding</code> , <a href="#">884</a>	<code>\gm@straightensec</code> , <a href="#">611</a> , <a href="#">618</a>	<code>\gm@LastSkip</code> , <a href="#">1147</a> ,
<code>\f@series</code> , <a href="#">690</a>	<code>\gm@targetheading</code> , <a href="#">389</a> , <a href="#">392</a>	<a href="#">1148</a> , <a href="#">1150</a>
<code>\fakern</code> , <a href="#">1261</a>	<code>\gma</code> , <a href="#">1275</a>	<code>\gm@luzniej</code> , <a href="#">1073</a> , <a href="#">1075</a> ,
<code>\fakesc@extrascap</code> ,	<code>\gma@arrowdash</code> , <a href="#">1279</a> ,	<a href="#">1076</a>
<a href="#">1045</a> , <a href="#">1058</a>	<a href="#">1287</a> , <a href="#">1293</a> , <a href="#">1294</a>	<code>\gm@nl@reserveda</code> , <a href="#">337</a> ,
<code>\fakesc@cap</code> , <a href="#">1027</a>	<code>\gma@bare</code> , <a href="#">1271</a> , <a href="#">1273</a>	<a href="#">340</a> , <a href="#">342</a> , <a href="#">345</a>
<code>\fakesc@score</code> , <a href="#">1011</a> , <a href="#">1029</a>	<code>\gma@bracket</code> , <a href="#">1273</a> , <a href="#">1274</a>	<code>\gm@numeratorkern</code> ,
<code>\fakesc@extrascap</code> , <a href="#">1058</a>	<code>\gma@checkbracket</code> ,	<a href="#">779</a> , <a href="#">803</a> , <a href="#">804</a> , <a href="#">947</a>
<code>\file</code> , <a href="#">219</a>	<a href="#">1272</a> , <a href="#">1276</a>	<code>\gm@prevsec</code> , <a href="#">491</a> , <a href="#">493</a> ,
<code>\fixcopyright</code> , <a href="#">940</a>	<code>\gma@dollar</code> , <a href="#">1270</a> , <a href="#">1271</a> , <a href="#">1276</a>	<a href="#">507</a> , <a href="#">512</a> , <a href="#">536</a>
<code>\fontencoding</code> , <a href="#">935</a>	<code>\gma@gmathhook</code> , <a href="#">1263</a> ,	<code>\gm@printsashes</code> , <a href="#">219</a> ,
<code>\fooatletter</code> , <a href="#">161</a>	<a href="#">1268</a> , <a href="#">1269</a> , <a href="#">1282</a>	<a href="#">220</a> , <a href="#">220</a> , <a href="#">222</a> , <a href="#">225</a>
<code>\foone</code> , <a href="#">159</a> , <a href="#">161</a> , <a href="#">164</a> , <a href="#">166</a> ,	<code>\gma@quantifierhook</code> ,	<code>\gm@resa</code> , <a href="#">900</a> , <a href="#">902</a>
<a href="#">172</a> , <a href="#">178</a> , <a href="#">187</a> , <a href="#">189</a> ,	<a href="#">1198</a> , <a href="#">1201</a> , <a href="#">1265</a> ,	<code>\gm@reserveda</code> , <a href="#">90</a> , <a href="#">92</a> ,
<a href="#">191</a> , <a href="#">231</a> , <a href="#">845</a> , <a href="#">847</a> ,	<a href="#">1267</a> , <a href="#">1289</a> , <a href="#">1295</a>	<a href="#">96</a> , <a href="#">98</a> , <a href="#">280</a> , <a href="#">281</a> , <a href="#">284</a> ,
<a href="#">1129</a> , <a href="#">1140</a> , <a href="#">1213</a>	<code>\gma@tempa</code> , <a href="#">1187</a> , <a href="#">1189</a> ,	<a href="#">483</a> , <a href="#">485</a> , <a href="#">508</a> , <a href="#">509</a> ,
