

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.lpp1.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.

```
82 \NeedsTeXFormat{LaTeX2e}
83 \ProvidesPackage{gmutils}
84 [2008/08/30_v0.93_some_rather_TeXnical_macros,_some_of_them_
tricky_(GM)]
```

Contents

Intro	2	Third person pronouns	19
Installation	2	Improvements to mwcls sectioning	
Contents of the gmutils.zip archive	2	commands	20
Compiling of the documentation	2	An improvement of MW's	
A couple of abbreviations	3	\SetSectionFormatting	22
\firstofone and the queer		Negative \addvspace	23
\catcodes	3	My heading setup for mwcls	25
Global Boolean switches	4	Compatibilising standard and mwcls	
Ampmulex Compressa-like		sectionings	26
modifications of macros	6	enumerate* and itemize*	28
\@ifnextcat, \@ifnextac	7	The logos	28
\afterfi and pals	9	Expandable turning stuff all into 'other'	31
Environments redefined	10	Brave New World of X_YT_EX	31
Almost an environment or		Fractions	32
redefinition of \begin	10	\resizegraphics	33
\@ifenvir and improvement of \end	10	Varia	34
From relsize	11	\@ifempty	38
Some 'other' stuff	12	\include not only .tex's	38
Metasymbols	13	Faked small caps	39
Macros for printing macros and		See above/see below	40
filenames	14	luzniej and napa-	
Storing and restoring the meanings		pierki—environments	
of cses	16	used in page breaking for money	40
Not only preamble!	19	Settings for mathematics in main font	43

* This file has version number v0.93 dated 2008/08/30.

Typesetting dates in my memoirs	46	Storing and restoring the catcodes of specials	53
Minion and Garamond Premier kerning and ligature fixes	50	hyperref's <code>\nolinkurl</code> into <code>\url*</code> . . .	53
A left-slanted font	51	Change History	53
Thousand separator	52	Index	55

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 branch. Creating a texmf/tex/latex/gm directory may be advisable if you consider using other packages written by me.

Then you should refresh your T_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.

```

156 \ifx\XeTeXversion\relax
157   \let\XeTeXversion\@undefined% If someone earlier used \@ifundefined{%
      % XeTeXversion} to test whether the engine is XeTeX, then \XeTeXversion
      is defined in the sense of  $\varepsilon$ -TeX tests. In that case we \let it to something
      really undefined. Well, we might keep sticking to \@ifundefined, but it's
      a macro and it eats its arguments, freezing their catcodes, which is not what
      we want in line 2939.
164 \fi
166 \ifdefined\XeTeXversion
167 \XeTeXinputencoding=utf-8% we use Unicode dashes later in this file.
168 \fi% and if we are not in XeTeX, we skip them thanks to XeTeX-test.

```

A couple of abbreviations

```

\@xa 174 \let\@xa\expandafter
\@nx 175 \let\@nx\noexpand
\pdef 178 \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 185 \long\def\gobble#1{}
\@gobble 187 \let\@gobble\gobble
\gobbletwo 188 \let\gobbletwo\@gobbletwo

\provide 192 \long\pdef\provide#1{%
193   \ifdefined#1%
194     \ifx\relax#1\afterfifi{\def#1}%
195     \else\afterfifi{\gmu@gobdef}%
196     \fi
197   \else\afterfi{\def#1}%
198   \fi}

\gmu@gobdef 201 \long\def\gmu@gobdef#1#2{%
202   \def\gmu@tempa{}}% it's a junk macro assignment to absorb possible prefixes.
204   \@gobble}

\pprovide 207 \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 ex-
pandable. If the condition(s) is false (#1 is defined) then the prefixes are absorbed by
a junk assignment.

Note moreover that unlike LATEX's \providecommand, our \{(p)provide allow any
parameters string just like \def (because they just expand to \def).

\@nameedef 220 \long\def\@nameedef#1#2{%
221   \@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.

