

Grzegorz Murzynowski

The gmutils Package*

Written by Grzegorz Murzynowski,
natror at o2 dot pl

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

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

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

LPPL status: "author-maintained".

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

```
84 \NeedsTeXFormat{LaTeX2e}
85 \ProvidesPackage{gmutils}
86 [2008/10/04_v0.95_some_rather_TeXnical_macros,_some_of_them_
tricky_(GM)]
```

Contents

Intro	2	Macros for printing macros and filenames	19
Installation	2	Storing and restoring the meanings of cses	21
Contents of the gmutils.zip archive . .	2	Not only preamble!	24
Compiling of the documentation . . .	2	Third person pronouns	25
A couple of abbreviations	3	Improvements to mwcls sectioning commands	26
\firstofone and the queer \catcodes .	4	An improvement of MW's \SetSectionFormatting	28
Global Boolean switches	4	Negative \addvspace	29
\gm@ifundefined—a test that doesn't create any hash entry unlike \@ifundefined	6	My heading setup for mwcls	30
Storing and restoring the catcodes of specials	7	Compatibilising standard and mwcls sectionings	31
From the ancient xparse of T_EXLive 2007 .	7	enumerate* and itemize*	33
Ampulex Compressa-like modifications of macros	11	The logos	34
\@ifnextcat, \@ifnextac	13	Expandable turning stuff all into 'other' .	36
\afterfi and pals	15	Brave New World of X_YT_EX	36
Environments redefined	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 .	38
From relsize	16	Minion and Garamond Premier kerning and ligature fixes	42
Some 'other' stuff	18	Varia	42
Metasymbols	19	\@ifempty	46

* This file has version number v0.95 dated 2008/10/04.

<code>\include not only .tex's</code>	46	Typesetting dates in my memoirs	51
Faked small caps	47	A left-slanted font	56
See above/see below	48	Thousand separator	57
luzniej and napa-		hyperref's <code>\nolinkurl</code> into <code>\url*</code> . .	58
pierki—environments		Change History	58
used in page breaking for money .	49	Index	60

Intro

The `gmutils.sty` package provides some macros that are analogous to the standard L^AT_EX 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^A)T_EX-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 T_EX 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 L^AT_EX 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, T_EX will find it), then `MakeIndex` on the `gmutils.idx` file, and then L^AT_EX 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 `gm-docc.cls`), `gmverb.sty`, `gmutils.sty`, `gmiflink.sty` and also some standard packages: `hyperref.sty`, `color.sty`, `geometry.sty`, `multicol.sty`, `lmodern.sty`, `fontenc.sty` that should be installed on your computer by default.

If you had not installed the `mwcls` classes (available on CTAN and present in T_EX Live e.g.), the result of your compilation might differ a bit from the `.pdf` provided in this `.zip`

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

```

158 \ifx\XeTeXversion\relax
159   \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 3438.
166 \fi
168 \ifdefined\XeTeXversion
169 \XeTeXinputencoding\utf-8% we use Unicode dashes later in this file.
170 \fi% and if we are not in XeTeX, we skip them thanks to XeTeX-test.

```

A couple of abbreviations

```

\@xa 176 \let\@xa\expandafter
\@nx 177 \let\@nx\noexpand
\@xau 179 \def\@xau{\@xa\unexpanded\@xa}
\pdef 183 \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 190 \long\def\gobble#1{}
\@gobble 192 \let\@gobble\gobble
\gobbletwo 193 \let\gobbletwo\@gobbletwo
\provide 197 \long\pdef\provide#1{%
198   \ifdefined#1%
199     \ifx\relax#1\afterfifi{\def#1}%
200     \else\afterfifi{\gmu@gobdef}%
201     \fi
202   \else\afterfi{\def#1}%
203   \fi}
\gmu@gobdef 206 \long\def\gmu@gobdef#1#1{%
207   \def\gmu@tempa{}}% it's a junk macro assignment to absorb possible prefixes.
209   \@gobble}
\pprovide 212 \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^AT_EX's \providecommand, our \{(p)provide allow any parameters string just like \def (because they just *expand* to \def).

```

\@nameedef 225 \long\def\@nameedef#1#2{%
226   \@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_EX's eyes', as my T_EX Guru taught me.

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

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

```
\foone 242 \long\def\foone#1{\begingroup#1\egfirstofone}
```

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

```
\foeatletter 246 \long\def\foeatletter{\foone\makeatletter}
```

Global Boolean switches

The `\newgif` declaration's effect is used even in the L^AT_EX 2_ε 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^AT_EX's `\newif` modulo the letter *g* and the `\global` prefix. (File d: ltdefs.dtx Date: 2004/02/20 Version v1.3g, lines 139–150)

```
\newgif 260 \pdef\newgif#1{%
261   {\escapechar\m@ne
262     \global\let#1\iffalse
263     \@gif#1\iftrue
264     \@gif#1\iffalse
265   }}
```

'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 271 \def\@gif#1#2{%
272   \protected\@xa\gdef\csname\@xa\@gobbletwo\string#1%
273   g% the letter g for 'global'.
274   \@xa\@gobbletwo\string#2\endcsname
275   {\global\let#1#2}}
```

```
277 \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.
```

```
284   \count@\escapechar\escapechar\m@ne
285   \let#1\iffalse
286   \@if#1\iftrue
287   \@if#1\iffalse
288   \escapechar\count@}
```

```
\@if 290 \def\@if#1#2{%
291   \protected\@xa\def\csname\@xa\@gobbletwo\string#1%
292   \@xa\@gobbletwo\string#2\endcsname
293   {\let#1#2}}
```

```
\hidden@iffalse 296 \pdef\hidden@iffalse{\iffalse}
```

```
\hidden@iftrue 297 \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 `\newgif`.

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

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

Warning. I use it because of very special reasons in `gmdoc` and in general it is probably not a good idea to make `\refstepcounter` global since it is contrary to the original L^AT_EX approach.

```
\grefstepcounter 318 \pdef\grefstepcounter#1{%
319   {\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 333 \pdef\hgrefstepcounter#1{%
334   \ifhmode\leavevmode\fi\grefstepcounter{#1}}
```

By the way I read some lines from *The T_EXbook* 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 341 \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 351 \def\glet{\global\let}
```

L^AT_EX 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 359 \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 364 \long\def\addto@macro#1#2{%
365   \toks@{\xa{#1#2}}%
366   \edef#1{\the\toks@}%
367 }% (\toks@ is a scratch register, namely \tokso.)
```

And for use in the very document,

```
\addtomacro 371 \let\addtomacro=\addto@macro
```

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

```
\prependtomacro 374 \long\def\prependtomacro#1#2{%
```

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

Note that `\prependtomacro` can be prefixed.

```
\addtotoks 380 \long\def\addtotoks#1#2{%
381 #1=\@xa{\the#1#2}}
```

```
\@emptyify 384 \newcommand*\@emptyify[1]{\let#1=\@empty}
\emptyify 385 \@ifdefinable\emptyify{\let\emptyify\@emptyify}
```

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

```
\g@emptyify 389 \newcommand*\g@emptyify{\global\@emptyify}
\gemptyify 390 \@ifdefinable\gemptyify{\let\gemptyify\g@emptyify}

\@relaxen 393 \newcommand\@relaxen[1]{\let#1=\relax}
\relaxen 394 \@ifdefinable\relaxen{\let\relaxen\@relaxen}
```

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

```
\g@relaxen 398 \newcommand*\g@relaxen{\global\@relaxen}
\grelaxen 399 \@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 408 \long\def\gm@ifundefined#1{% not \protected because expandable.
414 \ifcsname#1\endcsname% defined
415 \@xa\ifx\csname_#1\endcsname\relax% but as \relax
416 \afterfifi\@firstoftwo%
417 \else% defined and not \relax
418 \afterfifi\@secondoftwo%
419 \fi
420 \else% not defined
421 \afterfi\@firstoftwo%
422 \fi}
```

```
\gm@testdefined 425 \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.
430 \csname
431 \ifdefined#1%
432 \ifx#1\relax
433 \iffalse%
434 \else\iftrue%
435 \fi
436 \else\iffalse%
437 \fi\endcsname
438 }
```

```
\gm@testundefined 440 \long\def\gm@testundefined#1\iftrue{% we expand the last macro two levels.
We repeat the parameter string to force the proper syntax.
443 \@xa\@xa\@xa\unless\gm@testdefined#1\iftrue}
```

Storing and restoring the catcodes of specials

```

\gmu@storespecials 448 \newcommand*\gmu@storespecials[1] [] {% we provide a possibility of adding
                    stuff. For usage see line 2673.
450   \def\do##1{\catcode`\@nx##1=\the\catcode`##1\relax}%
451   \edef\gmu@restorespecials{\dospecials\do\^~M#1}}

\gmu@septify 453 \newcommand*\gmu@septify[1] [] {% restoring the standard catcodes of specials.
              The name is the opposite of 'sanitize' :-). It restores also the original catcode
              of ^^M
456   \def\do{\relax\catcode`}%
457   \do\_\do\_\do\_\do\_{1\do\}2\do\$_3\do\&4%
458   \do\#6\do\^7\do\_8\do\%14\do\~13\do\^~M5#1\relax}

```

From the ancient xparse of T_EX 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_EX Live 2007-13, in Ubuntu packages at least. It's a stub 'im Erwartung' (Schönberg) for the L^AT_EX₃ 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

```

478 \gm@testundefined\DeclareDocumentCommand\iftrue
480 \unless\ifdefined\@temptokenb
\@temptokenb 481 \newtoks\@temptokenb
482 \fi
\xparsed@args 484 \newtoks\xparsed@args
\DeclareDocumentCommand 486 \long\def\DeclareDocumentCommand#1#2#3{%
                    % #1 command to be defined,
                    % #2 arguments specification,
                    % #3 definition body.
492   \@tempcnta\z@
493   \toks@{}%
494   \@temptokena\toks@
495   \@temptokenb\toks@
496   \@ddc#2X% X is the guardian of parsing.
497   \protected\edef#1{% The \protected prefix is my (GM) addition.
498     \@nx\@ddc@
499     {\the\toks@}%
500     \@xa\@nx\csname\string#1\endcsname
501     \@nx#1%
502   }%
503   \long\@xa\def\csname\string#1\@xa\endcsname
504   \the\@temptokena{#3}}

DeclareDocumentEnvironment 506 \long\def\DeclareDocumentEnvironment#1#2#3#4{%
507   \@xa\DeclareDocumentCommand\csname#1\endcsname{#2}{%
508     \xparsed@args\toks@

```

```

509     #3}%
510     \@xa\let\csname_#1\endcsname\@parsed@endenv
511     \long\@xa\def\csname_#1\endcsname
512     \the\@temptokena{#4}}
\@parsed@endenv 514 \def\@parsed@endenv{%
515     \@xa\@parsed@endenv\the\xparsed@args}
\@parsed@endenv@ 517 \def\@parsed@endenv@#1{%
518     \csname_#1\endcsname}
\@ddc@ 520 \def\@ddc@#1#2#3{%
521     \ifx\protect\@typeset@protect
522     \@xa\@firstofone
523     \else
524     \protect#3\@xa\@gobble
525     \fi
526     {\toks@{#2}#1\the\toks@}}
\@ddc 528 \def\@ddc#1{%
529     \ifx#1X%
530     \else
531     \ifx#1m%
532     \addto@hook\@temptokenb_m%
533     \else
534     \toks@\@xa{%
535         \the\@xa\toks@
536         \csname_@ddc@\the\@temptokenb\@xa\endcsname
537         \csname_@ddc@#1\endcsname}%
538     \@temptokenb{}}%
539     \fi
540     \advance\@tempcnta\@ne
541     \@temptokena\@xa{%
542         \the\@xa\@temptokena\@xa##\the\@tempcnta}%
543     \@xa
544     \@ddc
545     \fi}
\@ddc@s 547 \long\def\@ddc@s#1\toks@{%
548     \@ifstar
549     {\addto@hook\toks@\BooleanTrue#1\toks@}%
550     {\addto@hook\toks@\BooleanFalse#1\toks@}}
\@ddc@m 552 \long\def\@ddc@m#1\toks@#2{%
553     \addto@hook\toks@{{#2}}#1\toks@}%
\@ddc@o 555 \long\def\@ddc@o#1\toks@{%
556     \@ifnextchar[%
557     {\@ddc@o@{#1}}
558     {\addto@hook\toks@\NoValue#1\toks@}}
\@ddc@o@ 560 \long\def\@ddc@o@#1[#2]{%
561     \addto@hook\toks@{{#2}}#1\toks@}
\@ddc 563 \def\@ddc#1{%
564     \ifx#1X%
565     \perhaps@grab@ms
566     \else

```



```

567 \ifx#1m%
568 \addto@hook\@temptokenb_m%
569 \else
570 \toks@{\@xa{%
571 \the\@xa\toks@
572 \csname_\@ddc@x\the\@temptokenb\@xa\endcsname
573 \csname_\@ddc@#1\endcsname}%
574 \@temptokenb{}}%
575 \ifx#1O%
576 \let\next@ddc\grab@default
577 \else
578 \ifx#1C%
579 \let\next@ddc\grab@default
580 \fi
581 \fi
582 \fi
583 \advance\@tempcnta\@ne
584 \@temptokena\@xa{%
585 \the\@xa\@temptokena\@xa##\the\@tempcnta}%
586 \@xa
587 \next@ddc
588 \fi
589 }%
\grab@default
591 \let\next@ddc\@ddc
592 \def\grab@default#1{%
593 \toks@{\@xa{%
594 \the\toks@
595 {#1}}}%
596 \let\next@ddc\@ddc
597 \@ddc
598 }
\@ddc@O 600 \long\def\@ddc@O#1#2\toks@{%
601 \@ifnextchar[%
602 {\@ddc@O@{#2}}%
603 {\addto@hook\toks@{{#1}}#2\toks@}}
\@ddc@c 605 \long\def\@ddc@c#1\toks@{%
606 \@ifnextchar(%
607 {\@ddc@c@{#1}%
608 {\PackageError{xparse}{Missing~coordinate~argument}%
609 {A~value~of~(o,o)~is~assumed}}%
610 \addto@hook\toks@{{oo}}#1\toks@}%
611 }
\@ddc@c@ 613 \long\def\@ddc@c@#1(#2,#3){%
614 \addto@hook\toks@{{#2}{#3}}#1\toks@}
\@ddc@C 616 \long\def\@ddc@C#1#2\toks@{%
617 \@ifnextchar(%
618 {\@ddc@c@{#2}%
619 {\addto@hook\toks@{{#1}}#2\toks@}}
621 \let\perhaps@grab@ms\relax
\grab@ms 622 \def\grab@ms{%
623 \toks@\@xa{%

```

```

624     \the\@xa\toks@
625     \csname_\@ddc@x\the\@temptokenb\endcsname
626   }}
627
628   \let\@ddc@m\undefined
629
630   \@ddc@xm \long\def\@ddc@xm#1\toks@#2{%
631     \addto@hook\toks@{{#2}}#1\toks@}
632
633   \@ddc@xmm \long\def\@ddc@xmm#1\toks@#2#3{%
634     \addto@hook\toks@{{#2}{#3}}#1\toks@}
635
636   \@ddc@xmnm \long\def\@ddc@xmnm#1\toks@#2#3#4{%
637     \addto@hook\toks@{{#2}{#3}{#4}}#1\toks@}
638
639   \@ddc@xmnmnm \long\def\@ddc@xmnmnm#1\toks@#2#3#4#5{%
640     \addto@hook\toks@{{#2}{#3}{#4}{#5}}#1\toks@}
641
642   \@ddc@xmnmnmnm \long\def\@ddc@xmnmnmnm#1\toks@#2#3#4#5#6{%
643     \addto@hook\toks@{{#2}{#3}{#4}{#5}{#6}}#1\toks@}
644
645   \@ddc@xmnmnmnmnm \long\def\@ddc@xmnmnmnmnm#1\toks@#2#3#4#5#6#7{%
646     \addto@hook\toks@{{#2}{#3}{#4}{#5}{#6}{#7}}#1\toks@}
647
648   \@ddc@xmnmnmnmnmnm \long\def\@ddc@xmnmnmnmnmnm#1\toks@#2#3#4#5#6#7#8{%
649     \addto@hook\toks@{{#2}{#3}{#4}{#5}{#6}{#7}{#8}}#1\toks@}
650
651   \@ddc@xmnmnmnmnmnmnm \long\def\@ddc@xmnmnmnmnmnmnm#1\toks@#2#3#4#5#6#7#8#9{%
652     \addto@hook\toks@{{#2}{#3}{#4}{#5}{#6}{#7}{#8}{#9}}#1\toks@}
653
654   \@ddc@xmnmnmnmnmnmnmnm \long\def\@ddc@xmnmnmnmnmnmnmnm\the\toks@#1#2#3#4#5#6#7#8#9{%
655     \addto@hook\toks@{{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}{#9}}\the%
656       \toks@}
657
658   \let\@ddc@x\relax
659
660   \DeclareDocumentEnvironment
661     \long\def\DeclareDocumentEnvironment#1#2#3#4{%
662       \@xa\DeclareDocumentCommand\csname#1\endcsname{#2}{%
663         #3}%
664       \@namedef{end#1}{#4}%
665     }
666
667   \let\@parsed@endenv\undefined
668   \let\@parsed@endenv@v\undefined
669
670   \IfSomethingTF
671     \something@in
672     \def\IfSomethingTF#1{\def\something@in{#1}\If@SomethingTF}
673     \def\IfSomethingT#1#2#3{\def\something@in{#1}%
674       \If@SomethingTF{#2}{#3}\@empty}
675     \something@in
676     \def\IfSomethingF#1#2#3{\def\something@in{#1}%
677       \If@SomethingTF{#2}\@empty{#3}}
678     \something@in
679     \def\If@SomethingTF#1{%
680       \def\something@tmp{#1}%
681       \ifx\something@tmp\something@in
682         \@xa\@secondofthree
683       \else
684         \@xa\def\@xa\something@tmpb\@xa{#1}%
685         \ifx\something@tmp\something@tmpb
686           \@xa\@xa\@xa\@thirdofthree
687         \else
688           \@xa\@xa\@xa\@firstofone
689         \fi
690       \fi

```

```

685   {\@xa\If@SomethingTF\@xa{#1}}%
686 }
\@secondofthree 688 \long\def\@secondofthree#1#2#3{#2}
\@thirdofthree 689 \long\def\@thirdofthree#1#2#3{#3}
\NoValue 690 \def\NoValue{-NoValue-}
\NoValueInIt 691 \def\NoValueInIt{\NoValue}
\IfNoValueTF 692 \def\IfNoValueTF{\IfSomethingTF\NoValue}
\IfNoValueT 693 \def\IfNoValueT{\IfSomethingT\NoValue}
\IfNoValueF 694 \def\IfNoValueF{\IfSomethingF\NoValue}
\IfValueTF 695 \def\IfValueTF#1#2#3{\IfNoValueTF{#1}{#3}{#2}}
696 \let\IfValueT\IfNoValueF
697 \let\IfValueF\IfNoValueT
\BooleanFalse 698 \def\BooleanFalse{TF}
\BooleanTrue 699 \def\BooleanTrue{TT}
\IfBooleanTF 700 \def\IfBooleanTF#1{%
701   \if#1%
702   \@xa\@firstoftwo
703   \else
704   \@xa\@secondoftwo
705   \fi
706 }
\IfBooleanT 708 \def\IfBooleanT#1#2{%
709   \IfBooleanTF{#1}{#2}\@empty
710 }
\IfBooleanF 712 \def\IfBooleanF#1{%
713   \IfBooleanTF{#1}\@empty
714 }
716 \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 \LaTeX 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 1707.