<code>\forall</code> , <a href="#">1198</a>	<a href="#">1192</a> , <a href="#">1194</a> , <a href="#">1233</a> ,	<a href="#">510</a> , <a href="#">656</a> , <a href="#">658</a> , <a href="#">660</a> ,
<code>\FormatHangHeading</code> ,	<a href="#">1237</a> , <a href="#">1256</a> , <a href="#">1259</a>	<a href="#">661</a> , <a href="#">662</a> , <a href="#">743</a> , <a href="#">745</a> ,
<a href="#">560</a> , <a href="#">566</a> , <a href="#">571</a> , <a href="#">576</a>	<code>\gma@tempb</code> , <a href="#">1234</a> , <a href="#">1237</a>	<a href="#">747</a> , <a href="#">748</a> , <a href="#">750</a> , <a href="#">751</a> ,
<code>\FormatRunInHeading</code> ,	<code>\gmath</code> , <a href="#">1154</a> , <a href="#">1270</a> , <a href="#">1274</a>	<a href="#">960</a> , <a href="#">961</a>
<a href="#">580</a> , <a href="#">584</a>	<code>\gmathhook</code> , <a href="#">1269</a>	<code>\gm@RPif</code> , <a href="#">892</a> , <a href="#">898</a> , <a href="#">904</a> , <a href="#">908</a>
<code>\freeze@actives</code> , <a href="#">875</a>		<code>\gm@scalar</code> , <a href="#">1033</a> , <a href="#">1037</a> ,
		<a href="#">1038</a> , <a href="#">1043</a>

<code>\gmu@scalematchX</code> , 1029, 1035, 1057	<code>\himher</code> , 370	<code>\leftmargin</code> , 628, 644
<code>\gmu@scapLetters</code> , 1007, 1016, 1020	<code>\HisHer</code> , 373	<code>\leftrightharpoon</code> , 1209
<code>\gmu@scapSpaces</code> , 1018, 1023, 1026	<code>\hisher</code> , 369	<code>\linebreak</code> , 1423
<code>\gmu@scapss</code> , 1025, 1028	<code>\HisHers</code> , 375	<code>\linedate</code> , 1386, 1394, 1395
<code>\gmu@scscale</code> , 1051, 1056	<code>\hishers</code> , 371	<code>\list</code> , 627, 643
<code>\gmu@setheading</code> , 543, 548, 549	<code>\hrule</code> , 528	<code>\listparindent</code> , 632, 648
<code>\gmu@setsetSMglobal</code> , 270, 273, 309	<code>\hyphenpenalty</code> , 216, 1091, 1408	<code>\liturgiques</code> , 891
<code>\gmu@setSMglobal</code> , 275, 277, 288	<code>\if@afterindent</code> , 497	<code>\longpauza</code> , 1123, 1124
<code>\gmu@SMdo@scope</code> , 323, 325, 327, 328, 334	<code>\if@filesw</code> , 861, 975, 987, 995	<code>\looseness</code> , 1077, 1081
<code>\gmu@SMdo@setscope</code> , 321, 326, 332	<code>\if@mainmatter</code> , 395	<code>\lpauza</code> , 1103
<code>\gmu@SMglobalfalse</code> , 260, 267, 277, 282, 299, 306, 330	<code>\if@nobreak</code> , 494	<code>luzniej</code> , 1078
<code>\gmu@SMglobaltrue</code> , 251, 275	<code>\if@openright</code> , 397	<code>luzniej*</code> , 1079
<code>\gmu@smtempa</code> , 262, 266, 301, 305	<code>\if@specialpage</code> , 414	<code>\luzniejcore</code> , 1074, 1078
<code>\gmu@tempa</code> , 1060, 1062, 1063, 1064	<code>\if@twoside</code> , 437	<code>\macro</code> , 744
