

Grzegorz Murzynowski

# The gmutils Package\*

Written by Grzegorz Murzynowski,  
natror at o2 dot pl

© 2005–2008 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.

```
88 \NeedsTeXFormat{LaTeX2e}
89 \ProvidesPackage{gmutils}
90 [2008/11/22_v0.97_some_rather_TeXnical_macros,_some_of_them_
tricky_(GM)]
```

## Contents

<b>Intro</b> . . . . .	2	<b>Macros for printing macros and filenames</b> . . . . .	19
Installation . . . . .	2	<b>Storing and restoring the meanings of cses</b> . . . . .	21
Contents of the gmutils.zip archive . . . . .	2	<b>Not only preamble!</b> . . . . .	24
Compiling of the documentation . . . . .	2	<b>Third person pronouns</b> . . . . .	25
<b>A couple of abbreviations</b> . . . . .	3	<b>Improvements to mwcls sectioning commands</b> . . . . .	26
<b>\firstofone and the queer \catcodes</b> . . . . .	4	An improvement of MW's \SetSectionFormatting . . . . .	28
<b>Global Boolean switches</b> . . . . .	4	Negative \addvspace . . . . .	29
<b>\gm@ifundefined—a test that doesn't create any hash entry unlike \@ifundefined</b> . . . . .	6	My heading setup for mwcls . . . . .	30
Storing and restoring the catcodes of specials . . . . .	7	Compatibilising standard and mwcls sectionings . . . . .	31
<b>From the ancient xparse of T<sub>E</sub>XLive 2007</b> . . . . .	7	<b>enumerate* and itemize*</b> . . . . .	33
<b>Ampulex Compressa-like modifications of macros</b> . . . . .	11	<b>The logos</b> . . . . .	34
<b>\@ifnextcat, \@ifnextac</b> . . . . .	13	<b>Expandable turning stuff all into 'other'</b> . . . . .	36
<b>\afterfi and pals</b> . . . . .	15	<b>Brave New World of X<sub>Y</sub>T<sub>E</sub>X</b> . . . . .	36
<b>Environments redefined</b> . . . . .	15	Fractions . . . . .	37
Almost an environment or redefinition of \begin . . . . .	15	\resizegraphics . . . . .	38
\@ifenvir and improvement of \end . . . . .	16	Settings for mathematics in main font . . . . .	39
<b>From relsize</b> . . . . .	16	Minion and Garamond Premier kerning and ligature fixes . . . . .	46
<b>Some 'other' stuff</b> . . . . .	18	<b>Varia</b> . . . . .	46
<b>Metasymbols</b> . . . . .	19	\@ifempty . . . . .	51

\* This file has version number v0.97 dated 2008/11/22.

<code>\include not only .tex's</code> . . . . .	51	Typesetting dates in my memoirs . . .	56
Faked small caps . . . . .	52	A left-slanted font . . . . .	60
See above/see below . . . . .	53	Thousand separator . . . . .	61
<code>luzniej</code> and <code>napa-</code> <code>pierki</code> —environments		<code>hyperref's \nolinkurl</code> into <code>\url*</code> . .	63
used in page breaking for money .	53	<b>Change History</b> . . . . .	63
		<b>Index</b> . . . . .	65

## Intro

The `gmutils.sty` package provides some macros that are analogous to the standard  $\text{\LaTeX}$  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) $\text{\TeX}$ -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/gm` directory. Creating a `texmf/tex/latex/gm` directory may be advisable if you consider using other packages written by me.

Then you should refresh your  $\text{\TeX}$  distribution's files' database most probably.

## Contents of the `gmutils.zip` archive

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

```
gmutils.sty
README
gmutils.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  $\text{\LaTeX}$  on the `gmutils.sty` file twice (`xelatex gmutils.sty` in the directory you wish the documentation to be in, you don't have copy the `.sty` file there,  $\text{\TeX}$  will find it), then `MakeIndex` on the `gmutils.idx` file, and then  $\text{\LaTeX}$  on `gmutils.sty` 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 `gmdoc.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  $\text{\TeX}$  Live e.g.), the result of your compilation might differ a bit from the `.pdf` provided in this `.zip`

archive in formatting: If you had not installed mwcls, the standard article.cls class would be used.

```

162 \ifx\XeTeXversion\relax
163   \let\XeTeXversion\@undefined% If someone earlier used \@ifundefined{%
      %XeTeXversion} to test whether the engine is XeTeX, then \XeTeXversion
      is defined in the sense of  $\epsilon$ -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 3749.
170 \fi
172 \ifdefined\XeTeXversion
173 \XeTeXinputencoding $\_utf-8$ % we use Unicode dashes later in this file.
174 \fi% and if we are not in XeTeX, we skip them thanks to XeTeX-test.

```

## A couple of abbreviations

```

\@xa 180 \let\@xa\expandafter
\@nx 181 \let\@nx\noexpand
\@xau 183 \def\@xau{\@xa\unexpanded\@xa}
\pdef 187 \def\pdef{\protected\def}

```

And this one is defined, I know, but it's not `\long` with the standard definition and I want to be able to `\gobble` a `\par` sometimes.

```

\gobble 194 \long\def\gobble#1{}
\@gobble 196 \let\@gobble\gobble
\gobbletwo 197 \let\gobbletwo\@gobbletwo
\provide 201 \long\pdef\provide#1{%
202   \ifdefined#1%
203     \ifx\relax#1\afterfifi{\def#1}%
204     \else\afterfifi{\gmu@gobdef}%
205     \fi
206   \else\afterfi{\def#1}%
207   \fi}
\gmu@gobdef 210 \long\def\gmu@gobdef#1#1{%
211   \def\gmu@tempa{}}% it's a junk macro assignment to absorb possible prefixes.
213   \@gobble}
\pprovide 216 \def\pprovide{\protected\provide}

```

Note that both `\provide` and `\pprovide` may be prefixed with `\global`, `\outer`, `\long` and `\protected` because the prefixes stick to `\def` because all before it is expandable. If the condition(s) is false (`#1` is defined) then the prefixes are absorbed by a junk assignment.

Note moreover that unlike L<sup>A</sup>T<sub>E</sub>X's `\providecommand`, our `\(p)provide` allow any parameters string just like `\def` (because they just *expand* to `\def`).

```

\@nameedef 229 \long\def\@nameedef#1#2{%
230   \@xa\edef\csname#1\endcsname{#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.

```
241 \long\def\firstofone#1{#1}
```

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

```
\foone 246 \long\def\foone#1{\begingroup#1\egfirstofone}
```

```
248 \long\def\egfirstofone#1{\endgroup#1}
```

```
\foeatletter 250 \long\def\foeatletter{\foone\makeatletter}
```

## Global Boolean switches

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: ltxdefns.dtx Date: 2004/02/20 Version v1.3g, lines 139–150)

```
\newgif 264 \pdef\newgif#1{%
265   {\escapechar\m@ne
266    \global\let#1\iffalse
267    \@gif#1\iftrue
268    \@gif#1\iffalse
269   }}
```

'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 275 \def\@gif#1#2{%
276   \protected\@xa\gdef\csname\@xa\@gobbletwo\string#1%
277   g% the letter g for 'global'.
278   \@xa\@gobbletwo\string#2\endcsname
279   {\global\let#1#2}}
```

```
281 \pdef\newif#1{% We not only make \newif \protected but also make it to define
      \protected assignments so that premature expansion doesn't affect \if...
      % \fi nesting.
```

```
288   \count@\escapechar\@escapechar\m@ne
289   \let#1\iffalse
290   \@if#1\iftrue
291   \@if#1\iffalse
292   \escapechar\count@}
```

```
\@if 294 \def\@if#1#2{%
295   \protected\@xa\def\csname\@xa\@gobbletwo\string#1%
296   \@xa\@gobbletwo\string#2\endcsname
297   {\let#1#2}}
```

```
\hidden@iffalse 300 \pdef\hidden@iffalse{\iffalse}
```

```
\hidden@iftrue 301 \pdef\hidden@iftrue{\iftrue}
```

After `\newif\iffoo` you may type `{\foogtrue}` and the `\iffoo` 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 `\newif`.

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 `LATEX` approach.

```
\grefstepcounter 322 \pdef\grefstepcounter#1{%
                 323   {\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`.

2008/08/10 I spent all the night debugging `\penalty 10000` that was added after a `hypertarget` in vertical mode. I didn't dare to touch `hyperref`'s guts, so I worked it around with ensuring every `\grefstepcounter` to be in `hmode`:

```
\hgrefstepcounter 337 \pdef\hgrefstepcounter#1{%
                 338   \ifhmode\leavevmode\fi\grefstepcounter{#1}}
```

By the way I read some lines from *The T<sub>E</sub>Xbook* and was reminded that `\unskip` strips any last skip, whether horizontal or vertical. And I use `\unskip` mostly to replace a blank space with some fixed skip. Therefore define

```
\hunskip 345 \pdef\hunskip{\ifhmode\unskip\fi}
```

Note the two macros defined above are `\protected`. I think it's a good idea to make `\protected` all the macros that contain assignments. There is one more thing with `\ifhmode`: it can be different at the point of `\edef` and at the point of execution.

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

```
\glet 355 \def\glet{\global\let}
```

`LATEX` 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 363 \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 368 \long\def\addto@macro#1#2{%
                 369   \toks@\@xa{#1#2}%
                 370   \edef#1{\the\toks@}%
                 371 }% (\toks@ is a scratch register, namely \tokso.)
```

And for use in the very document,

```
\addtomacro 375 \let\addtomacro=\addto@macro
```

2008/08/09 I need to prepend something not add at the end—so

```
\prependtomacro 378 \long\def\prependtomacro#1#2{%
```

```
380 \edef#1{\unexpanded{#2}\@xa\unexpanded\@xa{#1}}
```

Note that `\prependtomacro` can be prefixed.

```
\addtotoks 384 \long\def\addtotoks#1#2{%
385 #1=\@xa{\the#1#2}}
```

```
\@emptyify 388 \newcommand*\@emptyify[1]{\let#1=\@empty}
\emptyify 389 \@ifdefinable\emptyify{\let\emptyify\@emptyify}
```

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

```
\g@emptyify 393 \newcommand*\g@emptyify{\global\@emptyify}
\gemptify 394 \@ifdefinable\gemptify{\let\gemptify\g@emptyify}
```

```
\@relaxen 397 \newcommand\@relaxen[1]{\let#1=\relax}
\relaxen 398 \@ifdefinable\relaxen{\let\relaxen\@relaxen}
```

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

```
\g@relaxen 402 \newcommand*\g@relaxen{\global\@relaxen}
\grelaxen 403 \@ifdefinable\grelaxen{\let\grelaxen\g@relaxen}
```

### `\gm@ifundefined`—a test that doesn't create any hash entry unlike `\@ifundefined`

I define it under another name not redefine `\@ifundefined` because I can imagine an odd case when something works thanks to `\@ifundefined`'s 'relaxation effect'.

```
\gm@ifundefined 412 \long\def\gm@ifundefined#1{% not \protected because expandable.
418 \ifcsname#1\endcsname% defined
419 \@xa\ifx\csname_#1\endcsname\relax% but as \relax
420 \afterfifi\@firstoftwo%
421 \else% defined and not \relax
422 \afterfifi\@secondoftwo%
423 \fi
424 \else% not defined
425 \afterfi\@firstoftwo%
426 \fi}
```

```
\gm@testdefined 429 \long\def\gm@testdefined#1\iftrue{% This is a macro that expands to \iftrue
or \iffalse depending on whether it's argument is defined in the LATEX
sense. Its syntax requires an \iftrue to balance \ifs with \fis.
434 \csname
435 \ifdefined#1%
436 \ifx#1\relax
437 iffalse%
438 \else_iftrue%
439 \fi
440 \else_iffalse%
441 \fi\endcsname
442 }
```

```
\gm@testundefined 444 \long\def\gm@testundefined#1\iftrue{% we expand the last macro two levels.
We repeat the parameter string to force the proper syntax.
447 \@xa\@xa\@xa\unless\gm@testdefined#1\iftrue}
```

## Storing and restoring the catcodes of specials

```
\gmu@storespecialchars 452 \newcommand*\gmu@storespecialchars[1][\relax]{% we provide a possibility of adding
                        stuff. For usage see line ??.
454   \def\do##1{\catcode`\@nx##1=\the\catcode`##1\relax}%
455   \edef\gmu@restorespecials{\dospecials\do\^M#1}}

\gmu@septify 457 \pdef\gmu@septify{% restoring the standard catcodes of specials. The name is the
                opposite of 'sanitize' :-). It restores also the original catcode of ^M
460   \def\do{\relax\catcode`}%
461   \do\_\do\o\do\o\do\{1\do\}2\do\$\do\&4%
462   \do\#\do\^7\do\_8\do\%14\do\~13\do\^M5\relax}
```

## From the ancient xparse of T<sub>E</sub>X Live 2007

The code of this section is rewritten contents of the xparse package version 0.17 dated 1999/09/10, the version available in T<sub>E</sub>X Live 2007-13, in Ubuntu packages at least. It's a stub 'im Erwartung' (Schönberg) for the L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> bundle and it does what I want, namely defines `\DeclareDocumentCommand`. I rewrote the code to use the usual catcodes (only with @ a letter) and not to use the ldcsetup package (which caused an error of undefined `cs\KV@def`).

Well, I add the `\protected` prefix to the first macro.

After exchange of some mails with Morten Høgholm and trying xparse of 2008/08/03 svn 748 (which beautifully spoils the catcodes) I wrap the ancient code in a conditional to avoid name collision if someone loads xparse before gmutils

```
482 \gmu@testundefined\DeclareDocumentCommand\iftrue
484 \unless\ifdefined\@temptokenb
\@temptokenb 485 \newtoks\@temptokenb
486 \fi

\xparsed@args 488 \newtoks\xparsed@args

\DeclareDocumentCommand 490 \long\def\DeclareDocumentCommand#1#2#3{%
                    % #1 command to be defined,
                    % #2 arguments specification,
                    % #3 definition body.
496   \@tempcnta\z@
497   \toks@{}%
498   \@temptokena\toks@
499   \@temptokenb\toks@
500   \@ddc#2X% X is the guardian of parsing.
501   \protected\edef#1{% The \protected prefix is my (GM) addition.
502     \@nx\@ddc@
503     {\the\toks@}%
504     \@xa\@nx\csname\string#1\endcsname
505     \@nx#1%
506   }%
507   \long\@xa\def\csname\string#1\@xa\endcsname
508   \the\@temptokena{#3}}

DeclareDocumentEnvironment 510 \long\def\DeclareDocumentEnvironment#1#2#3#4{%
511   \@xa\DeclareDocumentCommand\csname#1\endcsname{#2}{%
512     \xparsed@args\toks@
513     #3}%
```

```

514 \@xa\let\csname_end#1\endcsname\@parsed@endenv
515 \long\@xa\def\csname_end\string\#1\@xa\endcsname
516 \the\@temptokena{#4}}
\@parsed@endenv 518 \def\@parsed@endenv{%
519 \@xa\@parsed@endenv@\the\xparsed@args}
\@parsed@endenv@ 521 \def\@parsed@endenv@#1{%
522 \csname_end\string#1\endcsname}
\@ddc@ 524 \def\@ddc@#1#2#3{%
525 \ifx\protect\@typeset@protect
526 \@xa\@firstofone
527 \else
528 \protect#3\@xa\@gobble
529 \fi
530 {\toks@{#2}#1\the\toks@}}
\@ddc 532 \def\@ddc#1{%
533 \ifx#1X%
534 \else
535 \ifx#1m%
536 \addto@hook\@temptokenb_m%
537 \else
538 \toks@\@xa{%
539 \the\@xa\toks@
540 \csname_@ddc@\the\@temptokenb\@xa\endcsname
541 \csname_@ddc@#1\endcsname}%
542 \@temptokenb{}}%
543 \fi
544 \advance\@tempcnta\@ne
545 \@temptokena\@xa{%
546 \the\@xa\@temptokena\@xa##\the\@tempcnta}%
547 \@xa
548 \@ddc
549 \fi}
\@ddc@s 551 \long\def\@ddc@s#1\toks@{%
552 \@ifstar
553 {\addto@hook\toks@\BooleanTrue#1\toks@}%
554 {\addto@hook\toks@\BooleanFalse#1\toks@}}
\@ddc@m 556 \long\def\@ddc@m#1\toks@#2{%
557 \addto@hook\toks@{#2}#1\toks@}%
\@ddc@o 559 \long\def\@ddc@o#1\toks@{%
560 \@ifnextchar[%
561 {\@ddc@o@{#1}}
562 {\addto@hook\toks@\NoValue#1\toks@}}
\@ddc@o@ 564 \long\def\@ddc@o@#1[#2]{%
565 \addto@hook\toks@{#2}#1\toks@}
\@ddc 567 \def\@ddc#1{%
568 \ifx#1X%
569 \perhaps@grab@ms
570 \else
571 \ifx#1m%

```



```

572 \addto@hook\@temptokenb_m%
573 \else
574 \toks@\@xa{%
575   \the\@xa\toks@
576   \csname_@ddc@x\the\@temptokenb\@xa\endcsname
577   \csname_@ddc@#1\endcsname}%
578 \@temptokenb{}}%
579 \ifx#10%
580 \let\next@ddc\grab@default
581 \else
582 \ifx#1C%
583 \let\next@ddc\grab@default
584 \fi
585 \fi
586 \fi
587 \advance\@tempcnta\@ne
588 \@temptokena\@xa{%
589   \the\@xa\@temptokena\@xa##\the\@tempcnta}%
590 \@xa
591 \next@ddc
592 \fi
593 }%
\grab@default
595 \let\next@ddc\@ddc
596 \def\grab@default#1{%
597   \toks@\@xa{%
598     \the\toks@
599     {#1}}%
600   \let\next@ddc\@ddc
601   \@ddc
602 }
\@ddc@0
604 \long\def\@ddc@0#1#2\toks@{%
605   \@ifnextchar[%
606     {\@ddc@o@{#2}}%
607     {\addto@hook\toks@{#{#1}}#2\toks@}}
\@ddc@c
609 \long\def\@ddc@c#1\toks@{%
610   \@ifnextchar(%
611     {\@ddc@c@{#1}}%
612     {\PackageError{gmutils/xparse}{Missing~coordinate~argument}%
613       {A~value~of~(o,o)~is~assumed}}%
614     \addto@hook\toks@{{oo}}#1\toks@}%
615 }
\@ddc@c@
617 \long\def\@ddc@c@#1(#2,#3){%
618   \addto@hook\toks@{#{#2}{#3}}#1\toks@}
\@ddc@C
620 \long\def\@ddc@C#1#2\toks@{%
621   \@ifnextchar(%
622     {\@ddc@c@{#2}}%
623     {\addto@hook\toks@{#{#1}}#2\toks@}}
\grab@ms
625 \let\perhaps@grab@ms\relax
626 \def\grab@ms{%
627   \toks@\@xa{%
628     \the\@xa\toks@

```

```

629     \csname_@ddc@x\the\@temptokenb\endcsname
630   }}
631
632 \let\@ddc@m\undefined
\@ddc@xm 634 \long\def\@ddc@xm#1\toks@#2{%
635   \addto@hook\toks@{#{#2}}#1\toks@}
\@ddc@xmm 637 \long\def\@ddc@xmm#1\toks@#2#3{%
638   \addto@hook\toks@{#{#2}{#3}}#1\toks@}
\@ddc@xmxxx 640 \long\def\@ddc@xmxxx#1\toks@#2#3#4{%
641   \addto@hook\toks@{#{#2}{#3}{#4}}#1\toks@}
\@ddc@xmxxxx 643 \long\def\@ddc@xmxxxx#1\toks@#2#3#4#5{%
644   \addto@hook\toks@{#{#2}{#3}{#4}{#5}}#1\toks@}
\@ddc@xmxxxxx 646 \long\def\@ddc@xmxxxxx#1\toks@#2#3#4#5#6{%
647   \addto@hook\toks@{#{#2}{#3}{#4}{#5}{#6}}#1\toks@}
\@ddc@xmxxxxxx 649 \long\def\@ddc@xmxxxxxx#1\toks@#2#3#4#5#6#7{%
650   \addto@hook\toks@{#{#2}{#3}{#4}{#5}{#6}{#7}}#1\toks@}
\@ddc@xmxxxxxxx 652 \long\def\@ddc@xmxxxxxxx#1\toks@#2#3#4#5#6#7#8{%
653   \addto@hook\toks@{#{#2}{#3}{#4}{#5}{#6}{#7}{#8}}#1\toks@}
\@ddc@xmxxxxxxx 655 \long\def\@ddc@xmxxxxxxx#1\toks@#2#3#4#5#6#7#8#9{%
656   \addto@hook\toks@{#{#2}{#3}{#4}{#5}{#6}{#7}{#8}{#9}}#1\toks@}
\@ddc@xmxxxxxxx 658 \long\def\@ddc@xmxxxxxxx\the\toks@#1#2#3#4#5#6#7#8#9{%
659   \addto@hook\toks@{#{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}{#9}}\the%
\toks@}
661 \let\@ddc@x\relax
DeclareDocumentEnvironment 663 \long\def\DeclareDocumentEnvironment#1#2#3#4{%
664   \@xa\DeclareDocumentCommand\csname#1\endcsname{#2}{%
665     #3}%
666   \@namedef{end#1}{#4}%
667 }
668 \let\@parsed@endenv\undefined
669 \let\@parsed@endenv@\undefined
\IfSomethingTF 670 \def\IfSomethingTF#1{\def\something@in{#1}\If@SomethingTF}
\something@in 671 \def\IfSomethingT#1#2#3{\def\something@in{#1}%
\IfSomethingT 672 \If@SomethingTF{#2}{#3}\@empty}
\something@in 674 \def\IfSomethingF#1#2#3{\def\something@in{#1}%
\IfSomethingF 675 \If@SomethingTF{#2}\@empty{#3}}
\something@in 677 \def\If@SomethingTF#1{%
\If@SomethingTF 678 \def\something@tmp{#1}%
\something@tmp 679 \ifx\something@tmp\something@in
680 \@xa\@secondofthree
681 \else
682 \@xa\def\@xa\something@tmpb\@xa{#1}%
683 \ifx\something@tmp\something@tmpb
684 \@xa\@xa\@xa\@thirdofthree
685 \else
686 \@xa\@xa\@xa\@firstofone
687 \fi
688 \fi
689 {\@xa\If@SomethingTF\@xa{#1}}%

```

```

690 }
\@secondofthree 692 \long\def\@secondofthree#1#2#3{#2}
\@thirdofthree 693 \long\def\@thirdofthree#1#2#3{#3}
\NoValue 694 \def\NoValue{-NoValue-}
\NoValueInIt 695 \def\NoValueInIt{\NoValue}
\IfNoValueTF 696 \def\IfNoValueTF{\IfSomethingTF\NoValue}
\IfNoValueT 697 \def\IfNoValueT{\IfSomethingT\NoValue}
\IfNoValueF 698 \def\IfNoValueF{\IfSomethingF\NoValue}
\IfValueTF 699 \def\IfValueTF#1#2#3{\IfNoValueTF{#1}{#3}{#2}}
700 \let\IfValueT\IfNoValueF
701 \let\IfValueF\IfNoValueT
\BooleanFalse 702 \def\BooleanFalse{TF}
\BooleanTrue 703 \def\BooleanTrue{TT}
\IfBooleanTF 704 \def\IfBooleanTF#1{%
705 \if#1%
706 \@xa\@firstoftwo
707 \else
708 \@xa\@secondoftwo
709 \fi
710 }
\IfBooleanT 712 \def\IfBooleanT#1#2{%
713 \IfBooleanTF{#1}{#2}\@empty
714 }
\IfBooleanF 716 \def\IfBooleanF#1{%
717 \IfBooleanTF{#1}\@empty
718 }
720 \fi% of \unless\ifdefined\DeclareDocumentCommand.

```

## Ampulex Compressa-like modifications of macros

Ampulex Compressa is a wasp that performs brain surgery on its victim cockroach to lead it to its lair and keep alive for its larva. Well, all we do here with the internal L<sup>A</sup>T<sub>E</sub>X macros resembles Ampulex's actions but here is a tool for a replacement of part of macro's definition.

The `\ampulexdef` command takes its #2 which has to be a macro and replaces part of its definition delimited with #5 and #6 with the replacement #7. The redefinition may be prefixed with #1. #2 may have parameters and for such a macro you have to set the parameters string and arguments string (the stuff to be taken by the one-step expansion of the macro) as the optional [#3] and [#4]. . If `\ampulexdef` doesn't find the start and end tokens in the meaning of the macro, it does nothing to it. You have to write #### instead of # or you can use `\ampulexhash` as well. For an example use see line 1718.

```

\ampulexdef 755 \DeclareDocumentCommand\ampulexdef{0-}m0-{}0-{}mmm}{%
% [#1] definition prefix, empty by default,
% #2 macro to be redefined,
% [#3] \def parameters string, empty by default,
% [#4] definition body parameters to be taken in a one-step expansion of the
% redefined macro, empty by default,
% #5 start token(s),
% #6 end token(s)
% #7 replacement.

```

For the example of usage see [1718](#).

```

\gmu@tempa 770 \def\gmu@tempa{#5}%
\gmu@tempb 771 \def\gmu@tempb{#6}%
\gmu@tempc 772 \def\gmu@tempc{#7}% we wrap the start, end and replacement tokens in macros
              to avoid unbalanced \ifs.
774 \edef\gmu@tempd{%
775     \long\def\@nx\gmu@tempd
776     #####1\all@other\gmu@tempa
777     #####2\all@other\gmu@tempb
778     #####3\@nx\gmu@tempd{%
779         \@ifempty{#####3}{\hidden@iffalse}{\hidden@iftrue}}}%
781 \gmu@tempd% it defines \gmu@tempc to produce an open \if depending on whether
              the start and end token(s) are found in the meaning of #2.