```

\ampulexdef 751 \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 [1707](#).

```

\gmu@tempa 766 \def\gmu@tempa{#5}%
\gmu@tempb 767 \def\gmu@tempb{#6}%
\gmu@tempc 768 \def\gmu@tempc{#7}% we wrap the start, end and replacement tokens in macros
              to avoid unbalanced \ifs.
770 \edef\gmu@tempd{%
771   \long\def\@nx\gmu@tempd
772   #####1\all@other\gmu@tempa
773   #####2\all@other\gmu@tempb
774   #####3\@nx\gmu@tempd{%
775     \@ifempty{#####3}{\hidden@iffalse}{\hidden@iftrue}}}%
777 \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.

781 \edef\gmu@tempe{%
782   \@nx\gmu@tempd\all@other#2%
783   \all@other\gmu@tempa
784   \all@other\gmu@tempb\@nx\gmu@tempd
785 }%
787 \gmu@tempe% we apply the checker and it produces an open \if.
789 \edef\gmu@tempd{%
790   \long\def\@nx\gmu@tempd
791   #####1\@xa\unexpanded\@xa{\gmu@tempa}%
792   #####2\@xa\unexpanded\@xa{\gmu@tempb}%
793   #####3\@nx\ampulexdef{% we define a temporary macro with the parameters
                          delimited with the 'start' and 'end' parameters of \ampulexdef.
796     \@nx\unexpanded{#####1}%
797     \@nx\@xa\@nx\unexpanded
798     \@nx\@xa{\@nx\gmu@tempc}% we replace the part of the redefined macro's
                          meaning with the replacement text.
800     \@nx\unexpanded{#####3}}}%
802 \gmu@tempd
805 \edef\gmu@tempf{#4}%
806 \edef\gmu@tempe{%
807   #1\def\@nx#2#3{%
808     \@xa\@xa\@xa\gmu@tempd\@xa#2\gmu@tempf\ampulexdef}}}%
809 \gmu@tempe
810 \fi}

\ampulexhash 812 \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 820 \def\gmshowlists{\showboxdepth=1000\showboxbreadth=1000\%
              \showlists}

\nameshow 824 \newcommand\nameshow[1]{\@xa\show\csname#1\endcsname}
\nameshowthe 825 \newcommand\nameshowthe[1]{\@xa\showthe\csname#1\endcsname}

```

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 836 \def\xiistring#1{%
            837   \if\@nx#1\xiispace
            838     \xiispace
            839   \else
            840     \string#1%
            841   \fi}

```

`\@ifnextcat`, `\@ifnextac`

As you guess, we `\def \@ifnextcat` à la `\@ifnextchar`, see L^AT_EX₂ ϵ 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_EXbook*, 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 858 \long\def\@ifnextcat#1#2#3{%
            862   \def\reserved@d{#1}%
            863   \def\reserved@a{#2}%
            864   \def\reserved@b{#3}%
            865   \futurelet\@let@token\@ifncat}

\@ifncat    868 \def\@ifncat{%
            869   \ifx\@let@token\@sptoken
            870     \let\reserved@c\@xifncat
            871   \else
            872     \ifcat\reserved@d\@nx\@let@token
            873       \let\reserved@c\reserved@a
            874     \else
            875       \let\reserved@c\reserved@b
            876     \fi
            877   \fi
            878   \reserved@c}

880 {\def\:\{\let\@sptoken=\ }\global\:\}% this makes \@sptoken a space token.
883 \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 894 \long\pdef\@ifnextif#1#2#3{%
            898   \def\reserved@d{#1}%
            899   \def\reserved@a{#2}%
            900   \def\reserved@b{#3}%
            901   \futurelet\@let@token\@ifnif}

\@ifnif    904 \def\@ifnif{%
            905   \ifx\@let@token\@sptoken
            906     \let\reserved@c\@xifnif

```

```

907 \else
908   \if\reserved@d\@nx\@let@token
909     \let\reserved@c\reserved@a
910   \else
911     \let\reserved@c\reserved@b
912   \fi
913 \fi
914 \reserved@c}
917 {\def\:{\let\@sptoken=\ }\:}% this makes \@sptoken a space token.
919 \def\:{\@xifnif}\@xa\gdef\:{\futurelet\@let@token\@ifnif}}

```

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

```

\@ifnextcat 930 \long\pdef\@ifnextac#1#2{%
931   \@ifnextcat\bgroup{#2}{\gm@ifnac{#1}{#2}}}
\gm@ifnac 933 \long\def\gm@ifnac#1#2#3{%
934   \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, `_10` that is. First define a cs `let` such a space. The assignment needs a little trick (*The T_EXbook* appendix D) since `\let`'s syntax includes one optional space after `=`.

```

947 \let\gmu@reserveda\*%
\* 948 \def\*{%
949   \let\*\gmu@reserveda
950   \let\gm@letspace=\}%
951 \*\%
\@ifnextspace 954 \def\@ifnextspace#1#2{%
955   \let\gmu@reserveda\*%
\* 956 \def\*{%
957   \let\*\gmu@reserveda
958   \ifx\@let@token\gm@letspace\afterfi{#1}%
959   \else\afterfi{#2}%
960   \fi}%
961 \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.

```

970 \foone\obeylines{%
\@ifnextMac 971 \long\pdef\@ifnextMac#1#2{%
972   \@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 `\fi . . .` as their arguments. If you know how many macros are there, you may type a couple of `\expandafters` and not to care how terrible it looks. But if you don't know how many tokens will there be, you seem to be in a real trouble. There's the Knuthian trick with `\next`. And here another, revealed to me by my \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`.

```
\afterfi 997 \long\def\afterfi#1#2\fi{\fi#1}
```

And two more of that family:

```
\afterfifi 999 \long\def\afterfifi#1#2\fi#3\fi{\fi\fi#1}
\afteriffifi 1001 \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 1005 \long\def\afteriffiffifi#1#2\fi#3\fi#4\fi{\fi#1}
\afteriffiffifi 1006 \long\def\afteriffiffifi#1#2\fi#3\fi#4\fi{\fi\fi#1}
\afteriffiffifi 1007 \long\def\afteriffiffifi#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\@endpfalse\reserved@a}
```

```
\@begnamedgroup 1039 \long\def\@begnamedgroup#1{%
1040   \@ignorefalse% not to ignore blanks after group
1041   \begingroup\@endpfalse
1042   \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 1087.
1046   \edef\@currenvline{\on@line}%
```

1047 \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 T_EX, this line will act as \relax.)

Let us make it the starred version of \begin.

```
\begin* 1056 \def\begin{\@ifstar{\@begnamedgroup}{%
\begin 1057   \@begnamedgroup@ifcs}}
\@begnamedgroup@ifcs 1060 \def\@begnamedgroup@ifcs#1{%
1061   \ifcsname#1\endcsname\afterfi{\@begnamedgroup{#1}}%
1062   \else\afterfi{\@latexerror{Environment_#1_undefined}\@eha}%
1063   \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^AT_EX 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, ₁, ₂ 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 1087 \long\def\@ifenvir#1#2#3{%
1089   \edef\gmu@reserveda{\@xa\string\csname\@currentvir\endcsname}%
1090   \edef\gmu@reservedb{\@xa\string\csname#1\endcsname}%
1091   \ifx\gmu@reserveda\gmu@reservedb\afterfi{#2}%
1092   \else\afterfi{#3}%
1093   \fi}
\@checkend 1095 \def\@checkend#1{\@ifenvir{#1}{\@badend{#1}}}
```

Thanks to it you may write \begin{macrocode*} with *₁₂ and end it with \end{% macrocode*} with *₁₁ (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^AT_EX 2_ε version of these macros was written by Donald Arseneau asnd@triumf.ca and Matt Swift swift@bu.edu after the L^AT_EX 2.09 smaller.sty style file written by Bernie Cosell cosell@WILMA.BBN.COM.

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

```
\relsize     You declare the font size with \relsize{<n>} where <n> gives the number of steps
("mag-step" = factor of 1.2) to change the size by. E.g., n = 3 changes from \normalsize
\smaller    to \LARGE size. Negative n selects smaller fonts. \smaller == \relsize{-1};
\larger     \larger == \relsize{1}. \smallerr(my addition) == \relsize{-2}; \largerr
\smallerr   guess yourself.
\largerr
```


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

```

\relsize 1135 \pdef\relsize#1{%
1136   \ifmmode_\@nomath\relsize\else
1137     \begingroup
1138     \@tempcnta_\% assign number representing current font size
1139     \ifx\@currsz\@normalsize_4\else_\% funny order is to have most
        ...
1140     \ifx\@currsz\small_3\else_\% ...likely sizes checked first
1141     \ifx\@currsz\footnotesize_2\else
1142     \ifx\@currsz\large_5\else
1143     \ifx\@currsz\Large_6\else
1144     \ifx\@currsz\LARGE_7\else
1145     \ifx\@currsz\scriptsize_1\else
1146     \ifx\@currsz\tiny_0\else
1147     \ifx\@currsz\huge_8\else
1148     \ifx\@currsz\Huge_9\else
1149       4\rs@unknown@warning_\% unknown state: \normalsize as
        starting point
1150     \fi\fi\fi\fi\fi\fi\fi\fi\fi
    Change the number by the given increment:
1152     \advance\@tempcnta#1\relax
    watch out for size underflow:
1154     \ifnum\@tempcnta<\z@\rs@size@warning{small}{\string\tiny}%
        \@tempcnta\z@\fi
1155     \@xa\endgroup
1156     \ifcase\@tempcnta_\% set new size based on altered number
1157     \tiny_\or_\scriptsize_\or_\footnotesize_\or_\small_\or_\%
        \normalsize_\or
1158     \large_\or_\Large_\or_\LARGE_\or_\huge_\or_\Huge_\else
1159     \rs@size@warning{large}{\string\Huge}\Huge
1160 \fi\fi}% end of \relsize.

\rs@size@warning 1162 \providecommand*\rs@size@warning[2]{\PackageWarning{gmutils_
    (relsize)}{%
1163   Size_requested_is_too_#1.\MessageBreak_Using_#2_instead}}

\rs@unknown@warning 1165 \providecommand*\rs@unknown@warning{\PackageWarning{gmutils_
    (relsize)}{Current_font_size
1166   is_unknown!(Why?!?)\MessageBreak_Assuming_\string\normalsize}}

    And a handful of shorthands:
    \larger 1170 \DeclareRobustCommand*\larger[1][\@ne]{\relsize{+#1}}
    \smaller 1171 \DeclareRobustCommand*\smaller[1][\@ne]{\relsize{-#1}}
    \textlarger 1172 \DeclareRobustCommand*\textlarger[2][\@ne]{\relsize{+#1}#2}}
    \textsmaller 1173 \DeclareRobustCommand*\textsmaller[2][\@ne]{\relsize{-#1}#2}}
    \largerr 1174 \pdef\largerr{\relsize{+2}}
    \smallerr 1175 \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`.

```

1185 \foone{\catcode`\_ =8_}%
\subs 1186 {\let\subs=_}

1188 \foone{\@makeother\_}%
\xiiunder 1189 {\def\xiiunder{_%}

1191 \ifdefined\XeTeXversion
\xiiunder 1192 \def\xiiunder{\char"005F_}%
1193 \let\_ \xiiunder
1194 \fi

1196 \foone{\catcode`\ [=1_ \@makeother\{
1197 \catcode`\ ] =2_ \@makeother\}}%
1198 [%
\xiilbrace 1199 \def\xiilbrace[{]%
\xiirbrace 1200 \def\xiirbrace[}%
1201 ]% of \firstofone

```

Note that L^AT_EX’s `\@charlb` and `\@charrb` are of catcode 11 (‘letter’), cf. The L^AT_EX 2_ε 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.

```

1212 \foone{\catcode`\_ =\active}
1213 {%
\smartunder 1214 \newcommand*\smartunder{%
1215 \catcode`\_ =\active
1216 \def_{\ifmmode\subs\else\_ \fi}}}% We define it as \_ not just as \xiiunder
because some font encodings don’t have _ at the \char`\_ position.

1222 \foone{\catcode`\! =o
1223 \@makeother\!}
\xiibackslash 1224 {\!newcommand*\xiibackslash{\!}}

\bslash 1228 \let\bslash=\xiibackslash

1232 \foone{\@makeother\}%
\xiipercen 1233 {\def\xiipercen{%%}

1236 \foone{\@makeother\&}%
\xiiand 1237 {\def\xiiand{&}}

1239 \foone{\@makeother\_}%
\xiispace 1240 {\def\xiispace{_%}

```

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

```

1249 \AtBeginDocument{%
1250 \ifdefined\xxt@visible
1251 \let\visible\xxt@visible
1252 \else

```

```

1253     \let\visiblepace\xiispace
1254     \fi}

```

Metasymbols

I fancy also another Knuthian trick for typesetting *metasymbols* in *The T_EXbook*. 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 1275 \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.”

```

1283     \ensuremath\langle
1284     \ifmmode\@xa\nfss@text\fi
1285     {%
1286     \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.

```

1294     #1\%/
1296   }\ensuremath\rangle
1297 }

```

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 1305 \def\meta@font@select{\it}

```

The below `\meta`’s drag¹ is a version of *The T_EXbook*’s one.

```

\<...> 1311 \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_EX primitive has three arguments: #1 ‘before break’, #2 ‘after break’, #3 ‘without break’, remember?

```

\discre 1322 \def\discre#1#2#3{\leavevmode\kernosp%
1323     \discretionary{#1}{#2}{#3}\penalty10000\hskiposp\relax}
\discret 1324 \def\discret#1{\leavevmode\kernosp%
1325     \discretionary{#1}{#1}{#1}\penalty10000\hskiposp\relax}

```

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

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

```
1329 \def\:{\ifmmode\afterfi{\mskip\medmuskip}\else\afterfi{\discret{%
    }}\fi}
```

```
\vs 1331 \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 1337 \def\printspaces#1{\let~=\vs\let\_\vs\gm@pswords#1\_\@nil}}
```

```
\gm@pswords 1339 \def\gm@pswords#1_\#2\_\@nil{%
```

```
1340 \ifx\relax#1\relax\else#1\fi
```

```
1341 \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 1346 \pdef\sfname#1{\textsf{\printspaces{#1}}}
```

```
\gm@discretionaryslash 1348 \def\gm@discretionaryslash{\discre{/}{\hbox{}}{/{/}}% the
second pseudo-argument nonempty to get \hyphenpenalty
not \exhyphenpenalty.
```

```
\file 1353 \pdef\file#1{\gm@printslashes#1/\gm@printslashes}
```

```
\gm@printslashes 1355 \def\gm@printslashes#1/#2\gm@printslashes{%
```

```
1356 \sfname{#1}%
```

```
1357 \ifx\gm@printslashes#2\gm@printslashes
```

```
1358 \else
```

```
1359 \textsf{\gm@discretionaryslash}%
```

```
1360 \afterfi{\gm@printslashes#2\gm@printslashes}\fi}
```

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

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

```
\pk 1368 \pdef\pk#1{\textsf{\textup{#1}}}
```

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

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

```
\cs 1379 \DeclareRobustCommand*\cs[2][\bslash]{\%
```

```
\- 1380 \def\-\{\discretionary{\rmfamily-}{\}{\}}%
```

```
1381 \def\{\{\char`\}\}\def\{\{\char`\}\}\ttfamily_\#1#2}}
```

```
\env 1385 \pdef\env#1{\cs[] {#1}}
```

And for the special sequences like ^~A:

```
1388 \foone{\@makeother\^}
```

```
\hathat 1389 {\pdef\hathat#1{\cs[^^] {#1}}}
```

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

```
\possfil 1394 \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 1404 \def\cmd#1{\cs{\@xa\cmd@to\cs\string#1}}
```

```

\cmd@to@cs 1406 \def\cmd@to@cs#1#2{\char\number`#2\relax}
               \marg{text} prints {\text}, ‘mandatory argument’.

\marg 1410 \def\marg#1{\tfamily\char`\{ }\meta{#1}{\tfamily\char`\}}
               \oarg{text} prints [{text}], ‘optional argument’. Also \oarg{text} does that.

\oarg 1415 \def\oarg{\@ifnextchar[\@oargsq\@oarg}
\@oarg 1417 \def\@oarg#1{\tfamily[]\meta{#1}{\tfamily}}
\@oargsq 1418 \def\@oargsq[#1]{\@oarg{#1}}
               \parg{te,xt} prints (<te,xt>), ‘picture mode argument’.

\parg 1422 \def\parg{\@ifnextchar(\@pargp\@parg}
\@parg 1424 \def\@parg#1{\tfamily{}\meta{#1}{\tfamily}}
\@pargp 1425 \def\@pargp(#1){\@parg{#1}}

```

But we can have all three in one command.

```

1429 \AtBeginDocument{%
\arg 1430 \let\math@arg\arg
\arg 1431 \def\arg{\ifmmode\math@arg\else\afterfi{%
1432 \ifnextchar[%
1433 \oargsq{\ifnextchar(
1434 \pargp\marg}}\fi}%
1435 }

```

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 1450 \newif\ifgmu@SMglobal
\SMglobal 1452 \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 1463 \pdef\StoreMacro{%
1464 \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 1587.

```

\egStore@Macro 1469 \long\def\egStore@Macro#1{\endgroup\Store@Macro{#1}}
\egStore@MacroSt 1470 \long\def\egStore@MacroSt#1{\endgroup\Store@MacroSt{#1}}
\Store@Macro 1472 \long\def\Store@Macro#1{%
1473 \escapechar92
1474 \ifgmu@SMglobal\afterfi\global\fi

```

```

1475 \@xa\let\csname_/gmu/store/string#1\endcsname#1%
1476 \global\gmu@SMglobalfalse}
\Store@MacroSt 1479 \long\def\Store@MacroSt#1{%
1480 \edef\gmu@smtempa{%
1481 \ifgmu@SMglobal\global\fi
1482 \@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.
1488 \@xa\@nx\csname#1\endcsname}
1489 \gmu@smtempa
1490 \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_#1\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 1506 \long\pdef\StoreMacros{\begingroup\makeatletter\Store@Macros}
\Store@Macros 1507 \long\def\Store@Macros#1{\endgroup
1508 \gmu@setsetSMglobal
1509 \let\gml@storeCS\Store@Macro
1510 \gml@storemacros#1.}
\gmu@setsetSMglobal 1513 \def\gmu@setsetSMglobal{%
1514 \ifgmu@SMglobal
1515 \let\gmu@setSMglobal\gmu@SMglobaltrue
1516 \else
1517 \let\gmu@setSMglobal\gmu@SMglobalfalse
1518 \fi}

```

And the inner iterating macro:

```

\gml@storemacros 1521 \long\def\gml@storemacros#1{%
\gmu@reserveda 1522 \def\gmu@reserveda{\@nx#1}% My TEX Guru's trick to deal with \fi and such,
        i.e., to hide #1 from TEX when it is processing a test's branch without expand-
        ing.
1525 \if\gmu@reserveda.% a dot finishes storing.
1526 \global\gmu@SMglobalfalse
1527 \else
1528 \if\gmu@reserveda,% The list this macro is put before may contain commas
        and that's O.K., we just continue the work.
1530 \afterfifi\gml@storemacros
1531 \else% what is else this shall be stored.
1532 \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.
1535 \afterfifi{\gmu@setSMglobal\gml@storemacros}%
1536 \fi
1537 \fi}

```

And for the restoring

```

\RestoreMacro 1543 \pdef\RestoreMacro{%

```

```

1544 \begingroup\makeatletter\@ifstar\egRestore@MacroSt%
      \egRestore@Macro}
\egRestore@Macro 1546 \long\def\egRestore@Macro#1{\endgroup\Restore@Macro{#1}}
\egRestore@MacroSt 1547 \long\def\egRestore@MacroSt#1{\endgroup\Restore@MacroSt{#1}}
\Restore@Macro 1549 \long\def\Restore@Macro#1{%
1550 \escapechar92
1551 \ifgmu@SMglobal\afterfi\global\fi
1552 \@xa\let\@xa#1\csname_/gmu/store/string#1\endcsname
1553 \global\gmu@SMglobalfalse}
\Restore@MacroSt 1555 \long\def\Restore@MacroSt#1{%
1556 \edef\gmu@smtmpa{%
1557 \ifgmu@SMglobal\global\fi
1558 \@nx\let\@xa\@nx\csname#1\endcsname
1559 \@xa\@nx\csname/gmu/store/bslash#1\endcsname}% cf. the commentary
      in line 1482.
1561 \gmu@smtmpa
1562 \global\gmu@SMglobalfalse}
\RestoreMacros 1565 \long\pdef\RestoreMacros{\begingroup\makeatletter\Restore@Macros}
\Restore@Macros 1567 \long\def\Restore@Macros#1{\endgroup
1568 \gmu@setsetSMglobal
1569 \let\gml@storeCS\Restore@Macro% we direct the core cs towards restoring
      and call the same iterating macro as in line 1510.
1572 \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 1578 \def\StoredMacro{\begingroup\makeatletter\Stored@Macro}
\Stored@Macro 1579 \long\def\Stored@Macro#1{\endgroup\Restore@Macro#1#1}

```

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