```

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

```

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

```
\foone 237 \long\def\foone#1{\bgroup#1\egroupfirstofone}
239 \long\def\egroupfirstofone#1{\egroup#1}
\foeatletter 241 \long\def\foeatletter{\foone\makeatletter}
```

Global Boolean switches

The `\newgif` declaration's effect is used even in the \LaTeX 2_ϵ source by redefining some particular user defined ifs (UD-ifs henceforth) step by step. The goal is to make the UD-ifs assignment global. I needed it at least twice during `gmdoc` writing so I make it a macro. It's an almost verbatim copy of \LaTeX 's `\newif` modulo the letter `g` and the `\global` prefix. (File `d:ltdefns.dtx` Date: 2004/02/20 Version v1.3g, lines 139–150)

```
\newgif 255 \pdef\newgif#1{%
256   {\escapechar\m@ne
257     \global\let#1\iffalse
258     \@gif#1\iftrue
259     \@gif#1\iffalse
260   }}
```

'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 266 \def\@gif#1#2{%
267   \protected\@xa\gdef\csname\@xa\@gobbletwo\string#1%
268   g% the letter g for '\global'.
269   \@xa\@gobbletwo\string#2\endcsname
270   {\global\let#1#2}}
272 \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.
279   \count@\escapechar\escapechar\m@ne
280   \let#1\iffalse
281   \@if#1\iftrue
282   \@if#1\iffalse
283   \escapechar\count@}
\@if 285 \def\@if#1#2{%
286   \protected\@xa\def\csname\@xa\@gobbletwo\string#1%
287   \@xa\@gobbletwo\string#2\endcsname
288   {\let#1#2}}
\hidden@iffalse 291 \pdef\hidden@iffalse{\iffalse}
\hidden@iftrue 292 \pdef\hidden@iftrue{\iftrue}
```

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

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

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

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

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

```
\grefstepcounter 313 \pdef\grefstepcounter#1{%
314   {\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 328 \pdef\hgrefstepcounter#1{%
329   \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 336 \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 `\expandafte`s e.g.

```
\glet 346 \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 354 \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 359 \long\def\addto@macro#1#2{%
360   \toks@{\@xa{#1#2}}%
361   \edef#1{\the\toks@}%
362 }% (\toks@ is a scratch register, namely \tokso.)
```

And for use in the very document,

```
\addtomacro 366 \let\addtomacro=\addto@macro
```

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

```
\prependtomacro 369 \long\def\prependtomacro#1#2{%
370   \edef#2{\unexpanded{#1}\@xa\unexpanded\@xa{#2}}}
```

Note that `\prependtomacro` can be prefixed.

```
\addtotoks 374 \long\def\addtotoks#1#2{%
375   #1=\@xa{\the#1#2}}
```

```
\@emptyify 378 \newcommand*\@emptyify[1]{\let#1=\@empty}
\emptyify 379 \@ifdefinable\emptyify{\let\emptyify\@emptyify}
```

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

```
\g@emptyify 383 \newcommand*\g@emptyify{\global\@emptyify}
```

```

\gemptyify 384 \@ifdefinable\gemptyify{\let\gemptyify\g@emptyify}
\@relaxen 387 \newcommand\@relaxen[1]{\let#1=\relax}
\relaxen 388 \@ifdefinable\relaxen{\let\relaxen\@relaxen}

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

\g@relaxen 392 \newcommand*\g@relaxen{\global\@relaxen}
\grelaxen 393 \@ifdefinable\grelaxen{\let\grelaxen\g@relaxen}

```

Ampulex Compressa-like modifications of macros

Ampulex Compressa is a wasp that performs brain surgery on its victim cockroach to lead it to its lair and keep alive for its larva. Well, all we do here with the internal L^AT_EX 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 #3 and #4) with the replacement #5. 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 with the `\ampulexset` declaration. If `\ampulexdef` doesn't find the start and end tokens in the meaning of the macro, it does nothing to it. For an example use see line 1373.

```

\ampulexdef 411 \newcommand\ampulexdef[5][\relax]{%
    % #1 definition prefix,
    % #2 macro to be redefined,
    % #3 start token(s),
    % #4 end token(s)
    % #5 replacement.

\gmu@tempa 420 \def\gmu@tempa{#3}%
\gmu@tempb 421 \def\gmu@tempb{#4}%
\gmu@tempc 422 \def\gmu@tempc{#5}% we wrap the start, end and replacement tokens in macros
    to avoid unbalanced \ifs.

424 \edef\gmu@tempd{%
425     \long\def\@nx\gmu@tempd
426     #####1\all@other\gmu@tempa
427     #####2\all@other\gmu@tempb
428     #####3\@nx\gmu@tempd{%
429         \@ifempty{#####3}{\hidden@iffalse}{\hidden@iftrue}}}%
431 \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.