785 \edef\gmu@tempe{%
786     \@nx\gmu@tempd\all@other#2%
787     \all@other\gmu@tempa
788     \all@other\gmu@tempb\@nx\gmu@tempd
789 }%
791 \gmu@tempe% we apply the checker and it produces an open \if.
793 \edef\gmu@tempd{%
794     \long\def\@nx\gmu@tempd
795     #####1\@xa\unexpanded\@xa{\gmu@tempa}%
796     #####2\@xa\unexpanded\@xa{\gmu@tempb}%
797     #####3\@nx\ampulexdef{% we define a temporary macro with the parameters
              delimited with the 'start' and 'end' parameters of \ampulexdef.
800         \@nx\unexpanded{#####1}%
801         \@nx\@xa\@nx\unexpanded
802         \@nx\@xa{\@nx\gmu@tempc}% we replace the part of the redefined macro's
              meaning with the replacement text.
804         \@nx\unexpanded{#####3}}}%
806 \gmu@tempd
809 \edef\gmu@tempf{#4}%
810 \edef\gmu@tempe{%
811     #1\def\@nx#2#3{%
812         \@xa\@xa\@xa\gmu@tempd\@xa#2\gmu@tempf\ampulexdef}}}%
813 \gmu@tempe
814 \fi}

\ampulexhash 816 \def\ampulexhash{#####}% for your convenience (not to count the hashes).

```

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 824 \def\gmshowlists{\showboxdepth=1000\showboxbreadth=1000\%
              \showlists}

\nameshow 828 \newcommand\nameshow[1]{\@xa\show\cename#1\endcename}
\nameshowthe 829 \newcommand\nameshowthe[1]{\@xa\showthe\cename#1\endcename}

```

Note that to get proper `\showthe\my@dimen14` in the 'other' @'s scope you write `\nameshowthe{my@dimen}14`.

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 840 \def\xiistring#1{%
            841   \if\@nx#1\xiispace
            842     \xiispace
            843   \else
            844     \string#1%
            845   \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 862 \long\def\@ifnextcat#1#2#3{%
            866   \def\reserved@d{#1}%
            867   \def\reserved@a{#2}%
            868   \def\reserved@b{#3}%
            869   \futurelet\@let@token\@ifncat}

```

```

\@ifncat 872 \def\@ifncat{%
            873   \ifx\@let@token\@sptoken
            874     \let\reserved@c\@xifncat
            875   \else
            876     \ifcat\reserved@d\@nx\@let@token
            877       \let\reserved@c\reserved@a
            878     \else
            879       \let\reserved@c\reserved@b
            880     \fi
            881   \fi
            882   \reserved@c}

```

```
884 {\def\:\{\let\@sptoken= }\global\:\}% this makes \@sptoken a space token.
```

```
887 \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 898 \long\pdef\@ifnextif#1#2#3{%
            902   \def\reserved@d{#1}%
            903   \def\reserved@a{#2}%
            904   \def\reserved@b{#3}%
            905   \futurelet\@let@token\@ifnif}

```

```

\@ifnif 908 \def\@ifnif{%
            909   \ifx\@let@token\@sptoken
            910     \let\reserved@c\@xifnif

```

```

911 \else
912   \if\reserved@d\@nx\@let@token
913     \let\reserved@c\reserved@a
914   \else
915     \let\reserved@c\reserved@b
916   \fi
917 \fi
918 \reserved@c}
921 {\def\:\{\let\@sptoken=\_}\:\_}% this makes \@sptoken a space token.
923 \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.

```

\@ifnextac 934 \long\pdef\@ifnextac#1#2{%
935   \@ifnextcat\bgroup{#2}\gm@ifnac{#1}{#2}}
\gm@ifnac 937 \long\def\gm@ifnac#1#2#3{%
938   \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, `\_`<sub>10</sub> 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 `=`.

```

951 \let\gmu@reserveda\*%
\* 952 \def\*{%
953   \let\*\gmu@reserveda
954   \let\gm@letspace=\_}%
955 \*_%
\@ifnextspace 958 \def\@ifnextspace#1#2{%
959   \let\gmu@reserveda\*%
\* 960 \def\*{%
961   \let\*\gmu@reserveda
962   \ifx\@let@token\gm@letspace\afterfi{#1}%
963   \else\afterfi{#2}%
964   \fi}%
965 \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.

Now a test if the next token is an active line end. I use it in `gmdoc` and later in this package for active long dashes.

```

974 \foone\obeylines{%
\@ifnextMac 975 \long\pdef\@ifnextMac#1#2{%
976   \@ifnextchar^~M{#1}{#2}}

```

## `\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 `\if...` 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  $\TeX$  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`.

```
\longafterfi 1001 \def\longafterfi{%
  \afterfi    1002   \long\def\afterfi##1##2\fi{\fi##1}}
1003 \longafterfi
```

And two more of that family:

```
\afterfifi 1005 \long\def\afterfifi#1#2\fi#3\fi{\fi\fi#1}
\afteriffifi 1007 \long\def\afteriffifi#1#2\fi#3\fi{\fi#1}
```

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

```
\afteriffiffifi 1011 \long\def\afteriffiffiffifi#1#2\fi#3\fi#4\fi{\fi#1}
\afteriffiffifi 1012 \long\def\afteriffiffiffifi#1#2\fi#3\fi#4\fi{\fi\fi#1}
\afterfiffifi 1013 \long\def\afterfiffifi#1#2\fi#3\fi#4\fi{\fi\fi\fi#1}
```

## Environments redefined

### 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  $\LaTeX$ '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 1045 \long\def\@begnamedgroup#1{%
1046   \@ignorefalse% not to ignore blanks after group
1047   \begingroup\@endpefalse
1048   \edef\@currenvir{#1}% We could do recatcoding through \string but all the
      name 'other' could affect a thousand packages so we don't do that and we'll
      recatcode in a testing macro, see line 1093.
```

```

1052 \edef\@currentvline{\on@line}%
1053 \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.)

```

Let us make it the starred version of \begin.

```

\begin* 1062 \def\begin{\@ifstar{\@begnamedgroup}{%
\begin 1063   \@begnamedgroup@ifcs}}
\@begnamedgroup@ifcs 1066 \def\@begnamedgroup@ifcs#1{%
1067   \ifcsname#1\endcsname\afterfi{\@begnamedgroup{#1}}%
1068   \else\afterfi{\@latex@error{Environment_#1_undefined}\@eha}%
1069   \fi}%

```

### \@ifenvir and improvement of \end

It's very clever and useful that \end checks whether its argument is \ifx-equivalent \@currentvir. However, in standard L<sup>A</sup>T<sub>E</sub>X 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 catcodes of chars are irrelevant (until they are \active, <sub>1</sub>, <sub>2</sub> etc.). Thus should be also in the \end's test and therefore we ensure the compared texts are both expanded and made all 'other'.

First a (not expandable) macro that checks whether current environment is as given in #1. Why is this macro \long?—you may ask. It's \long to allow environments such as \string\par.

```

\@ifenvir 1093 \long\def\@ifenvir#1#2#3{%
1095   \edef\gmu@reserveda{\@xa\string\csname\@currentvir\endcsname}%
1096   \edef\gmu@reservedb{\@xa\string\csname#1\endcsname}%
1097   \ifx\gmu@reserveda\gmu@reservedb\afterfi{#2}%
1098   \else\afterfi{#3}%
1099   \fi}
\@checkend 1101 \def\@checkend#1{\@ifenvir{#1}{}{\@badend{#1}}

```

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*}.

```

You might also 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 == \relsize{1}. \smallerr(my addition) == \relsize{-2}; \largerr
\smallerr
\largerr

```



guess yourself.

(Since `\DeclareRobustCommand` doesn't issue an error if its argument has been defined and it only informs about redefining, loading `relsize` remains allowed.)

```
\relsize 1141 \pdef\relsize#1{%
1142   \ifmmode\@nomath\relsize\else
1143     \begingroup
1144       \@tempcnta\% assign number representing current font size
1145       \ifx\@currsz\normalsize\else\% funny order is to have most
           ...
1146       \ifx\@currsz\small\else\% ...likely sizes checked first
1147       \ifx\@currsz\footnotesize\else
1148       \ifx\@currsz\large\else
1149       \ifx\@currsz\Large\else
1150       \ifx\@currsz\LARGE\else
1151       \ifx\@currsz\scriptsize\else
1152       \ifx\@currsz\tiny\else
1153       \ifx\@currsz\huge\else
1154       \ifx\@currsz\Huge\else
1155         4\rs@unknown@warning\% unknown state: \normalsize as
           starting point
1156     \fi\fi\fi\fi\fi\fi\fi\fi\fi
Change the number by the given increment:
1158   \advance\@tempcnta#1\relax
watch out for size underflow:
1160   \ifnum\@tempcnta<\z\rs@size@warning{small}{\string\tiny}%
           \@tempcnta\z\fi
1161   \@xa\endgroup
1162   \ifcase\@tempcnta\% set new size based on altered number
1163     \tiny\or\scriptsize\or\footnotesize\or\small\or\%
           \normalsize\or
1164     \large\or\Large\or\LARGE\or\huge\or\Huge\else
1165     \rs@size@warning{large}{\string\Huge}\Huge
1166 \fi\fi}% end of \relsize.
\rs@size@warning 1168 \providecommand*\rs@size@warning[2]{\PackageWarning{gmutils
           (relsize)}{%
1169   Size requested is too #1. \MessageBreak Using #2 instead}}
\rs@unknown@warning 1171 \providecommand*\rs@unknown@warning{\PackageWarning{gmutils
           (relsize)}{Current font size
1172   is unknown! (Why?!?)\MessageBreak Assuming\string\normalsize}}
And a handful of shorthands:
\larger 1176 \DeclareRobustCommand*\larger[1][\@one]{\relsize{+#1}}
\smaller 1177 \DeclareRobustCommand*\smaller[1][\@one]{\relsize{-#1}}
\textlarger 1178 \DeclareRobustCommand*\textlarger[2][\@one]{\relsize{+#1}#2}}
\textsmaller 1179 \DeclareRobustCommand*\textsmaller[2][\@one]{\relsize{-#1}#2}}
\largerr 1180 \pdef\largerr{\relsize{+2}}
\smallerr 1181 \pdef\smallerr{\relsize{-2}}
```

## 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`.

```

1191 \foone{\catcode`\_ =8_}%
\subs 1192 {\let\subs=_}

1194 \foone{\@makeother\_}%
\xiiunder 1195 {\def\xiiunder{_{}}

1197 \ifdefined\XeTeXversion
\xiiunder 1198 \def\xiiunder{\char"005F_}%
1199 \let\_ \xiiunder
1200 \fi

1202 \foone{\catcode`\ [=1_ \@makeother\{
1203 \catcode`\ ] =2_ \@makeother\}}%
1204 [%
\xiilbrace 1205 \def\xiilbrace[{}%
\xiirbrace 1206 \def\xiirbrace[}%
1207 ]% 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.

```

1218 \foone{\catcode`\_ =\active}
1219 {_%
\smartunder 1220 \newcommand*\smartunder{%
1221 \catcode`\_ =\active
1222 \def_{\ifmmode\subs\else\_ \fi}}}% We define it as \_ not just as \xiiunder
because some font encodings don’t have _ at the \char`\_ position.

1228 \foone{\catcode`\! =0
1229 \@makeother\!}
\xiibackslash 1230 {\!newcommand*\xiibackslash{\!}}
\bslash 1234 \let\bslash=\xiibackslash

1238 \foone{\@makeother\}%
\xiipercent 1239 {\def\xiipercent{}}

1242 \foone{\@makeother\&%}
\xiiand 1243 {\def\xiiand{&}}

1245 \foone{\@makeother\_}%
\xiispace 1246 {\def\xiispace{_}}

1248 \foone{\@makeother\#}%
\xiihash 1249 {\def\xiihash{#}}

```

We introduce `\visiblespace` from Will Robertson’s `xltxtra` if available. It’s not sufficient `\@ifpackageloaded{xltxtra}` since `\xxt@visiblespace` 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.

```

1258 \AtBeginDocument{%

```

```

1259 \ifdefined\xxt@visiblepace
1260 \let\visiblepace\xxt@visiblepace
1261 \else
1262 \let\visiblepace\xiispace
1263 \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 1284 \pdef\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.”

```

1292 \ensuremath\langle
1293 \ifmmode\@xa\@nfss@text\fi
1294 {%
1295 \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.

```

1303 #1\}%
1305 }\ensuremath\rangle
1306 }

```

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 1314 \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.

```
\<...> 1320 \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 1331 \def\discre#1#2#3{\leavevmode\kernosp%
1332 \discretionary{#1}{#2}{#3}\penalty10000\hskiposp\relax}
\discret 1333 \def\discret#1{\leavevmode\kernosp%

```

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

1334 `\discretionary{#1}{#1}{#1}\penalty10000\hskiposp\relax}`

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

1338 `\def\:-{\ifmmode\afterfi{\mskip\medmuskip}\else\afterfi{\discret{%`  
`}}\fi}`

`\vs` 1340 `\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` 1346 `\def\printspaces#1{{\let~=\vs\let\_=\vs\gm@pswords#1\@@nil}}`

`\gm@pswords` 1348 `\def\gm@pswords#1\#2\@@nil{%`

1349 `\ifx\relax#1\relax\else#1\fi`

1350 `\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` 1355 `\pdef\sfname#1{\textsf{\printspaces{#1}}}`

`\gm@discretionaryslash` 1357 `\def\gm@discretionaryslash{\discre{/}{\hbox{}}{/}}% the`  
 second pseudo-argument nonempty to get `\hyphenpenalty`  
 not `\exhyphenpenalty`.

`\file` 1362 `\pdef\file#1{\gm@printslashes#1/\gm@printslashes}`

`\gm@printslashes` 1364 `\def\gm@printslashes#1/#2\gm@printslashes{%`

1365 `\sfname{#1}%`

1366 `\ifx\gm@printslashes#2\gm@printslashes`

1367 `\else`

1368 `\textsf{\gm@discretionaryslash}%`

1369 `\afterfi{\gm@printslashes#2\gm@printslashes}\fi}`

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

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

`\pk` 1376 `\pdef\pk#1{\textsf{#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` 1390 `\DeclareRobustCommand*{\cs}[2][\slash]{%`

`\-` 1391 `\def\-\{\discretionary{\rmfamily-}{-}{-}}%`

1392 `\def\{\{\char`\}\def\{\char`\}\}\ttfamily\#1#2}}`

`\env` 1396 `\pdef\env#1{\cs[] {#1}}`

And for the special sequences like `^^A`:

1399 `\foone{\@makeother\^}`

`\hathat` 1400 `{\pdef\hathat#1{\cs[^^]{#1}}}`

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

`\possfil` 1405 `\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 1415 \def\cmd#1{\cs{\@xa\cmd@to@cs\string#1}}
\cmd@to@cs 1417 \def\cmd@to@cs#1#2{\char\number`#2\relax}

\marg 1421 \def\marg#1{\tffamily\char`\{ }\meta{#1}{\tffamily\char`\}}
\oarg 1426 \def\oarg{\@ifnextchar[\@oargsq\@oarg}
\@oarg 1428 \def\@oarg#1{\tffamily[]\meta{#1}{\tffamily}}
\@oargsq 1429 \def\@oargsq[#1]{\@oarg{#1}}

\parg 1433 \def\parg{\@ifnextchar(\@pargp\@parg}
\@parg 1435 \def\@parg#1{\tffamily()\meta{#1}{\tffamily}}
\@pargp 1436 \def\@pargp(#1){\@parg{#1}}

```

But we can have all three in one command.

```

1440 \AtBeginDocument{%
\arg 1441 \let\math@arg\arg
\arg 1442 \def\arg{\ifmmode\math@arg\else\afterfi{%
1443 \ifnextchar[%
1444 \@oargsq{\@ifnextchar(
1445 \@pargp\marg}}\fi}%
1446 }

```

Now you can write

`\arg{mand.␣arg}` to get  $\langle mand. arg \rangle$ ,  
`\arg[opt.␣arg]` for  $[\langle opt. arg \rangle]$  and  
`\arg{pict.␣arg}` for  $\langle pict. arg \rangle$ .  
And  $\$ \arg(1+i) = \pi/4$  for  $\arg(1 + i) = \pi/4$ .

## Storing and restoring the meanings of cses

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

```

\ifgmu@SMglobal 1461 \newif\ifgmu@SMglobal
\SMglobal 1463 \pdef\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 1474 \pdef\StoreMacro{%
1475 \begingroup\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 1598.

```

\egStore@Macro 1480 \long\def\egStore@Macro#1{\endgroup\Store@Macro{#1}}
\egStore@MacroSt 1481 \long\def\egStore@MacroSt#1{\endgroup\Store@MacroSt{#1}}
\Store@Macro 1483 \long\def\Store@Macro#1{%
1484 \escapechar92

```

```

1485 \ifgmu@SMglobal\afterfi\global\fi
1486 \@xa\let\csname_/gmu/store/string#1\endcsname#1%
1487 \global\gmu@SMglobalfalse}

```

```

\Store@MacroSt 1490 \long\def\Store@MacroSt#1{%
1491 \edef\gmu@smtempa{%
1492 \ifgmu@SMglobal\global\fi
1493 \@nx\let\@xa\@nx\csname/gmu/store/bslash#1\endcsname% we add
backslash 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.
1499 \@xa\@nx\csname#1\endcsname}
1500 \gmu@smtempa
1501 \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 cses and stores each of them. The cs may be separated with commas but they don't have to.

```

\StoreMacros 1517 \long\pdef\StoreMacros{\begingroup\makeatletter\Store@Macros}
\Store@Macros 1518 \long\def\Store@Macros#1{\endgroup
1519 \gmu@setsetSMglobal
1520 \let\gml@StoreCS\Store@Macro
1521 \gml@storemacros#1.}
\gmu@setsetSMglobal 1524 \def\gmu@setsetSMglobal{%
1525 \ifgmu@SMglobal
1526 \let\gmu@setSMglobal\gmu@SMglobaltrue
1527 \else
1528 \let\gmu@setSMglobal\gmu@SMglobalfalse
1529 \fi}

```

And the inner iterating macro:

```

\gml@storemacros 1532 \long\def\gml@storemacros#1{%
\gmu@reserveda 1533 \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 expanding.
1536 \if\gmu@reserveda.% a dot finishes storing.
1537 \global\gmu@SMglobalfalse
1538 \else
1539 \if\gmu@reserveda,% The list this macro is put before may contain commas
and that's O.K., we just continue the work.
1541 \afterfifi\gml@storemacros
1542 \else% what is else this shall be stored.
1543 \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.
1546 \afterfifi{\gmu@setSMglobal\gml@storemacros}%
1547 \fi
1548 \fi}

```

And for the restoring

```

\RestoreMacro 1554 \pdef\RestoreMacro{%
1555   \begingroup\makeatletter\@ifstar\egRestore@MacroSt%
           \egRestore@Macro}

\egRestore@Macro 1557 \long\def\egRestore@Macro#1{\endgroup\Restore@Macro{#1}}
\egRestore@MacroSt 1558 \long\def\egRestore@MacroSt#1{\endgroup\Restore@MacroSt{#1}}

\Restore@Macro 1560 \long\def\Restore@Macro#1{%
1561   \escapechar92
1562   \ifgmu@SMglobal\afterfi\global\fi
1563   \@xa\let\@xa#1\csname_/gmu/store/string#1\endcsname
1564   \global\gmu@SMglobalfalse}

\Restore@MacroSt 1566 \long\def\Restore@MacroSt#1{%
1567   \edef\gmu@smtmpa{%
1568     \ifgmu@SMglobal\global\fi
1569     \@nx\let\@xa\@nx\csname#1\endcsname
1570     \@xa\@nx\csname/gmu/store/bslash#1\endcsname}% cf. the commentary
           in line 1493.
1572   \gmu@smtmpa
1573   \global\gmu@SMglobalfalse}

\RestoreMacros 1576 \long\pdef\RestoreMacros{\begingroup\makeatletter\Restore@Macros}
\Restore@Macros 1578 \long\def\Restore@Macros#1{\endgroup
1579   \gmu@setsetSMglobal
1580   \let\gml@StoreCS\Restore@Macro% we direct the core cs towards restoring
           and call the same iterating macro as in line 1521.
1583   \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 1589 \def\StoredMacro{\begingroup\makeatletter\Stored@Macro}
\Stored@Macro 1590 \long\def\Stored@Macro#1{\endgroup\Restore@Macro#1#1}

```

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

```

\storedcsname 1593 \def\storedcsname#1{%
1594   \csname_/gmu/store/bslash#1\endcsname}

2008/08/03 we need to store also an environment.

```

```

\StoreEnvironment 1598 \pdef\StoreEnvironment#1{%
1600   \StoreMacro*{#1}\StoreMacro*{end#1}}

```

```

\RestoreEnvironment 1602 \pdef\RestoreEnvironment#1{%
1604   \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 1619 \long\def\StoringAndRelaxingDo{%
1620   \gmu@SMdo@setscope
1621   \long\def\do##1{%
1622     \gmu@SMdo@scope
1623     \@xa\let\csname_/gmu/store/string##1\endcsname##1%

```

```

1624     \gmu@SMdo@scope\let##1\relax}}
\gmu@SMdo@setscope 1626 \def\gmu@SMdo@setscope{%
1627     \ifgmu@SMglobal\let\gmu@SMdo@scope\global
1628     \else\let\gmu@SMdo@scope\relax
1629     \fi
1630     \global\gmu@SMglobalfalse}

```

And here is the counter-definition for restore.