```

\storedcsname 1582 \def\storedcsname#1{%
1583 \csname_/gmu/store/bslash#1\endcsname}

```

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

```

\StoreEnvironment 1587 \pdef\StoreEnvironment#1{%
1589 \StoreMacro*{#1}\StoreMacro*{end#1}}
\RestoreEnvironment 1591 \pdef\RestoreEnvironment#1{%
1593 \RestoreMacro*{#1}\RestoreMacro*{end#1}}

```

It happended (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 1608 \long\def\StoringAndRelaxingDo{%
1609 \gmu@SMdo@setscope
1610 \long\def\do##1{%
1611 \gmu@SMdo@scope
1612 \@xa\let\csname_/gmu/store/string##1\endcsname##1%
1613 \gmu@SMdo@scope\let##1\relax}}

```

```

\gmu@SMdo@setscope 1615 \def\gmu@SMdo@setscope{%
1616   \ifgmu@SMglobal\let\gmu@SMdo@scope\global
1617   \else\let\gmu@SMdo@scope\relax
1618   \fi
1619   \global\gmu@SMglobalfalse}

```

And here is the counter-definition for restore.

```

\RestoringDo 1628 \long\def\RestoringDo{%
1629   \gmu@SMdo@setscope
1630   \long\def\do##1{%
1631     \gmu@SMdo@scope
1632     \@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 1640 \def\n@melet#1#2{%
1641   \edef\gmu@nl@reserveda{%
1642     \let\@xa\@nx\csname#1\endcsname
1643     \@xa\@nx\csname#2\endcsname}%
1644   \gmu@nl@reserveda}

```

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

```

\gn@melet 1648 \def\gn@melet#1#2{%
1649   \edef\gmu@nl@reserveda{%
1650     \global\let\@xa\@nx\csname#1\endcsname
1651     \@xa\@nx\csname#2\endcsname}%
1652   \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 1669 \newcommand\not@onlypreamble[1]{\{%
1670   \def\do##1{\ifx##1\else\@nx\do\@nx##1\fi}%
1671   \xdef\@preamblecmds{\@preamblecmds}}
1673 \not@onlypreamble\@preamblecmds
1674 \not@onlypreamble\@ifpackageloaded
1675 \not@onlypreamble\@ifclassloaded
1676 \not@onlypreamble\@ifl@aded
1677 \not@onlypreamble\@pkgextension

```

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

```

\gm@notprerr 1682 \def\gm@notprerr{\can be used only in preamble(\on@line)}
1684 \AtBeginDocument{%
1685   \def\do#1{\@nx\do\@nx#1}%
1686   \edef\@preamblecmds{%
1687     \def\@nx\do##1{%

```



```

1688 \def##1{\@nx\PackageError{gmutils/LaTeX}%
1689 {\@nx\string##1_\@nx\gm@notprerr}\@nx\@eha}}%
1690 \@preamblecmds}}

```

A subtle error raises: the L^AT_EX 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^AT_EX’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 1707 \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.
1714 \ampulexdef[]\nocite[####1][{{####1}}]% note the double brace around
% #3.
1716 \ifx
1717 {\@onlypreamble\document}%
1718 \iftrue}
1720 \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 1747 \newcounter{gm@PronounGender}
\gm@atppron 1749 \newcommand*\gm@atppron[2]{%
1750 \stepcounter{gm@PronounGender}% remember \stepcounter is global.
1751 \ifodd\value{gm@PronounGender}#1\else#2\fi}
\heshe 1753 \newcommand*\heshe{\gm@atppron{he}{she}}
\hisher 1754 \newcommand*\hisher{\gm@atppron{his}{her}}
\himher 1755 \newcommand*\himher{\gm@atppron{him}{her}}
\hishers 1756 \newcommand*\hishers{\gm@atppron{his}{hers}}
\HeShe 1758 \newcommand*\HeShe{\gm@atppron{He}{She}}
\HisHer 1759 \newcommand*\HisHer{\gm@atppron{His}{Her}}
\HimHer 1760 \newcommand*\HimHer{\gm@atppron{Him}{Her}}
\HisHers 1761 \newcommand*\HisHers{\gm@atppron{His}{Hers}}

```

Improvements to mwcls sectioning commands

That is, ‘Expe-ri-mente’² 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.

1780 \AtBeginDocument{% because we don’t know when exactly hyperref is loaded and maybe after this package.

NoNumSecs 1782 \@ifpackageloaded{hyperref}{\newcounter{NoNumSecs}%

1783 \setcounter{NoNumSecs}{617}% to make \refing to an unnumbered section visible (and funny?).

1785 \def\gm@hyperrefstepcounter{\refstepcounter{NoNumSecs}}%

1786 \pdef\gm@targetheading#1{%

1787 \hypertarget{#1}{#1}}}% end of then

1788 {\def\gm@hyperrefstepcounter{}%

1789 \def\gm@targetheading#1{#1}}}% end of else

1790 }% of \AtBeginDocument

Auxiliary macros for the kernel sectioning macro:

1793 \def\gm@dontnumbersectionsoutofmainmatter{%

1794 \if@mainmatter\else\HeadingNumberedfalse\fi}

1795 \def\gm@clearpagesduetoopenright{%

1796 \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 1809 \long\def\@ifnotmw#1#2{\gm@ifundefined{mw@sectionxx}{#1}{#2}}

The kernel of MW’s sectioning commands:

1834 \@ifnotmw{}{%

\mw@sectionxx 1835 \def\mw@sectionxx#1#2[#3]#4{%

1836 \edef\mw@HeadingLevel{\csname#1@level\endcsname

1837 \space}% space delimits level number!

1838 \ifHeadingNumbered

1839 \ifnum\mw@HeadingLevel>\c@secnumdepth%

\HeadingNumberedfalse\fi

line below is in \gm@ifundefined to make it work in classes other than mwbk

1842 \gm@ifundefined{if@mainmatter}{%}

\gm@dontnumbersectionsoutofmainmatter}

1843 \fi

% \ifHeadingNumbered

% \refstepcounter{#1}%

² A. Berg, Wozzeck.

```

%      \protected@edef\HeadingNumber{\csname
%      the#1\endcsname\relax}%
%      \else
%      \let\HeadingNumber\@empty
%      \fi

\HeadingRHeadText 1852 \def\HeadingRHeadText{#2}%
\HeadingTOCText    1853 \def\HeadingTOCText{#3}%
\HeadingText       1854 \def\HeadingText{#4}%
\mw@HeadingType    1855 \def\mw@HeadingType{#1}%
1856 \if\mw@HeadingBreakBefore
1857   \if@specialpage\else\thispagestyle{closing}\fi
1858   \gm@ifundefined{if@openright}{\fi
%      \gm@clearpagesduetoopenright}%
1859   \if\mw@HeadingBreakAfter
1860     \thispagestyle{blank}\else
1861     \thispagestyle{opening}\fi
1862     \global\@topnum\z@
1863   \fi% of \if\mw@HeadingBreakBefore
placement of \refstep suggested by me (GM):
1866   \ifHeadingNumbered
1867     \refstepcounter{#1}%
1868     \protected@edef\HeadingNumber{\csname the#1\endcsname\relax}%
1869   \else
1870     \let\HeadingNumber\@empty
1871     \gm@hyperrefstepcounter
1872   \fi% of \ifHeadingNumbered
1874   \if\mw@HeadingRunIn
1875     \mw@runinheading
1876   \else
1877     \if\mw@HeadingWholeWidth
1878       \if@twocolumn
1879         \if\mw@HeadingBreakAfter
1880         \onecolumn
1881         \mw@normalheading
1882         \pagebreak\relax
1883         \if@twoside
1884           \null
1885           \thispagestyle{blank}%
1886           \newpage
1887         \fi% of \if@twoside
1888       \twocolumn
1889     \else
1890       \@topnewpage[\mw@normalheading]%
1891     \fi% of \if\mw@HeadingBreakAfter
1892   \else
1893     \mw@normalheading
1894     \if\mw@HeadingBreakAfter\pagebreak\relax\fi
1895   \fi% of \if@twocolumn
1896 \else
1897   \mw@normalheading
1898   \if\mw@HeadingBreakAfter\pagebreak\relax\fi
1899 \fi% of \if\mw@HeadingWholeWidth

```

```

1900 \fi% of \if\mw@HeadingRunIn
1901 }

```

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

```

1925 \relaxen\SetSectionFormatting
\SetSectionFormatting 1926 \newcommand*\SetSectionFormatting[5][\empty]{%
1927 \ifx\empty#1\relax\else% empty (not \empty!) #1 also launches \else.
1928 \def\mw@HeadingRunIn{10}\def\mw@HeadingBreakBefore{10}%
1929 \def\mw@HeadingBreakAfter{10}\def\mw@HeadingWholeWidth{10}%
1930 \@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.
1933 \fi
1934 \gm@ifundefined{#2}{\@namedef{#2}{\mw@section{#2}}}{}%
1935 \mw@secdef{#2}{@preskip}{#3}{2\oblig.}%
1936 \mw@secdef{#2}{@head}{#4}{3\oblig.}%
1937 \mw@secdef{#2}{@postskip}{#5}{4\oblig.}%
1938 \ifx\empty#1\relax
1939 \mw@secundef{#2@flags}{1\optional)}%
1940 \else\mw@setflags{#2}%
1941 \fi}
\mw@secdef 1943 \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.
1950 \@ifempty{#3}
1951 {\mw@secundef{#1#2}{#4}}
1952 {\@namedef{#1#2}{#3}}}
\mw@secundef 1954 \def\mw@secundef#1#2{%
1955 \gm@ifundefined{#1}{%
1956 \ClassError{mwcls/gm}{%
1957 command\backslash#1\undefined\MessageBreak
1958 after\backslashSetSectionFormatting!!!\MessageBreak}{%
1959 Provide the #2 argument of \backslash
SetSectionFormatting.}}{}}
First argument is a sectioning command (wo. the backslash) and second the stuff to be
added at the beginning of the heading declarations.
\addtoheading 1964 \def\addtoheading#1#2{%
1965 \n@melet{gmu@reserveda}{#1@head}%

```

```

1966 \edef\gmu@reserveda{\unexpanded{#2}\@xa\unexpanded{%
      \gmu@reserveda}}}%
1967 \n@melet{#1@head}{\gmu@reserveda}%
1969 }
1971 }% 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.

```

1983 \@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 1988 \def\@afterheading{%
1989   \@nobreaktrue
1990   \xdef\gmu@prevsec{\mw@HeadingType}% added now
1991   \everypar{%
1992     \grelaxen\gmu@prevsec% added now. All the rest is original LATEX.
1993     \if@nobreak
1994     \@nobreakfalse
1995     \clubpenalty_\@M
1996     \if@afterindent_\else
1997     {\setbox\z@\lastbox}%
1998     \fi
1999     \else
2000     \clubpenalty_\@clubpenalty
2001     \everypar{}%
2002     \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 2009 \def\gmu@checkaftersec{%
2010   \gmu@ifundefined{gmu@prevsec}{}{%
2011     \ifgmu@postsec% an additional switch that is true by default but may be
      turned into an \ifdim in special cases, see line 2047.
2014     {\@xa\mw@getflags\@xa{\gmu@prevsec}%
2015     \glet\gmu@reserveda\mw@HeadingBreakAfter}%
2016     \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.
2019     \if\gmu@reserveda\else
2020     \penalty10000\relax
2021     \skip\z@=\csname\gmu@prevsec_\@postskip\endcsname\relax
2022     \skip\tw@=\csname\mw@HeadingType_\@preskip\endcsname\relax
2023     \gmu@ifundefined{\mw@HeadingType_\@twoheadskip}{%

```

```

2024 \ifdim\skip\z@>\skip\tw@
2025 \vskip-\skip\z@% we strip off the post-skip of previous header if it's bigger
      than current pre-skip
2027 \else
2028 \vskip-\skip\tw@% we strip off the current pre-skip otherwise
2029 \fi}{% But if the two-header-skip is defined, we put it
2031 \penalty10000
2032 \vskip-\skip\z@
2033 \penalty10000
2034 \vskip-\skip\tw@
2035 \penalty10000
2036 \vskip\csname\mw@HeadingType_\@twoheadskip\endcsname
2037 \relax}%
2038 \penalty10000
2039 \hrule\height\z@\relax% to hide the last (un)skip before
      subsequent \addvspaces.
2041 \penalty10000
2042 \fi
2043 \fi
2044 }% of \gm@ifundefined{gmu@prevsec} 'else'.
2045 }% of \def\gmu@checkaftersec.
\ParanoidPostsec 2047 \def\ParanoidPostsec{% this version of \ifgmu@postsec is intended for the spe-
      cial case of sections may contain no normal text, as while gmdocing.
\ifgmu@postsec 2050 \def\ifgmu@postsec{% note this macro expands to an open \if.
2051 \skip\z@=\csname\gmu@prevsec_\@postskip\endcsname\relax
2052 \ifdim\lastskip=\skip\z@\relax% we play with the vskip only if the last
      skip is the previous heading's postskip (a counter-example I met while
      gmdocing).
2056 }}
2058 \let\ifgmu@postsec\iftrue
\gmu@getaddvs 2060 \def\gmu@getaddvs#1\addvspace#2\gmu@getaddvs{%
2061 \toks\z@={#1}%
2062 \toks\tw@={#2}}
      And the modification of the inner macros at last:
\gmu@setheading 2065 \def\gmu@setheading#1{%
2066 \@xa\gmu@getaddvs#1\gmu@getaddvs
2067 \edef#1{%
2068 \the\toks\z@\@nx\gmu@checkaftersec
2069 \@nx\addvspace\the\toks\tw@}}
2071 \gmu@setheading\mw@normalheading
2072 \gmu@setheading\mw@runinheading
\SetTwoheadSkip 2074 \def\SetTwoheadSkip#1#2{\@namedef{#1@twoheadskip}{#2}}
2076 }% 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.

```

2088 \@ifnotmw{}{% We define this declaration only when in mwcls.
\WPheadings 2089 \def\WPheadings{%
2090   \SetSectionFormatting[breakbefore,wholewidth]
2091     {part}{\z@\@plus1fill}{\z@\@plus3fill}%
2093   \gm@ifundefined{chapter}{}{%
2094     \SetSectionFormatting[breakbefore,wholewidth]
2095       {chapter}
2096       {66\p@}% {67\p@} for Adventor/Schola 0,95.
2097       {\FormatHangHeading{\LARGE}}
2098       {27\p@\@plus0,2\p@\@minus1\p@}%
2099   }%
2101   \SetTwoheadSkip{section}{27\p@\@plus0,5\p@}%
2102   \SetSectionFormatting{section}
2103     {24\p@\@plus0,5\p@\@minus5\p@}%
2104     {\FormatHangHeading{\Large}}
2105     {10\p@\@plus0,5\p@}% ed. Krajewska of "Wiedza Powszechna", as we un-
        understand her, wants the skip between a heading and text to be rigid.
2109   \SetTwoheadSkip{subsection}{11\p@\@plus0,5\p@\@minus1\p@}%
2110   \SetSectionFormatting{subsection}
2111     {19\p@\@plus0,4\p@\@minus6\p@}
2112     {\FormatHangHeading{\large}}% 12/14 pt
2113     {6\p@\@plus0,3\p@}% after-skip 6 pt due to p.12, not to squeeze the before-
        skip too much.
2116   \SetTwoheadSkip{subsubsection}{10\p@\@plus1,75\p@\@minus1\p@}%
2117   \SetSectionFormatting{subsubsection}
2118     {10\p@\@plus0,2\p@\@minus1\p@}
2119     {\FormatHangHeading{\normalsize}}
2120     {3\p@\@plus0,1\p@}% those little skips should be smaller than you calcu-
        late out of a geometric progression, because the interline skip enlarges
        them.
2124   \SetSectionFormatting[runin]{paragraph}
2125     {7\p@\@plus0,15\p@\@minus1\p@}
2126     {\FormatRunInHeading{\normalsize}}
2127     {2\p@}%
2129   \SetSectionFormatting[runin]{subparagraph}
2130     {4\p@\@plus1\p@\@minus0,5\p@}
2131     {\FormatRunInHeading{\normalsize}}
2132     {\z@}%
2133 }% of \WPheadings
2134 }% 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.)

```

2175 \@ifnotmw{% we are not in mwcls and want to handle mwcls-like sectionings i.e.,
      those written with two optionals.
\gm@secini 2178 \def\gm@secini{\gm@la}%
\gm@secxx 2180 \def\gm@secxx#1#2[#3]#4{%
      2181 \ifx\gm@secstar\@empty
      2182 \n@melet{\gm@true@#1mark}{#1mark}% a little trick to allow a special ver-
              sion of the heading just to the running head.
      2184 \@namedef{#1mark}##1{% we redefine \<sec>mark to gobble its argument
              and to launch the stored true marking command on the appropriate
              argument.
      2187 \csname_\gm@true@#1mark\endcsname{#2}%
      2188 \n@melet{#1mark}{\gm@true@#1mark}% after we've done what we
              wanted we restore original \#1mark.
      2190 }%
\gm@secstar 2191 \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.
      2196 \fi
      2197 \@xa\@xa\csname\gm@secini#1\endcsname
      2198 \gm@secstar{#4}}%
2200 }{% 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 2203 \def\gm@secini{\gm@mw}%
      2205 \let\gm@secmarkh\@gobble% in mwcls there's no need to make tricks for special
              version to running headings.
\gm@secxx 2208 \def\gm@secxx#1#2[#3]#4{%
      2209 \@xa\@xa\csname\gm@secini#1\endcsname
      2210 \gm@secstar[#2][#3]{#4}}%
      2211 }
\gm@sec 2213 \def\gm@sec#1{\@dblarg{\gm@secx{#1}}}
\gm@secx 2214 \def\gm@secx#1[#2]{%
      2215 \@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 2219 \def\gm@straightensec#1{% the parameter is for the command's name.
      2220 \gm@ifundefined{#1}{\@}% we don't change the ontological status of the com-
              mand because someone may test it.