435 \edef\gmu@tempe{%
436     \@nx\gmu@tempd\all@other#2%
437     \all@other\gmu@tempa
438     \all@other\gmu@tempb\@nx\gmu@tempd
439 }%
441 \gmu@tempe% we apply the checker and it produces an open \if.
443 \edef\gmu@tempd{%
444     \long\def\@nx\gmu@tempd
445     #####1\@xa\unexpanded\@xa{\gmu@tempa}%
446     #####2\@xa\unexpanded\@xa{\gmu@tempb}%
447     #####3\@nx\ampulexdef{% we define a temporary macro with the parameters
        delimited with the 'start' and 'end' parameters of \ampulexdef.
450         \@nx\unexpanded{#####1}%
451         \@nx\@xa\@nx\unexpanded

```

452 \@nx\@xa{\@nx\gmu@tempc}% we replace the part of the redefined macro's
 meaning with the replacement text.

454 \@nx\unexpanded{####3}}}%

455 \gmu@tempd

457 \edef\gmu@tempe{%

458 #1\def\@nx#2\gmu@ampulexpa{%

459 \@xa\@xa\@xa\gmu@tempd\@xa#2\gmu@ampulexpb

460 \ampulexdef}}}%

461 \gmu@tempe

462 \fi}

\ampulexset 465 \long\def\ampulexset#1#2{%

\gmu@ampulexpa 466 \def\gmu@ampulexpa{#1}% it's the parameter string for definition

\gmu@ampulexpb 467 \def\gmu@ampulexpb{#2}% it's the arguments string for the first expansion. For
 the example of usage see 1373.

469 }

471 \ampulexset{}{}

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

\nameshow 483 \newcommand\nameshow[1]{\@xa\show\csname#1\endcsname}

\nameshowthe 484 \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 495 \def\xiistring#1{%
 496 \if\@nx#1\xiispace
 497 \xiispace
 498 \else
 499 \string#1%
 500 \fi}

\@ifnextcat, \@ifnextac

As you guess, we \def \@ifnextcat à la \@ifnextchar, see L^AT_EX 2_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 517 \long\def\@ifnextcat#1#2#3{%
 521 \def\reserved@d{#1}%
 522 \def\reserved@a{#2}%

```

523 \def\reserved@b{#3}%
524 \futurelet\@let@token\@ifncat}
\@ifncat 527 \def\@ifncat{%
528 \ifx\@let@token\@sptoken
529 \let\reserved@c\@xifncat
530 \else
531 \ifcat\reserved@d\@nx\@let@token
532 \let\reserved@c\reserved@a
533 \else
534 \let\reserved@c\reserved@b
535 \fi
536 \fi
537 \reserved@c}

```

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

```

542 \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 553 \long\pdef\@ifnextif#1#2#3{%
557 \def\reserved@d{#1}%
558 \def\reserved@a{#2}%
559 \def\reserved@b{#3}%
560 \futurelet\@let@token\@ifnif}
\@ifnif 563 \def\@ifnif{%
564 \ifx\@let@token\@sptoken
565 \let\reserved@c\@xifnif
566 \else
567 \if\reserved@d\@nx\@let@token
568 \let\reserved@c\reserved@a
569 \else
570 \let\reserved@c\reserved@b
571 \fi
572 \fi
573 \reserved@c}
576 {\def\:{\let\@sptoken= }\:}% this makes \@sptoken a space token.
578 \def\:{\@xifnif}\@xa\gdef\:{\futurelet\@let@token\@ifnif}}

```

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

```

\@ifnextac 589 \long\pdef\@ifnextac#1#2{%
590 \@ifnextcat\bgroup{#2}{\gm@ifnac{#1}{#2}}}
\gm@ifnac 592 \long\def\gm@ifnac#1#2#3{%
593 \ifcat\@nx~\@nx#3\afterfi{#1#3}\else\afterfi{#2#3}\fi}

```

Yes, it won't work for an active char \let to {₁, but it *will* work for an active char \let to a char of catcode ≠ 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 `=`.