```

\RestoringDo 1639 \long\def\RestoringDo{%
1640     \gmu@SMdo@setscope
1641     \long\def\do##1{%
1642         \gmu@SMdo@scope
1643         \@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 1651 \def\n@melet#1#2{%
1652     \edef\gmu@nl@reserveda{%
1653         \let\@xa\@nx\csname#1\endcsname
1654         \@xa\@nx\csname#2\endcsname}%
1655     \gmu@nl@reserveda}

```

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

```

\gn@melet 1659 \def\gn@melet#1#2{%
1660     \edef\gmu@nl@reserveda{%
1661         \global\let\@xa\@nx\csname#1\endcsname
1662         \@xa\@nx\csname#2\endcsname}%
1663     \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 1680 \newcommand\not@onlypreamble[1]{%
1681     \def\do##1{\ifx##1\else\@nx\do\@nx##1\fi}%
1682     \xdef\@preamblecmds{\@preamblecmds}}
1684 \not@onlypreamble\@preamblecmds
1685 \not@onlypreamble\@ifpackageloaded
1686 \not@onlypreamble\@ifclassloaded
1687 \not@onlypreamble\@ifl@aded
1688 \not@onlypreamble\@pkgextension

```

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

```

\gm@notprerr 1693 \def\gm@notprerr{\can_be_used_only_in_preamble(\on@line)}
1695 \AtBeginDocument{%
1696     \def\do#1{\@nx\do\@nx#1}%

```



```

1697 \edef\@preamblecmds{%
1698 \def\@nx\do##1{%
1699 \def##1{\@nx\PackageError{gmutils/LaTeX}%
1700 {\@nx\string##1_\@nx\gm@notprerr}\@nx\@eha}}%
1701 \@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 with is standard L<sup>A</sup>T<sub>E</sub>X’s \nocite that checks \ifx\onlypreamble\document. So hoping this is a rare problem, we circumvent in with. 2008/08/29 a bug is reported by Edd Barrett that with natbib an ‘extra }’ error occurs so we wrap the fix in a conditional.

```

\gm@nocite@ampulex 1718 \def\gm@nocite@ampulex{% we wrap the stuff in a macro to hide an open \if.
                    And not to make the begin-input hook not too large. the first is the parameters
                    string and the second the argument for one-level expansion of \nocite so it
                    has to consist of two times less hashes than the first. Both hash strings are
                    doubled to pass the first \def.
1725 \ampulexdef []\nocite [####1] [{{####1}}]% note the double brace around
                    % #3.
1727 \ifx
1728 {\onlypreamble\document}%
1729 \iftrue}
1731 \AtBeginDocument\gm@nocite@ampulex

```

### 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 1758 \newcounter{gm@PronounGender}
\gm@atppron 1760 \newcommand*\gm@atppron [2] {%
1761 \stepcounter{gm@PronounGender}% remember \stepcounter is global.
1762 \ifodd\value{gm@PronounGender}#1\else#2\fi}
\heshe 1764 \newcommand*\heshe{\gm@atppron{he}{she}}
\hisher 1765 \newcommand*\hisher{\gm@atppron{his}{her}}
\himher 1766 \newcommand*\himher{\gm@atppron{him}{her}}
\hishers 1767 \newcommand*\hishers{\gm@atppron{his}{hers}}
\HeShe 1769 \newcommand*\HeShe{\gm@atppron{He}{She}}
\HisHer 1770 \newcommand*\HisHer{\gm@atppron{His}{Her}}
\HimHer 1771 \newcommand*\HimHer{\gm@atppron{Him}{Her}}
\HisHers 1772 \newcommand*\HisHers{\gm@atppron{His}{Hers}}

```

## 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.

1791 \AtBeginDocument{% because we don’t know when exactly hyperref is loaded and maybe after this package.

NoNumSecs 1793 \@ifpackageloaded{hyperref}{\newcounter{NoNumSecs}%  
1794 \setcounter{NoNumSecs}{617}% to make \refing to an unnumbered section visible (and funny?).

\gm@hyperrefstepcounter 1796 \def\gm@hyperrefstepcounter{\refstepcounter{NoNumSecs}}%  
\gm@targetheading 1797 \pdef\gm@targetheading#1{%  
1798 \hypertarget{#1}{#1}}}% end of then  
\gm@hyperrefstepcounter 1799 {\def\gm@hyperrefstepcounter{}%  
\gm@targetheading 1800 \def\gm@targetheading#1{#1}}}% end of else  
1801 }% of \AtBeginDocument

Auxiliary macros for the kernel sectioning macro:

bersectionsoutofmainmatter 1804 \def\gm@dontnumbersectionsoutofmainmatter{%  
1805 \ifmainmatter\else\HeadingNumberedfalse\fi}  
gm@clearpagesduetoopenright 1806 \def\gm@clearpagesduetoopenright{%  
1807 \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 1820 \long\def\@ifnotmw#1#2{\gm@ifundefined{mw@sectionxx}{#1}{#2}}

The kernel of MW’s sectioning commands:

1845 \@ifnotmw}{}%  
\mw@sectionxx 1846 \def\mw@sectionxx#1#2[#3]#4{%  
1847 \edef\mw@HeadingLevel{\csname#1@level\endcsname  
1848 \space}% space delimits level number!  
1849 \ifHeadingNumbered  
1850 \ifnum\mw@HeadingLevel>\c@secnumdepth\%  
\HeadingNumberedfalse\fi  
line below is in \gm@ifundefined to make it work in classes other than mwbk  
1853 \gm@ifundefined{ifmainmatter}{}{%  
\gm@dontnumbersectionsoutofmainmatter}  
1854 \fi  
% \ifHeadingNumbered  
% \refstepcounter{#1}%

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

```

%       \protected@edef\HeadingNumber{\csname
%       the#1\endcsname\relax}%
%       \else
%       \let\HeadingNumber\@empty
%       \fi
\HeadingRHeadText 1863 \def\HeadingRHeadText{#2}%
\HeadingTOCText   1864 \def\HeadingTOCText{#3}%
\HeadingText      1865 \def\HeadingText{#4}%
\mw@HeadingType   1866 \def\mw@HeadingType{#1}%
1867 \if\mw@HeadingBreakBefore
1868 \if@specialpage\else\thispagestyle{closing}\fi
1869 \gm@ifundefined{if@openright}{\}%
%       \gm@clearpagesduetoopenright}%
1870 \if\mw@HeadingBreakAfter
1871 \thispagestyle{blank}\else
1872 \thispagestyle{opening}\fi
1873 \global\@topnum\z@
1874 \fi% of \if\mw@HeadingBreakBefore
placement of \refstep suggested by me (GM):
1877 \ifHeadingNumbered
1878 \refstepcounter{#1}%
1879 \protected@edef\HeadingNumber{\csname_ the#1\endcsname\relax}%
1880 \else
1881 \let\HeadingNumber\@empty
1882 \gm@hyperrefstepcounter
1883 \fi% of \ifHeadingNumbered
1885 \if\mw@HeadingRunIn
1886 \mw@runinheading
1887 \else
1888 \if\mw@HeadingWholeWidth
1889 \if@twocolumn
1890 \if\mw@HeadingBreakAfter
1891 \onecolumn
1892 \mw@normalheading
1893 \pagebreak\relax
1894 \if@twoside
1895 \null
1896 \thispagestyle{blank}%
1897 \newpage
1898 \fi% of \if@twoside
1899 \twocolumn
1900 \else
1901 \@topnewpage[\mw@normalheading]%
1902 \fi% of \if\mw@HeadingBreakAfter
1903 \else
1904 \mw@normalheading
1905 \if\mw@HeadingBreakAfter\pagebreak\relax\fi
1906 \fi% of \if@twocolumn
1907 \else
1908 \mw@normalheading
1909 \if\mw@HeadingBreakAfter\pagebreak\relax\fi
1910 \fi% of \if\mw@HeadingWholeWidth

```

```

1911 \fi% of \if\mw@HeadingRunIn
1912 }

```

### 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] the flags, e.g. breakbefore, breakafter;
#2 the sectioning name, e.g. chapter, part;
#3 preskip;
#4 heading type;
#5 postskip

```

```

\SetSectionFormatting 1936 \relaxen\SetSectionFormatting
1937 \newcommand*\SetSectionFormatting[5] [\empty] {%
1938 \ifx\empty#1\relax\else% empty (not \empty!) #1 also launches \else.
1939 \def\mw@HeadingRunIn{10}\def\mw@HeadingBreakBefore{10}%
1940 \def\mw@HeadingBreakAfter{10}\def\mw@HeadingWholeWidth{10}%
1941 \@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.
1944 \fi
1945 \gm@ifundefined{#2}{\@namedef{#2}{\mw@section{#2}}}{}%
1946 \mw@secdef{#2}{@preskip}{#3}{2\oblig.}%
1947 \mw@secdef{#2}{@head}{#4}{3\oblig.}%
1948 \mw@secdef{#2}{@postskip}{#5}{4\oblig.}%
1949 \ifx\empty#1\relax
1950 \mw@secundef{#2@flags}{1\optional}%
1951 \else\mw@setflags{#2}%
1952 \fi}
\mw@secdef 1954 \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.
1961 \@ifempty{#3}
1962 {\mw@secundef{#1#2}{#4}}
1963 {\@namedef{#1#2}{#3}}}
\mw@secundef 1965 \def\mw@secundef#1#2{%
1966 \gm@ifundefined{#1}{%
1967 \ClassError{mwcls/gm}{%
1968 command\backslash#1\undefined\MessageBreak
1969 after\backslashSetSectionFormatting!!!\MessageBreak}{%
1970 Provide the #2 argument of \backslash
SetSectionFormatting.}}{}}
\addtoheading 1975 \def\addtoheading#1#2{%
1976 \n@melet{gmu@reserveda}{#1@head}%

```

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

```

1977 \edef\gmu@reserveda{\unexpanded{#2}\@xa\unexpanded{#1}
      \gmu@reserveda}}%
1978 \n@melet{#1@head}{\gmu@reserveda}%
1980 }
1982 }% of \@ifnotmw's else.

```

### 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.

```

1994 \@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 1999 \def\@afterheading{%
2000   \@nobreaktrue
2001   \xdef\gmu@prevsec{\mw@HeadingType}% added now
2002   \everypar{%
2003     \grelaxen\gmu@prevsec% added now. All the rest is original LATEX.
2004     \if@nobreak
2005     \@nobreakfalse
2006     \clubpenalty_\@M
2007     \if@afterindent_\else
2008     {\setbox\z@\lastbox}%
2009     \fi
2010     \else
2011     \clubpenalty_\@clubpenalty
2012     \everypar{}%
2013     \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 2020 \def\gmu@checkaftersec{%
2021   \gmu@ifundefined{gmu@prevsec}{}{%
2022     \ifgmu@postsec% an additional switch that is true by default but may be
      turned into an \ifdim in special cases, see line 2058.
2025     {\@xa\mw@getflags\@xa{\gmu@prevsec}%
2026     \glet\gmu@reserveda\mw@HeadingBreakAfter}%
2027     \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.
2030     \if\gmu@reserveda\else
2031     \penalty10000\relax
2032     \skip\z@=\csname\gmu@prevsec_\@postskip\endcsname\relax
2033     \skip\tw@=\csname\mw@HeadingType_\@preskip\endcsname\relax
2034     \gmu@ifundefined{\mw@HeadingType_\@twoheadskip}{%

```

```

2035     \ifdim\skip\z@>\skip\tw@
2036     \vskip-\skip\z@% we strip off the post-skip of previous header if it's bigger
           than current pre-skip
2038     \else
2039     \vskip-\skip\tw@% we strip off the current pre-skip otherwise
2040     \fi}{% But if the two-header-skip is defined, we put it
2042     \penalty10000
2043     \vskip-\skip\z@
2044     \penalty10000
2045     \vskip-\skip\tw@
2046     \penalty10000
2047     \vskip\curname\mw@HeadingType_\twoheadskip\endcurname
2048     \relax}%
2049     \penalty10000
2050     \hrule\height\z@\relax% to hide the last (un)skip before
           subsequent \advspaces.
2052     \penalty10000
2053     \fi
2054     \fi
2055     }% of \gm@ifundefined{gmu@prevsec} 'else'.
2056 }% of \def\gmu@checkaftersec.
\ParanoidPostsec 2058 \def\ParanoidPostsec{% this version of \ifgmu@postsec is intended for the special
           case of sections may contain no normal text, as while gmdocing.
\ifgmu@postsec 2061 \def\ifgmu@postsec{% note this macro expands to an open \if.
2062     \skip\z@=\curname\gmu@prevsec_\postskip\endcurname\relax
2063     \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).
2067     }}
2069 \let\ifgmu@postsec\iftrue
\gmu@getadvvs 2071 \def\gmu@getadvvs#1\advspace#2\gmu@getadvvs{%
2072     \toks\z@={#1}%
2073     \toks\tw@={#2}}
           And the modification of the inner macros at last:
\gmu@setheading 2076 \def\gmu@setheading#1{%
2077     \@xa\gmu@getadvvs#1\gmu@getadvvs
2078     \edef#1{%
2079         \the\toks\z@\@nx\gmu@checkaftersec
2080         \@nx\advspace\the\toks\tw@}}
2082 \gmu@setheading\mw@normalheading
2083 \gmu@setheading\mw@runinheading
\SetTwoheadSkip 2085 \def\SetTwoheadSkip#1#2{\@namedef{#1@twoheadskip}{#2}}
2087 }% of \@ifnotmw's else.

```

### 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.

```

2099 \@ifnotmw{}{% We define this declaration only when in mwcls.
\WPheadings
2100 \def\WPheadings{%
2101   \SetSectionFormatting[breakbefore,wholewidth]
2102     {part}{\z@\@plus1fill}{\z@\@plus3fill}%
2104   \gm@ifundefined{chapter}{}{%
2105     \SetSectionFormatting[breakbefore,wholewidth]
2106       {chapter}
2107       {66\p@}% {67\p@} for Adventor/Schola 0,95.
2108       {\FormatHangHeading{\LARGE}}
2109       {27\p@\@pluso,2\p@\@minus1\p@}%
2110   }%
2112   \SetTwoheadSkip{section}{27\p@\@pluso,5\p@}%
2113   \SetSectionFormatting{section}
2114     {24\p@\@pluso,5\p@\@minus5\p@}%
2115     {\FormatHangHeading_{\Large}}
2116     {10\p@\@pluso,5\p@}% ed. Krajewska of "Wiedza Powszechna", as we un-
      understand her, wants the skip between a heading and text to be rigid.
2120   \SetTwoheadSkip{subsection}{11\p@\@pluso,5\p@\@minus1\p@}%
2121   \SetSectionFormatting{subsection}
2122     {19\p@\@pluso,4\p@\@minus6\p@}
2123     {\FormatHangHeading_{\large}}% 12/14 pt
2124     {6\p@\@pluso,3\p@}% after-skip 6 pt due to p.12, not to squeeze the before-
      skip too much.
2127   \SetTwoheadSkip{subsubsection}{10\p@\@plus1,75\p@\@minus1\p@}%
2128   \SetSectionFormatting{subsubsection}
2129     {10\p@\@pluso,2\p@\@minus1\p@}
2130     {\FormatHangHeading_{\normalsize}}
2131     {3\p@\@pluso,1\p@}% those little skips should be smaller than you calcu-
      late out of a geometric progression, because the interline skip enlarges
      them.
2135   \SetSectionFormatting[runin]{paragraph}
2136     {7\p@\@pluso,15\p@\@minus1\p@}
2137     {\FormatRunInHeading{\normalsize}}
2138     {2\p@}%
2140   \SetSectionFormatting[runin]{subparagraph}
2141     {4\p@\@plus1\p@\@minuso,5\p@}
2142     {\FormatRunInHeading{\normalsize}}
2143     {\z@}%
2144 }% of \WPheadings
2145 }% 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.)

```

2186 \@ifnotmw{% we are not in mwcls and want to handle mwcls-like sectionings i.e.,
      those written with two optionals.
\gm@secini 2189 \def\gm@secini{gm@1a}%
\gm@secxx 2191 \def\gm@secxx#1#2[#3]#4{%
      2192 \ifx\gm@secstar\@empty
      2193 \n@melet{gm@true@#1mark}{#1mark}% a little trick to allow a special ver-
              sion of the heading just to the running head.
      2195 \@namedef{#1mark}##1{% we redefine \<sec>mark to gobble its argument
              and to launch the stored true marking command on the appropriate
              argument.
      2198 \csname_gm@true@#1mark\endcsname{#2}%
      2199 \n@melet{#1mark}{gm@true@#1mark}% after we've done what we
              wanted we restore original \#1mark.
      2201 }%
\gm@secstar 2202 \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.
      2207 \fi
      2208 \@xa\@xa\csname\gm@secini#1\endcsname
      2209 \gm@secstar{#4}}%
      2211 }{% 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 2214 \def\gm@secini{gm@mw}%
      2216 \let\gm@secmarkh\@gobble% in mwcls there's no need to make tricks for special
              version to running headings.
\gm@secxx 2219 \def\gm@secxx#1#2[#3]#4{%
      2220 \@xa\@xa\csname\gm@secini#1\endcsname
      2221 \gm@secstar[#2][#3]{#4}}%
      2222 }
\gm@sec 2224 \def\gm@sec#1{\@dblarg{\gm@secx{#1}}}
\gm@secx 2225 \def\gm@secx#1[#2]{%
      2226 \@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 2230 \def\gm@straightensec#1{% the parameter is for the command's name.
      2231 \gm@ifundefined{#1}{\@empty}{% we don't change the ontological status of the com-
              mand because someone may test it.

```



```

2233     \n@melet{\gm@secini#1}{#1}%
2234     \@namedef{#1}{%
\gm@secstar 2235         \@ifstar{\def\gm@secstar{*}\gm@sec{#1}}{%
\gm@secstar 2236         \def\gm@secstar{}\gm@sec{#1}}}%
2237     }%

2239     \let\do\gm@straightensec
2240     \do{part}\do{chapter}\do{section}\do{subsection}\do{%
        subsection}
2241     \@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* 2257 \@namedef{enumerate*}{%
2258     \ifnum\@enumdepth>\thr@@
2259         \@toodeep
2260     \else
2261         \advance\@enumdepth\@ne
2262         \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
2263         \@xa\list\csname\label\@enumctr\endcsname{%
2264             \partopsep\topsep\topsep\z@\leftmargin\z@
2265             \itemindent\@parindent\advance\itemindent\labelsep
2266             \labelwidth\@parindent
2267             \advance\labelwidth-\labelsep
2268             \listparindent\@parindent
2269             \usecounter\@enumctr
2270             \def\makelabel##1{##1\hfil}}%
2271         \fi}
2272 \@namedef{endenumerate*}{\endlist}

itemize* 2275 \@namedef{itemize*}{%
2276     \ifnum\@itemdepth>\thr@@
2277         \@toodeep
2278     \else
2279         \advance\@itemdepth\@ne
2280         \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
2281         \@xa\list\csname\@itemitem\endcsname{%
2282             \partopsep\topsep\topsep\z@\leftmargin\z@
2283             \itemindent\@parindent
2284             \labelwidth\@parindent
2285             \advance\labelwidth-\labelsep
2286             \listparindent\@parindent
2287             \def\makelabel##1{##1\hfil}}%
2288         \fi}
2289 \@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.

```
2298 \let\oldLaTeX\LaTeX
2299 \let\oldLaTeXe\LaTeXe
2301 \def\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}
\DeclareLogo 2303 \newcommand*\DeclareLogo[3][\relax]{%
    % [#1] is for non-LATEX spelling and will be used in the PD1 encoding (to make
    % pdf bookmarks);
    % #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 2311 \ifx\relax#1\def\gmu@reserveda{\@xa\@gobble\string#2}%
2312 \else
\gmu@reserveda 2313 \def\gmu@reserveda{#1}%
2314 \fi
2315 \edef\gmu@reserveda{%
2316 \@nx\DeclareTextCommand\@nx#2{PD1}{\gmu@reserveda}}
2317 \gmu@reserveda
2318 \DeclareTextCommandDefault#2{#3}%
\pdef 2319 \pdef#2{#3}% added for XYLATEX.
2320 }
\DeclareLogo 2323 \DeclareLogo\LaTeX{%
2324 {%
2325 L%
2326 \setbox\z@\hbox{\check@mathfonts
2327 \fontsize\sf@size\z@
2328 \math@fontsfalse\selectfont
2329 A}%
2330 \kern-.57\wd\z@
2331 \sbox\tw@_T%
2332 \vbox_1to\ht\tw@{\copy\z@_1\vss}%
2333 \kern-.2\wd\z@}% originally -, 15 em for T.
2334 {%
2335 \ifdim\fontdimen1\font=\z@
2336 \else
2337 \count\z@=\fontdimen5\font
2338 \multiply\count\z@_by_64\relax
2339 \divide\count\z@_by_p@
2340 \count\tw@=\fontdimen1\font
2341 \multiply\count\tw@_by\count\z@
2342 \divide\count\tw@_by_64\relax
2343 \divide\count\tw@_by\tw@
2344 \kern-\the\count\tw@_sp\relax
2345 \fi}%
2346 \TeX}
\LaTeXe 2349 \DeclareLogo\LaTeXe{\mbox{\m@th_1if
2350 b\expandafter\@car\fontseries\@nil\boldmath\fi
2351 \LaTeX\kern.15em2$_{\textstyle\varepsilon}$}}
2353 \StoreMacro\LaTeX
2354 \StoreMacro*\LaTeX_1}
```

'(L<sup>A</sup>)T<sub>E</sub>X' in my opinion better describes what I work with/in than just 'L<sup>A</sup>T<sub>E</sub>X'.

```

\LaTeXpar 2360 \DeclareLogo[(La)TeX]{\LaTeXpar}{%
2361   {%
2362     \setbox\z@\hbox{({}%
2363     \copy\z@
2364     \kern-.2\wd\z@_L%
2365     \setbox\z@\hbox{\check@mathfonts
2366     \fontsize\sf@size\z@
2367     \math@fontsfalse\selectfont
2368     A}%
2369     \kern-.57\wd\z@
2370     \sbox\tw@_T%
2371     \vbox_\to\ht\tw@{\box\z@%
2372     \vss}%
2373   }%
2374   \kern-.07em% originally -, 15 em for T.
2375   {%(
2376     \sbox\z@)%
2377     \kern-.2\wd\z@\copy\z@
2378     \kern-.2\wd\z@}\TeX
2379 }

```

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

```

2386 \gm@ifundefined{AmSTeX}
\AmSTeX 2387   {\def\AmSTeX{\leavevmode\hbox{\mathcal_A\kern-.2em%
2388     \lower.376ex%
2389     \hbox{\mathcal_M}\kern-.2em\mathcal_S-\TeX}}}{}
\BibTeX 2390 \DeclareLogo\BibTeX{\rmfamily_B\kern-.05em%
2391   \textsc{i{\kern-.025em}b}\kern-.08em% the kern is wrapped in braces
2392   for my \fakescaps’ sake.
2393   \TeX}}
\SliTeX 2396 \DeclareLogo\SliTeX{\rmfamily_S\kern-.06emL\kern-.18em%
2397   \raise.32ex\hbox
2398   {\scshape_i}\kern_\-.03em\TeX}}
\PlainTeX 2399 \DeclareLogo\PlainTeX{\textsc{Plain}\kern2pt\TeX}
\Web 2401 \DeclareLogo\Web{\textsc{Web}}

```

There’s also the  $\text{\LaTeX}$  logo got with the  $\text{\LaTeXpar}$  macro provided by  $\text{\gmutils}$ . And here *The  $\text{\TeX}$ book’s* logo:

```

\TeXbook 2404 \DeclareLogo[The_\TeX_\book]\TeXbook{\textsl{The_\TeX_\book}}
2405 \let\TB\TeXbook% TUG Boat uses this.
\TeX 2407 \DeclareLogo[e-TeX]\eTeX{%
2408   \iffontchar\font"03B5{\itshape_\} \else
2409   \ensuremath{\varepsilon}\fi-\kern-.125em\TeX}% definition sent by Karl
2410   Berry from TUG Boat itself.
2411 \StoreMacro\eTeX
\pdfTeX 2414 \DeclareLogo[pdf-e-TeX]\pdfTeX{pdf\eTeX}
\pdfTeX 2416 \DeclareLogo\pdfTeX{pdf\TeX}
\pdfLaTeX 2417 \DeclareLogo\pdfLaTeX{pdf\LaTeX}

```

```

2420 \gm@ifundefined{XeTeX}{%
\XeTeX 2421   \DeclareLogo\XeTeX{X\kern-.125em\relax
2422     \gm@ifundefined{reflectbox}{%
2423       \lower.5ex\hbox{E}\kern-.1667em\relax}{%
2424       \lower.5ex\hbox{\reflectbox{E}}\kern-.1667em\relax}%
2425     \TeX}}{}

2427 \gm@ifundefined{XeLaTeX}{%
\XeLaTeX 2428   \DeclareLogo\XeLaTeX{X\kern-.125em\relax
2429     \gm@ifundefined{reflectbox}{%
2430       \lower.5ex\hbox{E}\kern-.1667em\relax}{%
2431       \lower.5ex\hbox{\reflectbox{E}}\kern-.1667em\relax}%
2432     \LaTeX}}

```

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

```

\LuaTeX 2440 \DeclareLogo[LuaTeX]\LuaTeX{\textsc{Lua}\TeX}

```

### Expandable 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 cses: they could'nt be used within `\csname... \endcsname`. My  $\TeX$ Guru 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 2459 \long\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:

```

2464 \edef\gmu@tempa{%
\gm@gobmacro 2465   \def\@nx\gm@gobmacro##1\@xa\@gobble\string\macro:##2->{}}
2466 \gmu@tempa

```

### Brave New World of $\XeTeX$

```

\@ifXeTeX 2483 \newcommand\@ifXeTeX[2]{%
2484   \ifdefined\XeTeXversion
2485   \unless\ifx\XeTeXversion\relax\afterfifi{#1}\else\afterfifi{%
      #2}\fi
2486   \else\afterfi{#2}\fi}

\XeTeXthree 2489 \DeclareDocumentCommand\XeTeXthree{o}{%
2493   \@ifXeTeX{%
2494     \IfValueT{#1}{\PassOptionsToPackage{#1}{fontspec}}%
2495     \@ifpackageloaded{gmverb}{\StoreMacro\verb}{}}%
2496     \RequirePackage{xltxtra}% since v 0.4 (2008/07/29) this package redef-
      fines \verb and verbatim*, and quite elegantly provides an option to
      suppress the redefinitions, but unfortunately that option excludes also
      a nice definition of \xxt@visiblespace which I fancy.

```

```

2503     \@ifpackageloaded{gmverb}{\RestoreMacro\verb}{}%
2504     \AtBeginDocument{%
2505         \RestoreMacro\LaTeX\RestoreMacro*{LaTeX_}% my version of the
                LATEX logo has been stored just after defining, in line 2354.
2508     \RestoreMacro\eTeX}%
2509 }{}

```

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

```

2514 \AtBeginDocument{%
2515     \@ifpackageloaded{fontspec}{%
\udigits 2516         \pdf\udigits{%
2517             \addfontfeature{Numbers=Uppercase}}%
2518         }{%
2519         \emptify\udigits}}

```

## Fractions

```

\Xedekfracc 2524 \def\Xedekfracc{\@ifstar\gmu@xedekfraccstar\gmu@xedekfraccplain}

```

(plain) The starless version turns the font feature `frac` on.

(\*) But nor my modification of Minion Pro 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 `\gmu@dekfracc`).

(\*\*) 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@xedekfraccstar 2539 \def\gmu@xedekfraccstar{%
\gmu@xefraccdef 2540     \def\gmu@xefraccdef##1##2{%
2541         \iffontchar\font_##2
2542         \@namedef{gmu@xefracc##1}{\char##2_}%
2543         \else
2544         \n@melet{gmu@xefracc##1}{relax}%
2545         \fi}%
\gmu@dekfracc 2547     \def\gmu@dekfracc##1/##2{%
2548         {\addfontfeature{VerticalPosition=Numerator}##1}%
                \gmu@numeratorkern
2549         \char"2044_\gmu@denominatorkern
2550         {\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.

```

2554     \gmu@xefraccdef{1/4}{"BC}%
2555     \gmu@xefraccdef{1/2}{"BD}%
2556     \gmu@xefraccdef{3/4}{"BE}%
2557     \gmu@xefraccdef{1/3}{"2153}%
2558     \gmu@xefraccdef{2/3}{"2154}%
2559     \gmu@xefraccdef{1/8}{"215B}%
2560     \gmu@xefraccdef{3/8}{"215C}%
2561     \gmu@xefraccdef{5/8}{"215D}%
2562     \gmu@xefraccdef{7/8}{"215E}%
\dekfracc@args 2563     \pdf\dekfracc@args##1/##2{%
\gm@duppa 2564         \def\gm@duppa{##1/##2}%

```

```

2565     \gm@ifundefined{gmu@xfracc\all@other\gm@duppa}{%
2566     \gmu@dekfracc{##1}/{##2}}{%
2567     \csname_gmu@xfracc\all@other\gm@duppa\endcsname}%
2568     \if@gmu@mmhbox\egroup\fi
2569     }% of \dekfracc@args.
2570     \@ifstar{\let\gmu@dekfracc\gmu@dekfracccsimple}{}%
2571     }
\gmu@xedekfraccplain 2573 \def\gmu@xedekfraccplain{% 'else' of the main \@ifstar
\dekfracc@args      2574 \pdef\dekfracc@args##1/##2{%
2575     \ifmmode\hbox\fi{%
2576     \addfontfeature{Fractions=0n}%
2577     ##1/##2}%
2578     \if@gmu@mmhbox\egroup\fi
2579     }% of \dekfracc@args
2580     }
\if@gmu@mmhbox      2582 \newif\if@gmu@mmhbox% we'll use this switch for \dekfracc and also for \thous
                    (hacky thousand separator).
\dekfracc           2585 \pdef\dekfracc{%
2587     \ifmmode\hbox\bgroup\@gmu@mmhboxtrue\fi
2588     \dekfracc@args}
\gmu@numeratorkern 2591 \def\gmu@numeratorkern{\kern-.05em\relax}
2592 \let\gmu@denominator\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  $\text{\LaTeX} 2_{\epsilon}$  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 3439.