<code>\gmu@testdash</code> , 1376, 1385, 1393	<code>\ifdate</code> , 1383, 1385, 1386	<code>\MakeUppercase</code> , 1011
<code>\gmu@tilde</code> , 839, 841, 844	<code>\ifdefined</code> , 7, 168, 194, 753, 1083, 1128	<code>\mapsto</code> , 1286
<code>\gmu@whonly</code> , 1004, 1005	<code>\iffontchar</code> , 773	<code>\marg</code> , 236, 248
<code>\gmu@xedekfracplain</code> , 770, 798	<code>\ifgmu@dash</code> , 1372, 1376, 1382, 1385, 1393	<code>maszynopis</code> , 1401
<code>\gmu@xedekfracstar</code> , 770, 771	<code>\ifgmu@postsec</code> , 506, 535, 539	<code>\math@arg</code> , 244, 245
<code>\gmu@xefraccddef</code> , 772, 782, 783, 784, 785, 786, 787, 788, 789, 790	<code>\ifgmu@SMglobal</code> , 250, 258, 263, 274, 297, 302, 327	<code>\mathbin</code> , 1178, 1182, 1205, 1206, 1227, 1228, 1254, 1255, 1288, 1295
<code>\gn@melet</code> , 341	<code>\ifHeadingNumbered</code> , 405, 421	<code>\mathchoice</code> , 1185, 1203, 1214, 1247, 1284
<code>\gobble</code> , 162, 1257	<code>\ifodd</code> , 367	<code>\mathclose</code> , 1225, 1226, 1240, 1242, 1244, 1246, 1253
<code>\gobbletwo</code> , 163	<code>\ifSecondClass</code> , 943	<code>\mathfrak</code> , 908
<code>\grefstepcounter</code> , 23	<code>\ikern</code> , 956	<code>\mathit</code> , 1155, 1164
<code>\grelaxen</code> , 41, 41, 493	<code>\IMO</code> , 931	<code>\mathop</code> , 1185
<code>\hathat</code> , 232	<code>\in</code> , 1295	<code>\mathopen</code> , 1218, 1226, 1239, 1241, 1243, 1245, 1252
<code>\HeadingNumber</code> , 423, 425	<code>\inclasthook</code> , 992, 1006	<code>\mathpunct</code> , 1217
<code>\HeadingNumberedfalse</code> , 395, 406	<code>\includegraphics</code> , 828	<code>\mathrel</code> , 1179, 1183, 1207, 1208, 1209, 1227, 1228, 1229, 1258, 1287, 1293, 1294
<code>\HeadingRHeadText</code> , 409	<code>\infty</code> , 1177	<code>\mathrm</code> , 1161, 1166, 1172, 1178, 1179, 1181, 1182, 1183, 1196
<code>\HeadingText</code> , 411	<code>\itemindent</code> , 629, 645	<code>\Mathstrutbox@</code> , 1223
<code>\HeadingTOCText</code> , 410	<code>itemize*</code> , 637	<code>\medmuskip</code> , 211
<code>\HeShe</code> , 372	<code>\iteracro</code> , 909, 912	<code>\meta</code> , 199, 208, 236, 238, 241
<code>\heshe</code> , 368	<code>\justified</code> , 1412	<code>\meta@font@select</code> , 203, 207
<code>\hfillneg</code> , 853	<code>\labelsep</code> , 631, 647	<code>\mkern</code> , 1261
<code>\HimHer</code> , 374	<code>\labelwidth</code> , 630, 631, 646, 647	<code>\mskip</code> , 211
	<code>\larger</code> , <i>p.</i> 8, 152, 1188, 1193, 1239, 1240, 1243, 1244, 1245, 1246	<code>\multiply</code> , 680, 683, 1049, 1076, 1080
	<code>\largerr</code> , <i>p.</i> 8, 156, 1241, 1242	<code>\mw@getflags</code> , 507
	<code>\LaTeXe</code> , 653, 689	<code>\mw@HeadingBreakAfter</code> , 416, 433, 448, 452, 460, 508