```

606 \let\gmu@reserveda\*%
\* 607 \def\*{%
608   \let\*\gmu@reserveda
609   \let\gm@letspace=\_10}%
610 \*\_10%
\ifnextspace 613 \def\@ifnextspace#1#2{%
614   \let\gmu@reserveda\*%
\* 615   \def\*{%
616     \let\*\gmu@reserveda
617     \ifx\@let@token\gm@letspace\afterfi{#1}%
618     \else\afterfi{#2}%
619     \fi}%
620   \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.

```

629 \foone\obeylines{%
\@ifnextMac 630   \long\pdef\@ifnextMac#1#2{%
631     \@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 T_EX 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 656 \long\def\afterfi#1#2\fi{\fi#1}

```

And two more of that family:

```

\afterfifi 658 \long\def\afterfifi#1#2\fi#3\fi{\fi\fi#1}
\afteriffifi 659 \long\def\afteriffifi#1#2\if#3\fi#4\fi{\fi#1}

```

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

```

\afteriffiffifi 663 \long\def\afteriffiffifi#1#2\fi#3\fi#4\fi{\fi#1}
\afteriffifi 664 \long\def\afteriffifi#1#2\fi#3\fi#4\fi{\fi\fi#1}
\afterfifi 665 \long\def\afterfifi#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 L^AT_EX's `\begin`:

```
\def\begin#1{%
  \@ifundefined{#1}%
    {\def\reserved@a{\@latex@error{Environment #1
      undefined}\@eha}}%
    {\def\reserved@a{\def\@currenvir{#1}%
      \edef\@currenvline{\on@line}%
      \csname #1\endcsname}}%
  \@ignorefalse
  \begingroup\@endpefalse\reserved@a}
```

```
\@begnamedgroup 696 \long\def\@begnamedgroup#1{%
697   \@ignorefalse% not to ignore blanks after group
698   \begingroup\@endpefalse
699   \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 751.
703   \edef\@currenvline{\on@line}%
704   \csname_#1\endcsname}% if the argument is a command's name (an environ-
      ment's e.g.), this command will now be executed. (If the corresponding
      control sequence hasn't been known to TEX, this line will act as \relax.)
```

For back compatibility with my earlier works

```
\bnamegroup 712 \let\bnamegroup\@begnamedgroup
```

And for the ending

```
\enamegroup 714 \def\enamegroup#1{\end{#1}}
```

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

```
\begin* 720 \def\begin{\@ifstar{\@begnamedgroup}{%
\begin 721   \@begnamedgroup@ifcs}}
```

```
\@begnamedgroup@ifcs 724 \def\@begnamedgroup@ifcs#1{%
725   \ifcsname#1\endcsname\afterfi{\@begnamedgroup{#1}}%
726   \else\afterfi{\@latex@error{Environment_#1_undefined}\@eha}%
727   \fi}%
```

`\@ifenvir` and improvement of `\end`

It's very clever and useful that `\end` checks whether its argument is `\ifx`-equivalent `\@currenvir`. 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 751 \long\def\@ifenvir#1#2#3{%
753   \edef\gmu@reserveda{\@xa\string\csize\@currentenv\endcsize}%
754   \edef\gmu@reservedb{\@xa\string\csize#1\endcsize}%
755   \ifx\gmu@reserveda\gmu@reservedb\afterfi{#2}%
756   \else\afterfi{#3}%
757   \fi}
\@checkend 759 \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 == \relsize{1}. \smallerr(my addition) == \relsize{-2}; \largerr
\smallerr guess yourself.
\largerr (Since \DeclareRobustCommand doesn't issue an error if its argument has been de-
fined and it only informs about redefining, loading relsize remains allowed.)

\relsize 799 \pdef\relsize#1{%
800   \ifmmode_\@nomath\relsize\else
801     \begingroup
802     \@tempcnta_ % assign number representing current font size
803     \ifx\@currsize\normalsize_4\else_ _ % funny order is to have most
...
804     \ifx\@currsize\small_3\else_ _ _ _ _ % ...likely sizes checked first
805     \ifx\@currsize\footnotesize_2\else
806     \ifx\@currsize\large_5\else
807     \ifx\@currsize\Large_6\else
808     \ifx\@currsize\LARGE_7\else
809     \ifx\@currsize\scriptsize_1\else
810     \ifx\@currsize\tiny_0\else
811     \ifx\@currsize\huge_8\else
812     \ifx\@currsize\Huge_9\else
813     4\rs@unknown@warning_ % unknown state: \normalsize as
starting point
814     \fi\fi\fi\fi\fi\fi\fi\fi\fi
```