```

\resizegraphics
\resizegraphics 2614 \def\resizegraphics#1#2#3{%
2617     \resizebox{#1}{#2}{%
2618     \includegraphics{#3}}}
\GMtextsuperscript 2620 \def\GMtextsuperscript{%
2621     \@ifXeTeX{%
\textsuperscript 2622     \def\textsuperscript##1{%
2623         \addfontfeature{VerticalPosition=Numerator}##1}}%
2624     }{\truetextsuperscript}}
\truetextsuperscript 2626 \def\truetextsuperscript{%
\textsuperscript 2627     \pdef\textsuperscript##1{%
2628         \@textsuperscript{\selectfont##1}}%
\@textsuperscript 2629     \def\@textsuperscript##1{%
2630         {\m@th\ensuremath{\sim\mbox{\fontsize\sf@size\z@##1}}}}}}

```

## Settings for mathematics in main font

```

\gmath I used these terrible macros while typesetting E. Szarzyński's Letters in 2008. The \gmath
\garamath declaration introduces math-active digits and binary operators and redefines greek let-
ters and parentheses, the \garamath declaration redefines the quantifiers and is more
Garamond Premier Pro-specific.

\gmu@getfontstring 2644 \def\gmu@getfontstring{%
2645 \xdef\gmu@fontstring{%
2646 \gmu@fontstring@}}

\gmu@fontstring@ 2648 \def\gmu@fontstring@{%
2649 \@xa\@xa\@xa\gmu@quotedstring\@xa\meaning\the\font\@@nil}

\gmu@quotedstring 2651 \def\gmu@quotedstring#1"#2"#3\@@nil{"#2"}

\gmu@getfontscale 2653 \def\gmu@getfontscale#1Scale#2=#3,{%
2654 \ifx\gmu@getfontscale#3\else
2655 \gdef\gmu@fontscale{[#3]_}%
2656 \afterfi\gmu@getfontscale\fi
2657 }

\gmu@getfontdata 2660 \def\gmu@getfontdata#1{%
2661 \global\emptify\gmu@fontscale
2662 \begingroup
2663 #1%
2664 \@xa\@xa\@xa\gmu@getfontscale
2665 \csname_zf@family@options\fi@family\endcsname
2666 ,Scale=\gmu@getfontscale,%
2667 \gmu@getfontstring
2668 \xdef\gmu@theskewchar{\the\skewchar\font}%
2669 \endgroup}

\gmu@stripchar 2672 \def\gmu@stripchar#1"{"}

\gmath@getfamnum 2674 \def\gmath@getfamnum{%
2675 \edef\gmath@famnum{\@xa\gmu@stripchar\meaning\gmath@fam}%
2677 }

\XeTeXmathcode<char slot> [(<=)] <type> <family> <char slot>

\gmathbase 2681 \pdef\gmathbase{%
2682 \gmu@getfontdata{\rmfamily\itshape}%
2684 \edef\gmu@tempa{%
2685 \@nx\DeclareSymbolFont{letters}{\encodingdefault}{gmathit}{%
m}{it}%
2686 \@nx\DeclareFontFamily{\encodingdefault}{gmathit}{%
\skewchar\font\gmu@theskewchar\space}%
2687 \@nx\DeclareFontShape{\encodingdefault}{gmathit}{m}{it}{%
2688 <->_ \gmu@fontscale_ \gmu@fontstring}{}%
2689 } \gmu@tempa\typeout{@@@_gmathit_(letters):_ \meaning\gmu@tempa}%
2690 \gmu@getfontdata{\rmfamily\upshape}%
2693 \edef\gmu@tempa{%
2694 \@nx\DeclareSymbolFont{gmathroman}{\encodingdefault}{%
gmathrm}{m}{n}%
2695 \@nx\DeclareFontFamily{\encodingdefault}{gmathrm}{%
\skewchar\font\gmu@theskewchar\space}%
2696 \@nx\DeclareFontShape{\encodingdefault}{gmathrm}{m}{n}{%
2697 <->_ \gmu@fontscale_ \gmu@fontstring}{}%
2698 }

```

```

2699 } \gmu@tempa\typeout{@@@_gmathrm_(upright_symbols):
2700   \meaning\gmu@tempa}%
2701 \font\gmath@font=\gmu@fontstring\relax
\gmath@do 2703 \DeclareDocumentCommand\gmath@do{mom}{%
           % #1 the character or cs to be declared,
           % [#2] the Unicode to be assigned,
           % #3 math type (cs like \mathord etc.)
2710   \gmath@getfamnum
2711   \IfValueTF{##2}{%
2712     \edef\gmu@tempa{%
2713       =_ \mathchar@type##3\space
2714       \gmath@famnum\space
2715       "##2\relax}%
2716   \if\relax\@nx##1%
2717     \edef\gmu@tempa{%
2718       \XeTeXmathchardef_ \@nx##1\gmu@tempa}%
2719   \else
2720     \edef\gmu@tempa{%
2721       \XeTeXmathcode_ `##1_ \gmu@tempa}
2722   \fi%
2723 }%
2724 {%
2725   \edef\gmu@tempa{%
2726     \XeTeXmathcode_ `##1_ =
2727     \mathchar@type##3\space
2728     \gmath@famnum\space
2729     `##1\relax}%
2730 }%
2731 \gmu@tempa
2732 \typeout{@@@_ \@nx##1}%
2733 \typeout{@@@_ \meaning\gmu@tempa}%
2734 }% of \gmath@do
\gmath@doif 2738 \DeclareDocumentCommand\gmath@doif{mmmoo}{%
           % #1 the Unicode of char enquired,
           % #2 the char or cs to be declared,
           % #3 math type cs(\mathord etc.),
           % [#4] second-choice Unicode (taken if first-choice is absent),
           % [#5] third-choice Unicode (as above if second-choice is absent from
           font).
2748   \iffontchar\gmath@font"##1_ \gmath@do##2 [##1] ##3%
2749   \else\IfValueT{##4}{%
2750     \iffontchar\gmath@font"##4_ \gmath@do##2 [##4] ##3%
2751     \else\IfValueT{##5}{%
2752       \iffontchar\gmath@font"##5_ \gmath@do##2 [##5] ##3%
2753       \fi}%
2754     \fi}%
2755   \fi}%
2756 \iffalse_ doesn't work in a non-math font.
\gmath@delc 2758 \DeclareDocumentCommand\gmath@delc{mo}{%
           % #1 the char or cs to be declared,
           % [#2] the Unicode (if not the same as the char).
2764   \gmath@getfamnum

```



```

2765 \IfValueTF{##2}{%
2766 \edef\gmu@tempa{%
2767 =\gmath@famnum\space"##2\relax}%
2768 \edef\gmu@tempa{%
2769 \XeTeXdelcode`##1\gmu@tempa}
2770 }%
2771 {%
2772 \edef\gmu@tempa{%
2773 \XeTeXdelcode`##1=
2774 \gmath@famnum\space
2776 `##1\relax}%
2777 }%
2778 \gmu@tempa
2779 \typeout{@@@\@nx##1}%
2780 \typeout{@@@\@meaning\gmu@tempa}%
2781 }% of \gmath@delc
\gmath@delcif 2783 \def\gmath@delcif##1##2{%
% #1 the Unicode enquired,
% #2 the char to be delcode-declared
2789 \iffontchar\gmath@font"##1\gmath@delc##2[##1]\fi}
2790 \fi}% of iffalse
\gmath@delimif 2792 \def\gmath@delimif##1##2##3{%
% #1 the Unicode enquired,
% #2 the cs defined as \XeTeXdelimiter,
% #3 the math type cs (probably \mathopen or \mathclose).
2799 \iffontchar\gmath@font"##1
2800 \gmath@getfamnum
2801 \protected\edef##2{\@nx\ensuremath{%
2802 \XeTeXdelimiter\mathchar@type##3\space
2803 \gmath@famnum\space"##1\relax}}%
2804 \fi}% of \gmath@delimif.
\gmu@dogmathbase 2806 \pdef\gmu@dogmathbase{%
2808 \let\gmath@fam\symgmathroman
2810 \typeout{@@@\gmuutils.sty:taking some math chars from the
font^^J\gmu@fontstring@}%
2811 \gmath@do+\mathbin
2812 \gmath@doif{2212}-\mathbin[2013]% minus sign if present or else en dash
2813 \gmath@do=\mathrel
2814 \gmath@doo\mathord
2815 \gmath@do1\mathord
2816 \gmath@do2\mathord
2817 \gmath@do3\mathord
2818 \gmath@do4\mathord
2819 \gmath@do5\mathord
2820 \gmath@do6\mathord
2821 \gmath@do7\mathord
2822 \gmath@do8\mathord
2823 \gmath@dog\mathord
2825 \gmath@doif{2A7D}\xleq\mathrel
2826 \gmath@doif{2A7E}\xgeq\mathrel
2827 \@ifpackageloaded{polski}{%
2828 \ifdefined\xleq

```

```

2829     \let\leq=\xleq
2830     \let\le=\leq
2831     \fi
2832     \ifdefined\xgeq
2833     \let\geq=\xgeq
2834     \let\ge=\geq
2835     \fi}{}%
2837     \gmath@do.\mathpunct
2838     \gmath@do,\mathpunct
2839     \gmath@do;\mathpunct
2840     \gmath@do...\mathpunct
2841     \gmath@do(\mathopen
2844     \gmath@do)\mathclose
2846     \gmath@do[\mathopen
2848     \gmath@do]\mathclose
2851     \gmath@doif{00D7}* \mathbin
2852     \gmath@do:\mathrel
2853     \gmath@doif{00B7}\cdot\mathbin
2854     \gmath@doif{22C6}* \mathbin
2855     \gmath@doif{2300}\varnothing\mathord
2856     \gmath@doif{221E}\infty\mathord
2857     \gmath@doif{2248}\approx\mathrel
2858     \gmath@doif{2260}\neq\mathrel
2859     \let\ne\neq
2860     \gmath@doif{00AC}\neg\mathbin
2861     \gmath@do/\mathop
2862     \gmath@do<\mathrel
2864     \gmath@do>\mathrel
2866     \gmath@doif{2329}\langle\mathopen
2867     \gmath@doif{232A}\rangle\mathclose
2868     \gmath@doif{2202}\partial\mathord
2869     \gmath@doif{00B1}\pm\mathbin
2870     \gmath@doif{007E}\sim\mathrel
2871     \gmath@doif{2190}\leftarrow\mathrel
2872     \gmath@doif{2192}\rightarrow\mathrel
2873     \gmath@doif{2194}\leftrightarrow\mathrel% if not present, \gmathfurther
        will take care of it if left and right arrows are present.
2876     \gmath@doif{2191}\uparrow\mathrel% it should be a delimiter (declared
        with \gmath@delimif) but in a non-math font the delimiters don't work
        (2008/11/19) and I don't think I'll ever need up- and down- arrows as
        delimiters.
2880     \gmath@doif{2193}\downarrow\mathrel
2882     \gmath@doif{2208}\in\mathrel[o3F5][o454]%

```

As a fan of modal logics I allow redefinition of `\lozenge` and `\square` iff both are in the font. I don't accept the 'ballot box' U+2610.

```

2886     \if\iffontchar\gmath@font"25CA_o\else_1\fi
2887     \iffontchar\gmath@font"25FB_o\else\iffontchar%
        \gmath@font"25A1_o\else_2\fi\fi
2888     \gmath@do\lozenge[25CA]\mathord
2889     \gmath@doif{25FB}\square\mathord[25A1]% 'medium white square (modal
        operator)' of just 'white square'.
2891     \fi

```

```

2892 \gmath@doif{EBo8}\bigcircle\mathbin
2893 \gmath@doif{2227}\wedge\mathbin
2894 \gmath@doif{2228}\vee\mathbin
2896 \gmath@doif{0393}\Gamma\mathalpha
2897 \gmath@doif{0394}\Delta\mathalpha
2898 \gmath@doif{0398}\Theta\mathalpha
2899 \gmath@doif{039B}\Lambda\mathalpha
2900 \gmath@doif{039E}\Xi\mathalpha
2901 \gmath@doif{03A3}\Sigma\mathalpha
2902 \gmath@doif{03A5}\Upsilon\mathalpha
2903 \gmath@doif{03 6}\Phi\mathalpha
2904 \gmath@doif{03A8}\Psi\mathalpha
2905 \gmath@doif{03A9}\Omega\mathalpha
2907 \let\gmath@fam\symletters
2909 \gmath@doif{03B1}\alpha\mathalpha
2910 \gmath@doif{03B2}\beta\mathalpha
2911 \gmath@doif{03B3}\gamma\mathalpha
2912 \gmath@doif{03B4}\delta\mathalpha
2913 \gmath@doif{03F5}\epsilon\mathalpha
2914 \gmath@doif{03B5}\varepsilon\mathalpha
2915 \gmath@doif{03B6}\zeta\mathalpha
2916 \gmath@doif{03B7}\eta\mathalpha
2917 \gmath@doif{03B8}\theta\mathalpha
2918 \gmath@doif{03D1}\vartheta\mathalpha
2919 \gmath@doif{03B9}\iota\mathalpha
2920 \gmath@doif{03BA}\kappa\mathalpha
2921 \gmath@doif{03BB}\lambda\mathalpha
2922 \gmath@doif{03BC}\mu\mathalpha
2923 \gmath@doif{03BD}\nu\mathalpha
2924 \gmath@doif{03BE}\xi\mathalpha
2925 \gmath@doif{03C0}\pi\mathalpha
2926 \gmath@doif{03A0}\Pi\mathalpha
2927 \gmath@doif{03C1}\rho\mathalpha
2928 \gmath@doif{03C3}\sigma\mathalpha
2929 \gmath@doif{03DA}\varsigma\mathalpha% 03C2?
2930 \gmath@doif{03C4}\tau\mathalpha
2931 \gmath@doif{03C5}\upsilon\mathalpha
2932 \gmath@doif{03D5}\phi\mathalpha
2933 \gmath@doif{03C8}\psi\mathalpha
2934 \gmath@doif{03C9}\omega\mathalpha
2936 \if_1_1%
2937 \iffontchar\gmath@font"221A
2938 \fontdimen61\gmath@font=1pt
2939 \edef\sqrtsign{%
2940 \XeTeXradical_@\xa\gmu@stripchar\meaning\symgmathroman%
\space_"221A\relax}%
2941 \fi
2942 \fi% of if 1 1.
2943 }%
2944 \AtBeginDocument{\gmu@dogmathbase\let\gmathbase%
\gmu@dogmathbase}%
2945 \not@onlypreamble\gmathbase
2946 }% of \gmathbase

```

2948 \@onlypreamble\gmathbase

It's a bit tricky: if \gmathbase occurs first time in a document inside document then an error error is raised. But if \gmathbase occurs first time in the preamble, then it removes itself from the only-preamble list and redefines itself to be only the inner macro of the former itself.

```
\gmathfurther 2955 \pdef\gmathfurther{%
2962   \def\do##1##2##3{\def##1{%
2963     \mathop{\mathchoice{\hbox{%
2964       \rm
2965       \edef\gma@tempa{\the\fontdimen8\font}%
2966       \larger[3]%
2967       \lower\dimexpr(\fontdimen8\font-\gma@tempa)/2\fontdimen8\font-
2968       \hbox{##2}}{\hbox{%
2969         \rm
2970         \edef\gma@tempa{\the\fontdimen8\font}%
2971         \larger[2]%
2972         \lower\dimexpr(\fontdimen8\font-\gma@tempa)/2\fontdimen8\font-
2973         \hbox{##2}}}%
2974       {\mathrm{##2}}{\mathrm{##2}}##3}}%
2975   \iffontchar\gmath@font"2211\do\sum{\char"2211}\fi%
2976   \do\forall{\gma@quantifierhook\rotatebox[origin=c]{180}{A}%
2977     \gmu@forallkerning
2978   }\nolimits}%
2979   \def\gmu@forallkerning{\setboxo=\hbox{A}\setbox2=\hbox{%
2980     \scriptsize_x}%
2981     \kern\dimexpr\ht2/3*2\wdo/2\relax}% to be able to redefine it when
2982     the big quantifier is Bauhaus-like.
2983   \do\exists{\rotatebox[origin=c]{180}{\gma@quantifierhook_E}}%
2984     \nolimits%
2985   \def\do##1##2##3{\def##1{##3{%
2986     \mathchoice{\hbox{\rm##2}}{\hbox{\rm##2}}%
2987     {\hbox{\rm\scriptsize##2}}{\hbox{\rm\tiny##2}}}}}%
2988   \unless\iffontchar\gmath@font"2227
2989     \do\vee{\rotatebox[origin=c]{90}{<}}\mathbin%
2990   \fi
2991   \unless\iffontchar\gmath@font"2228
2992     \do\wedge{\rotatebox[origin=c]{-90}{<}}\mathbin
2993   \fi
2994   \unless\iffontchar\gmath@font"2194
2995     \if\iffontchar\gmath@font"2190\else1\fi
2996     \iffontchar\gmath@font"2192\else2\fi
2997     \do\leftrightharpoon{\char"2190\kern-0,1em\char"2192}%
2998     \mathrel
3000   \fi\fi
3001   \def\do##1##2##3{\def##1{##2{\hbox{%
3002     \rm
3003     \setboxo=\hbox{##1}%
3004     \edef\gma@tempa{\the\hto}%
3005     \edef\gma@tempb{\the\dpo}%
3006     ##3%
3007     \setboxo=\hbox{##1}%
3008     \setboxo=\hbox{##1}}}}
```

```

3009         \lower\dimexpr(\hto_+ \dpo)/2-\dpo_-(\gma@tempa+%
           \gma@tempb)/2-\gma@tempb)_%
3010     \boxo}}}}%
3011     \do\bigl\mathopen\larger
3012     \do\bigr\mathclose\larger
3013     \do\Bigl\mathopen\largerr
3014     \do\Bigr\mathclose\largerr
3015     \do\biggl\mathopen{\larger[3]}%
3016     \do\biggr\mathclose{\larger[3]}%
3017     \do\Biggl\mathopen{\larger[4]}%
3018     \do\Biggr\mathclose{\larger[4]}%
3021     \addtotoks\everymath{%
3024         \def\do##1##2{\def##1{\ifmmode##2{\mathchoice
3025             {\hbox{\rm\char`##1}}{\hbox{\rm\char`##1}}%
3026             {\hbox{\rm\scriptsize\char`##1}}{\hbox{\rm\tiny%
               \char`##1}}}%
3027         \else\char`##1\fi}}%
3029     \do{\mathopen
3030     \do{\mathclose
3032     \def\={\mathbin{=}}%
3033     \def\neqb{\mathbin{\neq}}%
3034     \let\neb\neqb
3035     \def\do##1{\edef\gma@tempa{%
3036         \def\@xa\@nx\csname_\@xa\gobble\string##1r\endcsname{%
3037             \@nx\mathrel{\@nx##1}}}%
3038         \gma@tempa}%
3039     \do\vee_ \do\wedge_ \do\neg
3040     \def\fakern{\mkern-3mu}%
3041     \thickmuskip=8mu_ plus_4mu\relax
3043     \gma@gmathhook
3044 }% of \everymath.
3045 \everydisplay\everymath
3046 \ifdefined\Url
3047     \ampulexdef\Url{\let\do}\@makeother
3048     {\everymath}\let\do\@makeother}% I don't know why but the url package's
           % \url typesets the argument inside a math which caused digits not to
           % be typewriter but Roman and lowercase.
3052     \fi% of ifdefined Url.
3053 }% of \def\gmathfurther.

\gmath 3055 \def\gmath{\gmathbase\gmathfurther}

\gmathscripts 3057 \pdef\gmathscripts{%
3058     \addtotoks\everymath{\catcode`\^=7\relax_ \catcode`\_ =8\relax_}%
3059     \everydisplay\everymath}

\gmathcats 3061 \pdef\gmathcats{%
3062     \addtotoks\everymath{\gmu@septify}%
3063     \everydisplay\everymath}

\quantifierhook 3065 \emptify\gma@quantifierhook
\gma@quantifierhook 3066 \def\quantifierhook#1{%
3067     \def\gma@quantifierhook{#1}}

\gmathhook 3069 \emptify\gma@gmathhook
\gma@gmathhook 3070 \def\gmathhook#1{\addtomacro\gma@gmathhook{#1}}

```

```

\gma@dollar 3073 \def\gma@dollar$#1$${\gmath$#1$}}%
\gma@bare 3074 \def\gma@bare#1{\gma@dollar$#1$}%
\gma@checkbracket 3075 \def\gma@checkbracket{\@ifnextchar\[%
3076 \gma@bracket\gma@bare}
\gma@bracket 3077 \def\gma@bracket\[#1\]{\gmath\[#1\]}\@ifnextchar\par}{\%
\noindent}}
\gma 3078 \def\gma{\@ifnextchar$%
3079 \gma@dollar\gma@checkbracket}
\garamath 3085 \def\garamath{%
3086 \addtotoks\everymath{%
3087 \quantifierhook{\addfontfeature{OpticalSize=800}}%
\gma@arrowdash 3089 \def\gma@arrowdash{%
3090 \setboxo=\hbox{\char"2192}\copyo\kern-0,6\wdo
3091 \bgcolor\rule[-\dpo]{0,6\wdo}{\dimexpr\hto+\dpo}%
\kern-0,6\wdo}}%
\gma@gmathhook 3093 \def\gma@gmathhook{%
3094 \def\do####1####2####3{\def####1{####3{%
\mathchoice 3095 \mathchoice{\hbox{\rm####2}}{\hbox{\rm####2}}%
3096 {\hbox{\rm\scriptsize####2}}{\hbox{\rm%
\tiny####2}}}}}%
3097 \do\mapsto{\rule[0,4ex]{0,1ex}{0,4ex}\kern-0,05em%
3098 \gma@arrowdash\kern-0,05em\char"2192}\mathrel
3099 \do\cup{\scshape_u}\mathbin
3100 \do\varnothing{\setboxo=\hbox{\gma@quantifierhook%
\addfontfeature{Scale=1.272727}o}%
3101 \setbox2=\hbox{\char"2044}}%
3102 \copyo_\kern-0,5\wdo_\kern-0,5\wd2_\lowero,125\wdo_\copy2
3103 \kerno,5\wdo\kern-0,5\wd2}{}}%
3104 \do\leftarrow{\char"2190\kern-0,05em\gma@arrowdash}\mathrel
3105 \do\rightarrow{\gma@arrowdash\kern-0,05em\char"2192}%
\mathrel
3106 \do\in{\gma@quantifierhook\char"0454}\mathbin
3107 }}%
3108 \everydisplay\everymath}

```

### Minion and Garamond Premier kerning and ligature fixes

»Ws« shall not make long »s« because long »s« looks ugly next to »W«.

```

\gmu@tempa 3117 \def\gmu@tempa{\kern-0,08em\penalty10000\hskiposp\relax
3118 s\penalty10000\hskiposp\relax}
3120 \protected\edef\Vs{V\gmu@tempa}
3122 \protected\edef\Ws{W\gmu@tempa}
\Wz 3124 \pdef\Wz{W\kern-0,05em\penalty10000\hskiposp\relax_z}

```

### Varia

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

```

\gmu@tilde 3133 \def\gmu@tilde{%
3134 \leavevmode\lower.8ex\hbox{\$, \widetilde{\mbox{_\}}\, \$}}

```

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

```
\* 3138 \pdef\*{\gmu@tilde}
3144 \AtBeginDocument{% to bypass redefinition of \~ as a text command with various
encodings
\texttilde 3146 \pdef\texttilde{%
3149 \@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 `\activeate` the space but also will (re)define it as the `\_` primitive. So define `\gmobeyspaces` that obeys this requirement.