```



```

2222 \n@melet{\gm@secini#1}{#1}%
2223 \@namedef{#1}{%
\gm@secstar 2224 \@ifstar{\def\gm@secstar{*}\gm@sec{#1}}{%
\gm@secstar 2225 \def\gm@secstar{}\gm@sec{#1}}}%
2226 }%

2228 \let\do\gm@straightensec
2229 \do{part}\do{chapter}\do{section}\do{subsection}\do{%
subsubsection}
2230 \@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^AT_EX’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* 2246 \@namedef{enumerate*}{%
2247 \ifnum\@enumdepth>\thr@@
2248 \toodeep
2249 \else
2250 \advance\@enumdepth\@ne
2251 \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
2252 \@xa\list\csname\label\@enumctr\endcsname{%
2253 \partopsep\topsep\topsep\z@\leftmargin\z@
2254 \itemindent\@parindent\advance\itemindent\labelsep
2255 \labelwidth\@parindent
2256 \advance\labelwidth-\labelsep
2257 \listparindent\@parindent
2258 \usecounter\@enumctr
2259 \def\makelabel##1{##1\hfil}}%
2260 \fi}
2261 \@namedef{endenumerate*}{\endlist}

itemize* 2264 \@namedef{itemize*}{%
2265 \ifnum\@itemdepth>\thr@@
2266 \toodeep
2267 \else
2268 \advance\@itemdepth\@ne
2269 \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
2270 \@xa\list\csname\@itemitem\endcsname{%
2271 \partopsep\topsep\topsep\z@\leftmargin\z@
2272 \itemindent\@parindent
2273 \labelwidth\@parindent
2274 \advance\labelwidth-\labelsep
2275 \listparindent\@parindent
2276 \def\makelabel##1{##1\hfil}}%
2277 \fi}
2278 \@namedef{enditemize*}{\endlist}

```

The logos

We'll modify The L^AT_EX logo now to make it fit better to various fonts.

```

2287 \let\oldLaTeX\LaTeX
2288 \let\oldLaTeXe\LaTeXe
2290 \def\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}
\DeclareLogo 2292 \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 2300 \ifx\relax#1\def\gmu@reserveda{\@xa\@gobble\string#2}%
2301 \else
\gmu@reserveda 2302 \def\gmu@reserveda{#1}%
2303 \fi
2304 \edef\gmu@reserveda{%
2305 \@nx\DeclareTextCommand\@nx#2{PD1}{\gmu@reserveda}}
2306 \gmu@reserveda
2307 \DeclareTextCommandDefault#2{#3}%
\pdef 2308 \pdef#2{#3}}% added for XYLATEX
\DeclareLogo 2311 \DeclareLogo\LaTeX{%
2312 {%
2313 L%
2314 \setbox\z@\hbox{\check@mathfonts
2315 \fontsize\sf@size\z@
2316 \math@fontsfalse\selectfont
2317 A}%
2318 \kern-.57\wd\z@
2319 \sbox\tw@T%
2320 \vbox\t@to\ht\tw@{\copy\z@\vss}%
2321 \kern-.2\wd\z@}% originally -, 15 em for T.
2322 {%
2323 \ifdim\fontdimen1\font=\z@
2324 \else
2325 \count\z@=\fontdimen5\font
2326 \multiply\count\z@by64\relax
2327 \divide\count\z@by\p@
2328 \count\tw@=\fontdimen1\font
2329 \multiply\count\tw@by\count\z@
2330 \divide\count\tw@by64\relax
2331 \divide\count\tw@by\tw@
2332 \kern-\the\count\tw@sp\relax
2333 \fi}%
2334 \TeX}
2335
\LaTeXe 2337 \DeclareLogo\LaTeXe{\mbox{\math@if
2338 b\expandafter\@car\fontseries\@nil\boldmath\fi
2339 \LaTeX\kern.15em2$_{\textstyle\varepsilon}$}}
2341 \StoreMacro\LaTeX
2342 \StoreMacro*\LaTeX_

```

'(L^A)T_EX' in my opinion better describes what I work with/in than just 'L^AT_EX'.

```

\LaTeXpar 2348 \DeclareLogo[(La)TeX]{\LaTeXpar}{%
2349   {%
2350     \setbox\z@\hbox{({}%
2351     \copy\z@
2352     \kern-.2\wd\z@_L%
2353     \setbox\z@\hbox{\check@mathfonts
2354       \fontsize\sf@size\z@
2355       \math@fontsfalse\selectfont
2356       A}%
2357     \kern-.57\wd\z@
2358     \sbox\tw@_T%
2359     \vbox_\to\ht\tw@{\box\z@%
2360       \vss}%
2361   }%
2362   \kern-.07em% originally -, 15 em for T.
2363   {% (
2364     \sbox\z@)%
2365     \kern-.2\wd\z@\copy\z@
2366     \kern-.2\wd\z@}\TeX
2367 }

```

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

```

2374 \gm@ifundefined{AmSTeX}
\AmSTeX 2375   {\def\AmSTeX{\leavevmode\hbox{${\mathcal\_A}\kern-.2em%
2376     \lower.376ex%
2377     \hbox{${\mathcal\_M}$}\kern-.2em\mathcal\_S$-\TeX}}}{%
\BibTeX 2378 \DeclareLogo\BibTeX{{\rmfamily\_B\kern-.05em%
2379   \textsc{i{\kern-.025em}b}\kern-.08em% the kern is wrapped in braces
2380   for my \fakescaps’ sake.
2381   \TeX}}
\SLiTeX 2384 \DeclareLogo\SLiTeX{{\rmfamily\_S\kern-.06emL\kern-.18em%
2385   \raise.32ex\hbox
2386   {\scshape\_i}\kern_\-.03em\TeX}}
\PlainTeX 2387 \DeclareLogo\PlainTeX{\textsc{Plain}\kern2pt\TeX}
\Web 2389 \DeclareLogo\Web{\textsc{Web}}

There’s also the  $\text{\LaTeX}$  logo got with the  $\text{\LaTeXpar}$  macro provided by gmutils. And
here The  $\text{\TeX}$ book’s logo:
\TeXbook 2392 \DeclareLogo[The\_TeX\_book]\TeXbook{\textsl{The\_TeX\_book}}
2393 \let\TB\TeXbook% TUG Boat uses this.
\TeX 2395 \DeclareLogo[e-TeX]\eTeX{%
2396   \ensuremath{\varepsilon}-\kern-.125em\TeX}% definition sent by Karl Berry
2397   from TUG Boat itself.
\pdfTeX 2399 \DeclareLogo[pdfe-TeX]\pdfTeX{pdf\eTeX}
\pdfTeX 2401 \DeclareLogo\pdfTeX{pdf\TeX}
2402 \gm@ifundefined{XeTeX}{%
\XeTeX 2404   \DeclareLogo\XeTeX{X\kern-.125em\relax
2405   \gm@ifundefined{reflectbox}{%

```

```

2406      \lower.5ex\hbox{E}\kern-.1667em\relax}{%
2407      \lower.5ex\hbox{\reflectbox{E}}\kern-.1667em\relax}%
2408      \TeX}}{}
2410 \gm@ifundefined{XeLaTeX}{%
\XeLaTeX 2411   \DeclareLogo\XeLaTeX{X\kern-.125em\relax
2412   \gm@ifundefined{reflectbox}{%
2413   \lower.5ex\hbox{E}\kern-.1667em\relax}{%
2414   \lower.5ex\hbox{\reflectbox{E}}\kern-.1667em\relax}%
2415   \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.

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 cs: 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 2440 \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:

```

2445 \edef\gmu@tempa{%
\gm@gobmacro 2446   \def\@nx\gm@gobmacro##1\@xa\@gobble\string\macro:##2->{}}
2447 \gmu@tempa

```

Brave New World of \XeTeX

```

\ifXeTeX 2464 \newcommand\@ifXeTeX[2]{%
2465   \ifdefined\XeTeXversion
2466   \unless\ifx\XeTeXversion\relax\afterfifi{#1}\else\afterfifi{%
      #2}\fi
2467   \else\afterfi{#2}\fi}
\XeTeXthree 2470 \def\XeTeXthree{%
2472   \@ifXeTeX{%
2473     \@ifpackageloaded{gmverb}{\StoreMacro\verb}{}}%
2474     \RequirePackage{xltxtra}% since v 0.4 (2008/07/29) this package redef-
      ines \verb and verbatim*, and quite elegantly provides an option to
      suppress the redefinitions, but unfortunately that option excludes also
      a nice definition of \xxt@visible space which I fancy.
2481     \@ifpackageloaded{gmverb}{\RestoreMacro\verb}{}}%
2482     \AtBeginDocument{%
2483       \RestoreMacro\LaTeX\RestoreMacro*\LaTeX_{}}% my version of the
      LATEX logo has been stored just after defining, in line 2342.
2486   }{}}

```

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

```

2491 \AtBeginDocument{%
2492   \@ifpackageloaded{fontspec}{%
\udigits 2493     \pdf\udigits{%
2494       \addfontfeature{Numbers=Uppercase}}%
2495     }{%
2496       \emptyify\udigits}}

```

Fractions

```

\Xedekfracc 2501 \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_EX 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 2516 \def\gmu@xedekfraccstar{%
\gmu@xefracccdef 2517   \def\gmu@xefracccdef##1##2{%
2518     \iffontchar\font_##2
2519     \@namedef{gmu@xefraccc##1}{\char##2_}%
2520     \else
2521     \n@melet{gmu@xefraccc##1}{relax}%
2522     \fi}%
\gmu@dekfracc 2523   \def\gmu@dekfracc##1/##2{%
2524     {\addfontfeature{VerticalPosition=Numerator}##1}%
2525     \gmu@numeratorkern
2526     \char"2044_\gmu@denominatorkern
2527     {\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.

```

2531   \gmu@xefracccdef{1/4}{"BC}%
2532   \gmu@xefracccdef{1/2}{"BD}%
2533   \gmu@xefracccdef{3/4}{"BE}%
2534   \gmu@xefracccdef{1/3}{"2153}%
2535   \gmu@xefracccdef{2/3}{"2154}%
2536   \gmu@xefracccdef{1/8}{"215B}%
2537   \gmu@xefracccdef{3/8}{"215C}%
2538   \gmu@xefracccdef{5/8}{"215D}%
2539   \gmu@xefracccdef{7/8}{"215E}%
\dekfracc@args 2540 \pdf\dekfracc@args##1/##2{%
\gm@duppa 2541   \def\gm@duppa{##1/##2}%
2542   \gm@ifundefined{gmu@xefraccc}{all@other\gm@duppa}{%
2543     \gmu@dekfracc{##1}/{##2}}{%
2544     \csname_\gmu@xefraccc\all@other\gm@duppa\endcsname}%
2545   \if@gmu@mmhbox\egroup\fi
2546   }% of \dekfracc@args.
2547   \@ifstar{\let\gmu@dekfracc\gmu@dekfracccsimple}{}%

```

```

2548 }
\gmu@xedekfraccplain 2550 \def\gmu@xedekfraccplain{% 'else' of the main \@ifstar
\dekfracc@args 2551 \pdef\dekfracc@args##1/##2{%
2552 \ifmmode\hbox\fi{%
2553 \addfontfeature{Fractions=On}%
2554 ##1/##2}%
2555 \if@gmu@mmhbox\egroup\fi
2556 }% of \dekfracc@args
2557 }
\if@gmu@mmhbox 2559 \newif\if@gmu@mmhbox% we'll use this switch for \dekfracc and also for \thous
(hacky thousand separator).
\dekfracc 2562 \pdef\dekfracc{%
2564 \ifmmode\hbox\bgroup@gmu@mmhboxtrue\fi
2565 \dekfracc@args}
\gmu@numeratorkern 2568 \def\gmu@numeratorkern{\kern-.05em\relax}
2569 \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 Unicones of built-in fractions and `\dekfracc` command is defined to use them.

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

Note that in the first argument of `\@ifstar` we wrote 8 (eight) #s to get the correct definition and in the second argument 'only' 4. (The L^AT_EX 2_ε 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 3128.

```

\resizegraphics
\resizegraphics 2591 \def\resizegraphics#1#2#3{%
2594 \resizebox{#1}{#2}{%
2595 \includegraphics{#3}}}%
\GMtextsuperscript 2597 \def\GMtextsuperscript{%
2598 \@ifXeTeX{%
\textsuperscript 2599 \def\textsuperscript##1{%
2600 \addfontfeature{VerticalPosition=Numerator}##1}}%
2601 }{\truetextsuperscript}}
\truetextsuperscript 2603 \def\truetextsuperscript{%
\textsuperscript 2604 \pdef\textsuperscript##1{%
2605 \@textsuperscript{\selectfont##1}}%
\@textsuperscript 2606 \def\@textsuperscript##1{%
2607 {\m@th\ensuremath{\sim\mbox{\fontsize\sfontsize\z@##1}}}}}%

```

Settings for mathematics in main font

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

```

\gmath 2620 \pdef\gmath{%

```

```

2621 \everymath{%
2622   \relaxen\do
2623   \newcommand*\do[4][\mathit]{\def##2{##3{##1{\char"##4}}}%
2624   \do\alpha{}{03B1}%
2625   \do[\mathrm]\Delta{}{0394}%
2626   \do\varepsilon{}{03B5}%
2627   \do\vartheta{}{03D1}%
2628   \do\nu{}{03BD}%
2629   \do\pi{}{03C0}%
2630   \do\phi{}{03D5}%
2631   \do[\mathrm]\Phi{}{0424}%
2632   \do\sigma{}{03C3}%
2633   \do\varsigma{}{03DA}%
2634   \do\psi{}{03C8}%
2635   \do\omega{}{03C9}%
2636   \do\infty{}{221E}%
2637   \do[\mathrm]\neg{\mathbin}{00AC}%
2638   \do[\mathrm]\neq{\mathrel}{2260}%
2639   \do\partial{}{2202}%
2640   \do[\mathrm]\pm{}{00B1}%
2641   \do[\mathrm]\pm{\mathbin}{00B1}%
2642   \do[\mathrm]\sim{\mathrel}{007E}%
2643   \def\do##1##2##3{\def##1{%
2644     \mathop{\mathchoice{\hbox{%
2645       \rm
2646       \edef\gma@tempa{\the\fontdimen8\font}%
2647       \larger[3]%
2648       \lower\dimexpr(\fontdimen8\font-\gma@tempa)/2\fontdimen8\font}{\hbox{##2}}}{\hbox{%
2649         \rm
2650         \edef\gma@tempa{\the\fontdimen8\font}%
2651         \larger[2]%
2652         \lower\dimexpr(\fontdimen8\font-\gma@tempa)/2\fontdimen8\font}{\hbox{##2}}}}}%
2653     {\mathrm{##2}}{\mathrm{##2}}}}##3}%
2654   \do\sum{\char"2211}{}%
2655   \do\forall{\gma@quantifierhook\rotatebox[origin=c]{180}{A}%
2656     \setbox0=\hbox{A}\setbox2=\hbox{\scriptsize x}%
2657     \kern\dimexpr\ht2/3*2-\wd0/2\relax}{\nolimits}%
2658   \do\exists{\rotatebox[origin=c]{180}{\gma@quantifierhook E}}%
2659     \nolimits%
2660   \def\do##1##2##3{\def##1{##3{%
2661     \mathchoice{\hbox{\rm##2}}{\hbox{\rm##2}}%
2662     {\hbox{\rm\scriptsize##2}}{\hbox{\rm\tiny##2}}}}}%
2663   \do\vee{\rotatebox[origin=c]{90}{<}}\mathbin
2664   \do\wedge{\rotatebox[origin=c]{-90}{<}}\mathbin
2665   \do\leftarrow{\char"2190}\mathrel
2666   \do\rightarrow{\char"2192}\mathrel
2667   \do\leftrightarrow{\char"2190\kern-0,1em\char"2192}\mathrel
2668   \gmu@storespecialchars[\do\` \do\' \do\=]%
2669   \gmu@septify[\do\` 12\do\' 12\do\=12]%
2670   \def\do##1##2##3{%
2671     \catcode`##1=12\relax\do\` ##2 to ensure ##2 be ‘other’ in the definition body.

```

```

2678 \scantokens{\mathcode`##1="8000\relax
2679 \foone{\catcode`##1=\active}{\def##1}{##3{%
2680 \mathchoice{\hbox{\rm##2}}{\hbox{\rm##2}}%
2681 {\hbox{\rm\scriptsize##2}}{\hbox{\rm\tiny##2}}}%
2682 \ignorespaces}}% to eat the lineend (scantokens acts as \read including
line end).
2684 \do..\mathpunct_\do,,\mathpunct_\do.....\mathpunct
2685 \do((\mathopen%
2686 \gm@ifundefined{resetMathstrut@}{\}% an error occurred 'bad mathchar
etc.' because amsmath.sty doesn't take account of a possibility of '(' being
math-active.
2690 \def\resetMathstrut@{%
2691 \setbox\z@\hbox{%
%%_\mathchardef\@tempa\mathcode`\(\relax
%%_\def\@tempb##1"##2##3{\the\textfont"##3\char"%
%%_\expandafter\@tempb\meaning\@tempa_\relax
2695 (%)
2696 }%
2697 \ht\Mathstrutbox@\ht\z@_\dp\Mathstrutbox@\dp\z@
2698 }}%
2699 \do))\mathclose
2700 \do[[\mathopen\do]]\mathclose
2701 \do-{\char"2212}\mathbin_\do++\mathbin_\do==\mathrel_\do%
\do\times\mathbin
2702 \do::\mathbin_\do..\mathbin_\do/\mathbin_\do<<\mathrel
2703 \do>>\mathrel
2704 \do00\mathord_\do11\mathord_\do22\mathord_\do33\mathord
2705 \do44\mathord_\do55\mathord_\do66\mathord_\do77\mathord
2706 \do88\mathord_\do99\mathord
2707 \gmu@restorespecials
2708 \def\do##1##2##3{\def##1####1{##2{\hbox{%
2710 \rm
2711 \setboxo=\hbox{####1}%
2712 \edef\gma@tempa{\the\hto}%
2713 \edef\gma@tempb{\the\dpo}%
2714 ##3%
2715 \setboxo=\hbox{####1}%
2716 \lower\dimexpr(\hto+_\dpo)/2-\dpo_-((\gma@tempa+%
\gma@tempb)/2-\gma@tempb)_%
2717 \boxo}}}%
2718 \do\bigl\mathopen\larger
2719 \do\bigl\mathclose\larger
2720 \do\Bigl\mathopen\largerr
2721 \do\Bigl\mathclose\largerr
2722 \do\biggl\mathopen{\larger[3]}%
2723 \do\biggr\mathclose{\larger[3]}%
2724 \do\Biggl\mathopen{\larger[4]}%
2725 \do\Biggr\mathclose{\larger[4]}%
2727 \def\do##1##2{\def##1{\ifmmode##2{\mathchoice
2728 {\hbox{\rm\char`##1}}{\hbox{\rm\char`##1}}%
2729 {\hbox{\rm\scriptsize\char`##1}}{\hbox{\rm\tiny%
\char`##1}}}%
2730 \else\char`##1\fi}}%

```



```

2732 \do\{\mathopen
2733 \do\}\mathclose
2735 \def\={\mathbin{=}}%
2736 \def\neqb{\mathbin{\neq}}%
2737 \def\do##1{\edef\gma@tempa{%
2738 \def\@xa\@nx\csname_\@xa\gobble\string##1r\endcsname{%
2739 \@nx\mathrel{\@nx##1}}}%
2740 \gma@tempa}%
2741 \do\vee_\do\wedge_\do\neg
2742 \def\fakern{\mkern-3mu}%
2743 \thickmuskip=8mu_\plus_4mu\relax
2745 \gma@gmathhook
2746 }% of \everymath.
2747 \everydisplay\everymath
2748 \ifdefined\Url
2749 \ampulexdef\Url{\let\do}\@makeother
2750 {\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.

2754 \fi
2755 }% of \def\gmath.

\quantifierhook 2757 \emptify\gma@quantifierhook
\gma@quantifierhook 2758 \def\quantifierhook#1{%
2759 \def\gma@quantifierhook{#1}}

2761 \emptify\gma@gmathhook
\gmathhook 2762 \def\gmathhook#1{\addtomacro\gma@gmathhook{#1}}

\gma@dollar 2765 \def\gma@dollar$#1${\gmath$#1$}%
\gma@bare 2766 \def\gma@bare#1{\gma@dollar$#1$}%
\gma@checkbracket 2767 \def\gma@checkbracket{\@ifnextchar\[%
2768 \gma@bracket\gma@bare}
\gma@bracket 2769 \def\gma@bracket\[#1\]{\gmath\[#1\]}\@ifnextchar\par{\}%
\noindent}}

\gma 2770 \def\gma{\@ifnextchar$%
2771 \gma@dollar\gma@checkbracket}

\garamath 2777 \def\garamath{%
2778 \addtotoks\everymath{%
2779 \quantifierhook{\addfontfeature{OpticalSize=800}}}%
\gma@arrowdash 2781 \def\gma@arrowdash{%
2782 \setboxo=\hbox{\char"2192}\copyo\kern-o,6\wdo
2783 \bgcolor\rule[-\dpo]{o,6\wdo}{\dimexpr\hto+\dpo}%
\kern-o,6\wdo}}%

\gma@gmathhook 2785 \def\gma@gmathhook{%
2786 \def\do####1####2####3{\def####1{####3{%
\mathchoice{\hbox{\rm####2}}{\hbox{\rm####2}}}%
\hbox{\rm\scriptsize####2}}{\hbox{\rm%
\tiny####2}}}}}%
2789 \do\mapsto{\rule[o,4ex]{o,1ex}{o,4ex}\kern-o,05em%
2790 \gma@arrowdash\kern-o,05em\char"2192}\mathrel
2791 \do\cup{\scshape_\u}\mathbin
2792 \do\varnothing{\setboxo=\hbox{\gma@quantifierhook%
\addfontfeature{Scale=1.272727}o}}%

```

```

2793         \setbox2=\hbox{\char"2044}%
2794         \copyo_\kern-0,5\wdo_\kern-0,5\wd2_\lowero,125\wdo_\copy2
2795         \kerno,5\wdo\kern-0,5\wd2}{}%
2796         \do\leftarrow{\char"2190\kern-0,05em\gma@arrowdash}\mathrel
2797         \do\rightarrow{\gma@arrowdash\kern-0,05em\char"2192}%
                \mathrel
2798         \do\in{\gma@quantifierhook\char"0454}\mathbin
2799     }}%
2800 \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 2808 \def\gmu@tempa{\kern-0,08em\penalty10000\hskiposp\relax
2809      s\penalty10000\hskiposp\relax}
2811 \protected\edef\Vs{V\gmu@tempa}
2813 \protected\edef\Ws{W\gmu@tempa}
\Wz 2815 \pdef\Wz{W\kern-0,05em\penalty10000\hskiposp\relax_z}

```

Varia

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

```

\gmu@tilde 2824 \def\gmu@tilde{%
2825     \leavevmode\lower.8ex\hbox{${\,\widetilde{\mbox{\_}}}\,,${}}

```

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

```

\* 2829 \pdef\*{\gmu@tilde}
2835 \AtBeginDocument{% to bypass redefinition of \~ as a text command with various
        encodings
\texttilde 2837 \pdef\texttilde{%
2840     \@ifnextchar/{\gmu@tilde\kern-0,1667em\relax}\gmu@tilde}}

```

We prepare the proper kerning for “~/”.

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

(This definition is repeated in `gmverb`.)