Change the number by the given increment:

```

816      \advance\@tempcnta#1\relax
      watch out for size underflow:
818      \ifnum\@tempcnta<\z@\rs@size@warning{small}{\string\tiny}%
          \@tempcnta\z@\fi
819      \@xa\endgroup
820      \ifcase\@tempcnta\% set new size based on altered number
821      \tiny\or\scriptsize\or\footnotesize\or\small\or\%
          \normalsize\or
822      \large\or\Large\or\LARGE\or\huge\or\Huge\else
823      \rs@size@warning{large}{\string\Huge}\Huge
824      \fi\fi}% end of \relsize.

\rs@size@warning 826 \providecommand*\rs@size@warning[2]{\PackageWarning{gmutils
          (relsize)}{%
827   Size requested is too #1.\MessageBreak Using #2 instead}}

\rs@unknown@warning 829 \providecommand*\rs@unknown@warning{\PackageWarning{gmutils
          (relsize)}{Current font size
830   is unknown!\(Why?!\)\MessageBreak Assuming\string\normalsize}}

      And a handful of shorthands:
      \larger 834 \DeclareRobustCommand*\larger[1][\@one]{\relsize{+#1}}
      \smaller 835 \DeclareRobustCommand*\smaller[1][\@one]{\relsize{-#1}}
      \textlarger 836 \DeclareRobustCommand*\textlarger[2][\@one]{\relsize{+#1}#2}}
      \textsmaller 837 \DeclareRobustCommand*\textsmaller[2][\@one]{\relsize{-#1}#2}}
      \largerr 838 \pdef\largerr{\relsize{+2}}
      \smallerr 839 \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`.

```

849 \foone{\catcode`\_ =8}%
\subs 850 {\let\subs=_}

852 \foone{\@makeother\_}%
\xiiunder 853 {\def\xiiunder{_{}}

855 \ifdefined\XeTeXversion
\xiiunder 856 \def\xiiunder{\char"005F}%
857 \let\_ \xiiunder
858 \fi

860 \foone{\catcode`\ [=1\@makeother\{
861 \catcode`\ ] =2\@makeother\}}%
862 [%
\xiilbrace 863 \def\xiilbrace[{]%
\xiirbrace 864 \def\xiirbrace[}%
865 ]% 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.

```

876 \foone{\catcode`\_=\active}
877 {%
\smartunder 878 \newcommand*\smartunder{%
879 \catcode`\_=\active
880 \def_{\ifmmode\subs\else\_ \fi}}}% We define it as \_ not just as \xiunder
because some font encodings don't have _ at the \char`\_ position.

886 \foone{\catcode`\!=o
887 \@makeother\}
\xiibackslash 888 {\!newcommand*\xiibackslash{\}}
\bslash 892 \let\bslash=\xiibackslash

896 \foone{\@makeother\}%
\xiipercant 897 {\def\xiipercant{\}}

900 \foone{\@makeother\&\}%
\xiiand 901 {\def\xiiand{\&}}

903 \foone{\@makeother\_ \}%
\xiispace 904 {\def\xiispace{\_ \}}

```

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

```

913 \AtBeginDocument{%
914 \ifdefined\xxt@visiblepace
915 \let\visiblepace\xxt@visiblepace
916 \else
917 \let\visiblepace\xiispace
918 \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 939 \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."

```

947 \ensuremath\langle
948 \ifmmode\_ \@xa\_ \nfss@text\_ \fi
949 {%

```

```
950 \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.

```
958 #1\/%
960 }\ensuremath\range
961 }
```

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 969 \def\meta@font@select{\it}
```

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

```
\<...> 975 \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 986 \def\discre#1#2#3{\leavevmode\kernosp%
987 \discretionary{#1}{#2}{#3}\penalty10000\hskiposp\relax}
\discret 988 \def\discret#1{\leavevmode\kernosp%
989 \discretionary{#1}{#1}{#1}\penalty10000\hskiposp\relax}
```

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

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

```
\vs 995 \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 1001 \def\printspaces#1{\let~=\vs\let\_\=\vs\gm@pswords#1_\@nil}}
\gm@pswords 1003 \def\gm@pswords#1_\#2_\@nil{%
1004 \ifx\relax#1\relax\else#1\fi
1005 \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 1010 \pdef\sfname#1{\textsf{\printspaces{#1}}}
```