(This definition is repeated in `gmverb`.)

```
3161 \foone{\catcode\_ \active}%
\gmobeyspaces 3162 {\def\gmobeyspaces{\let\_ \_ \catcode\_ \_ \active}}
```

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.

```
3169 \foone{\catcode\_ \^M \active}% the comment signs here are crucial.
\defobeylines 3170 {\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:

```
\deksmallskip 3179 \def\deksmallskip{\vskip\smallskipamount}
\undeksmallskip 3180 \def\undeksmallskip{\vskip-\smallskipamount}
\dekmedskip 3181 \def\dekmedskip{\vskip\medskipamount}
\dekbigskip 3182 \def\dekbigskip{\vskip\bigskipamount}
\hfillneg 3185 \def\hfillneg{\hskip\_opt\_plus\_ -1fill\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 3193 \long\def\@firstofmany#1#2\@nil{#1}
\@secondofmany 3195 \long\def\@secondofmany#1#2\@nil{#2}
```

A mark for the **TODO!**:

```
\TODO 3199 \newcommand*\TODO}[1] [] {%
3200 \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 3235 \newcommand*\twocoltoc{%
3236 \RequirePackage{multicol}%
\starttoc 3237 \def\@starttoc##1{%
3238 \begin{multicols}{2}\makeatletter\@input\_ {\jobname\_ .##1}%
3239 \if@filesw\_ \@xa\_ \newwrite\_ \csname\_ tf@##1\endcsname
3240 \immediate\_ \openout\_ \csname\_ tf@##1\endcsname\_ \jobname\_
.##1\relax
```

```

3241     \fi
3242     \@nbreakfalse\end{multicols}}
3244 \@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 3249 \newcommand*\enoughpage [1] {%
3250     \par
3251     \dimeno=\pagegoal
3252     \advance\dimeno_\by-\pagetotal
3253     \ifdim\dimeno<#1\relax\newpage\fi}

```

An equality sign properly spaced:

```

\equals 3262 \pdef\equals{\hunskip$={}={$}\ignorespaces}

```

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

```

\eequals 3264 \pdef\eequals{\hunskip${]=={$}\ignorespaces}
\cdot 3266 \pdef\cdot{\hunskip${\cdot}{}\ignorespaces}

```

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 3286 \def\freeze@actives{%
3287     \count\z@\z@
3289     \@whilenum\count\z@<\@cclvi\do{%
3290         \ifnum\catcode\count\z@=\active
\~ 3291             \uccode`~=\count\z@
3292             \uppercase{\let~\relax}%
3293         \fi
3294         \advance\count\z@\@ne}}

```

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

```

\ShowFont 3300 \newcommand*\ShowFont [1] [6] {%
3301     \begin{multicols}{#1}[The_\current_\font_(the_\f@encoding\_\_
        encoding):]
3302     \parindent\z@
3303     \count\z@\@m@ne
3304     \@whilenum\count\z@<\@cclv\do{
3305         \advance\count\z@\@ne
3306         \_\the\count\z@:\~\char\count\z@\par}
3307     \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 3315 \newcommand*\liturgiques [1] [red] {% Requires the color package.
3316     \gmu@RPfor{xcolor}\color%
\czerwo 3317     \newcommand*\czerwo{\small\color{#1}}% environment
\czer 3318     \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.

```
\* 3321 \def\*{\czer{*$*$}}
\+ 3322 \def\+{\czer{$\dag$}}
\nieczer 3323 \newcommand*\nieczer[1]{\textcolor{black}{##1}}
```

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

```
\gmu@RPfor 3328 \newcommand*\gmu@RPfor[3][]{%
3330 \ifx\relax#1\relax\emptify\gmu@resa
\gmu@resa 3331 \else\def\gmu@resa{[#1]}%
3332 \fi
3333 \@xa\RequirePackage\gmu@resa[#2]}
```

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

```
3339 \AtBeginDocument{%
\gmu@RPfor 3340 \renewcommand*\gmu@RPfor[3][]{%
3341 \unless\ifdefined#3%
3342 \@ifpackageloaded{#2}{}{%
3343 \typeout{^^J!\Package`#2' not loaded!!!(\on@line)^^J}}%
3344 \fi}}
```

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 3350 \provide\continuum{%
3351 \gmu@RPfor{eufrak}\mathfrak\ensuremath{\mathfrak{c}}}
```

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

```
\iteracro 3355 \def\iteracro{%
\acro 3356 \pdef\acro##1{\gmu@acrospace##1\gmu@acrospace}%
3357 }
```

```
3359 \iteracro
```

```
\gmu@acrospace 3361 \def\gmu@acrospace#1#2\gmu@acrospace{%
3362 \gmu@acroinner#1\gmu@acroinner
3363 \ifx\relax#2\relax\else
3364 \space
3365 \afterfi{\gmu@acrospace#2\gmu@acrospace}% when #2 is nonempty, it
is ended with a space. Adding one more space in this line resulted in an
infinite loop, of course.
3369 \fi}
```

```
\gmu@acroinner 3372 \def\gmu@acroinner#1{%
3373 \ifx\gmu@acroinner#1\relax\else
3374 \ifcat@a@nx#1\relax%
3375 \ifnum`#1=\ucode`#1%
3376 {\acrocore{#1}}%
3377 \else{#1}% tu byto \smallerr
3378 \fi
3379 \else#1%
3380 \fi
3381 \afterfi\gmu@acroinner
3382 \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 3386 \def\acrocore{\scshape\lowercase}
```

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

```
\IMO 3391 \newcommand*\IMO{\acro{IMO}}
```

```
\AKA 3392 \newcommand*\AKA{\acro{AKA}}
```

```
\usc 3394 \pdef\usc#1{\addfontfeature{Letters=UppercaseSmallCaps}#1}}
```

```
\uscacro 3396 \def\uscacro{\let\acro\usc}
```

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 3414 \def\secondclass{%
\ifSecondClass 3415 \newif\ifSecondClass
3416 \SecondClasstrue
3417 \@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 3429 \def\gmu@dekfraccsimple#1/#2{\leavevmode\kern.1em
3430 \raise.5ex\hbox{%
3431 \smaller[3]#1}\gmu@numeratorkern
3432 \dekfracslash\gmu@denominatorkern
3434 }%
3435 \smaller[3]#2}%
3436 \if@gmu@mmhbox\egroup\fi}
```

```
\dekfraccsimple 3439 \def\dekfraccsimple{%
3440 \let\dekfracc@args\gmu@dekfraccsimple
3441 }
```

```
\dekfracslash 3442 \@ifXeTeX{\def\dekfracslash{\char"2044}}{%
```

```
\dekfracslash 3443 \def\dekfracslash{/}}\% You can define it as the fraction
slash, \char"2044
```

```
3445 \dekfraccsimple
```

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

```
\ikern 3453 \newcommand*\ikern{\,\penalty10000\hskiposp\relax}
```

And a macro to forbid hyphenation of the next word:

```
\nohy 3457 \newcommand*\nohy{\leavevmode\kernosp\relax}
```

```
\yeshy 3458 \newcommand*\yeshy{\leavevmode\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 3464 \long\pdef\@ifempty#1#2#3{%
\gmu@reserveda 3465 \def\gmu@reserveda{#1}%
3466 \ifx\gmu@reserveda\@empty\afterfi{#2}%
3467 \else\afterfi{#3}\fi
3468 }

\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 3480 \def\gmu@getext#1.#2\@nil{%
3481 \def\gmu@filename{#1}%
3482 \def\gmu@fileext{#2}}

3484 \def\include#1{\relax
3485 \ifnum\@auxout=\@partaux
3486 \@latex@error{\string\include\space cannot be nested}\@eha
3487 \else\@include#1\fi}

\@include 3489 \def\@include#1{%
3490 \gmu@getext#1.\@nil
3492 \ifx\gmu@fileext\empty\def\gmu@fileext{tex}\fi
3493 \clearpage
3494 \if@filesw
3495 \immediate\write\@mainaux{\string\@input{\gmu@filename.aux}}%
3496 \fi
3497 \@tempswatrue
3498 \if@partsw
3499 \@tempswafalse
3500 \edef\reserved@b{#1}%
3501 \@for\reserved@a:=\@partlist\do{%
3502 \ifx\reserved@a\reserved@b\@tempswatrue\fi}%
3503 \fi
3504 \if@tempswa
3505 \let\@auxout\@partaux
3506 \if@filesw
3507 \immediate\openout\@partaux\gmu@filename.aux
3508 \immediate\write\@partaux{\relax}%
3509 \fi
3510 \@input@{\gmu@filename.\gmu@fileext}%
3511 \inclasthook
3512 \clearpage
3513 \@writeckpt{\gmu@filename}%
3514 \if@filesw
3515 \immediate\closeout\@partaux
3516 \fi
3517 \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.

3521 \deadcycles\z@

```

```

3522     \@nameuse{cp@\gmu@filename}%
3523     \fi
3524     \let\@auxout\@mainaux}
\whenonly 3527 \newcommand\whenonly[3]{%
\gmu@whonly 3528     \def\gmu@whonly{#1,}%
3529     \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 3533 \def\inclasthook{\thispagestyle{closing}}

```

### Faked small caps

```

\gmu@scapLetters 3539 \def\gmu@scapLetters#1{%
3540     \ifx#1\relax\relax\else% two \relaxes to cover the case of empty #1.
3541     \ifcat_a#1\relax
3542     \ifnum\the\lccode`#1=`#1\relax
3543     {\fakescapsscore\MakeUppercase{#1}}% not Plain \uppercase because
        that works bad with inputenc.
3545     \else#1%
3546     \fi
3547     \else#1%
3548     \fi%
3549     \@xa\gmu@scapLetters
3550     \fi}%
\gmu@scapSpaces 3552 \def\gmu@scapSpaces#1_#2\@nil{%
3553     \ifx#1\relax\relax
3554     \else\gmu@scapLetters#1\relax
3555     \fi
3556     \ifx#2\relax\relax
3557     \else\afterfi{\_ \gmu@scapSpaces#2\@nil}%
3558     \fi}
\gmu@scapss 3560 \def\gmu@scapss#1\@nil{{\def~{\nobreakspace}}%
\nobreakspace 3561     \gmu@scapSpaces#1_ \@nil}}% \_ \def\\{\newline}}\relax adding re-
        definition of \_ caused stack overflow. Note it disallows hyphenation
        except at \-.
\fakecaps 3565 \pdef\fakecaps#1{{\gmu@scapss#1\@nil}}
3567 \let\fakecapscore\gmu@scalematchX
Experimente z akcentami patrz no3.tex.
\tinycae 3570 \def\tinycae{{\tiny\AE}}% to use in \fakecaps[\tiny]{...}
3572 \RequirePackage{calc}
        wg \zf@calc@scale pakietu fontspec.
3576 \@ifpackageloaded{fontspec}{%
\gmu@scalar 3577     \def\gmu@scalar{1.0}%
\zf@scale 3578     \def\zf@scale{}}%
\gmu@scalematchX 3579     \def\gmu@scalematchX{%
3580     \begingroup
\gmu@scalar 3581     \ifx\zf@scale\empty\def\gmu@scalar{1.0}%
3582     \else\let\gmu@scalar\zf@scale\fi
3583     \setlength\@tempdima{\fontdimen5\font}% 5—ex height
3584     \setlength\@tempdimb{\fontdimen8\font}% 8—XTeX synthesized up-
        percase height.

```

```

3586 \divide\@tempdimb_\by1000\relax
3587 \divide\@tempdima_\by\@tempdimb
3588 \setlength{\@tempdima}{\@tempdima*\real{\gmu@scalar}}%
3589 \gmu@ifundefined{fakesc@extrascale}{}{}%
3590 \setlength{\@tempdima}{\@tempdima*\real{%
\
fakesc@extrascale}}}%
3591 \@tempcnta=\@tempdima
3592 \divide\@tempcnta_\by_1000\relax
3593 \@tempcntb=-1000\relax
3594 \multiply\@tempcntb_\by\@tempcnta
3595 \advance\@tempcntb_\by\@tempdima
3596 \xdef\gmu@scscale{\the\@tempcnta.%
\
ifnum\@tempcntb<100_\o\fi
\
ifnum\@tempcntb<10_\o\fi
\
the\@tempcntb}%
3599 \endgroup
3601 \addfontfeature{Scale=\gmu@scscale}%
3603 }}{\let\gmu@scalematchX\smallerr}

```

```

\fakecextrascala 3605 \def\fakecextrascala#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 or after.

```

\wyzejnizej 3611 \newcommand*\wyzejnizej[1]{%
3612 \edef\gmu@tempa{\gmu@ifundefined{r#1}{\arabic{page}}}%
3613 \@xa\@xa\@xa\@secondoftwo\csname_r#1\endcsname}}%
3614 \ifnum\gmu@tempa<\arabic{page}\relax_wy\zej\fi
3615 \ifnum\gmu@tempa>\arabic{page}\relax_ni\zej\fi
3616 \ifnum\gmu@tempa=\arabic{page}\relax_\@xa\ignorespaces\fi
3617 }

```

### 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 3627 \def\napapierkistretch{0,3pt}% It’s quite much for 11/13pt leading.
\napapierkicore 3629 \def\napapierkicore{\advance\baselineskip%
3630 by_\optplus\napapierkistretch\relax}
napapierki 3632 \newenvironment*{napapierki}{%
3633 \par\global\napapierkicore}{%
3634 \par\dimen\z@=\baselineskip
3635 \global\baselineskip=\dimen\z@}% so that you can use \endnapapierki in
interlacing environments.
\gmu@luzniej 3639 \newcount\gmu@luzniej
\luzniejcore 3641 \newcommand*\luzniejcore[1][1]{%
3642 \advance\gmu@luzniej_\@ne% We use this count to check whether we open the
environment or just set \looseness inside it again.
3644 \ifnum\gmu@luzniej=\@ne_\_ \multiply\tolerance_\by_\_2_\_ \fi
3645 \looseness=#1\relax}

```

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

luzniej 3649 `\newenvironment*{luzniej}{\par\luzniejcore}{\par}`

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

luzniej\* 3654 `\newenvironment*{luzniej*}[1][1]{%`  
 3655 `\multiply\tolerance_\by_\_2\relax`  
 3656 `\everypar{\looseness=#1\relax}}{\par}`

\nawj 3658 `\newcommand*\nawj{\kern,1em\relax}`% a kern to be put between parentheses and letters with descendants 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.

```

3666 \ifdefined\XeTeXversion
\pauza@skipcore 3667 \def\pauza@skipcore{\hskip.2em_\pluso.1em\relax
\pauzacore 3668 \pauzacore\hskip.2em_\pluso.1em\relax\ignorespaces}%
\ppauza@skipcore 3670 \def\ppauza@skipcore{\unskip\penalty10000\hskip.2em_\pluso.1em%
\relax
3671 -\hskip.2em_\pluso.1em\ignorespaces}
3673 \AtBeginDocument{% to be independent of moment of loading of polski.
\pauza 3674 \pdef\--{%
3675 \ifhmode
3676 \unskip\penalty10000
3677 \afterfi{%
3678 \@ifnextspace{\pauza@skipcore}%
3679 {\@ifnextMac\pauza@skipcore{%
3680 \pauzacore\penalty\hyphenpenalty\hskip\z@}}}%
3681 \else

```

According to *Instrukcja technologiczna. Skład ręczny i maszynowy* the dialogue dash should be followed by a rigid `hskip` of ½ em.

```

3685 \leavevmode\pauzacore\penalty10000\hskipo,5em\ignorespaces
3686 \fi}%

```

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

```

\pauza 3689 \pdef\pauza{%
3690 \ifhmode
3691 \unskip\penalty10000
3692 \hskip.2em_\pluso.1em\relax
3693 \pauzacore\hskip.2em_\pluso.1em\relax\ignorespaces%
3694 \else
3695 \pauzadial
3696 \fi}%

```

According to *Instrukcja technologiczna. Skład ręczny i maszynowy* the dialogue dash should be followed by a rigid `hskip` of ½ em.

```

\pauzadial 3701 \pdef\pauzadial{%
3702 \leavevmode\pauzacore\penalty10000\hskipo,5em\ignorespaces}

```

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

```

\lpauza 3706 \pdef\lpauza{%
3707 \pauzacore\hskip.2em_\pluso.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<sub>1</sub>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...`

```

\ 3715 \pdef\-%
3716 \ifvmode\PackageError{gmutils}{%
3717   command\backslashppauza(en dash) not intended for vmode.}{%
3718   Use\backslashppauza(en dash) only in number and numeral
      ranges.}%
3719 \else
3720 \afterfi{%
3721   \@ifnextspace{\ppauza@skipcore}{%
3722     \@ifnextMac\ppauza@skipcore{\unskip\discretionary{-}{-}{-}}%
      -}{-}}}%
3723   }%
3724 \fi
\ppauza 3725 }%
3727 \pdef\ppauza{%
3728 \ifvmode\PackageError{gmutils}{%
3729   command\backslashppauza(en dash) not intended for vmode.}{%
3730   Use\backslashppauza(en dash) only in number and numeral
      ranges.}%
3731 \else
3732 \unskip\discretionary{-}{-}{-}%
3733 \fi}%
\emdash 3735 \def\emdash{\char`-}
3736 }% of at begin document

\longpauza 3738 \def\longpauza{\def\pauzacore{-}}
\pauzacore 3739 \longpauza
\shortpauza 3740 \def\shortpauza{%
\pauzacore 3741 \def\pauzacore{-\kern,23em\relax\llap{-}}}
3742 \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 `-`.

```

3748 \ifdefined\XeTeXversion
3749 \foone{\catcode`-\active\catcode`-\active\catcode`-\active}{%
\adashes 3750 \def\adashes{\AtBeginDocument\adashes}% because \pauza is defined at
      begin document.
\adashes 3752 \AtBeginDocument{\def\adashes{%
3753   \catcode`-\active\let-\-%
3754   \catcode`-\active\let-\-%
3755   \addtomacro\dospecials{\do\-\do\-\}%
3756   \addtomacro\@sanitize{\@makeother\-\@makeother\-\}%
3757   \addtomacro\gmu@septify{\do\-\13\do\-\13\relax}%
3758 }}}}
3759 \else
3760 \relaxen\adashes
3762 \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 3771 \def\gmu@dywiz{\ifmmode-\else
3772   \ifvmode-\else\afterffifi\dywiz\fi\fi}%
3774 \foone{\catcode`-\active}{%
\ahyphen 3775   \def\ahyphen{\let-\gmu@dywiz\catcode`-\active}}
    To get current time. Works in  $\epsilon$ -TeXs, including XeTeX. \czas typesets 17.11 and
    \czas[:] typesets 17:11.
\czas 3780 \newcommand*\czas[1][.]{%
3781   \the\numexpr(\time-30)/60\relax#1%
3782   \@tempcnta=\numexpr\time-(\time-30)/60*60\relax
3783   \ifnum\@tempcnta<10\o\fi\the\@tempcnta}
3786 \@ifXeTeX{%
\textbullet 3787   \pdef\textbullet{%
3790     \iffontchar\font"2022\char"2022\else\ensuremath{\bullet}%
       \fi}%
\glyphname 3792   \pprovide\glyphname#1{%
3794     \XeTeXglyph\numexprXeTeXglyphindex"#1"\relax\relax}%since XeTeX
       ... \numexpr is redundant.
3796 }
\textbullet 3797 {\def\textbullet{\ensuremath{\bullet}}}
    tytulowa 3799 \newenvironment*{tytulowa}{\newpage}{\par\thispagestyle{empty}%
       \newpage}
    To typeset peoples' names on page 4 (the editorial page):
\nazwired 3802 \def\nazwired{\quad\textsc}

```

### 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 3816 \newcommand*\polskadata{%
\gmu@datef 3817   \def\gmu@datef##1-##2-##3##4,##5\gmu@datef{%
3818     \ifx\relax##2\relax##3##4%
3819     \else
3820       \ifnum##3\@firstofmany##4o\@nil=o\relax
3821       \else
3822         \ifnumo##3=o\relax
3823         \else##3%
3824         \fi##4%
3825       \fi
3826       \ifcase##2\relax\or\stycznia\or\lutego%
3827       \or\marca\or\kwietnia\or\maja\or\czerwca\or\lipca\or\
         sierpnia%
3828       \or\wrzeźnia\or\października\or\listopada\or\grudnia\else
3829       }%
3830       \fi
3831       \if\relax##1\relax\else\fi##1%

```



```

3832     \fi
3833     \gmu@datecomma{##5}}% of \gmu@datef.
\gmu@datefsl 3835 \def\gmu@datefsl##1/##2/##3##4,##5\gmu@datefsl{%
3836     \if\relax##2\relax##3##4%
3837     \else
3838     \ifnum##3\@firstofmany##4o\@nil=o\relax
3839     \else
3840     \ifnumo##3=o\relax
3841     \else##3%
3842     \fi##4%
3843     \fi
3844     \ifcase##2\relax\or\stycznia\or\lutego%
3845     \or\marca\or\kwietnia\or\maja\or\czerwca\or\lipca\or\
sierpnia%
3846     \or\września\or\października\or\listopada\or\grudnia\else
3847     }%
3848     \fi
3849     \if\relax##1\relax\else\fi##1%
3850     \fi
3851     \gmu@datecomma{##5}}%
3852 }% of \polskadata
3853 \polskadata
For documentation in English:
\englishdate 3860 \newcommand*\englishdate{%
\gmu@datef 3861 \def\gmu@datef##1-##2-##3##4,##5\gmu@datef{%
3862     \if\relax##2\relax##3##4%
3863     \else
3864     \ifcase##2\relax\or\January\or\February%
3865     \or\March\or\April\or\May\or\June\or\July\or\August%
3866     \or\September\or\October\or\November\or\December\else
3867     }%
3868     \fi
3869     \ifnum##3\@firstofmany##4o\@nil=o\relax
3870     \else
3871     \_%
3872     \ifnumo##3=o\relax
3873     \else##3%
3874     \fi##4%
3875     \ifcase##3\@firstofmany##4\relax\@nil\relax\or\st\or\nd%
\or\rd\else\th\fi
3876     \fi
3877     \ifx\relax##1\relax\else,\fi##1%
3878     \fi
\gmu@datefsl 3879 \gmu@datecomma{##5}}%
3881 \def\gmu@datefsl##1/##2/##3##4,##5\gmu@datefsl{%
3882     \if\relax##2\relax##3##4%
3883     \else
3884     \ifcase##2\relax\or\January\or\February%
3885     \or\March\or\April\or\May\or\June\or\July\or\August%
3886     \or\September\or\October\or\November\or\December\else
3887     }%

```

```

3888     \fi
3889     \ifnum##3\@firstofmany##4o\@nil=o\relax
3890     \else
3891         \_ %
3892         \ifnumo##3=o\relax
3893         \else##3%
3894         \fi##4%
3895         \ifcase##3\@firstofmany##4\relax\@nil\relax\or_st\or_nd%
           \or_rd\else_th\fi
3896     \fi
3897     \if\relax##1\relax\else,\_ \fi_##1%
3898 \fi
3899 \gmu@datecomma{##5}}%
3900 }
\gmu@datecomma 3903 \def\gmu@datecomma#1{% sometimes we want to typeset something like ‘11 wrześ-
           nia, czwartek’ so we add handling for comma in the \ldate’s argument.
3906     \ifx\gmu@datecomma#1\gmu@datecomma\else
3907         ,\gmu@stripcomma#1%
3908     \fi
3909 }% of \gmu@datecomma
\gmu@stripcomma 3911 \def\gmu@stripcomma#1,{#1}
\ifgmu@dash 3914 \newif\ifgmu@dash
\gmu@ifnodash 3916 \def\gmu@ifnodash#1-#2\@nil{%
\gmu@tempa 3917     \def\gmu@tempa{#2}%
3918     \ifx\gmu@tempa\@empty}
\gmu@testdash 3920 \pdef\gmu@testdash#1\ifgmu@dash{%
3921     \gmu@ifnodash#1-\@nil
3922     \gmu@dashfalse
3923     \else
3924     \gmu@dashtrue
3925     \fi
3926     \ifgmu@dash}

```

A word of explanation to the pair of macros above. `\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 3957 \newif\ifdate
\bidate 3959 \pdef\bidate#1{%
3960   \ifdate\gmu@testdash#1%
3961     \ifgmu@dash
3962       \gmu@datef#1,\gmu@datef
3963     \else
3964       \gmu@datefsl#1,\gmu@datefsl
3965     \fi\fi}

\linedate 3967 \pdef\linedate{\@ifstar\linedate@@\linedate@}
\linedate@@ 3968 \pdef\linedate@@#1{\linedate@{--}{#1}}
\linedate@ 3969 \pdef\linedate@#1{\par\ifdate\addvspace{\dateskip}%
3970   \date@line{\footnotesize\itshape\bidate{#1}}%
3971   \nopagebreak\else%%\ifnum\arabic{dateinsection}>o\dekbigskip\fi
3972   \addvspace{\bigskipamount}%
3973   \fi}% end of \linedate.

3975 \let\dateskip\medskipamount

\rdate 3983 \pdef\rdate{\let\date@line\rightline\linedate}
\ldate 3984 \pdef\ldate{%
\date@line 3986   \def\date@line##1{\par{\raggedright##1\par}}%
3987   \linedate}

\runindate 3988 \newcommand*\runindate[1]{%
3989   \paragraph{\footnotesize\itshape\gmu@datef#1\gmu@datef}%
3990   \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.

3993 \let\thedata\ldate

\zwrobcy 3996 \pdef\zwrobcy#1{\emph{#1}}\% ostinato, allegro con moto, garden party etc.,
także komplement

\tytul 3999 \pdef\tytul#1{\emph{#1}}

Maszynopis w świecie justowanym zrobi delikatną chorągiewkę. (The maszynopis
environment will make a delicate ragged right if called in a justified world.)

maszynopis 4005 \newenvironment{maszynopis}[1][\ttfamily
4006   \hyphenchar\font=45\relax% this assignment is global for the font.
4007   \@tempskipa=\glueexpr\rightskip+\leftskip\relax
4008   \ifdim\gluestretch\@tempskipa=\z@
4009   \tolerance900
it worked well with tolerance = 900.
4011   \advance\rightskip by\z@ pluso,5em\relax\fi
4012   \fontdimen3\font=\z@% we forbid stretching spaces...
%\fontdimen4\font=\z@ but allow shrinking them.
4014   \hyphenpenaltyo% not to make TEX nervous: in a typewriting this marvellous
algorithm of hyphenation should be turned off and every line broken at the
last allowable point.