	<code>\LaTeXpar</code> , 694	
	<code>\ldate</code> , 1395, 1398	
	<code>\leftarrow</code> , 1207, 1293	
	<code>\leftline</code> , 1395	

<code>\mw@HeadingBreakBefore,</code> 413, <u>459</u> , 509	<code>\PackageError</code> , 359, 1106, 1115	<code>\Restore@MacroSt</code> , 294, <u>300</u>
<code>\mw@HeadingLevel</code> , 403, 406	<code>\PackageWarning</code> , 148, 150	<code>\RestoreEnvironment</code> , <u>318</u>
<code>\mw@HeadingRunIn</code> , 428, <u>459</u>	<code>\pagebreak</code> , 436, 448, 452	<code>\RestoreMacro</code> , <u>291</u> , 401, 760, 762
<code>\mw@HeadingType</code> , <u>412</u> , 491, 513, 514, 525	<code>\pagegoal</code> , 868	<code>\RestoreMacro*</code> , 319, 762
<code>\mw@HeadingWholeWidth</code> , 431, <u>460</u>	<code>\pagetotal</code> , 869	<code>\RestoreMacros</code> , <u>307</u>
<code>\mw@normalheading</code> , 435, 444, 447, 451, 548	<code>\paragraph</code> , 1397	<code>\RestoringDo</code> , <u>331</u>
<code>\mw@processflags</code> , 461	<code>\ParanoidPostsec</code> , <u>534</u>	<code>\rightarrow</code> , 1208, 1294
<code>\mw@runinheading</code> , 429, 549	<code>\parg</code> , <u>240</u>	<code>\rightline</code> , 1394, 1419
<code>\mw@secdef</code> , 464, 465, 466, 471	<code>\partial</code> , 1180	<code>\romannumeral</code> , 626, 642
<code>\mw@section</code> , 463	<code>\partopsep</code> , 628, 644	<code>\rotatebox</code> , 1198, 1201, 1205, 1206
<code>\mw@sectionxx</code> , 402	<code>\pauza</code> , <u>1095</u>	<code>\rs@size@warning</code> , 141, 146, <u>148</u>
<code>\mw@secundef</code> , 468, 473, <u>475</u>	<code>\pauzacore</code> , <u>1090</u> , 1091, 1093, 1099, 1101, 1104, <u>1123</u> , <u>1126</u> , 1409, <u>1410</u>	<code>\rs@unknown@warning</code> , 138, <u>150</u>
<code>\mw@setflags</code> , 469	<code>\pdfTeX</code> , <u>728</u>	<code>\runindate</code> , <u>1396</u>
 	<code>\pdfTeX</code> , <u>729</u>	
<code>\n@melet</code> , <u>336</u> , 482, 486, 592, 595, 613, 776	<code>\Phi</code> , 1172	<code>\scantokens</code> , 1212
<code>\nameshow</code> , <u>43</u>	<code>\phi</code> , 1171	<code>\scshape</code> , 721, 930, 1288
<code>napapierki</code> , <u>1069</u>	<code>\pi</code> , 1170	<code>\secondclass</code> , <u>942</u>
<code>\napapierkicore</code> , <u>1067</u> , 1070	<code>\pk</code> , <u>226</u>	<code>\SecondClasstrue</code> , 944
<code>\napapierkistretch</code> , <u>1066</u> , 1068	<code>\PlainTeX</code> , <u>722</u>	<code>\sectionsign</code> , 1159
<code>\nawj</code> , <u>1082</u>	<code>\pm</code> , 1181, 1182	<code>\SetSectionFormatting</code> , 456, 457, 554, 557, 564, 569, 574, 578, 582
<code>\nazwired</code> , <u>1153</u>	<code>\polskadata</code> , <u>1297</u> , 1331	<code>\SetTwoheadSkip</code> , <u>550</u> , 563, 568, 573
<code>\neg</code> , 1178, 1260	<code>\possfil</code> , <u>233</u>	<code>\sfname</code> , <u>217</u> , 221