```
\gm@discretionaryslash 1012 \def\gm@discretionaryslash{\discre{/}{\hbox{}}{}}% the
second pseudo-argument nonempty to get \hyphenpenalty
not \exhyphenpenalty.
```

```
\file 1017 \pdef\file#1{\gm@printslashes#1/\gm@printslashes}
```

```
\gm@printslashes 1019 \def\gm@printslashes#1/#2\gm@printslashes{%
1020 \sfname{#1}%
```

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

```

1021 \ifx\gmu@printslashes#2\gmu@printslashes
1022 \else
1023 \textsf{\gmu@discretionaryslash}%
1024 \afterfi{\gmu@printslashes#2\gmu@printslashes}\fi}

```

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

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

```
\pk 1032 \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 1043 \DeclareRobustCommand*{\cs}[2][\bslash]{\{%
\ 1044 \def\-\{\discretionary{\rmfamily-}\}\}%
1045 \def\{\{\char`\}\}\def\{\char`\}\}\ttfamily\_#1#2}}
\env 1049 \pdef\env#1{\cs[] {#1}}

```

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

```

1052 \foone{\@makeother\^}
\hathat 1053 {\pdef\hathat#1{\cs[^^]{#1}}}
```

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

```
\possfil 1058 \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 1068 \def\cmd#1{\cs{\@xa\cmd@to@cs\string#1}}
\cmd@to@cs 1070 \def\cmd@to@cs#1#2{\char\number`#2\relax}
\marg{text} prints {\text}, 'mandatory argument'.

```

```

\marg 1074 \def\marg#1{\{\ttfamily\char`\}\meta{#1}\{\ttfamily\char`\}\}}
\oarg{text} prints [{text}], 'optional argument'. Also \oarg[text] does that.

```

```

\oarg 1079 \def\oarg{\@ifnextchar[\@oargsq\@oarg}
\@oarg 1081 \def\@oarg#1{\{\ttfamily[]\meta{#1}\{\ttfamily}\}}
\@oargsq 1082 \def\@oargsq[#1]{\@oarg{#1}}
\parg{te,xt} prints (\te,xt), 'picture mode argument'.

```

```

\parg 1086 \def\parg{\@ifnextchar(\@pargp\@parg}
\@parg 1088 \def\@parg#1{\{\ttfamily()\meta{#1}\{\ttfamily}\}}
\@pargp 1089 \def\@pargp(#1){\@parg{#1}}

```

But we can have all three in one command.

```

1093 \AtBeginDocument{%
\arg 1094 \let\math@arg\arg
\arg 1095 \def\arg{\ifmmode\math@arg\else\afterfi{%
1096 \ifnextchar[%
1097 \@oargsq{\@ifnextchar(
1098 \@pargp\marg}}\fi}%
1099 }

```

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 1114 \newif\ifgmu@SMglobal
\SMglobal 1116 \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 1127 \pdef\StoreMacro{%
1128 \bgroup\makeatletter\@ifstar\egStore@MacroSt\egStore@Macro}
```

The unstarred version takes a cs and the starred version a text, which is intended for special control sequences. For storing environments there is a special command in line 1251.

```
\egStore@Macro 1133 \long\def\egStore@Macro#1{\egroup\Store@Macro{#1}}
\egStore@MacroSt 1134 \long\def\egStore@MacroSt#1{\egroup\Store@MacroSt{#1}}
\Store@Macro 1136 \long\def\Store@Macro#1{%
1137 \escapechar92
1138 \ifgmu@SMglobal\afterfi\global\fi
1139 \@xa\let\csname/gmu/store/string#1\endcsname#1%
1140 \global\gmu@SMglobalfalse}
\Store@MacroSt 1143 \long\def\Store@MacroSt#1{%
1144 \edef\gmu@smtempa{%
1145 \ifgmu@SMglobal\global\fi
1146 \@nx\let\@xa\@nx\csname/gmu/store/bslash#1\endcsname% we add
backslash because to ensure compatibility between \StoreMacro
and \StoreMacro*, that is. to allow writing
e.g. \StoreMacro\kitten and then \RestoreMacro*\kitten to
restore the meaning of \kitten.
1152 \@xa\@nx\csname#1\endcsname}
1153 \gmu@smtempa
1154 \global\gmu@SMglobalfalse}% we wish the globality to be just once.
```