4017   \StoreMacro\pauzacore
\pauzacore 4018   \def\pauzacore{-\rlap{\kern-o,3em-}-}%
4019   }\par}

\justified 4023 \newcommand*\justified{%

```

```

4024 \leftskip=1\leftskip% to preserve the natural length and discard stretch and
      shrink.
4026 \rightskip=1\rightskip
4027 \parfillskip=1\parfillskip
4028 \advance\parfillskip_\by_osp_plus_1fil\relax
4029 \let\\\@normalcr}

```

To conform Polish recommendation for typesetting saying that a paragraph's last line leaving less than `\parindent` should be stretched to fill the text width:

```

\fullpar 4034 \newcommand*\fullpar{%
4035   \hunskip
4036   \bgroup\parfillskip\z@skip\par\egroup}

```

To conform Polish recommendation for typesetting saying that the last line of a paragraph has to be `2\parindent` long at least. The idea is to set `\parfillskip` naturally rigid and long as `\textwidth-2\parindent`, but that causes non-negligible shrinking of the interword spaces so we provide a declaration to catch the proper glue where the `parindent` is set (e.g. in footnotes `parindent` is 0 pt)

```

\twoparinit 4045 \newcommand*\twoparinit{% the name stands for 'last paragraph line's length
              minimum two \parindent.
4047   \edef\twopar{%
4048     \hunskip% it's \protected, remember?
4049     \bgroup
4050     \parfillskip=\the\glueexpr
4051     \dimexpr\textwidth-2\parindent\relax
4052     minus\dimexpr\textwidth-2\parindent\relax
4053     \relax% to delimit \glueexpr.
4054     \relax% to delimit the assignment.
4055     \par\egroup
4056   }% of \gmu@twoparfill
4061 }% of \twoparinit.

```

For dati under poems.

```

\wherncore 4068 \newcommand\wherncore[1]{%
4069   \rightline{%
4070     \parbox{0,7666\textwidth}{
4071       \leftskiposp_plus_\textwidth
4072       \parfillskiposp\relax
4073       \let\\\linebreak
4074       \footnotesize_#1}}}

\whern 4076 \def\whern{%
4077   \@ifstar\wherncore{\vskip\whernskip\wherncore}}

\whernskip 4080 \newskip\whernskip
4081 \whernskip2\baselineskip_minus_2\baselineskip\relax

\whernup 4083 \newcommand\whernup[1]{\par\wherncore{#1}}

```

## A left-slanted font

Or rather a left Italic *and* left slanted font. In both cases we sample the skewness of the `itshape` font of the current family, we reverse it and apply to `\itshape` in `\litshape` and `\textlit` and to `\sl` in `\lsl`. Note a slight asymmetry: `\litshape` and `\textlit` take the current family while `\lsl` and `\textlsl` the basic Roman family and basic (serif)

Italic font. Therefore we introduce the `\lit` declaration for symmetry, that declaration `left-slants \it`.

I introduced them first while typesetting E. Szarzyński's *Letters* to follow his (elaborate) hand-writing and now I copy them here when need left Italic for his *Albert Camus' The Plague* to avoid using bold font.

Of course it's rather esoteric so I wrap all that in a declaration.

```

\leftslanting 4107 \def\leftslanting{%
\litshape     4108 \pdef\litshape{%
              4110 \itshape
              4111 \@tempdima=-2\fontdimen1\font
              4112 \advance\leftskip_\by\strip@pt\fontdimen1\font_ex_% to assure at least
                  the lowercase letters not to overshoot to the (left) margin. Note this has
                  any effect only if there is a \par in the scope.
              4116 \edef\gmu@tempa{%
              4117 \@nx\addfontfeature{FakeSlant=\strip@pt\@tempdima}}}% when
                  not \edefed, it caused an error, which is perfectly understandable.
              4120 \gmu@tempa}%
\textlit      4123 \pdef\textlit##1{%
              4124 {\litshape##1}}}%
\lit          4126 \pdef\lit{\rm\litshape}%
\lsl         4129 \pdef\lsl{\it
              4132 \@tempdima=-\fontdimen1\font
              4133 \xdef\gmu@tempa{%
              4134 \@nx\addfontfeature{RawFeature={slant=\strip@pt%
                  \@tempdima}}}}}%
              4135 \rm_\_% Note in this declaration we left-slant the basic Roman font not the it-
                  shape of the current family.
              4137 \gmu@tempa}%

```

Now we can redefine `\em` and `\emph` to use left Italic for nested emphasis. In Polish typesetting there is bold in nested emphasis as I have heard but we don't like bold since it perturbs homogeneous greyness of a page. So we introduce a three-cycle instead of two-: Italic, left Italic, upright.

```

\em          4145 \pdef\em{%
              4146 \ifdim\fontdimen1\font=\z@\_ \itshape
              4147 \else
              4148 \ifdim\fontdimen1\font>\z@\_ \litshape
              4149 \else_\upshape
              4150 \fi
              4151 \fi}%
              4154 \pdef\emph##1{%
              4155 {\em##1}}}%
              4156 }% of \leftslanting.

```

### Thousand separator

```

\thousep    4160 \pdef\thousep#1{% a macro that'll put the thousand separator between every two
                  three-digit groups.

```

First we check whether we have at least five digits.

```

4164 \gmu@thou@fiver#1\relax\relax\relax\relax\relax% we put
      five \relaxes after the parameter to ensure the string will
      meet \gmu@thou@fiver's definition.
4167 \gmu@thou@fiver{#1}{% if more than five digits:

```

```

4168     \emptify\gmu@thou@put
4169     \relaxen\gmu@thou@o\relaxen\gmu@thou@i\relaxen\gmu@thou@ii
4170     \@tempcnta\z@
4171     \gmu@thou@putter#1\gmu@thou@putter
4172     \gmu@thou@put
4173   }}

\gmu@thou@fiver 4175 \def\gmu@thou@fiver#1#2#3#4#5\gmu@thou@fiver#6#7{%
4176   \ifx\relax#5\relax\afterfi{#6}\else\afterfi{#7}\fi}

\gmu@thou@putter 4178 \def\gmu@thou@putter#1#2{% we are sure to have at least five tokens before the
guardian \gmu@thou@putter.
4180   \advance\@tempcnta\@ne
4181   \@tempcntb\@tempcnta
4182   \divide\@tempcntb3\relax
4183   \@tempcnta=\numexpr\@tempcnta-\@tempcntb*3
4184   \edef\gmu@thou@put{\gmu@thou@put#1%
4185     \ifx\gmu@thou@putter#2\else
4186       \ifcase\@tempcnta
4187         \gmu@thou@o\or\gmu@thou@i\or\gmu@thou@ii% all three cses are
yet \relax so we may put them in an \edef safely.
4190     \fi
4191     \fi}% of \edef
4192   \ifx\gmu@thou@putter#2% if we are at end of the digits...
4193     \edef\gmu@tempa{%
4194       \ifcase\@tempcnta
4195         \gmu@thou@o\or\gmu@thou@i\or\gmu@thou@ii
4196       \fi}%
4197     \@xa\let\gmu@tempa\gmu@thousep% ... we set the proper cs...
4198   \else% or ...
4199     \afterfi{% iterate.
4200       \gmu@thou@putter#2}% of \afterfi
4201   \fi% of if end of digits.
4202 }% of \gmu@thou@putter.

\gmu@thousep 4204 \def\gmu@thousep{\,}% in Polish the recommended thousand separator is a thin
space.

So you can type \thousep{7123123123123} to get 7 123 123 123 123. But what if
you want to apply \thousep to a count register or a \numexpr? You should write one
or two \expandafters and \the. Let's do it only once for all:

\xathousep 4212 \pdef\xathousep#1{\@xa\thousep\@xa{\the#1}}

Now write \xathousep{\numexpr10*9*8*7*6*120} to get 3 628 800.

\shortthousep 4216 \def\shortthousep{%
\thous 4217   \pdef\thous{%
4218     \ifmmode\hbox\bgroup\@gmu@mmhboxtrue\fi
4219     \afterassignment\thous@inner
4220     \@tempcnta=}%

\thous@inner 4222 \def\thous@inner{%
4223   \ifnum\@tempcnta<0\$_-$_%
4224     \@tempcnta=-\@tempcnta
4225   \fi
4226   \xathousep\@tempcnta
4227   \if@gmu@mmhbox\egroup

```

```

4228     \else\afterfi{\@ifnextcat_a\space{}}%
4229     \fi}%
4230 }% of \shortthousep.

```

And now write `\thous_3628800` to get 3 628 800 even with a blank space (beware of the range of TeX's counts).

### hyperref's `\nolinkurl` into `\url*`