<code>\neq</code> , 1179, 1255	<code>\ppauza</code> , <u>1114</u>	<code>\shortpauza</code> , <u>1125</u>
<code>\neqb</code> , <u>1255</u>	<code>\printspaces</code> , 213, 217	<code>\showboxbreadth</code> , 42
<code>\newcount</code> , 1073	<code>\przeniesvskip</code> , <u>1146</u>	<code>\showboxdepth</code> , 42
<code>\newcounter</code> , 364, 386	<code>\psi</code> , 1175	<code>\ShowFont</code> , 883
<code>\newgif</code> , <u>12</u>	<code>\quad</code> , 1153	<code>\showlists</code> , 42
<code>\newskip</code> , 1428	<code>\quantifierhook</code> , <u>1266</u> , 1278	<code>\sigma</code> , 1173
<code>\newwrite</code> , 861	<code>\qxcopyright</code> , <u>936</u>	<code>\sim</code> , 1183
<code>\nfss@text</code> , 201	<code>\qxcopyrights</code> , <u>937</u> , 941	<code>\SliTeX</code> , <u>720</u>
<code>\nieczer</code> , <u>897</u>	<code>\qxenc</code> , <u>935</u> , 936, 939	<code>\smaller</code> , <i>p. 8</i> , <u>153</u> , 947, 949
<code>\nobreakspace</code> , <u>1025</u>	<code>\rdate</code> , <u>1394</u>	<code>\smallerr</code> , <i>p. 8</i> , <u>157</u> , 1057
<code>\nocite</code> , <u>362</u>	<code>\real</code> , 1043, 1045	<code>\smallskipamount</code> , 849, 850
<code>\nohy</code> , <u>957</u>	<code>\reflectbox</code> , 734, 740	<code>\smartunder</code> , <u>180</u>
<code>\nolimits</code> , 1200, 1201	<code>\relaxen</code> , <u>39</u> , 39, 456, 1136, 1163	<code>\SMglobal</code> , <u>251</u>
<code>NoNumSecs</code> , <u>386</u>	<code>\relsize</code> , <i>p. 8</i> , <u>124</u> , 125, 152, 153, 154, 155, 156, 157	<code>\stepnummacro</code> , <u>377</u>
<code>\not@onlypreamble</code> , 346, 349, 350, 351, 352, 353	<code>\renewcommand*</code> , 904	<code>\Store@Macro</code> , 254, <u>256</u> , 271
<code>\nu</code> , 1169	<code>\RequirePackage</code> , 759, 858, 902, 1031	<code>\Store@Macros</code> , 268, <u>269</u>
<code>\numexpr</code> , 1143, 1144	<code>\resetMathstrut@</code> , <u>1220</u>	<code>\Store@MacroSt</code> , 255, <u>261</u>
<code>\nummacro</code> , <u>376</u>	<code>\resizebox</code> , 827	<code>\Stored@Macro</code> , 312, <u>313</u>
 	<code>\resizegraphics</code> , <u>806</u> , <u>826</u>	<code>\storedcsname</code> , <u>314</u>
<code>\oarg</code> , <u>237</u>	<code>\Restore@Macro</code> , 293, 295, 310, 313	<code>\StoredMacro</code> , <u>312</u>
<code>\old@begin</code> , <u>118</u> , 119	<code>\Restore@Macros</code> , <u>307</u> , <u>308</u>	<code>\StoreEnvironment</code> , <u>316</u>
<code>\oldLaTeX</code> , 652		<code>\StoreMacro</code> , 252, 401, 692, 758, 1409
<code>\oldLaTeXe</code> , 653		<code>\StoreMacro*</code> , 317, 693
<code>\omega</code> , 1176		
<code>\OnAtLine</code> , <u>872</u>		

<code>\StoreMacros</code> , 268, 1251	<code>tytulowa</code> , <u>1152</u>	<code>\WPheadings</code> , <u>553</u>
<code>\StoringAndRelaxingDo</code> , <u>320</u>	<code>\udigits</code> , 766, 769, 947, 949	<code>\Ws</code> , <u>1431</u>