```

2852 \foone{\catcode`\_\active}%
\gmobeyspaces 2853 {\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.

```

2860 \foone{\catcode`\^~M\active}% the comment signs here are crucial.
\defobeylines 2861 {\def\defobeylines{\catcode`\^~M=13_\def^~M{\par}}}

```

Another thing I dislike in L^AT_EX yet is doing special things for `\...skip’s`, ‘cause I like the Knuthian simplicity. So I sort of restore Knuthian meanings:

```

\dekssmallskip 2870 \def\dekssmallskip{\vskip\smallskipamount}

```

```

\undeksmallskip 2871 \def\undeksmallskip{\vskip-\smallskipamount}
\dekmedskip      2872 \def\dekmedskip{\vskip\medskipamount}
\dekbigskip      2873 \def\dekbigskip{\vskip\bigskipamount}
\hfillneg        2876 \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 2884 \long\def\@firstofmany#1#2\@@nil{#1}
```

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

```

\TODO 2888 \newcommand*{\TODO}[1] [] {%
2889     \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}` out to the .toc file but it worked wrong in some cases. So I redefine the internal L^AT_EX macro instead.

```

\twocoltoc 2924 \newcommand*\twocoltoc{%
2925     \RequirePackage{multicol}%
\starttoc 2926 \def\@starttoc##1{%
2927     \begin{multicols}{2}\makeatletter\@input{\jobname.##1}%
2928     \if@filesw\@xa\newwrite\csname_tfc##1\endcsname
2929     \immediate\openout\csname_tfc##1\endcsname\jobname.
        .##1\relax
2930     \fi
2931     \@nobreakfalse\end{multicols}}
2933 \@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 2938 \newcommand*\enoughpage[1]{%
2939     \par
2940     \dimeno=\pagegoal
2941     \advance\dimeno by-\pagetotal
2942     \ifdim\dimeno<#1\relax\newpage\fi}

```

An equality sign properly spaced:

```
\equals 2951 \pdef\equals{\hunskip$={}$\ignorespaces}
```

And for the L^AT_EX's pseudo-code statements:

```

\eequals 2953 \pdef\eequals{\hunskip$==$}\ignorespaces}
\cdot 2955 \pdef\cdot{\hunskip$={}\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 2975 \def\freeze@actives{%
2976   \count\z@\z@
2978   \@whilenum\count\z@<\@ccclvi\do{%
2979     \ifnum\catcode\count\z@=\active
\~ 2980     \uccode`\~=\count\z@
2981     \uppercase{\let~\relax}%
2982     \fi
2983     \advance\count\z@\@ne}}

```

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

```

\ShowFont 2989 \newcommand*\ShowFont[1][6]{%
2990   \begin{multicols}{#1}[The\currentfont(the\fontencoding\
      encoding):]
2991   \parindent\z@
2992   \count\z@\m@ne
2993   \@whilenum\count\z@<\@ccclv\do{
2994     \advance\count\z@\@ne
2995     \the\count\z@:\~\char\count\z@\par}
2996   \end{multicols}}

```

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

```

\liturgiques 3004 \newcommand*\liturgiques[1][red]{% Requires the color package.
3005   \gmu@RPfor{color}\color%
\czerwo 3006   \newcommand*\czerwo{\small\color{#1}}% environment
\czer 3007   \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.
\* 3010   \def\*{\czer{*$*$}}
\+ 3011   \def\+{\czer{$\dag$}}
\nieczer 3012   \newcommand*\niezcer[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 3017 \newcommand*\gmu@RPfor[3][]{%
3019   \ifx\relax#1\relax
\gmu@resa 3020   \else\def\gmu@resa{[#1]}%
3021   \fi
3022   \@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.

```

3028 \AtBeginDocument{%
\gmu@RPfor 3029   \renewcommand*\gmu@RPfor[3][]{%
3030     \unless\ifdefined#3%
3031     \@ifpackageloaded{#2}{}%
3032     \typeout{^^J!\Package`#2' not loaded!!!(\on@line)^^J}}%
3033     \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 3039 \pprovide\continuum{%
3040   \gmu@RPfor{eufrak}\mathfrak\ensuremath{\mathfrak{c}}}

```

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

```

\iteracro 3044 \def\iteracro{%
\acro      3045 \pdef\acro##1{\gmu@acrospaces##1\gmu@acrospaces}%
3046 }

3048 \iteracro

\gmu@acrospaces 3050 \def\gmu@acrospaces#1\#2\gmu@acrospaces{%
3051 \gmu@acroinner#1\gmu@acroinner
3052 \ifx\relax#2\relax\else
3053 \space
3054 \afterfi{\gmu@acrospaces#2\gmu@acrospaces}% when #2 is nonempty, it
is ended with a space. Adding one more space in this line resulted in an
infinite loop, of course.
3058 \fi}

\gmu@acroinner 3061 \def\gmu@acroinner#1{%
3062 \ifx\gmu@acroinner#1\relax\else
3063 \ifcat\@nx#1\relax%
3064 \ifnum`#1=\uccode`#1%
3065 {\acrocore{#1}}%
3066 \else{#1}% tu bylo \smallerr
3067 \fi
3068 \else#1%
3069 \fi
3070 \afterfi\gmu@acroinner
3071 \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 3075 \def\acrocore{\scshape\lowercase}
```

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

```

\IMO 3080 \newcommand*\IMO{\acro{IMO}}
\AKA 3081 \newcommand*\AKA{\acro{AKA}}

\usc 3083 \pdef\usc#1{{\addfontfeature{Letters=UppercaseSmallCaps}#1}}

\uscacro 3085 \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 3103 \def\secondclass{%
\ifSecondClass 3104 \newif\ifSecondClass
3105 \SecondClasstrue
3106 \@fileswithoptions\@clsextension}% [outeroff,gmeometric]{gmdocc}
it's loading gmdocc.cls with all the bells and whistles except the error mes-
sage.

```

Cf. *The T_EXbook* exc. 11.6.

A line from L^AT_EX:

%\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 3118 \def\gmu@dekfraccsimple#1/#2{\leavevmode\kern.1em
```

```

3119 \raise.5ex\hbox{%
3120 \smaller[3]#1}\gmu@numeratorkern
3121 \dekfraccslash\gmu@denominator\kern
3122 {%
3123 \smaller[3]#2}%
3124 \if@gmu@mmhbox\egroup\fi}
\dekfraccsimple 3128 \def\dekfraccsimple{%
3129 \let\dekfracc@args\gmu@dekfraccsimple
3130 }
\dekfraccslash 3131 \@ifXeTeX{\def\dekfraccslash{\char"2044}}{%
\dekfraccslash 3132 \def\dekfraccslash{/}}\% You can define it as the fraction
slash, \char"2044
3134 \dekfraccsimple

```

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

```
\ikern 3142 \newcommand*\ikern{\,\penalty10000\hskip0.5pt\relax}
```

And a macro to forbid hyphenation of the next word:

```

\nohy 3146 \newcommand*\nohy{\leavevmode\kern0.5pt\relax}
\yeshy 3147 \newcommand*\yeshy{\leavevmode\penalty10000\hskip0.5pt\relax}

```

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

\@ifempty

```

\@ifempty 3153 \long\pdef\@ifempty#1#2#3{%
\gmu@reserveda 3154 \def\gmu@reserveda{#1}%
3155 \ifx\gmu@reserveda\@empty\afterfi{#2}%
3156 \else\afterfi{#3}\fi
3157 }

```

\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 3169 \def\gmu@getext#1.#2\@nil{%
3170 \def\gmu@filename{#1}%
3171 \def\gmu@fileext{#2}}
3173 \def\include#1{\relax
3174 \ifnum\@auxout=\@partaux
3175 \@latexerror{\string\include\space cannot be nested}\@eha
3176 \else\@include#1\fi}
\@include 3178 \def\@include#1{%
3179 \gmu@getext#1.\@nil
3181 \ifx\gmu@fileext\empty\def\gmu@fileext{tex}\fi
3182 \clearpage
3183 \if@filesw
3184 \immediate\write\@mainaux{\string\@input{\gmu@filename.aux}}%
3185 \fi

```

```

3186 \@tempwattrue
3187 \if@partsw
3188 \@tempwafalse
3189 \edef\reserved@b{#1}%
3190 \@for\reserved@a:=\@partlist\do{%
3191   \ifx\reserved@a\reserved@b\@tempwattrue\fi}%
3192 \fi
3193 \if@tempswa
3194   \let\@auxout\@partaux
3195   \if@filesw
3196     \immediate\openout\@partaux\gmufilename.aux
3197     \immediate\write\@partaux{\relax}%
3198   \fi
3199   \@input{\gmufilename.\gmufileext}%
3200   \inclasthook
3201   \clearpage
3202   \@writeckpt{\gmufilename}%
3203   \if@filesw
3204     \immediate\closeout\@partaux
3205   \fi
3206 \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.

```

3210   \deadcycles\z@
3211   \@nameuse{cp@\gmufilename}%
3212 \fi
3213 \let\@auxout\@mainaux}

```

```

\whenonly 3216 \newcommand\whenonly[3]{%
\gmu@whonly 3217 \def\gmu@whonly{#1,}%
3218 \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 3222 \def\inclasthook{\thispagestyle{closing}}

```

Faked small caps

```

\gmu@scapLetters 3228 \def\gmu@scapLetters#1{%
3229   \ifx#1\relax\relax\else% two \relaxes to cover the case of empty #1.
3230     \ifcat_a#1\relax
3231       \ifnum\the\lccode`#1=`#1\relax
3232         {\fakescapscore\MakeUppercase{#1}}% not Plain \uppercase because
           that works bad with inputenc.
3234       \else#1%
3235       \fi
3236     \else#1%
3237     \fi%
3238     \@xa\gmu@scapLetters
3239   \fi}%
\gmu@scapSpaces 3241 \def\gmu@scapSpaces#1_#2\@@nil{%
3242   \ifx#1\relax\relax
3243   \else\gmu@scapLetters#1\relax
3244   \fi

```

```

3245 \ifx#2\relax\relax
3246 \else\afterfi{\_\gmu@scapSpaces#2\@@nil}%
3247 \fi}
\gmu@scapss 3249 \def\gmu@scapss#1\@@nil{{\def~{{\nobreakspace}}%
\nobreakspace 3250 \gmu@scapSpaces#1_\@@nil}}%\_\def\\{{\newline}}\relax adding re-
definition of \_ caused stack overflow. Note it disallows hyphenation
except at \-.

\fakescaps 3254 \pdef\fakescaps#1{{\gmu@scapss#1\@@nil}}
3256 \let\fakescapscore\gmu@scalematchX
Experimente z akcentami patrz no3.tex.

\tinycap 3259 \def\tinycap{{\tiny\AE}}% to use in \fakescaps[\tiny]{...}
3261 \RequirePackage{calc}
wg \zf@calc@scale pakietu fontspec.

3265 \@ifpackageloaded{fontspec}{%
\gmu@scalar 3266 \def\gmu@scalar{1.0}%
\zf@scale 3267 \def\zf@scale{}%
\gmu@scalematchX 3268 \def\gmu@scalematchX{%
3269 \begingroup
\gmu@scalar 3270 \ifx\zf@scale\empty\def\gmu@scalar{1.0}%
3271 \else\let\gmu@scalar\zf@scale\fi
3272 \setlength\@tempdima{\fontdimen5\font}% 5—ex height
3273 \setlength\@tempdimb{\fontdimen8\font}% 8—XTeX synthesized up-
percase height.
3275 \divide\@tempdimb\_by1000\relax
3276 \divide\@tempdima\_by\_@tempdimb
3277 \setlength{\@tempdima}{\@tempdima*\real{\gmu@scalar}}%
3278 \gm@ifundefined{fakesc@extrascap}{\}%
3279 \setlength{\@tempdima}{\@tempdima*\real{%
\fakec@extrascap}}}%
3280 \@tempcnta=\@tempdima
3281 \divide\@tempcnta\_by\_1000\relax
3282 \@tempcntb=-1000\relax
3283 \multiply\@tempcntb\_by\_@tempcnta
3284 \advance\@tempcntb\_by\_@tempdima
3285 \xdef\gmu@scscale{\the\@tempcnta.%
3286 \ifnum\@tempcntb<100\_0\fi
3287 \ifnum\@tempcntb<10\_0\fi
3288 \the\@tempcntb}%
3290 \endgroup
3291 \addfontfeature{Scale=\gmu@scscale}%
3292 }}{\let\gmu@scalematchX\smallerr}

\fakecextrascap 3294 \def\fakecextrascap#1{\def\fakec@extrascap{#1}}
\fakec@extrascap

```

See above/see below

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

```

\wyzejnizej 3300 \newcommand*\wyzejnizej[1]{%
3301 \edef\gmu@tempa{\gm@ifundefined{r#1}{\arabic{page}}}%
3302 \@xa\@xa\@xa\@secondoftwo\csname\_r#1\endcsname}}%

```



```

3303 \ifnum\gmu@tempa<\arabic{page}\relax_wy\zej\fi
3304 \ifnum\gmu@tempa>\arabic{page}\relax_ni\zej\fi
3305 \ifnum\gmu@tempa=\arabic{page}\relax_\@xa\ignorespaces\fi
3306 }

```

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 3316 \def\napapierkistretch{0,3pt}% It's quite much for 11/13pt leading.
\napapierkicore    3318 \def\napapierkicore{\advance\baselineskip%
                    3319   by_optplus\napapierkistretch\relax}
napapierki         3321 \newenvironment*{napapierki}{%
                    3322   \par\global\napapierkicore}{%
                    3323   \par\dimen\z@=\baselineskip
                    3324   \global\baselineskip=\dimen\z@}% so that you can use \endnapapierki in
                        interlacing environments.
\gmu@luzniej       3328 \newcount\gmu@luzniej
\luzniejcore       3330 \newcommand*\luzniejcore[1][1]{%
                    3331   \advance\gmu@luzniej\@ne% We use this count to check whether we open the
                        environment or just set \looseness inside it again.
                    3333   \ifnum\gmu@luzniej=\@ne_\multiply\tolerance_by_2_\fi
                    3334   \looseness=#1\relax}

```

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

```

luzniej 3338 \newenvironment*{luzniej}{\par\luzniejcore}{\par}

```

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

```

luzniej* 3343 \newenvironment*{luzniej*}[1][1]{%
          3344   \multiply\tolerance_by_2\relax
          3345   \everypar{\looseness=#1\relax}}{\par}

```

```

\nawj 3347 \newcommand*\nawj{\kerno,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.

```

3355 \ifdefined\XeTeXversion
\pauza@skipcore 3356 \def\pauza@skipcore{\hskip.2em_plus.1em\relax
\pauzacore      3357   \pauzacore\hskip.2em_plus.1em\relax\ignorespaces}%
\ppauza@skipcore 3359 \def\ppauza@skipcore{\unskip\penalty10000\hskip.2em_plus.1em%
                    \relax
                    3360   -\hskip.2em_plus.1em\ignorespaces}
3362 \AtBeginDocument{% to be independent of moment of loading of polski.
\pdef\--{
3363   \ifhmode
3364     \unskip\penalty10000
3365     \afterfi{%
3366       \@ifnextspace{\pauza@skipcore}%
3367       {\@ifnextMac\pauza@skipcore{%
3368         \pauzacore\penalty\hyphenpenalty\hskip\z@}}}%
3369     \else
3370     \else

```

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

```
3374 \leavevmode\pauzacore\penalty10000\hskipo,5em\ignorespaces
3375 \fi}%
```

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

```
\pauza 3378 \pdef\pauza{%
3379 \ifhmode
3380 \unskip\penalty10000
3381 \hskipo.2em\pluso.1em\relax
3382 \pauzacore\hskip.2em\pluso.1em\relax\ignorespaces%
3383 \else
3384 \pauzadial
3385 \fi}%
```

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

```
\pauzadial 3390 \pdef\pauzadial{%
3391 \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 3395 \pdef\lpauza{%
3396 \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 space₁₀. 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...

```
\- 3404 \pdef\-%
3405 \ifvmode\PackageError{gmutils}{%
3406 command_\backslash_ppauza_(en_dash)_not_intended_for_vmode.}{%
3407 Use_\backslash_ppauza_(en_dash)_only_in_number_and_numerals_
ranges.}%
3408 \else
3409 \afterfi{%
3410 \@ifnextspace{\ppauza@skipcore}{%
3411 \@ifnextMac\ppauza@skipcore{\unskip\discretionary{-}{-}{-}}%
-}{-}}}%
3412 }%
3413 \fi
3414 }%
\ppauza 3416 \pdef\ppauza{%
3417 \ifvmode\PackageError{gmutils}{%
3418 command_\backslash_ppauza_(en_dash)_not_intended_for_vmode.}{%
3419 Use_\backslash_ppauza_(en_dash)_only_in_number_and_numerals_
ranges.}%
3420 \else
3421 \unskip\discretionary{-}{-}{-}%
3422 \fi}%
\emdash 3424 \def\emdash{\char`-}
3425 }% of at begin document
```

```

\longpauza 3427 \def\longpauza{\def\pauzacore{-}}
\pauzacore 3428 \longpauza
\shortpauza 3429 \def\shortpauza{%
\pauzacore 3430 \def\pauzacore{-\kern,23em\relax\llap{-}}}
3431 \fi% of if XYTeX.

```

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 –.

```

3437 \ifdefined\XeTeXversion
3438 \foone{\catcode`\- \active \catcode`\- \active \catcode`\- \active}{%
\adashes 3439 \def\adashes{\AtBeginDocument\adashes}% because \pauza is defined at
begin document.
\adashes 3441 \AtBeginDocument{\def\adashes{%
3442 \catcode`\- \active \let-\-%
3443 \catcode`\- \active \let-\-%
3445 }}}
3446 \else
3447 \relaxen\adashes
3448 \fi

```

The hyphen shouldn't be active IMO because it's used in T_EX 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 3457 \def\gmu@dywiz{\ifmmode-\else
3458 \ifvmode-\else\afterfifi\dywiz\fi\fi}%
3460 \foone{\catcode`\- \active}{%
\ahyphen 3461 \def\ahyphen{\let-\gmu@dywiz\catcode`\- \active}}

```

To get current time. Works in ε -T_EXs, including X_YTeX. \czas typesets 16:30 and \czas[:] typesets 16:30.