```

\urladdstar 4238 \def\urladdstar{%
4239     \AtBeginDocument{%
4240         \@ifpackageloaded{hyperref}{%
4241             \StoreMacro\url
\url 4242         \def\url{\@ifstar{\nolinkurl}{\storedcsname{url}}}%
4243         }{}}
4245 \@onlypreamble\urladdstar
4248 \endinput

```

## Change History

- |   |   |
|---|---|
| <p>vo.74</p> <p><code>\@begnamedgroup@ifcs:</code><br/> The catcodes of <code>\begin</code> and <code>\end</code> argument(s) don't have to agree strictly anymore: an environment is properly closed if the <code>\begin</code>'s and <code>\end</code>'s arguments result in the same <code>\csname</code>, 1069</p> <p>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, 4248</p> <p>vo.75</p> <p><code>\@ifnextcat:</code><br/> <code>\let</code> for #1 changed to <code>\def</code> to allow things like <code>\noexpand~</code>, 862</p> <p><code>\@ifnextif:</code><br/> <code>\let</code> for #1 changed to <code>\def</code> to allow things like <code>\noexpand~</code>, 898</p> <p><code>\@ifnif:</code><br/> added, 923</p> <p>vo.76</p> <p>General:<br/> A 'fixing' of <code>\dots</code> was rolled back since it came out they were o.k. and that was the qx encoding that prints them very tight, 4248</p> | <p><code>\freeze@actives:</code><br/> added, 3266</p> <p>vo.77</p> <p>General:<br/> <code>\afterfi</code> &amp; pals made two-argument as the Marcin Woliński's analogoi are. At this occasion some redundant macros of that family are deleted, 4248</p> <p>vo.78</p> <p>General:<br/> <code>\@namelet</code> renamed to <code>\n@melet</code> to solve a conflict with the beamer class. The package contents regrouped, 4248</p> <p>vo.79</p> <p><code>\not@onlypreamble:</code><br/> All the actions are done in a group and therefore <code>\xdef</code> used instead of <code>\edef</code> because this command has to use <code>\do</code> (which is contained in the <code>\@preamblecmds</code> list) and <code>\not@onlypreamble</code> itself should be able to be let to <code>\do</code>, 1663</p> <p>vo.80</p> <p>General:<br/> <code>Checksum</code> 1689, 0</p> <p><code>\hfillneg:</code><br/> added, 3185</p> <p>vo.81</p> <p><code>\dekfracslash:</code><br/> moved here from <code>pmlectionis.cls</code>, 3445</p> <p><code>\ifSecondClass:</code><br/> moved here from <code>pmlectionis.cls</code>, 3417</p> |
|---|---|

- vo.82  
`\ikern:`  
 added, 3453
- vo.83  
`\~:`  
 postponed to `\begin{document}` to  
 avoid overwriting by a text command  
 and made sensible to a subsequent `/`,  
 3138
- vo.84  
 General:  
 CheckSum 2684, 0
- vo.85  
 General:  
 CheckSum 2795, 0  
 fixed behaviour of too clever headings  
 with `gmdoc` by adding an `\ifdim`  
 test, 4248
- vo.86  
`\texttilde:`  
 renamed from `\~` since the latter is one  
 of L<sup>A</sup>T<sub>E</sub>X accents, 3146
- vo.87  
 General:  
 CheckSum 4027, 0  
 the package goes  $\epsilon$ -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>L<sup>A</sup>T<sub>E</sub>X-condition, 4248
- vo.88  
 General:  
 CheckSum 4040, 0  
`\RestoreEnvironment:`  
 added, 1602  
`\storedcsname:`  
 added, 1593  
`\StoreEnvironment:`  
 added, 1598
- vo.89  
 General:  
 removed obsolete adjustment of `pgf` for  
 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, 4248
- vo.90  
 General:  
 CheckSum 4035, 0  
`\XeTeXthree:`  
 adjusted to the redefinition of `\verb` in  
`xlxtra 2008/07/29`, 2489
- vo.91  
 General:  
 CheckSum 4055, 0  
 removed `\jobnamewoe` since  
`\jobname` is always without  
 extension. `\xiispace` forked to  
`\visiblespace` `\let` to  
`\xxt@visiblepace` of `xltxtra` if  
 available. The documentation driver  
 integrated with the `.sty` file, 4248
- vo.92  
`\@checkend:`  
 shortened thanks to `\@ifenvir`, 1101  
`\@gif:`  
 added redefinition so that now  
 switches defined with it are  
`\protected` so they won't expand to  
 a further expanding or unbalanced  
`\iftrue/false` in an `\edef`, 281  
`\@ifenvir:`  
 added, 1093  
 General:  
 CheckSum 4133, 0
- vo.93  
`\@nameedef:`  
 added, 229  
 General:  
 A couple of  
`\DeclareRobustCommand*` changed  
 to `\pdef`, 4248  
 CheckSum 4140, 0  
 CheckSum 4501, 0  
 The numerical macros commented out  
 as obsolete and never really used, 4248  
`\ampulexdef:`  
 added, 755  
`\em:`  
 added, 4145, 4154  
`\gmu@RPfor:`  
 renamed from `\gmu@RPif` and `#3`  
 changed from a `csname` to `cs`, 3328  
`\litshape:`  
 copied here from E. Szarzyński's *The*  
*Letters*, 4108  
`\lsl:`  
 copied here from E. Szarzyński's *The*  
*Letters*, 4129  
`\nocite:`  
 a bug fixed: with `natbib` an 'extra `}`  
 error. Now it fixes only the standard  
 version of `\nocite`, 1701  
`\pdef:`  
 added, 187  
`\pprovide:`  
 added, 216  
`\provide:`  
 added, 201  
`\textlit:`  
 added, 4123  
`\thousep:`  
 added, 4160
- vo.94  
`\@xau:`  
 added, 183  
 General:



`\bgroup` and `\egroup` in the macro storing commands and in `\foone` changed to `\begingroup` and `\endgroup` since the former produce an empty `\mathord` in math mode while the latter don't, 4248  
 CheckSum 4880, 0  
 removed `\unex@namedef` and `\unex@nameuse`, probably never really used since they were incomplete: `\edef@other` undefined, 4248  
 The code from ancient `xparse` (1999) of `TeXLive` 2007 rewritten here, 4248  
`\afterfifi`:  
   `\if` removed from parameters' string, 1005  
`\ampulexdef`:  
   made `xparse`-ish and `\ampulexset` removed, 755  
`\dekfracc`:  
   made to work also in math mode, even with math-active digits, 2585  
`\gm@ifundefined`:  
   added. All `\@ifundefineds` used by me changed to this, 412  
   made robust to unbalanced `\ifs` and `\fis` the same way as `LATEX`'s `\@ifundefined` (after a heavy debug :-), 412  
`\gmathfurther`:  
   removed definition of `\langle letter \rangle s` and `\langle digit \rangle s`, 2955  
`\ldate`:  
   `\leftline` replaced with `\par...\par` to work well with `floatflt`, 3984  
`\prependtomacro`:  
   order of arguments reversed, 378  
`\resizegraphics`:  
   `\includegraphics` works well in `XYTeX` so I remove the complicated version with `\XeTeXpicfile`, 2614  
`\textbullet`:  
   the `XYTeX` version enriched with `\iffontchar` due to lack of bullets with the default settings reported by Morten Høgholm and Edd Barrett, 3787  
 vo.95  
   General:  
     CheckSum 4908, 0  
   `\gm@testdefined`:  
     added, 429  
   `\gm@testundefined`:  
     added, 444  
 vo.96  
   General:  
     CheckSum 5363, 0  
     put to CTAN on 2008/11/21, 0, 4248  
   `\glyphname`:  
     moved here from my private document class, 3792  
   `\gmathfurther`:  
     Greek letters completed. Wrapped with `\addtotoks` to allow using in any order with `\garamath` and others, 2955  
     the `\everymath`'s left brace moved here: earlier all the stuff was put into `\everymath`, 3021  
   `\gmathscripts`:  
     added, 3057  
   `\LuaTeX`:  
     added, 2440  
   `\pk`:  
     `\textup` removed to allow slanting the name in titles (that are usually typeset in Italic font), 1376  
 vo.97  
   General:  
     CheckSum 5375, 0  
     put to CTAN on 2008/11/22, 0  
   `\pdfLaTeX`:  
     added, 2417

## 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> , 951, <u>952</u> , 953, 955, 959,	<code>\@@nil</code> , 1346, 1348, 1350,	3838, 3869, 3875,
<u>960</u> , 961, 965, <u>3138</u> , <u>3321</u>	2649, 2651, 3193,	3889, 3895, 3916, 3921
<code>\+</code> , <u>3322</u>	3195, 3480, 3490,	<code>\@M</code> , 2006
<code>\-</code> , <u>1391</u> , 3775	3552, 3557, 3560,	<code>\@afterheading</code> , <u>1999</u>
<code>\&lt;...&gt;</code> , <u>1320</u>	3561, 3565, 3820,	<code>\@badend</code> , 1101

\@begnamedgroup, 1045,  
1062, 1067  
\@begnamedgroup@ifcs,  
1063, 1066  
\@car, 2350  
\@cclv, 3304  
\@cclvi, 3289  
\@checkend, 1101  
\@clsextension, 3417  
\@clubpenalty, 2011  
\@currenvir, 1048, 1095  
\@currenvline, 1052  
\@currsize, 1145, 1146,  
1147, 1148, 1149,  
1150, 1151, 1152, 1153,  
1154  
\@ddc, 500, 532, 548, 567,  
595, 600, 601  
\@ddc@, 502, 524  
\@ddc@C, 620  
\@ddc@O, 604  
\@ddc@c, 609  
\@ddc@c@, 611, 617, 622  
\@ddc@m, 556, 632  
\@ddc@o, 559  
\@ddc@o@, 561, 564, 606  
\@ddc@s, 551  
\@ddc@x, 661  
\@ddc@xm, 634  
\@ddc@xmm, 637  
\@ddc@xmmm, 640  
\@ddc@xmmmm, 643  
\@ddc@xmmmmm, 646  
\@ddc@xmmmmmm, 649  
\@ddc@xmmmmmmm, 652  
\@ddc@xmmmmmmmm, 655  
\@ddc@xmmmmmmmmm, 658  
\@emptify, 388, 389, 393  
\@enumctr, 2262, 2263, 2269  
\@enumdepth, 2258, 2261,  
2262  
\@fileswithoptions, 3417  
\@firstofmany, 3193,  
3820, 3838, 3869,  
3875, 3889, 3895  
\@firstofone, 526, 686  
\@gif, 267, 268, 275  
\@gmu@mmhboxtrue, 2587,  
4218  
\@if, 290, 291, 294  
\@ifXeTeX, 2483, 2493,  
2621, 3442, 3786  
\@ifempty, 779, 1941, 1961,  
3464  
\@ifenvir, 1093, 1101  
\@ifl@aded, 1687  
\@ifncat, 869, 872, 887  
\@ifnextMac, 975, 3679, 3722  
\@ifnextac, 934  
\@ifnextcat, 862, 935, 4228  
\@ifnextif, 898  
\@ifnextspace, 958, 3678,  
3721  
\@ifnif, 905, 908, 923  
\@ifnotmw, 1820, 1845,  
1994, 2099, 2186, 2241  
\@include, 3487, 3489  
\@itemdepth, 2276, 2279,  
2280  
\@itemitem, 2280, 2281  
\@minus, 2109, 2114, 2120,  
2122, 2127, 2129,  
2136, 2141  
\@nameedef, 229  
\@nobreakfalse, 2005, 3242  
\@nobreaktrue, 2000  
\@normalcr, 4029  
\@nx, 181  
\@oarg, 1426, 1428, 1429  
\@oargsq, 1426, 1429, 1444  
\@onlypreamble, 1728,  
2948, 3244, 4245  
\@parg, 1433, 1435, 1436  
\@pargp, 1433, 1436, 1445  
\@parindent, 2265, 2266,  
2268, 2283, 2284, 2286  
\@parsed@endenv, 514,  
518, 668  
\@parsed@endenv@, 519,  
521, 669  
\@pkgextension, 1688  
\@preamblecmds, 1682,  
1684, 1697, 1701  
\@relaxen, 397, 398, 402  
\@secondofmany, 3195  
\@secondofthree, 680, 692  
\@starttoc, 3237  
\@tempcntb\*, 4183  
\@tempdima, 3583, 3587,  
3588, 3590, 3591,  
3595, 4111, 4117,  
4132, 4134  
\@tempdima\*, 3588, 3590  
\@tempdimb, 3584, 3586, 3587  
\@temptokena, 498, 508,  
516, 545, 546, 588, 589  
\@temptokenb, 484, 485,  
499, 536, 540, 542,  
572, 576, 578, 629  
\@textsuperscript,  
2628, 2629  
\@thirdofthree, 684, 693  
\@toodeep, 2259, 2277  
\@topnewpage, 1901  
\@typeset@protect, 525  
\@undefined, 163  
\@whilenum, 3289, 3304  
\@xa, 180, 183  
\@xau, 183  
\@xifncat, 874, 887  
\@xifnif, 910, 923  
\acro, 3356, 3391, 3392, 3396  
\acrocore, 3376, 3386  
\adashes, 3750, 3750,  
3752, 3761  
\addfontfeature, 2517,  
2548, 2550, 2576,  
2623, 3087, 3100,  
3394, 3602, 4117, 4134  
\addto@macro, 368, 375  
\addtoheading, 1975  
\addtomacro, 375, 3070,  
3756, 3757, 3758  
\addtotoks, 384, 3021,  
3058, 3062, 3086  
\AE, 3570  
\afterassignment, 4219  
\afterfi, 206, 425, 938,  
962, 963, 1002, 1067,  
1068, 1097, 1098,  
1338, 1369, 1442,  
1485, 1562, 2486,  
2656, 3365, 3381,  
3466, 3467, 3529,  
3557, 3677, 3720,  
4176, 4199, 4228  
\afterfifi, 203, 204, 420,  
422, 1005, 1541, 1546,  
2485, 3772  
\afterfififi, 1013  
\afteriffifi, 1007  
\afteriffififi, 1012  
\afteriffiffififi, 1011  
\ahyphen, 3775  
\AKA, 3392  
\all@other, 776, 777, 786,  
787, 788, 2459, 2565,  
2567  
\alpha, 2909

<code>\ampulexdef</code> , <a href="#">755</a> , <a href="#">797</a> , 812, <a href="#">1725</a> , <a href="#">3047</a>	<a href="#">3294</a> , <a href="#">3303</a> , <a href="#">3304</a> , <a href="#">3305</a> , <a href="#">3306</a>	<code>\discret</code> , <a href="#">1333</a> , <a href="#">1338</a>
<code>\ampulexhash</code> , <a href="#">816</a>	<code>\cs</code> , <a href="#">1390</a> , <a href="#">1396</a> , <a href="#">1400</a> , <a href="#">1415</a>	<code>\divide</code> , <a href="#">2340</a> , <a href="#">2343</a> , <a href="#">2344</a> , <a href="#">3586</a> , <a href="#">3587</a> , <a href="#">3592</a> , <a href="#">4182</a>
<code>\AmSTeX</code> , <a href="#">2387</a>	<code>\cup</code> , <a href="#">3099</a>	<code>\document</code> , <a href="#">1728</a>
<code>\approx</code> , <a href="#">2857</a>	<code>\czas</code> , <a href="#">3780</a>	<code>\downarrow</code> , <a href="#">2880</a>
<code>\arg</code> , <a href="#">1441</a> , <a href="#">1442</a>	<code>\czer</code> , <a href="#">3318</a> , <a href="#">3321</a> , <a href="#">3322</a>	<code>\dp</code> , <a href="#">3006</a> , <a href="#">3009</a> , <a href="#">3091</a>
<code>\AtBeginDocument</code> , <a href="#">1258</a> , <a href="#">1440</a> , <a href="#">1695</a> , <a href="#">1731</a> , <a href="#">1791</a> , <a href="#">2504</a> , <a href="#">2514</a> , <a href="#">2944</a> , <a href="#">3144</a> , <a href="#">3339</a> , <a href="#">3673</a> , <a href="#">3750</a> , <a href="#">3752</a> , <a href="#">4239</a>	<code>\czerwo</code> , <a href="#">3317</a> , <a href="#">3318</a>	<code>\dywiz</code> , <a href="#">3772</a>
<code>\begin</code> , <a href="#">1062</a>	<code>\dag</code> , <a href="#">3322</a>	<code>\eequals</code> , <a href="#">3264</a>
<code>\begin*</code> , <a href="#">1062</a>	<code>\date@line</code> , <a href="#">3970</a> , <a href="#">3983</a> , <a href="#">3986</a>	<code>\egfirstofone</code> , <a href="#">246</a> , <a href="#">248</a>
<code>\beta</code> , <a href="#">2910</a>	<code>\dateskip</code> , <a href="#">3969</a> , <a href="#">3975</a>	<code>\egRestore@Macro</code> , <a href="#">1555</a> , <a href="#">1557</a>
<code>\bgcolor</code> , <a href="#">3091</a>	<code>\deadcycles</code> , <a href="#">3521</a>	<code>\egRestore@MacroSt</code> , <a href="#">1555</a> , <a href="#">1558</a>
<code>\BibTeX</code> , <a href="#">2390</a>	<code>\DeclareDocumentCommand</code> , <a href="#">482</a> , <a href="#">490</a> , <a href="#">511</a> , <a href="#">664</a> , <a href="#">755</a> , <a href="#">2489</a> , <a href="#">2703</a> , <a href="#">2738</a> , <a href="#">2758</a>	<code>\egStore@Macro</code> , <a href="#">1475</a> , <a href="#">1480</a>
<code>\bidate</code> , <a href="#">3959</a> , <a href="#">3970</a>	<code>\DeclareDocumentEnvironment</code> , <a href="#">510</a> , <a href="#">663</a>	<code>\egStore@MacroSt</code> , <a href="#">1475</a> , <a href="#">1481</a>
<code>\bigcircle</code> , <a href="#">2892</a>	<code>\DeclareFontFamily</code> , <a href="#">2686</a> , <a href="#">2695</a>	<code>\em</code> , <a href="#">4145</a> , <a href="#">4155</a>
<code>\Biggl</code> , <a href="#">3017</a>	<code>\DeclareFontShape</code> , <a href="#">2688</a> , <a href="#">2697</a>	<code>\emdash</code> , <a href="#">3735</a>
<code>\biggl</code> , <a href="#">3015</a>	<code>\DeclareLogo</code> , <a href="#">2303</a> , <a href="#">2323</a> , <a href="#">2349</a> , <a href="#">2360</a> , <a href="#">2390</a> , <a href="#">2396</a> , <a href="#">2399</a> , <a href="#">2401</a> , <a href="#">2404</a> , <a href="#">2407</a> , <a href="#">2414</a> , <a href="#">2416</a> , <a href="#">2417</a> , <a href="#">2421</a> , <a href="#">2428</a> , <a href="#">2440</a>	<code>\emptify</code> , <a href="#">389</a> , <a href="#">389</a> , <a href="#">2519</a> , <a href="#">2661</a> , <a href="#">3065</a> , <a href="#">3069</a> , <a href="#">3330</a> , <a href="#">4168</a>
<code>\Biggr</code> , <a href="#">3018</a>	<code>\DeclareRobustCommand*</code> , <a href="#">1176</a> , <a href="#">1177</a> , <a href="#">1178</a> , <a href="#">1179</a> , <a href="#">1390</a>	<code>\encodingdefault</code> , <a href="#">2685</a> , <a href="#">2686</a> , <a href="#">2688</a> , <a href="#">2694</a> , <a href="#">2695</a> , <a href="#">2697</a>
<code>\biggr</code> , <a href="#">3016</a>	<code>\DeclareSymbolFont</code> , <a href="#">2685</a> , <a href="#">2694</a>	<code>\endlist</code> , <a href="#">2272</a> , <a href="#">2289</a>
<code>\Bigl</code> , <a href="#">3013</a>	<code>\DeclareTextCommand</code> , <a href="#">2316</a>	<code>\englishdate</code> , <a href="#">3860</a>
<code>\bigl</code> , <a href="#">3011</a>	<code>\DeclareTextCommandDefault</code> , <a href="#">2318</a>	<code>\enoughpage</code> , <a href="#">3249</a>
<code>\Bigr</code> , <a href="#">3014</a>	<code>\defobeylines</code> , <a href="#">3170</a>	<code>\ensuremath</code> , <a href="#">1292</a> , <a href="#">1305</a> , <a href="#">2409</a> , <a href="#">2630</a> , <a href="#">2801</a> , <a href="#">3351</a> , <a href="#">3790</a> , <a href="#">3797</a>
<code>\bigr</code> , <a href="#">3012</a>	<code>\dekbigskip</code> , <a href="#">3182</a>	<code>enumerate*</code> , <a href="#">2257</a>
<code>\bigskipamount</code> , <a href="#">3182</a> , <a href="#">3972</a>	<code>\dekfrac</code> , <a href="#">2585</a>	<code>\env</code> , <a href="#">1396</a>
<code>\boldmath</code> , <a href="#">2350</a>	<code>\dekfrac@args</code> , <a href="#">2563</a> , <a href="#">2574</a> , <a href="#">2588</a> , <a href="#">3440</a>	<code>\epsilon</code> , <a href="#">2913</a>
<code>\BooleanFalse</code> , <a href="#">554</a> , <a href="#">702</a>	<code>\dekfracssimple</code> , <a href="#">3439</a> , <a href="#">3445</a>	<code>\equals</code> , <a href="#">3262</a>
<code>\BooleanTrue</code> , <a href="#">553</a> , <a href="#">703</a>	<code>\dekfracslash</code> , <a href="#">3432</a> , <a href="#">3442</a> , <a href="#">3443</a>	<code>\eta</code> , <a href="#">2916</a>
<code>\box</code> , <a href="#">2371</a> , <a href="#">3010</a>	<code>\dekmedskip</code> , <a href="#">3181</a>	<code>\eTeX</code> , <a href="#">2407</a> , <a href="#">2412</a> , <a href="#">2414</a> , <a href="#">2508</a>
<code>\bslash</code> , <a href="#">1234</a> , <a href="#">1390</a> , <a href="#">1493</a> , <a href="#">1570</a> , <a href="#">1594</a> , <a href="#">1968</a> , <a href="#">1969</a> , <a href="#">1970</a> , <a href="#">3717</a> , <a href="#">3718</a> , <a href="#">3729</a> , <a href="#">3730</a>	<code>\deksmallskip</code> , <a href="#">3179</a>	<code>\everydisplay</code> , <a href="#">3045</a> , <a href="#">3059</a> , <a href="#">3063</a> , <a href="#">3108</a>
<code>\bullet</code> , <a href="#">3790</a> , <a href="#">3797</a>	<code>\Delta</code> , <a href="#">2897</a>	<code>\everymath</code> , <a href="#">3021</a> , <a href="#">3045</a> , <a href="#">3048</a> , <a href="#">3058</a> , <a href="#">3059</a> , <a href="#">3062</a> , <a href="#">3063</a> , <a href="#">3086</a> , <a href="#">3108</a>
<code>\c@gm@PronounGender</code> , <a href="#">1758</a>	<code>\delta</code> , <a href="#">2912</a>	<code>\everypar</code> , <a href="#">2002</a> , <a href="#">2012</a> , <a href="#">3656</a>
<code>\c@NoNumSecs</code> , <a href="#">1793</a>	<code>\dimen</code> , <a href="#">3251</a> , <a href="#">3252</a> , <a href="#">3253</a> , <a href="#">3634</a> , <a href="#">3635</a>	<code>\exists</code> , <a href="#">2982</a>
<code>\c@secnumdepth</code> , <a href="#">1850</a>	<code>\dimexpr</code> , <a href="#">2967</a> , <a href="#">2972</a> , <a href="#">2980</a> , <a href="#">3009</a> , <a href="#">3091</a> , <a href="#">4051</a> , <a href="#">4052</a>	
<code>\ClassError</code> , <a href="#">1967</a>	<code>\discre</code> , <a href="#">1331</a> , <a href="#">1340</a> , <a href="#">1357</a>	
<code>\cleardoublepage</code> , <a href="#">1807</a>		
<code>\clubpenalty</code> , <a href="#">2006</a> , <a href="#">2011</a>		
<code>\cmd</code> , <a href="#">1415</a>		
<code>\cmd@to@cs</code> , <a href="#">1415</a> , <a href="#">1417</a>		
<code>\color</code> , <a href="#">3316</a> , <a href="#">3317</a>		
<code>\continuum</code> , <a href="#">3350</a>		
<code>\copy</code> , <a href="#">2333</a> , <a href="#">2363</a> , <a href="#">2377</a> , <a href="#">3090</a> , <a href="#">3102</a>		
<code>\count</code> , <a href="#">2338</a> , <a href="#">2339</a> , <a href="#">2340</a> , <a href="#">2341</a> , <a href="#">2342</a> , <a href="#">2343</a> , <a href="#">2344</a> , <a href="#">2345</a> , <a href="#">3287</a> , <a href="#">3289</a> , <a href="#">3290</a> , <a href="#">3291</a> ,		

`\foeatletter`, [250](#)  
`\foone`, [246](#), [250](#), [974](#), [1191](#),  
[1194](#), [1202](#), [1218](#),  
[1238](#), [1242](#), [1245](#),  
[1248](#), [1399](#), [3161](#),  
[3169](#), [3749](#), [3774](#)  
`\forall`, [2976](#)  
`\FormatHangHeading`,  
[2108](#), [2115](#), [2123](#), [2130](#)  
`\FormatRunInHeading`,  
[2137](#), [2142](#)  
`\freezeactives`, [3286](#)  
`\fullpar`, [4034](#)  
  
`\g@empty`, [393](#), [394](#)  
`\g@relaxen`, [402](#), [403](#)  
`\gaddtomacro`, [363](#)  
`\Gamma`, [2896](#)  
`\gamma`, [2911](#)  
`\garamath`, *p.* [39](#), [3085](#)  
`\ge`, [2834](#)  
`\gempty`, [394](#), [394](#)  
`\geq`, [2833](#), [2834](#)  
`\glet`, [355](#), [2026](#)  
`\glueexpr`, [4007](#), [4050](#)  
`\gluestretch`, [4008](#)  
`\glyphname`, [3792](#)  
`\gm@atppron`, [1760](#), [1764](#),  
[1765](#), [1766](#), [1767](#),  
[1769](#), [1770](#), [1771](#), [1772](#)  
`\gm@clearpagesduetoopenright`,  
[1806](#), [1869](#)  
`\gm@dontnumbersectionsoutofmainmatter`,  
[1804](#), [1853](#)  
`\gm@duppa`, [2564](#), [2565](#), [2567](#)  
`\gm@gobmacro`, [2459](#), [2465](#)  
`\gm@hyperrefstepcounter`,  
[1796](#), [1799](#), [1882](#)  
`\gm@ifnac`, [935](#), [937](#)  
`\gm@ifundefined`, [412](#),  
[1820](#), [1853](#), [1869](#),  
[1945](#), [1966](#), [2021](#),  
[2034](#), [2104](#), [2231](#),  
[2386](#), [2420](#), [2422](#),  
[2427](#), [2429](#), [2565](#),  
[3589](#), [3612](#)  
`\gm@letspace`, [954](#), [962](#)  
`\gm@notprerr`, [1693](#), [1700](#)  
`gm@PronounGender`, [1758](#)  
`\gm@pswords`, [1346](#), [1348](#), [1350](#)  
`\gm@sec`, [2224](#), [2235](#), [2236](#)  
`\gm@secini`, [2189](#), [2208](#),  
[2214](#), [2220](#), [2233](#)  
`\gm@secmarkh`, [2216](#)  
  
`\gm@secstar`, [2192](#), [2202](#),  
[2209](#), [2221](#), [2235](#), [2236](#)  
`\gm@secx`, [2224](#), [2225](#)  
`\gm@secxx`, [2191](#), [2219](#), [2226](#)  
`\gm@straightensc`,  
[2230](#), [2239](#)  
`\gm@targetheading`,  
[1797](#), [1800](#)  
`\gm@testdefined`, [429](#), [447](#)  
`\gm@testundefined`, [444](#),  
[482](#)  
`\gma`, [3078](#)  
`\gma@arrowdash`, [3089](#),  
[3098](#), [3104](#), [3105](#)  
`\gma@bare`, [3074](#), [3076](#)  
`\gma@bracket`, [3076](#), [3077](#)  
`\gma@checkbracket`,  
[3075](#), [3079](#)  
`\gma@dollar`, [3073](#), [3074](#),  
[3079](#)  
`\gma@gmathhook`, [3043](#),  
[3069](#), [3070](#), [3093](#)  
`\gma@quantifierhook`,  
[2976](#), [2982](#), [3065](#),  
[3067](#), [3100](#), [3106](#)  
`\gma@tempa`, [2965](#), [2967](#),  
[2970](#), [2972](#), [3005](#),  
[3009](#), [3035](#), [3038](#)  
`\gma@tempb`, [3006](#), [3009](#)  
`\gmath`, *p.* [39](#), [3055](#), [3073](#), [3077](#)  
`\gmath@delc`, [2758](#), [2789](#)  
`\gmath@delc`, [2758](#), [2789](#)  
`\gmath@delif`, [2783](#)  
`\gmath@delimif`, [2792](#)  
`\gmath@do`, [2703](#), [2748](#),  
[2750](#), [2752](#), [2811](#),  
[2813](#), [2814](#), [2815](#),  
[2816](#), [2817](#), [2818](#),  
[2819](#), [2820](#), [2821](#),  
[2822](#), [2823](#), [2837](#),  
[2838](#), [2839](#), [2840](#),  
[2841](#), [2844](#), [2846](#),  
[2848](#), [2852](#), [2861](#),  
[2862](#), [2864](#), [2888](#)  
`\gmath@doif`, [2738](#), [2812](#),  
[2825](#), [2826](#), [2851](#),  
[2853](#), [2854](#), [2855](#),  
[2856](#), [2857](#), [2858](#),  
[2860](#), [2866](#), [2867](#),  
[2868](#), [2869](#), [2870](#),  
[2871](#), [2872](#), [2873](#),  
[2876](#), [2880](#), [2882](#),  
[2889](#), [2892](#), [2893](#),  
[2894](#), [2896](#), [2897](#),  
[2898](#), [2899](#), [2900](#),  
[2901](#), [2902](#), [2903](#),  
[2904](#), [2905](#), [2909](#),  
[2910](#), [2911](#), [2912](#),  
[2913](#), [2914](#), [2915](#),  
[2916](#), [2917](#), [2918](#),  
[2919](#), [2920](#), [2921](#),  
[2922](#), [2923](#), [2924](#),  
[2925](#), [2926](#), [2927](#),  
[2928](#), [2929](#), [2930](#),  
[2931](#), [2932](#), [2933](#), [2934](#)  
`\gmath@fam`, [2675](#), [2808](#), [2907](#)  
`\gmath@famnum`, [2675](#),  
[2714](#), [2729](#), [2767](#),  
[2774](#), [2803](#)  
`\gmath@font`, [2701](#), [2748](#),  
[2750](#), [2752](#), [2789](#),  
[2799](#), [2886](#), [2887](#),  
[2937](#), [2938](#), [2975](#),  
[2988](#), [2991](#), [2995](#),  
[2996](#), [2997](#)  
`\gmath@getfamnum`, [2674](#),  
[2710](#), [2764](#), [2800](#)  
`\gmathbase`, [2681](#), [2944](#),  
[2945](#), [2948](#), [3055](#)  
`\gmathcats`, [3061](#)  
`\gmathfurther`, [2955](#), [3055](#)  
`\gmathhook`, [3070](#)  
`\gmathscripts`, [3057](#)  
`\gml@StoreCS`, [1520](#), [1543](#),  
[1580](#)  
`\gml@storemacros`, [1521](#),  
[1532](#), [1541](#), [1546](#), [1583](#)  
`\gmobeyspaces`, [3162](#)  
`\gmshowlists`, [824](#)  
`\GMtextsuperscript`, [2620](#)  
`\gmu@acroinner`, [3362](#),  
[3372](#), [3373](#), [3381](#)  
`\gmu@acrospace`, [3356](#),  
[3361](#), [3361](#), [3365](#)  
`\gmu@checkaftersec`,  
[2020](#), [2079](#)  
`\gmu@dashfalse`, [3922](#)  
`\gmu@dashtrue`, [3924](#)  
`\gmu@datecomma`, [3833](#),  
[3851](#), [3879](#), [3899](#),  
[3903](#), [3906](#)  
`\gmu@datef`, [3817](#), [3817](#),  
[3861](#), [3861](#), [3962](#), [3989](#)  
`\gmu@datefsl`, [3835](#), [3835](#),  
[3881](#), [3881](#), [3964](#)  
`\gmu@dekfracc`, [2547](#),  
[2566](#), [2570](#)  
`\gmu@dekfraccsimple`,  
[2570](#), [3429](#), [3440](#)