<code>\subs</code> , <u>165</u> , 182	<code>\undeksmallskip</code> , <u>850</u>	<code>\wyzejnizej</code> , <u>1059</u>
<code>\sum</code> , 1197	<code>\unex@namedef</code> , <u>746</u>	<code>\Wz</code> , <u>1433</u>
<code>\TB</code> , 725	<code>\unex@nameuse</code> , <u>749</u>	<code>\Xedekfrac</code> , <u>770</u>
<code>\TeXbook</code> , <u>724</u> , 725	<code>\unless</code> , 754	<code>\XeLaTeX</code> , <u>737</u>
<code>\textbullet</code> , <u>1151</u> , 1151	<code>\usc</code> , <u>933</u> , 934	<code>\XeTeX</code> , <u>731</u>
<code>\textcolor</code> , 897	<code>\uscacro</code> , <u>934</u>	<code>\XeTeXinputencoding</code> , 8
<code>\textlarger</code> , <u>154</u>	<code>\usecounter</code> , 633	<code>\XeTeXpicfile</code> , 807, 824
<code>\textsl</code> , 724	<code>\value</code> , 367	<code>\XeTeXthree</code> , <u>756</u>
<code>\textsmaller</code> , <u>155</u>	<code>\varepsilon</code> , 691, 727, 1167	<code>\XeTeXversion</code> , 4, 5, 7, 168, 753, 754, 1083, 1128
<code>\textstyle</code> , 691	<code>\varnothing</code> , 1289	<code>\xiiand</code> , <u>190</u>
<code>\textsuperscript</code> , <u>831</u> , <u>835</u>	<code>\varsigma</code> , 1174	<code>\xiibackslash</code> , <u>185</u> , 186
<code>\texttilde</code> , <u>843</u>	<code>\vartheta</code> , 1168	<code>\xiilbrace</code> , <u>175</u>
<code>\textwidth</code> , 1420, 1421	<code>\vee</code> , 1205, 1260	<code>\xiipercen</code> , <u>188</u>
<code>\thedata</code> , 1398	<code>\verb</code> , 758, 760	<code>\xiirbrace</code> , <u>176</u>
<code>\thickmuskip</code> , 1262	<code>\visiblespace</code> , 195, 197, 212	<code>\xiispace</code> , 45, 46, <u>192</u> , 197
<code>\thr@@</code> , 622, 638	<code>\vs</code> , <u>212</u> , 213, 216	<code>\xiistring</code> , <u>44</u>
<code>\time</code> , 1143, 1144	<code>\vspace*</code> , 1149	<code>\xiiunder</code> , <u>167</u> , <u>169</u> , 170
<code>\tinycap</code> , <u>1030</u>	<code>\wd</code> , 672, 675, 698, 703, 711, 712, 810, 1200, 1280, 1281, 1291, 1292	<code>\xxt@visiblespace</code> , 194, 195
<code>\TODO</code> , <u>855</u>	<code>\Web</code> , <u>723</u>	<code>\yeshy</code> , <u>958</u>
<code>\toks</code> , 483, 484, 485, 541, 542, 546, 547	<code>\wedge</code> , 1206, 1260	<code>\zf@scale</code> , <u>1034</u> , 1037, 1038
<code>\tolerance</code> , 1076, 1080, 1405	<code>\whenonly</code> , <u>1003</u>	<code>\zwrobcy</code> , <u>1399</u>
<code>\tOnLine</code> , <u>871</u>	<code>\whern</code> , <u>1425</u>	<code>\-</code> , <u>1105</u> , 1133
<code>\truetextsuperscript</code> , <u>833</u> , <u>834</u>	<code>\wherncore</code> , <u>1418</u> , 1427, 1430	<code>\-</code> , <u>1085</u> , 1121, 1132
<code>\twocoltoc</code> , <u>857</u> , 865	<code>\whernskip</code> , <u>1426</u> , <u>1428</u> , 1429	
<code>\tytul</code> , <u>1400</u>	<code>\whernup</code> , <u>1430</u>	