```

\czas 3466 \newcommand*\czas[1][.]{%
3467 \the\numexpr(\time-30)/60\relax#1%
3468 \@tempcnta=\numexpr\time-(\time-30)/60*60\relax
3469 \ifnum\@tempcnta<10\o\fi\the\@tempcnta}
3472 \@ifXeTeX{%
\textbullet 3473 \pdef\textbullet{%
3476 \iffontchar\font"2022\char"2022\else\ensuremath{\bullet}%
\fi}}
\textbullet 3477 {\def\textbullet{\ensuremath{\bullet}}}
tytulowa 3479 \newenvironment*{tytulowa}{\newpage}{\par\thispagestyle{empty}%
\newpage}

```

To typeset peoples' names on page 4 (the editorial page):

```

\nazwired 3482 \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 SevSevo4.tex).

```

\polskadata 3496 \newcommand*\polskadata{%
\gmudatef 3497 \def\gmudatef##1-##2-##3##4,##5\gmudatef{%
3498 \ifx\relax##2\relax##3##4%
3499 \else
3500 \ifnum##3\@firstofmany##4o\@nil=o\relax
3501 \else
3502 \ifnumo##3=o\relax
3503 \else##3%
3504 \fi##4%
3505 \fi
3506 \ifcase##2\relax\or\_stycznia\or\_lutego%
3507 \or\_marca\or\_kwietnia\or\_maja\or\_czerwca\or\_lipca\or\_
sierpnia%
3508 \or\_wrzeźnia\or\_października\or\_listopada\or\_grudnia\else
3509 {}%
3510 \fi
3511 \if\relax##1\relax\else\_fi\_##1%
3512 \fi
3513 \gmudatecomma{##5}}% of \gmudatef.
\gmudatefsl 3515 \def\gmudatefsl##1/##2/##3##4,##5\gmudatefsl{%
3516 \if\relax##2\relax##3##4%
3517 \else
3518 \ifnum##3\@firstofmany##4o\@nil=o\relax
3519 \else
3520 \ifnumo##3=o\relax
3521 \else##3%
3522 \fi##4%
3523 \fi
3524 \ifcase##2\relax\or\_stycznia\or\_lutego%
3525 \or\_marca\or\_kwietnia\or\_maja\or\_czerwca\or\_lipca\or\_
sierpnia%
3526 \or\_wrzeźnia\or\_października\or\_listopada\or\_grudnia\else
3527 {}%
3528 \fi
3529 \if\relax##1\relax\else\_fi\_##1%
3530 \fi
3531 \gmudatecomma{##5}}%
3532 }% of \polskadata
3537 \polskadata

```

For documentation in English:

```

\englishdate 3540 \newcommand*\englishdate{%
\gmudatef 3541 \def\gmudatef##1-##2-##3##4,##5\gmudatef{%
3542 \if\relax##2\relax##3##4%
3543 \else
3544 \ifcase##2\relax\or\_January\or\_February%
3545 \or\_March\or\_April\or\_May\or\_June\or\_July\or\_August%
3546 \or\_September\or\_October\or\_November\or\_December\else
3547 {}%
3548 \fi

```

```

3549 \ifnum##3\@firstofmany##4o\@@nil=o\relax
3550 \else
3551 \_ %
3552 \ifnumo##3=o\relax
3553 \else##3%
3554 \fi##4%
3555 \ifcase##3\@firstofmany##4\relax\@@nil\relax\or\_st\or\_nd%
\or\_rd\else\_th\fi
3556 \fi
3557 \ifx\relax##1\relax\else,\_\fi\_##1%
3558 \fi
3559 \gmu@datecomma{##5}}%
\gmu@datefsl 3561 \def\gmu@datefsl##1/##2/##3##4,##5\gmu@datefsl{%
3562 \if\relax##2\relax##3##4%
3563 \else
3564 \ifcase##2\relax\or\_January\or\_February%
3565 \or\_March\or\_April\or\_May\or\_June\or\_July\or\_August%
3566 \or\_September\or\_October\or\_November\or\_December\else
3567 {}%
3568 \fi
3569 \ifnum##3\@firstofmany##4o\@@nil=o\relax
3570 \else
3571 \_ %
3572 \ifnumo##3=o\relax
3573 \else##3%
3574 \fi##4%
3575 \ifcase##3\@firstofmany##4\relax\@@nil\relax\or\_st\or\_nd%
\or\_rd\else\_th\fi
3576 \fi
3577 \if\relax##1\relax\else,\_\fi\_##1%
3578 \fi
3579 \gmu@datecomma{##5}}%
3580 }
\gmu@datecomma 3583 \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.
3586 \ifx\gmu@datecomma#1\gmu@datecomma\else
3587 ,\gmu@stripcomma#1%
3588 \fi
3589 }% of \gmu@datecomma
\gmu@stripcomma 3591 \def\gmu@stripcomma#1,{#1}
\ifgmu@dash 3594 \newif\ifgmu@dash
\gmu@ifnodash 3596 \def\gmu@ifnodash#1-#2\@@nil{%
\gmu@tempa 3597 \def\gmu@tempa{#2}%
3598 \ifx\gmu@tempa\@empty}
\gmu@testdash 3600 \pdef\gmu@testdash#1\ifgmu@dash{%
3601 \gmu@ifnodash#1-\@@nil
3602 \gmu@dashfalse
3603 \else
3604 \gmu@dashtrue
3605 \fi
3606 \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 technical 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 3637 \newif\ifdate
\bidate 3639 \pdef\bidate#1{%
3640   \ifdate\gmu@testdash#1%
3641   \ifgmu@dash
3642     \gmu@datef#1,\gmu@datef
3643   \else
3644     \gmu@datefsl#1,\gmu@datefsl
3645   \fi\fi}

\linedate 3647 \pdef\linedate{\@ifstar\linedate@@\linedate@}
\linedate@@ 3648 \pdef\linedate@@#1{\linedate@{--}{#1}}
\linedate@ 3649 \pdef\linedate@#1{\par\ifdate\addvspace{\dateskip}%
3650   \date@line{\footnotesize\itshape\bidate{#1}}%
3651   \nopagebreak\else%%\ifnum\arabic{dateinsection}>0\dekbigskip\fi
3652   \addvspace{\bigskipamount}%
3653   \fi}% end of \linedate.
3655 \let\dateskip\medskipamount

\rdate 3663 \pdef\rdate{\let\date@line\rightline\linedate}
\ldate 3664 \pdef\ldate{%
\date@line 3666   \def\date@line##1{\par{\raggedright##1\par}}%
3667   \linedate}
\runindate 3668 \newcommand*\runindate[1]{%
3669   \paragraph{\footnotesize\itshape\gmu@datef#1\gmu@datef}%
3670   \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.

```

3673 \let\thedata\ldate
\zwrobcy 3676 \pdef\zwrobcy#1{\emph{#1}}% ostinato, allegro con moto, garden party etc.,
           także komplement
\tytul 3679 \pdef\tytul#1{\emph{#1}}
```

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

```

maszynopis 3685 \newenvironment{maszynopis}[1][\ttfamily
3686 \hyphenchar\font=45\relax% this assignment is global for the font.
3687 \@tempskipa=\glueexpr\rightskip+\leftskip\relax
3688 \ifdim\gluestretch\@tempskipa=\z@
3689 \tolerance900
it worked well with tolerance = 900.
3691 \advance\rightskip_\by\z@_pluso,5em\relax\fi
3692 \fontdimen3\font=\z@% we forbid stretching spaces...
%_\fontdimen4\font=\z@ but allow shrinking them.
3694 \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.
3697 \StoreMacro\pauzacore
\pauzacore 3698 \def\pauzacore{-\rlap{\kern-o,3em-}-}%
3699 }{\par}
\justified 3703 \newcommand*\justified{%
3704 \leftskip=1\leftskip% to preserve the natural length and discard stretch and
shrink.
3706 \rightskip=1\rightskip
3707 \parfillskip=1\parfillskip
3708 \advance\parfillskip_\by_osp_plus_1fil\relax
3709 \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 3714 \newcommand*\fullpar{%
3715 \hunskip
3716 \bgroup\parfillskip\z@skip\par\egroup}
To conform Polish recommendation for typesetting saying that the last line of a para-
graph 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 3725 \newcommand*\twoparinit{% the name stands for 'last paragraph line's length
minimum two \parindent.
3727 \edef\twopar{%
3728 \hunskip% it's \protected, remember?
3729 \bgroup
3730 \parfillskip=\the\glueexpr
3731 \dimexpr\textwidth-2\parindent\relax
3732 minus\dimexpr\textwidth-2\parindent\relax
3733 \relax% to delimit \glueexpr.
3734 \relax% to delimit the assignment.
3735 \par\egroup
3736 }% of \gmu@twoparfill
3741 }% of \twoparinit.
For dati under poems.
\wherncore 3748 \newcommand\wherncore[1]{%
3749 \rightline{%
3750 \parbox{o,7666\textwidth}{
3751 \leftskiposp_plus_\textwidth
3752 \parfillskiposp\relax

```

```

3753         \let\\\linebreak
3754         \footnotesize_#1}}
\whern 3756 \def\whern{%
3757     \@ifstar\wherncore{\vskip\whernskip\wherncore}}
\whernskip 3760 \newskip\whernskip
3761 \whernskip2\baselineskip_minus_2\baselineskip\relax
\whernup 3763 \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 3787 \def\leftslanting{%
\litshape 3788     \pdef\litshape{%
3790         \itshape
3791         \@tempdima=-2\fontdimen1\font
3792         \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.
3796         \edef\gmu@tempa{%
3797             \@nx\addfontfeature{RawFeature={slant=\strip@pt%
                \@tempdima}}}% when not \edefed, it caused an error, which is
                perfectly understandable.
3800         \gmu@tempa}%
\textlit 3803     \pdef\textlit##1{%
3804         {\litshape##1}}%
\lit 3806     \pdef\lit{\rm\litshape}%
\lsl 3809     \pdef\lsl{\it
3812         \@tempdima=-\fontdimen1\font
3813         \xdef\gmu@tempa{%
3814             \@nx\addfontfeature{RawFeature={slant=\strip@pt%
                \@tempdima}}}%
3815         \rm_% Note in this declaration we left-slant the basic Roman font not the it-
            shape of the current family.
3817         \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 3825     \pdef\em{%
3826         \ifdim\fontdimen1\font=\z@_ \litshape
3827         \else
3828             \ifdim\fontdimen1\font>\z@_ \litshape

```



```

3829         \else_\upshape
3830         \fi
3831     \fi}%
3834     \pdef\emph##1{%
3835         {\em##1}}%
3836 }% of \leftslanting.

```

Thousand separator

\thousep 3840 \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.

```

3844     \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.
3847     \gmu@thou@fiver{#1}{% if more than five digits:
3848         \emptify\gmu@thou@put
3849         \relaxen\gmu@thou@o\relaxen\gmu@thou@i\relaxen\gmu@thou@ii
3850         \@tempcnta\z@
3851         \gmu@thou@putter#1\gmu@thou@putter
3852         \gmu@thou@put
3853     }}

```

```

\gmu@thou@fiver 3855 \def\gmu@thou@fiver#1#2#3#4#5\gmu@thou@fiver#6#7{%
3856     \ifx\relax#5\relax\afterfi{#6}\else\afterfi{#7}\fi}

```

```

\gmu@thou@putter 3858 \def\gmu@thou@putter#1#2{% we are sure to have at least five tokens before the
           guardian \gmu@thou@putter.
3860     \advance\@tempcnta\@ne
3861     \@tempcntb\@tempcnta
3862     \divide\@tempcntb3\relax
3863     \@tempcnta=\numexpr\@tempcnta-\@tempcntb*3
3864     \edef\gmu@thou@put{\gmu@thou@put#1%
3865         \ifx\gmu@thou@putter#2\else
3866             \ifcase\@tempcnta
3867                 \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.
3870             \fi
3871             \fi}% of \edef
3872     \ifx\gmu@thou@putter#2% if we are at end of the digits...
3873         \edef\gmu@tempa{%
3874             \ifcase\@tempcnta
3875                 \gmu@thou@o\or\gmu@thou@i\or\gmu@thou@ii
3876             \fi}%
3877         \@xa\let\gmu@tempa\gmu@thousep% ... we set the proper cs...
3878     \else% or ...
3879         \afterfi{% iterate.
3880             \gmu@thou@putter#2}% of \afterfi
3881     \fi% of if end of digits.
3882 }% of \gmu@thou@putter.

```

```

\gmu@thousep 3884 \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 3892 \pdef\xathousep#1{\@xa\xathousep\@xa{\the#1}}
```

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

```
\shortthousep 3896 \def\shortthousep{%
\thous 3897 \pdef\thous{%
3898 \ifmode\hbox\bgroup\@gmu@mmhboxtrue\fi
3899 \afterassignment\thous@inner
3900 \@tempcnta=}%
\thous@inner 3902 \def\thous@inner{%
3903 \ifnum\@tempcnta<0_\$-$$%
3904 \@tempcnta=-\@tempcnta
3905 \fi
3906 \xathousep\@tempcnta
3907 \if@gmu@mmhbox\egroup
3908 \else\afterfi{\@ifnextcat_a\space{}}}%
3909 \fi}%
3910 }% of \shortthousep.
```

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

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