\gmu@denominatorkern, 2549, 2592, 3432  
\gmu@discretionaryslash, 1357, 1368  
\gmu@dogmathbase, 2806, 2944  
\gmu@dywiz, 3771, 3775  
\gmu@fileext, 3482, 3492, 3510  
\gmu@filename, 3481, 3495, 3507, 3510, 3513, 3522  
\gmu@fontscale, 2655, 2661, 2689, 2698  
\gmu@fontstring, 2645, 2689, 2698, 2701  
\gmu@fontstring@, 2646, 2648, 2810  
\gmu@forallkerning, 2977, 2979  
\gmu@getaddvs, 2071, 2071, 2077  
\gmu@getext, 3480, 3490  
\gmu@getfontdata, 2660, 2682, 2692  
\gmu@getfontscale, 2653, 2654, 2656, 2664, 2666  
\gmu@getfontstring, 2644, 2667  
\gmu@gobdef, 204, 210  
\gmu@ifnodash, 3916, 3921  
\gmu@luzniej, 3639, 3642, 3644  
\gmu@nl@reserveda, 1652, 1655, 1660, 1663  
\gmu@nocite@ampulex, 1718, 1731  
\gmu@numeratorkern, 2548, 2591, 2592, 3431  
\gmu@prevsec, 2001, 2003, 2025, 2032, 2062  
\gmu@printslashes, 1362, 1364, 1364, 1366, 1369  
\gmu@quotedstring, 2649, 2651  
\gmu@resa, 3330, 3331, 3333  
\gmu@reserveda, 951, 953, 959, 961, 1095, 1097, 1533, 1536, 1539, 1977, 2026, 2027, 2030, 2311, 2313, 2315, 2316, 2317, 3465, 3466  
\gmu@reservedb, 1096, 1097  
\gmu@restorespecials, 455  
\gmu@RPfor, 3316, 3328, 3340, 3351  
\gmu@scalar, 3577, 3581, 3582, 3588  
\gmu@scalematchX, 3567, 3579, 3603  
\gmu@scapLetters, 3539, 3549, 3554  
\gmu@scapSpaces, 3552, 3557, 3561  
\gmu@scapss, 3560, 3565  
\gmu@scscale, 3596, 3602  
\gmu@septify, 457, 3062, 3758  
\gmu@setheading, 2076, 2082, 2083  
\gmu@setsetSMglobal, 1519, 1524, 1579  
\gmu@setSMglobal, 1526, 1528, 1546  
\gmu@SMdo@scope, 1622, 1624, 1627, 1628, 1642  
\gmu@SMdo@setscope, 1620, 1626, 1640  
\gmu@SMglobalfalse, 1487, 1501, 1528, 1537, 1564, 1573, 1630  
\gmu@SMglobaltrue, 1463, 1526  
\gmu@smtempa, 1491, 1500, 1567, 1572  
\gmu@storespecials, 452  
\gmu@stripchar, 2672, 2675, 2940  
\gmu@stripcomma, 3907, 3911  
\gmu@tempa, 211, 770, 776, 787, 795, 2464, 2466, 2684, 2690, 2693, 2699, 2700, 2712, 2718, 2719, 2721, 2722, 2726, 2733, 2735, 2766, 2768, 2769, 2772, 2778, 2780, 3117, 3120, 3122, 3612, 3614, 3615, 3616, 3917, 3918, 4116, 4120, 4133, 4137, 4193, 4197  
\gmu@tempb, 771, 777, 788, 796  
\gmu@tempc, 772, 802  
\gmu@tempd, 774, 775, 778, 781, 786, 788, 793, 794, 806, 812  
\gmu@tempe, 785, 791, 810, 813  
\gmu@tempf, 809, 812  
\gmu@testdash, 3920, 3960  
\gmu@theskewchar, 2668, 2687, 2696  
\gmu@thou@fiver, 4164, 4167, 4175, 4175  
\gmu@thou@i, 4169, 4187, 4195  
\gmu@thou@ii, 4169, 4187, 4195  
\gmu@thou@o, 4169, 4187, 4195  
\gmu@thou@put, 4168, 4172, 4184  
\gmu@thou@putter, 4171, 4178, 4185, 4192, 4200  
\gmu@thousep, 4197, 4204  
\gmu@tilde, 3133, 3138, 3149  
\gmu@whonly, 3528, 3529  
\gmu@xedekfracplain, 2524, 2573  
\gmu@xedekfracstar, 2524, 2539  
\gmu@xefraccdef, 2540, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562  
\gn@melet, 1659  
\gobble, 194, 196, 3036  
\gobbletwo, 197  
\grab@default, 580, 583, 596  
\grab@ms, 626  
\grefstepcounter, 322, 338  
\grelaxen, 403, 403, 2003  
\hathat, 1400  
\HeadingNumber, 1879, 1881  
\HeadingNumberedfalse, 1805, 1850  
\HeadingRHeadText, 1863  
\HeadingText, 1865  
\HeadingTOCText, 1864  
\HeShe, 1769  
\heshe, 1764  
\hfillneg, 3185  
\hgrefstepcounter, 337  
\hidden@iffalse, 300, 779  
\hidden@iftrue, 301, 779  
\HimHer, 1771  
\himher, 1766  
\HisHer, 1770

`\hisher`, [1765](#)  
`\HisHers`, [1772](#)  
`\hishers`, [1767](#)  
`\hrule`, [2050](#)  
`\hunskip`, [345](#), [3262](#), [3264](#),  
[3266](#), [4035](#), [4048](#)  
`\hyphenpenalty`, [1350](#),  
[3680](#), [4014](#)  
  
`\if@afterindent`, [2007](#)  
`\if@filesw`, [3239](#), [3494](#),  
[3506](#), [3514](#)  
`\if@gnu@mmhbox`, [2568](#),  
[2578](#), [2582](#), [3436](#), [4227](#)  
`\if@mainmatter`, [1805](#)  
`\if@nobreak`, [2004](#)  
`\if@openright`, [1807](#)  
`\If@SomethingTF`, [670](#),  
[672](#), [675](#), [677](#), [689](#)  
`\if@specialpage`, [1868](#)  
`\if@twoside`, [1894](#)  
`\IfBooleanF`, [716](#)  
`\IfBooleanT`, [712](#)  
`\IfBooleanTF`, [704](#), [713](#), [717](#)  
`\ifcsname`, [418](#), [1067](#)  
`\ifdate`, [3957](#), [3960](#), [3969](#)  
`\ifdefined`, [172](#), [202](#), [435](#),  
[484](#), [1197](#), [1259](#),  
[2484](#), [2828](#), [2832](#),  
[3046](#), [3341](#), [3666](#), [3748](#)  
`\iffontchar`, [2408](#), [2541](#),  
[2748](#), [2750](#), [2752](#),  
[2789](#), [2799](#), [2886](#),  
[2887](#), [2937](#), [2975](#),  
[2988](#), [2991](#), [2995](#),  
[2996](#), [2997](#), [3790](#)  
`\ifgnu@dash`, [3914](#), [3920](#),  
[3926](#), [3961](#)  
`\ifgnu@postsec`, [2022](#),  
[2061](#), [2069](#)  
`\ifgnu@SMglobal`, [1461](#),  
[1485](#), [1492](#), [1525](#),  
[1562](#), [1568](#), [1627](#)  
`\ifHeadingNumbered`,  
[1849](#), [1877](#)  
`\IfNoValueF`, [698](#), [700](#)  
`\IfNoValueT`, [697](#), [701](#)  
`\IfNoValueTF`, [696](#), [699](#)  
`\ifodd`, [1762](#)  
`\ifSecondClass`, [3415](#)  
`\IfSomethingF`, [674](#), [698](#)  
`\IfSomethingT`, [671](#), [697](#)  
`\IfSomethingTF`, [670](#), [696](#)  
`\IfValueF`, [701](#)

`\IfValueT`, [700](#), [2494](#),  
[2749](#), [2751](#)  
`\IfValueTF`, [699](#), [2711](#), [2765](#)  
`\ikern`, [3453](#)  
`\IMO`, [3391](#)  
`\in`, [2882](#), [3106](#)  
`\inclasthook`, [3511](#), [3533](#)  
`\includegraphics`, [2618](#)  
`\infty`, [2856](#)  
`\iota`, [2919](#)  
`\itemindent`, [2265](#), [2283](#)  
`itemize*`, [2275](#)  
`\iteracro`, [3355](#), [3359](#)  
  
`\justified`, [4023](#)  
  
`\kappa`, [2920](#)  
  
`\labelsep`, [2267](#), [2285](#)  
`\labelwidth`, [2266](#), [2267](#),  
[2284](#), [2285](#)  
`\Lambda`, [2899](#)  
`\lambda`, [2921](#)  
`\larger`, *p. 16*, [1176](#), [2966](#),  
[2971](#), [3011](#), [3012](#),  
[3015](#), [3016](#), [3017](#), [3018](#)  
`\largerr`, *p. 16*, [1180](#), [3013](#),  
[3014](#)  
`\LaTeXe`, [2299](#), [2349](#)  
`\LaTeXpar`, [2360](#)  
`\ldate`, [3984](#), [3993](#)  
`\le`, [2830](#)  
`\leftarrow`, [2871](#), [3104](#)  
`\leftmargin`, [2264](#), [2282](#)  
`\leftrightarrow`, [2873](#), [2998](#)  
`\leftslanting`, [4107](#)  
`\leq`, [2829](#), [2830](#)  
`\linebreak`, [4073](#)  
`\linedate`, [3967](#), [3983](#), [3987](#)  
`\linedate@`, [3967](#), [3968](#), [3969](#)  
`\linedate@@`, [3967](#), [3968](#)  
`\list`, [2263](#), [2281](#)  
`\listparindent`, [2268](#), [2286](#)  
`\lit`, [4126](#)  
`\litshape`, [4108](#), [4124](#),  
[4126](#), [4148](#)  
`\liturgiques`, [3315](#)  
`\longafterfi`, [1001](#), [1003](#)  
`\longpauza`, [3738](#), [3739](#)  
`\looseness`, [3645](#), [3656](#)  
`\lozenge`, [2888](#)  
`\lpauza`, [3706](#)  
`\lsl`, [4129](#)  
`\LuaTeX`, [2440](#)

`luzniej`, [3649](#)  
`luzniej*`, [3654](#)  
`\luzniejcore`, [3641](#), [3649](#)  
  
`\macro`, [2465](#)  
`\MakeUppercase`, [3543](#)  
`\mapsto`, [3097](#)  
`\marg`, [1421](#), [1445](#)  
`maszynopis`, [4005](#)  
`\math@arg`, [1441](#), [1442](#)  
`\mathalpha`, [2896](#), [2897](#),  
[2898](#), [2899](#), [2900](#),  
[2901](#), [2902](#), [2903](#),  
[2904](#), [2905](#), [2909](#),  
[2910](#), [2911](#), [2912](#),  
[2913](#), [2914](#), [2915](#),  
[2916](#), [2917](#), [2918](#),  
[2919](#), [2920](#), [2921](#),  
[2922](#), [2923](#), [2924](#),  
[2925](#), [2926](#), [2927](#),  
[2928](#), [2929](#), [2930](#),  
[2931](#), [2932](#), [2933](#), [2934](#)  
`\mathbin`, [2811](#), [2812](#), [2851](#),  
[2853](#), [2854](#), [2860](#),  
[2869](#), [2892](#), [2893](#),  
[2894](#), [2989](#), [2992](#),  
[3032](#), [3033](#), [3099](#), [3106](#)  
`\mathchar@type`, [2713](#),  
[2728](#), [2802](#)  
`\mathchoice`, [2963](#), [2985](#),  
[3024](#), [3095](#)  
`\mathclose`, [2844](#), [2848](#),  
[2867](#), [3012](#), [3014](#),  
[3016](#), [3018](#), [3030](#)  
`\mathfrak`, [3351](#)  
`\mathop`, [2861](#), [2963](#)  
`\mathopen`, [2841](#), [2846](#),  
[2866](#), [3011](#), [3013](#),  
[3015](#), [3017](#), [3029](#)  
`\mathord`, [2814](#), [2815](#), [2816](#),  
[2817](#), [2818](#), [2819](#),  
[2820](#), [2821](#), [2822](#),  
[2823](#), [2855](#), [2856](#),  
[2868](#), [2888](#), [2889](#)  
`\mathpunct`, [2837](#), [2838](#),  
[2839](#), [2840](#)  
`\mathrel`, [2813](#), [2825](#),  
[2826](#), [2852](#), [2857](#),  
[2858](#), [2862](#), [2864](#),  
[2870](#), [2871](#), [2872](#),  
[2873](#), [2876](#), [2880](#),  
[2882](#), [2998](#), [3037](#),  
[3098](#), [3104](#), [3105](#)  
`\mathrm`, [2974](#)  
`\medmuskip`, [1338](#)

<code>\meta</code> , <a href="#">1284</a> , <a href="#">1320</a> , <a href="#">1421</a> , <a href="#">1428</a> , <a href="#">1435</a>	<code>\newskip</code> , <a href="#">4080</a>	<a href="#">1376</a> , <a href="#">1396</a> , <a href="#">1400</a> , <a href="#">1463</a> , <a href="#">1474</a> , <a href="#">1517</a> ,
<code>\meta@font@select</code> , <a href="#">1295</a> , <a href="#">1314</a>	<code>\newtoks</code> , <a href="#">485</a> , <a href="#">488</a>	<a href="#">1554</a> , <a href="#">1576</a> , <a href="#">1598</a> ,
<code>\mkern</code> , <a href="#">3040</a>	<code>\newwrite</code> , <a href="#">3239</a>	<a href="#">1602</a> , <a href="#">1797</a> , <a href="#">2319</a> ,
<code>\mskip</code> , <a href="#">1338</a>	<code>\next@ddc</code> , <a href="#">580</a> , <a href="#">583</a> , <a href="#">591</a> , <a href="#">595</a> , <a href="#">600</a>	<a href="#">2516</a> , <a href="#">2563</a> , <a href="#">2574</a> ,
<code>\mu</code> , <a href="#">2922</a>	<code>\nfss@text</code> , <a href="#">1293</a>	<a href="#">2585</a> , <a href="#">2627</a> , <a href="#">2681</a> ,
<code>\multiply</code> , <a href="#">2339</a> , <a href="#">2342</a> , <a href="#">3594</a> , <a href="#">3644</a> , <a href="#">3655</a>	<code>\nieczer</code> , <a href="#">3323</a>	<a href="#">2806</a> , <a href="#">2955</a> , <a href="#">3057</a> ,
<code>\mw@getflags</code> , <a href="#">2025</a>	<code>\nobeckspace</code> , <a href="#">3560</a>	<a href="#">3061</a> , <a href="#">3124</a> , <a href="#">3138</a> ,
<code>\mw@HeadingBreakAfter</code> , <a href="#">1870</a> , <a href="#">1890</a> , <a href="#">1905</a> , <a href="#">1909</a> , <a href="#">1940</a> , <a href="#">2026</a>	<code>\nocite</code> , <a href="#">1725</a>	<a href="#">3146</a> , <a href="#">3262</a> , <a href="#">3264</a> ,
<code>\mw@HeadingBreakBefore</code> , <a href="#">1867</a> , <a href="#">1939</a> , <a href="#">2027</a>	<code>\nohy</code> , <a href="#">3457</a>	<a href="#">3266</a> , <a href="#">3356</a> , <a href="#">3394</a> ,
<code>\mw@HeadingLevel</code> , <a href="#">1847</a> , <a href="#">1850</a>	<code>\nolimits</code> , <a href="#">2978</a> , <a href="#">2982</a>	<a href="#">3464</a> , <a href="#">3565</a> , <a href="#">3674</a> ,
<code>\mw@HeadingRunIn</code> , <a href="#">1885</a> , <a href="#">1939</a>	<code>\nolinkurl</code> , <a href="#">4242</a>	<a href="#">3689</a> , <a href="#">3701</a> , <a href="#">3706</a> ,
<code>\mw@HeadingType</code> , <a href="#">1866</a> , <a href="#">2001</a> , <a href="#">2033</a> , <a href="#">2034</a> , <a href="#">2047</a>	NoNumSecs, <a href="#">1793</a>	<a href="#">3715</a> , <a href="#">3727</a> , <a href="#">3787</a> ,
<code>\mw@HeadingWholeWidth</code> , <a href="#">1888</a> , <a href="#">1940</a>	<code>\not@onlypreamble</code> , <a href="#">1680</a> , <a href="#">1684</a> , <a href="#">1685</a> , <a href="#">1686</a> , <a href="#">1687</a> , <a href="#">1688</a> , <a href="#">2945</a>	<a href="#">3920</a> , <a href="#">3959</a> , <a href="#">3967</a> ,
<code>\mw@normalheading</code> , <a href="#">1892</a> , <a href="#">1901</a> , <a href="#">1904</a> , <a href="#">1908</a> , <a href="#">2082</a>	<code>\NoValue</code> , <a href="#">562</a> , <a href="#">694</a> , <a href="#">695</a> , <a href="#">696</a> , <a href="#">697</a> , <a href="#">698</a>	<a href="#">3968</a> , <a href="#">3969</a> , <a href="#">3983</a> ,
<code>\mw@processflags</code> , <a href="#">1941</a>	<code>\NoValueInIt</code> , <a href="#">695</a>	<a href="#">3984</a> , <a href="#">3996</a> , <a href="#">3999</a> ,
<code>\mw@runinheading</code> , <a href="#">1886</a> , <a href="#">2083</a>	<code>\nu</code> , <a href="#">2923</a>	<a href="#">4108</a> , <a href="#">4123</a> , <a href="#">4126</a> ,
<code>\mw@secdef</code> , <a href="#">1946</a> , <a href="#">1947</a> , <a href="#">1948</a> , <a href="#">1954</a>	<code>\numexpr</code> , <a href="#">3781</a> , <a href="#">3782</a> , <a href="#">3794</a> , <a href="#">4183</a>	<a href="#">4129</a> , <a href="#">4145</a> , <a href="#">4154</a> ,
<code>\mw@section</code> , <a href="#">1945</a>	<code>\oarg</code> , <a href="#">1426</a>	<a href="#">4160</a> , <a href="#">4212</a> , <a href="#">4217</a>
<code>\mw@sectionxxx</code> , <a href="#">1846</a>	<code>\oldLaTeX</code> , <a href="#">2298</a>	<code>\pdfTeX</code> , <a href="#">2414</a>
<code>\mw@secundef</code> , <a href="#">1950</a> , <a href="#">1962</a> , <a href="#">1965</a>	<code>\oldLaTeXe</code> , <a href="#">2299</a>	<code>\pdfLaTeX</code> , <a href="#">2417</a>
<code>\mw@setflags</code> , <a href="#">1951</a>	<code>\Omega</code> , <a href="#">2905</a>	<code>\pdfTeX</code> , <a href="#">2416</a>
<code>\n@melet</code> , <a href="#">1651</a> , <a href="#">1976</a> , <a href="#">1978</a> , <a href="#">2193</a> , <a href="#">2199</a> , <a href="#">2233</a> , <a href="#">2544</a>	<code>\omega</code> , <a href="#">2934</a>	<code>\perhaps@grab@ms</code> , <a href="#">569</a> , <a href="#">625</a>
<code>\nameshow</code> , <a href="#">828</a>	<code>\PackageError</code> , <a href="#">612</a> , <a href="#">1699</a> , <a href="#">3716</a> , <a href="#">3728</a>	<code>\Phi</code> , <a href="#">2903</a>
<code>\nameshowthe</code> , <a href="#">829</a>	<code>\PackageWarning</code> , <a href="#">1168</a> , <a href="#">1171</a>	<code>\phi</code> , <a href="#">2932</a>
<code>napapierki</code> , <a href="#">3632</a>	<code>\pagebreak</code> , <a href="#">1893</a> , <a href="#">1905</a> , <a href="#">1909</a>	<code>\Pi</code> , <a href="#">2926</a>
<code>\napapierkicore</code> , <a href="#">3629</a> , <a href="#">3633</a>	<code>\pagegoal</code> , <a href="#">3251</a>	<code>\pi</code> , <a href="#">2925</a>
<code>\napapierkistretch</code> , <a href="#">3627</a> , <a href="#">3630</a>	<code>\pagetotal</code> , <a href="#">3252</a>	<code>\pk</code> , <a href="#">1376</a>
<code>\nawj</code> , <a href="#">3658</a>	<code>\paragraph</code> , <a href="#">3989</a>	<code>\PlainTeX</code> , <a href="#">2399</a>
<code>\nazwired</code> , <a href="#">3802</a>	<code>\ParanoidPostsec</code> , <a href="#">2058</a>	<code>\pm</code> , <a href="#">2869</a>
<code>\ne</code> , <a href="#">2859</a>	<code>\parg</code> , <a href="#">1433</a>	<code>\polskadata</code> , <a href="#">3816</a> , <a href="#">3857</a>
<code>\neb</code> , <a href="#">3034</a>	<code>\partial</code> , <a href="#">2868</a>	<code>\possfil</code> , <a href="#">1405</a>
<code>\neg</code> , <a href="#">2860</a> , <a href="#">3039</a>	<code>\partopsep</code> , <a href="#">2264</a> , <a href="#">2282</a>	<code>\ppauza</code> , <a href="#">3727</a>
<code>\neq</code> , <a href="#">2858</a> , <a href="#">2859</a> , <a href="#">3033</a>	<code>\PassOptionsToPackage</code> , <a href="#">2494</a>	<code>\ppauza@skipcore</code> , <a href="#">3670</a> , <a href="#">3721</a> , <a href="#">3722</a>
<code>\neqb</code> , <a href="#">3033</a> , <a href="#">3034</a>	<code>\pauza</code> , <a href="#">3689</a>	<code>\pprovide</code> , <a href="#">216</a> , <a href="#">3350</a> , <a href="#">3792</a>
<code>\newcount</code> , <a href="#">3639</a>	<code>\pauza@skipcore</code> , <a href="#">3667</a> , <a href="#">3678</a> , <a href="#">3679</a>	<code>\prependtomacro</code> , <a href="#">378</a>
<code>\newcounter</code> , <a href="#">1758</a> , <a href="#">1793</a>	<code>\pauzacore</code> , <a href="#">3668</a> , <a href="#">3680</a> , <a href="#">3685</a> , <a href="#">3693</a> , <a href="#">3702</a> , <a href="#">3707</a> , <a href="#">3738</a> , <a href="#">3741</a> , <a href="#">4017</a> , <a href="#">4018</a>	<code>\printsaces</code> , <a href="#">1346</a> , <a href="#">1355</a>
<code>\newgif</code> , <a href="#">264</a>	<code>\pauzadial</code> , <a href="#">3695</a> , <a href="#">3701</a>	<code>\protected</code> , <a href="#">187</a> , <a href="#">216</a> , <a href="#">276</a> , <a href="#">295</a> , <a href="#">501</a> , <a href="#">2801</a> , <a href="#">3120</a> , <a href="#">3122</a>
	<code>\pdef</code> , <a href="#">187</a> , <a href="#">201</a> , <a href="#">264</a> , <a href="#">281</a> , <a href="#">300</a> , <a href="#">301</a> , <a href="#">322</a> , <a href="#">337</a> , <a href="#">345</a> , <a href="#">457</a> , <a href="#">898</a> , <a href="#">934</a> , <a href="#">975</a> , <a href="#">1141</a> , <a href="#">1180</a> , <a href="#">1181</a> , <a href="#">1284</a> , <a href="#">1355</a> , <a href="#">1362</a> ,	<code>\provide</code> , <a href="#">201</a> , <a href="#">216</a>
	<a href="#">1376</a> , <a href="#">1396</a> , <a href="#">1400</a> , <a href="#">1463</a> , <a href="#">1474</a> , <a href="#">1517</a> , <a href="#">1554</a> , <a href="#">1576</a> , <a href="#">1598</a> , <a href="#">1602</a> , <a href="#">1797</a> , <a href="#">2319</a> , <a href="#">2516</a> , <a href="#">2563</a> , <a href="#">2574</a> , <a href="#">2585</a> , <a href="#">2627</a> , <a href="#">2681</a> , <a href="#">2806</a> , <a href="#">2955</a> , <a href="#">3057</a> , <a href="#">3061</a> , <a href="#">3124</a> , <a href="#">3138</a> , <a href="#">3146</a> , <a href="#">3262</a> , <a href="#">3264</a> , <a href="#">3266</a> , <a href="#">3356</a> , <a href="#">3394</a> , <a href="#">3464</a> , <a href="#">3565</a> , <a href="#">3674</a> , <a href="#">3689</a> , <a href="#">3701</a> , <a href="#">3706</a> , <a href="#">3715</a> , <a href="#">3727</a> , <a href="#">3787</a> , <a href="#">3920</a> , <a href="#">3959</a> , <a href="#">3967</a> , <a href="#">3968</a> , <a href="#">3969</a> , <a href="#">3983</a> , <a href="#">3984</a> , <a href="#">3996</a> , <a href="#">3999</a> , <a href="#">4108</a> , <a href="#">4123</a> , <a href="#">4126</a> , <a href="#">4129</a> , <a href="#">4145</a> , <a href="#">4154</a> , <a href="#">4160</a> , <a href="#">4212</a> , <a href="#">4217</a>	<code>\Psi</code> , <a href="#">2904</a>
		<code>\psi</code> , <a href="#">2933</a>
		<code>\quad</code> , <a href="#">3802</a>
		<code>\quantifierhook</code> , <a href="#">3066</a> , <a href="#">3087</a>
		<code>\rdate</code> , <a href="#">3983</a>
		<code>\real</code> , <a href="#">3588</a> , <a href="#">3590</a>
		<code>\reflectbox</code> , <a href="#">2424</a> , <a href="#">2431</a>
		<code>\relaxen</code> , <a href="#">398</a> , <a href="#">398</a> , <a href="#">1936</a> , <a href="#">3761</a> , <a href="#">4169</a>
		<code>\relsize</code> , <i>p. 16</i> , <a href="#">1141</a> , <a href="#">1142</a> , <a href="#">1176</a> , <a href="#">1177</a> , <a href="#">1178</a> , <a href="#">1179</a> , <a href="#">1180</a> , <a href="#">1181</a>

`\renewcommand*`, 3340  
`\RequirePackage`, 2496, 3236, 3333, 3572  
`\resizebox`, 2617  
`\resizegraphics`, 2614  
`\Restore@Macro`, 1557, 1560, 1580, 1590  
`\Restore@Macros`, 1576, 1578  
`\Restore@MacroSt`, 1558, 1566  
`\RestoreEnvironment`, 1602  
`\RestoreMacro`, 1554, 2503, 2505, 2508  
`\RestoreMacro*`, 1604, 2505  
`\RestoreMacros`, 1576  
`\RestoringDo`, 1639  
`\rho`, 2927  
`\rightarrow`, 2872, 3105  
`\rightline`, 3983, 4069  
`\romannumeral`, 2262, 2280  
`\rotatebox`, 2976, 2982, 2989, 2992  
`\rs@size@warning`, 1160, 1165, 1168  
`\rs@unknown@warning`, 1155, 1171  
`\runindate`, 3988  
  
`\scshape`, 2397, 3099, 3386  
`\secondclass`, 3414  
`\SecondClasstrue`, 3416  
`\SetSectionFormatting`, 1936, 1937, 2101, 2105, 2113, 2121, 2128, 2135, 2140  
`\SetTwoheadSkip`, 2085, 2112, 2120, 2127  
`\sfilename`, 1355, 1365  
`\shortpauza`, 3740  
`\shortthousep`, 4216  
`\showboxbreadth`, 824  
`\showboxdepth`, 824  
`\ShowFont`, 3300  
`\showlists`, 824  
`\showthe`, 829  
`\Sigma`, 2901  
`\sigma`, 2928  
`\sim`, 2870  
`\skewchar`, 2668, 2687, 2696  
`\SliTeX`, 2396  
`\smaller`, *p. 16*, 1177, 3431, 3435  
`\smallerr`, *p. 16*, 1181, 3603  
`\smallskipamount`, 3179, 3180  
  
`\smartunder`, 1220  
`\SMglobal`, 1463  
`\something@in`, 670, 671, 674, 679  
`\something@tmp`, 678, 679, 683  
`\something@tmpb`, 682, 683  
`\sqrtsign`, 2939  
`\square`, 2889  
`\Store@Macro`, 1480, 1483, 1520  
`\Store@Macros`, 1517, 1518  
`\Store@MacroSt`, 1481, 1490  
`\Stored@Macro`, 1589, 1590  
`\storedcsname`, 1593, 4242  
`\StoredMacro`, 1589  
`\StoreEnvironment`, 1598  
`\StoreMacro`, 1474, 2353, 2412, 2495, 4017, 4241  
`\StoreMacro*`, 1600, 2354  
`\StoreMacros`, 1517  
`\StoringAndRelaxingDo`, 1619  
`\strip@pt`, 4112, 4117, 4134  
`\subs`, 1192, 1222  
`\sum`, 2975  
`\symgmathroman`, 2808, 2940  
`\symletters`, 2907  
  
`\tau`, 2930  
`\TB`, 2405  
`\TeXbook`, 2404, 2405  
`\textbullet`, 3787, 3797  
`\textcolor`, 3323  
`\textlarger`, 1178  
`\textlit`, 4123  
`\textsl`, 2404  
`\textsmaller`, 1179  
`\textstyle`, 2351  
`\textsuperscript`, 2622, 2627  
`\texttilde`, 3146  
`\textwidth`, 4051, 4052, 4070, 4071  
`\thedata`, 3993  
`\Theta`, 2898  
`\theta`, 2917  
`\thickmuskip`, 3041  
`\thous`, 4217  
`\thous@inner`, 4219, 4222  
`\thousep`, 4160, 4212  
`\thr@@`, 2258, 2276  
`\time`, 3781, 3782  
`\tinycae`, 3570  
  
`\TODO`, 3199  
`\toks`, 2072, 2073, 2079, 2080  
`\tolerance`, 3644, 3655, 4009  
`\truetextsuperscript`, 2624, 2626  
`\twocoltoc`, 3235, 3244  
`\twopar`, 4047  
`\twoparinit`, 4045  
`\tytul`, 3999  
`tytulowa`, 3799  
  
`\udigits`, 2516, 2519  
`\undeksmallskip`, 3180  
`\unexpanded`, 183, 380, 795, 796, 800, 801, 804, 1977  
`\unless`, 447, 484, 2485, 2988, 2991, 2995, 3341  
`\uparrow`, 2876  
`\upshape`, 2692, 4149  
`\Upsilon`, 2902  
`\upsilon`, 2931  
`\Url`, 3046, 3047  
`\url`, 4241, 4242  
`\urladdstar`, 4238, 4245  
`\usc`, 3394, 3396  
`\uscacro`, 3396  
`\usecounter`, 2269  
  
`\value`, 1762  
`\varepsilon`, 2351, 2409, 2914  
`\varnothing`, 2855, 3100  
`\varsigma`, 2929  
`\vartheta`, 2918  
`\vee`, 2894, 2989, 3039  
`\verb`, 2495, 2503  
`\visiblepace`, 1260, 1262, 1340  
`\Vs`, 3120  
`\vs`, 1340, 1346, 1350  
  
`\wd`, 2331, 2334, 2364, 2369, 2377, 2378, 2980, 3090, 3091, 3102, 3103  
`\Web`, 2401  
`\wedge`, 2893, 2992, 3039  
`\whenonly`, 3527  
`\whern`, 4076  
`\wherncore`, 4068, 4077, 4083  
`\whernskip`, 4077, 4080, 4081  
`\whernup`, 4083  
`\WPheadings`, 2100  
`\Ws`, 3122  
`\wyzejnizej`, 3611



<code>\Wz</code> , <a href="#">3124</a>	<code>\XeTeXversion</code> , <a href="#">162</a> , <a href="#">163</a> , <a href="#">172</a> , <a href="#">1197</a> , <a href="#">2484</a> , <a href="#">2485</a> , <a href="#">3666</a> , <a href="#">3748</a>	<code>\xleq</code> , <a href="#">2825</a> , <a href="#">2828</a> , <a href="#">2829</a>
<code>\xathousep</code> , <a href="#">4212</a> , <a href="#">4226</a>	<code>\xgeq</code> , <a href="#">2826</a> , <a href="#">2832</a> , <a href="#">2833</a>	<code>\xparsed@args</code> , <a href="#">488</a> , <a href="#">512</a> , <a href="#">519</a>
<code>\Xedekfrac</code> , <a href="#">2524</a>	<code>\Xi</code> , <a href="#">2900</a>	<code>\xxt@visible</code> , <a href="#">1259</a> , <a href="#">1260</a>
<code>\XeLaTeX</code> , <a href="#">2428</a>	<code>\xi</code> , <a href="#">2924</a>	<code>\yesy</code> , <a href="#">3458</a>
<code>\XeTeX</code> , <a href="#">2421</a>	<code>\xiiand</code> , <a href="#">1243</a>	<code>\z@skip</code> , <a href="#">4036</a>
<code>\XeTeXdelcode</code> , <a href="#">2769</a> , <a href="#">2773</a>	<code>\xiibackslash</code> , <a href="#">1230</a> , <a href="#">1234</a>	<code>\zeta</code> , <a href="#">2915</a>
<code>\XeTeXdelimiter</code> , <a href="#">2802</a>	<code>\xiihash</code> , <a href="#">1249</a>	<code>\zf@scale</code> , <a href="#">3578</a> , <a href="#">3581</a> , <a href="#">3582</a>
<code>\XeTeXglyph</code> , <a href="#">3794</a>	<code>\xiilbrace</code> , <a href="#">1205</a>	<code>\zwrobcy</code> , <a href="#">3996</a>
<code>\XeTeXglyphindex</code> , <a href="#">3794</a>	<code>\xiipercent</code> , <a href="#">1239</a>	<code>\cdot</code> , <a href="#">3266</a>
<code>\XeTeXinputencoding</code> , <a href="#">173</a>	<code>\xiirbrace</code> , <a href="#">1206</a>	<code>\-</code> , <a href="#">3715</a> , <a href="#">3754</a> , <a href="#">3756</a> , <a href="#">3757</a> , <a href="#">3758</a>
<code>\XeTeXmathchardef</code> , <a href="#">2719</a>	<code>\xiispace</code> , <a href="#">841</a> , <a href="#">842</a> , <a href="#">1246</a> , <a href="#">1262</a>	<code>\-</code> , <a href="#">3674</a> , <a href="#">3735</a> , <a href="#">3753</a> , <a href="#">3756</a> , <a href="#">3757</a> , <a href="#">3758</a>
<code>\XeTeXmathcode</code> , <a href="#">2722</a> , <a href="#">2727</a>	<code>\xiistring</code> , <a href="#">840</a>	
<code>\XeTeXradical</code> , <a href="#">2940</a>	<code>\xiiunder</code> , <a href="#">1195</a> , <a href="#">1198</a> , <a href="#">1199</a>	
<code>\XeTeXthree</code> , <a href="#">2489</a>		