```
\urladdstar 3918 \def\urladdstar{%
3919 \AtBeginDocument{%
3920 \@ifpackageloaded{hyperref}{%
3921 \StoreMacro\url
\url 3922 \def\url{\@ifstar{\nolinkurl}{\storedcsname{url}}}%
3923 }{}}
3925 \@onlypreamble\urladdstar
3928 \endinput
```

Change History

vo.74

`\@begnamedgroup@ifcs:`

The catcodes of `\begin` and `\end` argument(s) don't have to agree strictly anymore: an environment is properly closed if the `\begin`'s and `\end`'s arguments result in the same `\csname`, 1063

General:

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, 3928

vo.75

`\@ifnextcat:`

`\let` for #1 changed to `\def` to allow things like `\noexpand~`, 858

`\@ifnextif:`

`\let` for #1 changed to `\def` to allow things like `\noexpand~`, 894

`\@ifnif:`

added, 919

- vo.76
General:
A ‘fixing’ of `\dots` was rolled back since it came out they were o.k. and that was the `qx` encoding that prints them very tight, 3928
`\freeze@actives`:
added, 2955
- vo.77
General:
`\afterfi` & `pals` made two-argument as the Marcin Woliński’s analogoi are. At this occasion some redundant macros of that family are deleted, 3928
- vo.78
General:
`\@namelet` renamed to `\n@melet` to solve a conflict with the beamer class. The package contents regrouped, 3928
- vo.79
`\not@onlypreamble`:
All the actions are done in a group and therefore `\xdef` used instead of `\edef` because this command has to use `\do` (which is contained in the `\@preamblecmds` list) and `\not@onlypreamble` itself should be able to be let to `\do`, 1652
- vo.80
General:
Checksum 1689, 0
`\hfillneg`:
added, 2876
- vo.81
`\dekfracslash`:
moved here from `pmlectionis.cls`, 3134
`\ifSecondClass`:
moved here from `pmlectionis.cls`, 3106
- vo.82
`\ikern`:
added, 3142
- vo.83
`\~`:
postponed to `\begin{document}` to avoid overwriting by a text command and made sensible to a subsequent `/`, 2829
- 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, 3928
- vo.86
`\texttilde`:
renamed from `\~` since the latter is one of L^AT_EX accents, 2837
- vo.87
General:
Checksum 4027, 0
the package goes ε -T_EX even more, making use of `\ifdefined` and the code using utf-8 chars is wrapped in a X_YT_EX-condition, 3928
- vo.88
General:
Checksum 4040, 0
`\RestoreEnvironment`:
added, 1591
`\storedcsname`:
added, 1582
`\StoreEnvironment`:
added, 1587
- vo.89
General:
removed obsolete adjustment of `pgf` for X_YT_EX, 3928
- vo.90
General:
Checksum 4035, 0
`\XeTeXthree`:
adjusted to the redefinition of `\verb` in `xlxtra` 2008/07/29, 2470
- vo.91
General:
Checksum 4055, 0
removed `\jobnamewoe` since `\jobname` is always without extension. `\xiispace` forked to `\visiblespace` `\let` to `\xxt@visiblespace` of `xlxtra` if available. The documentation driver integrated with the `.sty` file, 3928
- vo.92
`\@checkend`:
shortened thanks to `\@ifenvir`, 1095
`\@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`, 277
`\@ifenvir`:
added, 1087
General:
Checksum 4133, 0
- vo.93
`\@nameedef`:
added, 225
General:

<p>A couple of <code>\DeclareRobustCommand*</code> changed to <code>\pdef</code>, 3928 Checksum 4140, 0 Checksum 4501, 0 The numerical macros commented out as obsolete and never really used, 3928 <code>\ampulexdef</code>: added, 751 <code>\em</code>: added, 3825, 3834 <code>\gmu@RPfor</code>: renamed from <code>\gmu@RPif</code> and #3 changed from a cname to cs, 3017 <code>\litshape</code>: copied here from E. Szarzyński’s <i>The Letters</i>, 3788 <code>\lsl</code>: copied here from E. Szarzyński’s <i>The Letters</i>, 3809 <code>\nocite</code>: a bug fixed: with natbib an ‘extra ’ error. Now it fixes only the standard version of <code>\nocite</code>, 1690 <code>\pdef</code>: added, 183 <code>\pprovide</code>: added, 212 <code>\provide</code>: added, 197 <code>\textlit</code>: added, 3803 <code>\thousep</code>: added, 3840 vo.94 <code>\@xau</code>: added, 179 General: <code>\bgroup</code> and <code>\egroup</code> in the macro storing commands and in <code>\foone</code> changed to <code>\begingroup</code> and <code>\endgroup</code> since the former produce an empty <code>\mathord</code> in math mode while the latter don’t, 3928 Checksum 4880, 0</p>	<p>removed <code>\unex@namedef</code> and <code>\unex@nameuse</code>, probably never really used since they were incomplete: <code>\edef@other</code> undefined, 3928 The code from ancient xparse (1999) of T_EXLive 2007 rewritten here, 3928 <code>\afterfifi</code>: <code>\if</code> removed from parameters’ string, 999 <code>\ampulexdef</code>: made xparse-ish and <code>\ampulexset</code> removed, 751 <code>\dekfracc</code>: made to work also in math mode, even with math-active digits, 2562 <code>\gm@ifundefined</code>: added. All <code>\@ifundefineds</code> used by me changed to this, 408 made robust to unbalanced <code>\ifs</code> and <code>\fis</code> the same way as L^AT_EX’s <code>\@ifundefined</code> (after a heavy debug :-), 408 <code>\gmath</code>: removed definition of <code>\langle letter \rangle s</code> and <code>\langle digit \rangle s</code>, 2621 <code>\ldate</code>: <code>\leftline</code> replaced with <code>\par...\par</code> to work well with floatflt, 3664 <code>\prependtomacro</code>: order of arguments reversed, 374 <code>\resizegraphics</code>: <code>\includegraphics</code> works well in X_YL_AT_EX so I remove the complicated version with <code>\XeTeXpicfile</code>, 2591 <code>\textbullet</code>: the X_YL_AT_EX version enriched with <code>\iffontchar</code> due to lack of bullets with the default settings reported by Morten Høgholm and Edd Barrett, 3473 vo.95 General: Checksum 4908, 0 <code>\gm@testdefined</code>: added, 425 <code>\gm@testundefined</code>: added, 440</p>
--	--

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.

`*`, 947, 948, 949, 951, 955, 956, 957, 961, 2829, 3010 `\+`, 3011

<code>\-, 1380, 3461</code>	<code>\@firstofmany, 2884,</code>	<code>\@tempcntb*, 3863</code>
<code>\<...>, 1311</code>	<code>3500, 3518, 3549,</code>	<code>\@tempdima, 3272, 3276,</code>
<code>\@@nil, 1337, 1339, 1341,</code>	<code>3555, 3569, 3575</code>	<code>3277, 3279, 3280,</code>
<code>2884, 3169, 3179,</code>	<code>\@firstofone, 522, 682</code>	<code>3284, 3791, 3797,</code>
<code>3241, 3246, 3249,</code>	<code>\@gif, 263, 264, 271</code>	<code>3812, 3814</code>
<code>3250, 3254, 3500,</code>	<code>\@gmu@mmhboxtrue, 2564,</code>	<code>\@tempdima*, 3277, 3279</code>
<code>3518, 3549, 3555,</code>	<code>3898</code>	<code>\@tempdimb, 3273, 3275, 3276</code>
<code>3569, 3575, 3596, 3601</code>	<code>\@if, 286, 287, 290</code>	<code>\@temptokena, 494, 504,</code>
<code>\@M, 1995</code>	<code>\@ifXeTeX, 2464, 2472,</code>	<code>512, 541, 542, 584, 585</code>
<code>\@afterheading, 1988</code>	<code>2598, 3131, 3472</code>	<code>\@temptokenb, 480, 481,</code>
<code>\@badend, 1095</code>	<code>\@ifempty, 775, 1930,</code>	<code>495, 532, 536, 538,</code>
<code>\@begnamedgroup, 1039,</code>	<code>1950, 3153</code>	<code>568, 572, 574, 625</code>
<code>1056, 1061</code>	<code>\@ifenvir, 1087, 1095</code>	<code>\@textsuperscript,</code>
<code>\@begnamedgroup@ifcs,</code>	<code>\@ifl@aded, 1676</code>	<code>2605, 2606</code>
<code>1057, 1060</code>	<code>\@ifncat, 865, 868, 883</code>	<code>\@thirdofthree, 680, 689</code>
<code>\@car, 2338</code>	<code>\@ifnextMac, 971, 3368, 3411</code>	<code>\@toodeep, 2248, 2266</code>
<code>\@cclv, 2993</code>	<code>\@ifnextac, 930</code>	<code>\@topnewpage, 1890</code>
<code>\@cclvi, 2978</code>	<code>\@ifnextcat, 858, 931, 3908</code>	<code>\@typeset@protect, 521</code>
<code>\@checkend, 1095</code>	<code>\@ifnextif, 894</code>	<code>\@undefined, 159</code>
<code>\@clsextension, 3106</code>	<code>\@ifnextspace, 954, 3367,</code>	<code>\@whilenum, 2978, 2993</code>
<code>\@clubpenalty, 2000</code>	<code>3410</code>	<code>\@xa, 176, 179</code>
<code>\@currenvir, 1042, 1089</code>	<code>\@ifnif, 901, 904, 919</code>	<code>\@xau, 179</code>
<code>\@currenvline, 1046</code>	<code>\@ifnotmw, 1809, 1834,</code>	<code>\@xifncat, 870, 883</code>
<code>\@currsize, 1139, 1140,</code>	<code>1983, 2088, 2175, 2230</code>	<code>\@xifnif, 906, 919</code>
<code>1141, 1142, 1143,</code>	<code>\@include, 3176, 3178</code>	<code>\acro, 3045, 3080, 3081, 3085</code>
<code>1144, 1145, 1146,</code>	<code>\@itemdepth, 2265, 2268,</code>	<code>\acrocore, 3065, 3075</code>
<code>1147, 1148</code>	<code>2269</code>	<code>\adashes, 3439, 3439,</code>
<code>\@ddc, 496, 528, 544, 563,</code>	<code>\@itemitem, 2269, 2270</code>	<code>3441, 3447</code>
<code>591, 596, 597</code>	<code>\@minus, 2098, 2103, 2109,</code>	<code>\addfontfeature, 2494,</code>
<code>\@ddc@, 498, 520</code>	<code>2111, 2116, 2118,</code>	<code>2525, 2527, 2553,</code>
<code>\@ddc@C, 616</code>	<code>2125, 2130</code>	<code>2600, 2779, 2792,</code>
<code>\@ddc@D, 600</code>	<code>\@nameedef, 225</code>	<code>3083, 3291, 3797, 3814</code>
<code>\@ddc@c, 605</code>	<code>\@nobreakfalse, 1994, 2931</code>	<code>\addto@macro, 364, 371</code>
<code>\@ddc@c@, 607, 613, 618</code>	<code>\@nobreaktrue, 1989</code>	<code>\addtoheading, 1964</code>
<code>\@ddc@m, 552, 628</code>	<code>\@normalcr, 3709</code>	<code>\addtomacro, 371, 2762</code>
<code>\@ddc@o, 555</code>	<code>\@nx, 177</code>	<code>\addtotoks, 380, 2778</code>
<code>\@ddc@o@, 557, 560, 602</code>	<code>\@oarg, 1415, 1417, 1418</code>	<code>\AE, 3259</code>
<code>\@ddc@s, 547</code>	<code>\@oargsq, 1415, 1418, 1433</code>	<code>\afterassignment, 3899</code>
<code>\@ddc@x, 657</code>	<code>\@onlypreamble, 1717,</code>	<code>\afterfi, 202, 421, 934,</code>
<code>\@ddc@xm, 630</code>	<code>2933, 3925</code>	<code>958, 959, 997, 1061,</code>
<code>\@ddc@xmm, 633</code>	<code>\@parg, 1422, 1424, 1425</code>	<code>1062, 1091, 1092,</code>
<code>\@ddc@xmmm, 636</code>	<code>\@pargp, 1422, 1425, 1434</code>	<code>1329, 1360, 1431,</code>
<code>\@ddc@xmmmm, 639</code>	<code>\@parindent, 2254, 2255,</code>	<code>1474, 1551, 2467,</code>
<code>\@ddc@xmmmmm, 642</code>	<code>2257, 2272, 2273, 2275</code>	<code>3054, 3070, 3155,</code>
<code>\@ddc@xmmmmmm, 645</code>	<code>\@parsed@endenv, 510,</code>	<code>3156, 3218, 3246,</code>
<code>\@ddc@xmmmmmmm, 648</code>	<code>514, 664</code>	<code>3366, 3409, 3856,</code>
<code>\@ddc@xmmmmmmmm, 651</code>	<code>\@parsed@endenv@, 515,</code>	<code>3879, 3908</code>
<code>\@ddc@xmmmmmmmmm, 654</code>	<code>517, 665</code>	<code>\afterfifi, 199, 200, 416,</code>
<code>\@emptyify, 384, 385, 389</code>	<code>\@pkgextension, 1677</code>	<code>418, 999, 1530, 1535,</code>
<code>\@enumctr, 2251, 2252, 2258</code>	<code>\@preamblecmds, 1671,</code>	<code>2466, 3458</code>
<code>\@enumdepth, 2247, 2250,</code>	<code>1673, 1686, 1690</code>	<code>\afterfififi, 1007</code>
<code>2251</code>	<code>\@relaxen, 393, 394, 398</code>	<code>\afteriffifi, 1001</code>
<code>\@fileswithoptions, 3106</code>	<code>\@secondofthree, 676, 688</code>	<code>\afteriffififi, 1006</code>
	<code>\@starttoc, 2926</code>	<code>\afteriffiffififi, 1005</code>

<code>\ahyphen</code> , 3461	2332 , 2333 , 2976 ,	<code>\egfirstofone</code> , 242 , 244
<code>\AKA</code> , 3081	2978 , 2979 , 2980 ,	<code>\egRestore@Macro</code> , 1544 ,
<code>\all@other</code> , 772 , 773 , 782 ,	2983 , 2992 , 2993 ,	1546
783 , 784 , 2440 , 2542 ,	2994 , 2995	<code>\egRestore@MacroSt</code> ,
2544	<code>\cs</code> , 1379 , 1385 , 1389 , 1404	1544 , 1547
<code>\alpha</code> , 2625	<code>\cup</code> , 2791	<code>\egStore@Macro</code> , 1464 , 1469
<code>\ampulexdef</code> , 751 , 793 ,	<code>\czas</code> , 3466	<code>\egStore@MacroSt</code> , 1464 ,
808 , 1714 , 2749	<code>\czer</code> , 3007 , 3010 , 3011	1470
<code>\ampulexhash</code> , 812	<code>\czerwo</code> , 3006 , 3007	<code>\em</code> , 3825 , 3835
<code>\AmSTeX</code> , 2375	<code>\dag</code> , 3011	<code>\emdash</code> , 3424
<code>\arg</code> , 1430 , 1431	<code>\date@line</code> , 3650 , 3663 , 3666	<code>\emptify</code> , 385 , 385 , 2496 ,
<code>\AtBeginDocument</code> , 1249 ,	<code>\dateskip</code> , 3649 , 3655	2757 , 2761 , 3848
1429 , 1684 , 1720 ,	<code>\deadcycles</code> , 3210	<code>\endlist</code> , 2261 , 2278
1780 , 2482 , 2491 ,	<code>\DeclareDocumentCommand</code> ,	<code>\englishdate</code> , 3540
2835 , 3028 , 3362 ,	478 , 486 , 507 , 660 , 751	<code>\enoughpage</code> , 2938
3439 , 3441 , 3919	<code>\DeclareDocumentEnvironment</code> ,	<code>\ensuremath</code> , 1283 , 1296 ,
<code>\begin</code> , 1056	506 , 659	2396 , 2607 , 3040 ,
<code>\begin*</code> , 1056	<code>\DeclareLogo</code> , 2292 , 2311 ,	3476 , 3477
<code>\bgcolor</code> , 2783	2337 , 2348 , 2378 ,	<code>enumerate*</code> , 2246
<code>\BibTeX</code> , 2378	2384 , 2387 , 2389 ,	<code>\env</code> , 1385
<code>\bidate</code> , 3639 , 3650	2392 , 2395 , 2399 ,	<code>\equals</code> , 2951
<code>\Biggl</code> , 2724	2401 , 2404 , 2411	<code>\eTeX</code> , 2395 , 2399
<code>\biggl</code> , 2722	<code>\DeclareRobustCommand*</code> ,	<code>\everydisplay</code> , 2747 , 2800
<code>\Biggr</code> , 2725	1170 , 1171 , 1172 ,	<code>\everymath</code> , 2621 , 2747 ,
<code>\biggr</code> , 2723	1173 , 1379	2750 , 2778 , 2800
<code>\Bigl</code> , 2720	<code>\DeclareTextCommand</code> , 2305	<code>\everypar</code> , 1991 , 2001 , 3345
<code>\bigl</code> , 2718	<code>\DeclareTextCommandDefault</code> ,	<code>\exists</code> , 2662
<code>\Bigr</code> , 2721	2307	<code>\f@encoding</code> , 2990
<code>\bigr</code> , 2719	<code>\defobeylines</code> , 2861	<code>\f@series</code> , 2338
<code>\bigskipamount</code> , 2873 , 3652	<code>\dekbigskip</code> , 2873	<code>\fakern</code> , 2742
<code>\boldmath</code> , 2338	<code>\dekfracc</code> , 2562	<code>\fakesc@extrascap</code> ,
<code>\BooleanFalse</code> , 550 , 698	<code>\dekfracc@args</code> , 2540 ,	3279 , 3294
<code>\BooleanTrue</code> , 549 , 699	2551 , 2565 , 3129	<code>\fakescaps</code> , 3254
<code>\box</code> , 2359 , 2717	<code>\dekfraccsimple</code> , 3128 , 3134	<code>\fakescapscore</code> , 3232 , 3256
<code>\bslash</code> , 1228 , 1379 , 1482 ,	<code>\dekfraccslash</code> , 3121 ,	<code>\fakescextrascap</code> , 3294
1559 , 1583 , 1957 ,	3131 , 3132	<code>\file</code> , 1353
1958 , 1959 , 3406 ,	<code>\dekmedskip</code> , 2872	<code>\fooatletter</code> , 246
3407 , 3418 , 3419	<code>\deksmallskip</code> , 2870	<code>\foone</code> , 242 , 246 , 970 , 1185 ,
<code>\bullet</code> , 3476 , 3477	<code>\Delta</code> , 2626	1188 , 1196 , 1212 ,
<code>\c@gm@PronounGender</code> , 1747	<code>\dimen</code> , 2940 , 2941 , 2942 ,	1232 , 1236 , 1239 ,
<code>\c@NoNumSecs</code> , 1782	3323 , 3324	1388 , 2679 , 2852 ,
<code>\c@secnumdepth</code> , 1839	<code>\dimexpr</code> , 2650 , 2655 ,	2860 , 3438 , 3460
<code>\ClassError</code> , 1956	2661 , 2716 , 2783 ,	<code>\forall</code> , 2659
<code>\cleardoublepage</code> , 1796	3731 , 3732	<code>\FormatHangHeading</code> ,
<code>\clubpenalty</code> , 1995 , 2000	<code>\discre</code> , 1322 , 1331 , 1348	2097 , 2104 , 2112 , 2119
<code>\cmd</code> , 1404	<code>\discret</code> , 1324 , 1329	<code>\FormatRunInHeading</code> ,
<code>\cmd@to@cs</code> , 1404 , 1406	<code>\divide</code> , 2328 , 2331 , 2332 ,	2126 , 2131
<code>\color</code> , 3005 , 3006	3275 , 3276 , 3281 , 3862	<code>\freeze@actives</code> , 2975
<code>\continuum</code> , 3039	<code>\document</code> , 1717	<code>\fullpar</code> , 3714
<code>\copy</code> , 2321 , 2351 , 2365 ,	<code>\dp</code> , 2697 , 2713 , 2716 , 2783	<code>\g@emptify</code> , 389 , 390
2782 , 2794	<code>\dywiz</code> , 3458	<code>\g@relaxen</code> , 398 , 399
<code>\count</code> , 2326 , 2327 , 2328 ,	<code>\eequals</code> , 2953	<code>\gaddtomacro</code> , 359
2329 , 2330 , 2331 ,		

<code>\garamath</code> , <i>p.</i> 38, 2777	<code>\gma@gmathhook</code> , 2745 ,	<code>\gmu@nl@reserveda</code> ,
<code>\gemptify</code> , 390 , 390	2761 , 2762 , 2785	1641 , 1644 , 1649 , 1652
<code>\glet</code> , 351 , 2015	<code>\gma@quantifierhook</code> ,	<code>\gmu@nocite@ampulex</code> ,
<code>\glueexpr</code> , 3687 , 3730	2659 , 2662 , 2757 ,	1707 , 1720
<code>\gluestretch</code> , 3688	2759 , 2792 , 2798	<code>\gmu@numeratorkern</code> ,
<code>\gm@atppron</code> , 1749 , 1753 ,	<code>\gma@tempa</code> , 2648 , 2650 ,	2525 , 2568 , 2569 , 3120
1754 , 1755 , 1756 ,	2653 , 2655 , 2712 ,	<code>\gmu@prevsec</code> , 1990 , 1992 ,
1758 , 1759 , 1760 , 1761	2716 , 2737 , 2740	2014 , 2021 , 2051
<code>\gm@clearpagesduetoopenright</code> ,	<code>\gma@tempb</code> , 2713 , 2716	<code>\gmu@printslashes</code> ,
1795 , 1858	<code>\gm@math</code> , <i>p.</i> 38, 2620 , 2765 , 2769	1353 , 1355 , 1355 ,
<code>\gm@dontnumbersectionsoutofmainmatter</code> ,	<code>\gm@mathhook</code> , 2762	1357 , 1360
1793 , 1842	<code>\gml@storeCS</code> , 1509 , 1532 ,	<code>\gmu@cresa</code> , 3020 , 3022
<code>\gm@duppa</code> , 2541 , 2542 , 2544	1569	<code>\gmu@reserveda</code> , 947 ,
<code>\gm@gobmacro</code> , 2440 , 2446	<code>\gml@storemacros</code> , 1510 ,	949 , 955 , 957 , 1089 ,
<code>\gm@hyperrefstepcounter</code> ,	1521 , 1530 , 1535 , 1572	1091 , 1522 , 1525 ,
1785 , 1788 , 1871	<code>\gm@obeyspaces</code> , 2853	1528 , 1966 , 2015 ,
<code>\gm@ifnac</code> , 931 , 933	<code>\gm@showlists</code> , 820	2016 , 2019 , 2300 ,
<code>\gm@ifundefined</code> , 408 ,	<code>\GM@textsuperscript</code> , 2597	2302 , 2304 , 2305 ,
1809 , 1842 , 1858 ,	<code>\gmu@acroinner</code> , 3051 ,	2306 , 3154 , 3155
1934 , 1955 , 2010 ,	3061 , 3062 , 3070	<code>\gmu@reservedb</code> , 1090 , 1091
2023 , 2093 , 2220 ,	<code>\gmu@acrospace</code> , 3045 ,	<code>\gmu@restorespecials</code> ,
2374 , 2403 , 2405 ,	3050 , 3050 , 3054	451 , 2707
2410 , 2412 , 2542 ,	<code>\gmu@checkaftersec</code> ,	<code>\gmu@RPfor</code> , 3005 , 3017 ,
2686 , 3278 , 3301	2009 , 2068	3029 , 3040
<code>\gm@letspace</code> , 950 , 958	<code>\gmu@dashfalse</code> , 3602	<code>\gmu@scalar</code> , 3266 , 3270 ,
<code>\gm@notprerr</code> , 1682 , 1689	<code>\gmu@dashtrue</code> , 3604	3271 , 3277
<code>gm@PronounGender</code> , 1747	<code>\gmu@datecomma</code> , 3513 ,	<code>\gmu@scalematchX</code> , 3256 ,
<code>\gm@pswords</code> , 1337 , 1339 , 1341	3531 , 3559 , 3579 ,	3268 , 3292
<code>\gm@sec</code> , 2213 , 2224 , 2225	3583 , 3586	<code>\gmu@scapLetters</code> , 3228 ,
<code>\gm@secini</code> , 2178 , 2197 ,	<code>\gmu@datef</code> , 3497 , 3497 ,	3238 , 3243
2203 , 2209 , 2222	3541 , 3541 , 3642 , 3669	<code>\gmu@scapSpaces</code> , 3241 ,
<code>\gm@secmarkh</code> , 2205	<code>\gmu@datefsl</code> , 3515 , 3515 ,	3246 , 3250
<code>\gm@secstar</code> , 2181 , 2191 ,	3561 , 3561 , 3644	<code>\gmu@scapss</code> , 3249 , 3254
2198 , 2210 , 2224 , 2225	<code>\gmu@dekfracc</code> , 2524 ,	<code>\gmu@scscale</code> , 3285 , 3291
<code>\gm@secx</code> , 2213 , 2214	2543 , 2547	<code>\gmu@septify</code> , 453 , 2674
<code>\gm@secxx</code> , 2180 , 2208 , 2215	<code>\gmu@dekfraccsimple</code> ,	<code>\gmu@setheading</code> , 2065 ,
<code>\gm@straightensec</code> ,	2547 , 3118 , 3129	2071 , 2072
2219 , 2228	<code>\gmu@denominatorkern</code> ,	<code>\gmu@setsetSMglobal</code> ,
<code>\gm@targetheading</code> ,	2526 , 2569 , 3121	1508 , 1513 , 1568
1786 , 1789	<code>\gmu@discretionaryslash</code> ,	<code>\gmu@setSMglobal</code> , 1515 ,
<code>\gm@testdefined</code> , 425 , 443	1348 , 1359	1517 , 1535
<code>\gm@testundefined</code> , 440 ,	<code>\gmu@dywiz</code> , 3457 , 3461	<code>\gmu@SMdo@scope</code> , 1611 ,
478	<code>\gmu@fileext</code> , 3171 , 3181 ,	1613 , 1616 , 1617 , 1631
<code>\gma</code> , 2770	3199	<code>\gmu@SMdo@setscope</code> ,
<code>\gma@arrowdash</code> , 2781 ,	<code>\gmu@filename</code> , 3170 ,	1609 , 1615 , 1629
2790 , 2796 , 2797	3184 , 3196 , 3199 ,	<code>\gmu@SMglobalfalse</code> ,
<code>\gma@bare</code> , 2766 , 2768	3202 , 3211	1476 , 1490 , 1517 ,
<code>\gma@bracket</code> , 2768 , 2769	<code>\gmu@getaddvs</code> , 2060 ,	1526 , 1553 , 1562 , 1619
<code>\gma@checkbracket</code> ,	2060 , 2066	<code>\gmu@SMglobaltrue</code> ,
2767 , 2771	<code>\gmu@getext</code> , 3169 , 3179	1452 , 1515
<code>\gma@dollar</code> , 2765 , 2766 ,	<code>\gmu@gobdef</code> , 200 , 206	<code>\gmu@smtempa</code> , 1480 , 1489 ,
2771	<code>\gmu@ifnodash</code> , 3596 , 3601	1556 , 1561
	<code>\gmu@luzniej</code> , 3328 , 3331 ,	<code>\gmu@storespecials</code> ,
	3333	448 , 2673
		<code>\gmu@stripcomma</code> , 3587 , 3591

<code>\gmu@tempa</code> , 207, <u>766</u> , 772, 783, 791, 2445, 2447, <u>2808</u> , 2811, 2813, 3301, 3303, 3304, 3305, <u>3597</u> , 3598, 3796, 3800, 3813, 3817, 3873, 3877	<code>\HeShe</code> , <u>1758</u>	<code>\IfSomethingF</code> , <u>670</u> , 694
<code>\gmu@tempb</code> , <u>767</u> , 773, 784, 792	<code>\heshe</code> , <u>1753</u>	<code>\IfSomethingT</code> , <u>667</u> , 693
<code>\gmu@tempc</code> , <u>768</u> , 798	<code>\hfillneg</code> , 2876	<code>\IfSomethingTF</code> , <u>666</u> , 692
<code>\gmu@tempd</code> , 770, 771, 774, 777, 782, 784, 789, 790, 802, 808	<code>\hgrefstepcounter</code> , <u>333</u>	<code>\IfValueF</code> , 697
<code>\gmu@tempe</code> , 781, 787, 806, 809	<code>\hidden@iffalse</code> , <u>296</u> , 775	<code>\IfValueT</code> , 696
<code>\gmu@tempf</code> , 805, 808	<code>\hidden@iftrue</code> , <u>297</u> , 775	<code>\IfValueTF</code> , <u>695</u>
<code>\gmu@testdash</code> , 3600, 3640	<code>\HimHer</code> , <u>1760</u>	<code>\ikern</code> , <u>3142</u>
<code>\gmu@thou@fiver</code> , 3844, 3847, <u>3855</u> , 3855	<code>\himher</code> , <u>1755</u>	<code>\IMO</code> , <u>3080</u>
<code>\gmu@thou@i</code> , 3849, 3867, 3875	<code>\HisHer</code> , 1759	<code>\in</code> , 2798
<code>\gmu@thou@ii</code> , 3849, 3867, 3875	<code>\hisher</code> , <u>1754</u>	<code>\inclasthook</code> , 3200, <u>3222</u>
<code>\gmu@thou@o</code> , 3849, 3867, 3875	<code>\HisHers</code> , <u>1761</u>	<code>\includegraphics</code> , 2595
<code>\gmu@thou@put</code> , 3848, 3852, 3864	<code>\hishers</code> , <u>1756</u>	<code>\infty</code> , 2637
<code>\gmu@thou@putter</code> , 3851, <u>3858</u> , 3865, 3872, 3880	<code>\hrule</code> , 2039	<code>\itemindent</code> , 2254, 2272
<code>\gmu@thousep</code> , 3877, 3884	<code>\hunskip</code> , <u>341</u> , 2951, 2953, 2955, 3715, 3728	<code>itemize*</code> , <u>2264</u>
<code>\gmu@tilde</code> , <u>2824</u> , 2829, 2840	<code>\hyphenpenalty</code> , 1341, 3369, 3694	<code>\iteracro</code> , <u>3044</u> , 3048
<code>\gmu@whonly</code> , <u>3217</u> , 3218	<code>\if@afterindent</code> , 1996	<code>\justified</code> , <u>3703</u>
<code>\gmu@xedekfracplain</code> , 2501, <u>2550</u>	<code>\if@files</code> , 2928, 3183, 3195, 3203	<code>\labelsep</code> , 2256, 2274
<code>\gmu@xedekfracstar</code> , 2501, <u>2516</u>	<code>\if@gmu@mmhbox</code> , 2545, 2555, <u>2559</u> , 3125, 3907	<code>\labelwidth</code> , 2255, 2256, 2273, 2274
<code>\gmu@xfracdef</code> , <u>2517</u> , 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539	<code>\if@mainmatter</code> , 1794	<code>\larger</code> , <i>p.</i> 16, <u>1170</u> , 2649, 2654, 2718, 2719, 2722, 2723, 2724, 2725
<code>\gn@melet</code> , <u>1648</u>	<code>\if@nobreak</code> , 1993	<code>\largerr</code> , <i>p.</i> 16, <u>1174</u> , 2720, 2721
<code>\gobble</code> , 190, 192, 2738	<code>\if@openright</code> , 1796	<code>\LaTeXe</code> , 2288, <u>2337</u>
<code>\gobbletwo</code> , <u>193</u>	<code>\If@SomethingTF</code> , 666, 668, 671, <u>673</u> , 685	<code>\LaTeXpar</code> , <u>2348</u>
<code>\grab@default</code> , 576, 579, <u>592</u>	<code>\if@specialpage</code> , 1857	<code>\ldate</code> , 3664, 3673
<code>\grab@ms</code> , <u>622</u>	<code>\if@twoside</code> , 1883	<code>\leftarrow</code> , 2669, 2796
<code>\grefstepcounter</code> , 318, 334	<code>\IfBooleanF</code> , <u>712</u>	<code>\leftmargin</code> , 2253, 2271
<code>\grelaxen</code> , <u>399</u> , 399, 1992	<code>\IfBooleanT</code> , <u>708</u>	<code>\leftrightharrow</code> , 2671
<code>\hathat</code> , <u>1389</u>	<code>\IfBooleanTF</code> , <u>700</u> , 709, 713	<code>\leftslanting</code> , <u>3787</u>
<code>\HeadingNumber</code> , 1868, 1870	<code>\ifcsname</code> , 414, 1061	<code>\linebreak</code> , 3753
<code>\HeadingNumberedfalse</code> , 1794, 1839	<code>\ifdate</code> , <u>3637</u> , 3640, 3649	<code>\linedate</code> , <u>3647</u> , 3663, 3667
<code>\HeadingRHeadText</code> , <u>1852</u>	<code>\ifdefined</code> , 168, 198, 431, 480, 1191, 1250, 2465, 2748, 3030, 3355, 3437	<code>\linedate@</code> , 3647, 3648, <u>3649</u>
<code>\HeadingText</code> , 1854	<code>\iffontchar</code> , 2518, 3476	<code>\linedate@@</code> , <u>3647</u> , <u>3648</u>
<code>\HeadingTOCText</code> , <u>1853</u>	<code>\ifgmu@dash</code> , <u>3594</u> , 3600, 3606, 3641	<code>\list</code> , 2252, 2270
	<code>\ifgmu@postsec</code> , 2011, <u>2050</u> , 2058	<code>\listparindent</code> , 2257, 2275
	<code>\ifgmu@SMglobal</code> , <u>1450</u> , 1474, 1481, 1514, 1551, 1557, 1616	<code>\lit</code> , <u>3806</u>
	<code>\ifHeadingNumbered</code> , 1838, 1866	<code>\litshape</code> , <u>3788</u> , 3804, 3806, 3828
	<code>\IfNoValueF</code> , <u>694</u> , 696	<code>\liturgiques</code> , 3004
	<code>\IfNoValueT</code> , <u>693</u> , 697	<code>\longpauza</code> , <u>3427</u> , 3428
	<code>\IfNoValueTF</code> , <u>692</u> , 695	<code>\looseness</code> , 3334, 3345
	<code>\ifodd</code> , 1751	<code>\lpauza</code> , <u>3395</u>
	<code>\ifSecondClass</code> , <u>3104</u>	<code>\lsl</code> , <u>3809</u>
		<code>luzniej</code> , 3338
		<code>luzniej*</code> , 3343
		<code>\luzniejcore</code> , <u>3330</u> , 3338
		<code>\macro</code> , 2446
		<code>\MakeUppercase</code> , 3232
		<code>\mapsto</code> , 2789

<code>\marg</code> , 1410, 1434	<code>\mw@runinheading</code> , 1875, 2072	<code>\PackageWarning</code> , 1162, 1165
<code>maszynopis</code> , 3685	<code>\mw@secdef</code> , 1935, 1936, 1937, 1943	<code>\pagebreak</code> , 1882, 1894, 1898
<code>\math@arg</code> , 1430, 1431	<code>\mw@section</code> , 1934	<code>\pagegoal</code> , 2940
<code>\mathbin</code> , 2638, 2642, 2667, 2668, 2701, 2702, 2735, 2736, 2791, 2798	<code>\mw@sectionxx</code> , 1835	<code>\pagetotal</code> , 2941
<code>\mathchoice</code> , 2646, 2665, 2680, 2727, 2787	<code>\mw@secundef</code> , 1939, 1951, 1954	<code>\paragraph</code> , 3669
<code>\mathclose</code> , 2699, 2700, 2719, 2721, 2723, 2725, 2733	<code>\mw@setflags</code> , 1940	<code>\ParanoidPostsec</code> , 2047
<code>\mathfrak</code> , 3040	<code>\n@melet</code> , 1640, 1965, 1967, 2182, 2188, 2222, 2521	<code>\parg</code> , 1422
<code>\mathit</code> , 2624	<code>\nameshow</code> , 824	<code>\partial</code> , 2640
<code>\mathop</code> , 2646	<code>\nameshowthe</code> , 825	<code>\partopsep</code> , 2253, 2271
<code>\mathopen</code> , 2685, 2700, 2718, 2720, 2722, 2724, 2732	<code>napapierki</code> , 3321	<code>\pauza</code> , 3378
<code>\mathord</code> , 2704, 2705, 2706	<code>\napapierkicore</code> , 3318, 3322	<code>\pauza@skipcore</code> , 3356, 3367, 3368
<code>\mathpunct</code> , 2684	<code>\napapierkistretch</code> , 3316, 3319	<code>\pauzacore</code> , 3357, 3369, 3374, 3382, 3391, 3396, 3427, 3430, 3697, 3698
<code>\mathrel</code> , 2639, 2643, 2669, 2670, 2671, 2701, 2702, 2703, 2739, 2790, 2796, 2797	<code>\nawj</code> , 3347	<code>\pauzadial</code> , 3384, 3390
<code>\mathrm</code> , 2626, 2632, 2638, 2639, 2641, 2642, 2643, 2657	<code>\nazwired</code> , 3482	<code>\pdef</code> , 183, 197, 260, 277, 296, 297, 318, 333, 341, 894, 930, 971, 1135, 1174, 1175, 1275, 1346, 1353, 1368, 1385, 1389, 1452, 1463, 1506, 1543, 1565, 1587, 1591, 1786, 2308, 2493, 2540, 2551, 2562, 2604, 2620, 2815, 2829, 2837, 2951, 2953, 2955, 3045, 3083, 3153, 3254, 3363, 3378, 3390, 3395, 3404, 3416, 3473, 3600, 3639, 3647, 3648, 3649, 3663, 3664, 3676, 3679, 3788, 3803, 3806, 3809, 3825, 3834, 3840, 3892, 3897
<code>\Mathstrutbox@</code> , 2697	<code>\neg</code> , 2638, 2741	<code>\pdfTeX</code> , 2399
<code>\medmuskip</code> , 1329	<code>\neq</code> , 2639, 2736	<code>\pdfTeX</code> , 2401
<code>\meta</code> , 1275, 1311, 1410, 1417, 1424	<code>\neqb</code> , 2736	<code>\perhaps@grab@ms</code> , 565, 621
<code>\meta@font@select</code> , 1286, 1305	<code>\newcount</code> , 3328	<code>\Phi</code> , 2632
<code>\mkern</code> , 2742	<code>\newcounter</code> , 1747, 1782	<code>\phi</code> , 2631
<code>\mskip</code> , 1329	<code>\newgif</code> , 260	<code>\pi</code> , 2630
<code>\multiply</code> , 2327, 2330, 3283, 3333, 3344	<code>\newskip</code> , 3760	<code>\pk</code> , 1368
<code>\mw@getflags</code> , 2014	<code>\newtoks</code> , 481, 484	<code>\PlainTeX</code> , 2387
<code>\mw@HeadingBreakAfter</code> , 1859, 1879, 1894, 1898, 1929, 2015	<code>\newwrite</code> , 2928	<code>\pm</code> , 2641, 2642
<code>\mw@HeadingBreakBefore</code> , 1856, 1928, 2016	<code>\next@ddc</code> , 576, 579, 587, 591, 596	<code>\polskadata</code> , 3496, 3537
<code>\mw@HeadingLevel</code> , 1836, 1839	<code>\nfss@text</code> , 1284	<code>\possfil</code> , 1394
<code>\mw@HeadingRunIn</code> , 1874, 1928	<code>\nieczer</code> , 3012	<code>\ppauza</code> , 3416
<code>\mw@HeadingType</code> , 1855, 1990, 2022, 2023, 2036	<code>\nobreakspace</code> , 3249	<code>\ppauza@skipcore</code> , 3359, 3410, 3411
<code>\mw@HeadingWholeWidth</code> , 1877, 1929	<code>\nocite</code> , 1714	<code>\pprovide</code> , 212, 3039
<code>\mw@normalheading</code> , 1881, 1890, 1893, 1897, 2071	<code>\nohy</code> , 3146	
<code>\mw@processflags</code> , 1930	<code>\nolimits</code> , 2661, 2662	
	<code>\nolinkurl</code> , 3922	
	<code>NoNumSecs</code> , 1782	
	<code>\not@onlypreamble</code> , 1669, 1673, 1674, 1675, 1676, 1677	
	<code>\NoValue</code> , 558, 690, 691, 692, 693, 694	
	<code>\NoValueInIt</code> , 691	
	<code>\nu</code> , 2629	
	<code>\numexpr</code> , 3467, 3468, 3863	
	<code>\oarg</code> , 1415	
	<code>\oldLaTeX</code> , 2287	
	<code>\oldLaTeXe</code> , 2288	
	<code>\omega</code> , 2636	
	<code>\PackageError</code> , 608, 1688, 3405, 3417	

<code>\prependtomacro</code> , 374	<code>\SetTwoheadSkip</code> , 2074 , 2101 , 2109 , 2116	<code>\textsuperscript</code> , 2599 , 2604
<code>\printsaces</code> , 1337 , 1346	<code>\sfname</code> , 1346 , 1356	<code>\texttilde</code> , 2837
<code>\protected</code> , 183 , 212 , 272 , 291 , 497 , 2811 , 2813	<code>\shortpauza</code> , 3429	<code>\textwidth</code> , 3731 , 3732 , 3750 , 3751
<code>\provide</code> , 197 , 212	<code>\shortthousep</code> , 3896	<code>\thedata</code> , 3673
<code>\psi</code> , 2635	<code>\showboxbreadth</code> , 820	<code>\thickmuskip</code> , 2743
<code>\quad</code> , 3482	<code>\showboxdepth</code> , 820	<code>\thous</code> , 3897
<code>\quantifierhook</code> , 2758 , 2779	<code>\ShowFont</code> , 2989	<code>\thous@inner</code> , 3899 , 3902
<code>\rdate</code> , 3663	<code>\showlists</code> , 820	<code>\thousep</code> , 3840 , 3892
<code>\real</code> , 3277 , 3279	<code>\showthe</code> , 825	<code>\thr@@</code> , 2247 , 2265
<code>\reflectbox</code> , 2407 , 2414	<code>\sigma</code> , 2633	<code>\time</code> , 3467 , 3468
<code>\relaxen</code> , 394 , 394 , 1925 , 2623 , 3447 , 3849	<code>\sim</code> , 2643	<code>\tinycas</code> , 3259
<code>\relsize</code> , <i>p.</i> 16 , 1135 , 1136 , 1170 , 1171 , 1172 , 1173 , 1174 , 1175	<code>\SliTeX</code> , 2384	<code>\TODO</code> , 2888
<code>\renewcommand*</code> , 3029	<code>\smaller</code> , <i>p.</i> 16 , 1171 , 3120 , 3124	<code>\toks</code> , 2061 , 2062 , 2068 , 2069
<code>\RequirePackage</code> , 2474 , 2925 , 3022 , 3261	<code>\smallerr</code> , <i>p.</i> 16 , 1175 , 3292	<code>\tolerance</code> , 3333 , 3344 , 3689
<code>\resetMathstrut@</code> , 2690	<code>\smallskipamount</code> , 2870 , 2871	<code>\truetextsuperscript</code> , 2601 , 2603
<code>\resizebox</code> , 2594	<code>\smartunder</code> , 1214	<code>\twocoltoc</code> , 2924 , 2933
<code>\resizegraphics</code> , 2591	<code>\SMglobal</code> , 1452	<code>\twopar</code> , 3727
<code>\Restore@Macro</code> , 1546 , 1549 , 1569 , 1579	<code>\something@in</code> , 666 , 667 , 670 , 675	<code>\twoparinit</code> , 3725
<code>\Restore@Macros</code> , 1565 , 1567	<code>\something@tmp</code> , 674 , 675 , 679	<code>\tytul</code> , 3679
<code>\Restore@MacroSt</code> , 1547 , 1555	<code>\something@tmpb</code> , 678 , 679	<code>tytulowa</code> , 3479
<code>\RestoreEnvironment</code> , 1591	<code>\Store@Macro</code> , 1469 , 1472 , 1509	<code>\udigits</code> , 2493 , 2496
<code>\RestoreMacro</code> , 1543 , 2481 , 2483	<code>\Store@Macros</code> , 1506 , 1507	<code>\undeksmallskip</code> , 2871
<code>\RestoreMacro*</code> , 1593 , 2483	<code>\Store@MacroSt</code> , 1470 , 1479	<code>\unexpanded</code> , 179 , 376 , 791 , 792 , 796 , 797 , 800 , 1966
<code>\RestoreMacros</code> , 1565	<code>\Stored@Macro</code> , 1578 , 1579	<code>\unless</code> , 443 , 480 , 2466 , 3030
<code>\RestoringDo</code> , 1628	<code>\storedcsname</code> , 1582 , 3922	<code>\upshape</code> , 3829
<code>\rightarrow</code> , 2670 , 2797	<code>\StoredMacro</code> , 1578	<code>\Url</code> , 2748 , 2749
<code>\rightline</code> , 3663 , 3749	<code>\StoreEnvironment</code> , 1587	<code>\url</code> , 3921 , 3922
<code>\romannumeral</code> , 2251 , 2269	<code>\StoreMacro</code> , 1463 , 2341 , 2473 , 3697 , 3921	<code>\urladdstar</code> , 3918 , 3925
<code>\rotatebox</code> , 2659 , 2662 , 2667 , 2668	<code>\StoreMacro*</code> , 1589 , 2342	<code>\usc</code> , 3083 , 3085
<code>\rs@size@warning</code> , 1154 , 1159 , 1162	<code>\StoreMacros</code> , 1506	<code>\uscacro</code> , 3085
<code>\rs@unknown@warning</code> , 1149 , 1165	<code>\StoringAndRelaxingDo</code> , 1608	<code>\usecounter</code> , 2258
<code>\runindate</code> , 3668	<code>\strip@pt</code> , 3792 , 3797 , 3814	<code>\value</code> , 1751
<code>\scantokens</code> , 2678	<code>\subs</code> , 1186 , 1216	<code>\varepsilon</code> , 2339 , 2396 , 2627
<code>\scshape</code> , 2385 , 2791 , 3075	<code>\sum</code> , 2658	<code>\varnothing</code> , 2792
<code>\secondclass</code> , 3103	<code>\TB</code> , 2393	<code>\varsigma</code> , 2634
<code>\SecondClasstrue</code> , 3105	<code>\TeXbook</code> , 2392 , 2393	<code>\vartheta</code> , 2628
<code>\SetSectionFormatting</code> , 1925 , 1926 , 2090 , 2094 , 2102 , 2110 , 2117 , 2124 , 2129	<code>\textbullet</code> , 3473 , 3477	<code>\vee</code> , 2667 , 2741
	<code>\textcolor</code> , 3012	<code>\verb</code> , 2473 , 2481
	<code>\textlarger</code> , 1172	<code>\visiblespace</code> , 1251 , 1253 , 1331
	<code>\textlit</code> , 3803	<code>\Vs</code> , 2811
	<code>\textsl</code> , 2392	<code>\vs</code> , 1331 , 1337 , 1341
	<code>\textsmaller</code> , 1173	<code>\wd</code> , 2319 , 2322 , 2352 , 2357 , 2365 , 2366 , 2661 , 2782 , 2783 , 2794 , 2795
	<code>\textstyle</code> , 2339	<code>\Web</code> , 2389

<code>\wedge</code> , 2668, 2741	<code>\XeTeX</code> , 2404	<code>\xiiunder</code> , 1189, 1192, 1193
<code>\whenonly</code> , 3216	<code>\XeTeXinputencoding</code> , 169	<code>\xparsed@args</code> , 484, 508, 515
<code>\whern</code> , 3756	<code>\XeTeXthree</code> , 2470	<code>\xxt@visiblespace</code> ,
<code>\wherncore</code> , 3748, 3757, 3763	<code>\XeTeXversion</code> , 158, 159,	1250, 1251
<code>\whernskip</code> , 3757, 3760, 3761	168, 1191, 2465, 2466,	
<code>\whernup</code> , 3763	3355, 3437	<code>\yeshy</code> , 3147
<code>\WPheadings</code> , 2089	<code>\xiiand</code> , 1237	<code>\z@skip</code> , 3716
<code>\Ws</code> , 2813	<code>\xiibackslash</code> , 1224, 1228	<code>\zf@scale</code> , 3267, 3270, 3271
<code>\wyzejnizej</code> , 3300	<code>\xiilbrace</code> , 1199	<code>\zwrobcy</code> , 3676
<code>\Wz</code> , 2815	<code>\xiipercents</code> , 1233	
	<code>\xiirbrace</code> , 1200	<code>\cdot</code> , 2955
<code>\xathousep</code> , 3892, 3906	<code>\xiispace</code> , 837, 838, 1240,	<code>\-</code> , 3404, 3443
<code>\Xedekfracc</code> , 2501	1253	<code>\-</code> , 3363, 3424, 3442
<code>\XeLaTeX</code> , 2411	<code>\xiistring</code> , 836	