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

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

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

```
\StoreMacros 1170 \long\pdef\StoreMacros{\bgroup\makeatletter\Store@Macros}
\Store@Macros 1171 \long\def\Store@Macros#1{\egroup
1172 \gmu@setsetSMglobal
```



```

1173 \let\gml@storeCS\Store@Macro
1174 \gml@storemacros#1.}

\gmu@setsetSMglobal 1177 \def\gmu@setsetSMglobal{%
1178 \ifgmu@SMglobal
1179 \let\gmu@setSMglobal\gmu@SMglobaltrue
1180 \else
1181 \let\gmu@setSMglobal\gmu@SMglobalfalse
1182 \fi}

And the inner iterating macro:

\gml@storemacros 1185 \long\def\gml@storemacros#1{%
\gmu@reserveda 1186 \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.
1189 \if\gmu@reserveda.% a dot finishes storing.
1190 \global\gmu@SMglobalfalse
1191 \else
1192 \if\gmu@reserveda,% The list this macro is put before may contain commas
and that's O.K., we just continue the work.
1194 \afterfifi\gml@storemacros
1195 \else% what is else this shall be stored.
1196 \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.
1199 \afterfifi{\gmu@setSMglobal\gml@storemacros}%
1200 \fi
1201 \fi}

And for the restoring

\RestoreMacro 1207 \pdef\RestoreMacro{%
1208 \bgroup\makeatletter\@ifstar\egRestore@MacroSt\egRestore@Macro}

\egRestore@Macro 1210 \long\def\egRestore@Macro#1{\egroup\Restore@Macro{#1}}
\egRestore@MacroSt 1211 \long\def\egRestore@MacroSt#1{\egroup\Restore@MacroSt{#1}}

\Restore@Macro 1213 \long\def\Restore@Macro#1{%
1214 \escapechar92
1215 \ifgmu@SMglobal\afterfi\global\fi
1216 \@xa\let\@xa#1\csname_/gmu/store/string#1\endcsname
1217 \global\gmu@SMglobalfalse}

\Restore@MacroSt 1219 \long\def\Restore@MacroSt#1{%
1220 \edef\gmu@smtempa{%
1221 \ifgmu@SMglobal\global\fi
1222 \@nx\let\@xa\@nx\csname#1\endcsname
1223 \@xa\@nx\csname/gmu/store/bslash#1\endcsname}% cf. the commentary
in line 1146.
1225 \gmu@smtempa
1226 \global\gmu@SMglobalfalse}

\RestoreMacros 1229 \long\pdef\RestoreMacros{\bgroup\makeatletter\Restore@Macros}

\Restore@Macros 1231 \long\def\Restore@Macros#1{\egroup
1232 \gmu@setsetSMglobal
1233 \let\gml@storeCS\Restore@Macro% we direct the core cs towards restoring
and call the same iterating macro as in line 1174.
1236 \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 1242 \def\StoredMacro{\bgroup\makeatletter\Stored@Macro}
\Stored@Macro 1243 \long\def\Stored@Macro#1{\egroup\Restore@Macro#1#1}
```

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

```
\storedcsname 1246 \def\storedcsname#1{%
1247   \csname_/gmu/store\slash#1\endcsname}
2008/08/03 we need to store also an environment.
```

```
\StoreEnvironment 1251 \pdef\StoreEnvironment#1{%
1253   \StoreMacro*{#1}\StoreMacro*{end#1}}
```

```
\RestoreEnvironment 1255 \pdef\RestoreEnvironment#1{%
1257   \RestoreMacro*{#1}\RestoreMacro*{end#1}}
```

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

```
\StoringAndRelaxingDo 1272 \long\def\StoringAndRelaxingDo{%
1273   \gmu@SMdo@setscope
1274   \long\def\do##1{%
1275     \gmu@SMdo@scope
1276     \@xa\let\csname_/gmu/store/string##1\endcsname##1%
1277     \gmu@SMdo@scope\let##1\relax}}
\gmu@SMdo@setscope 1279 \def\gmu@SMdo@setscope{%
1280   \ifgmu@SMglobal\let\gmu@SMdo@scope\global
1281   \else\let\gmu@SMdo@scope\relax
1282   \fi
1283   \global\gmu@SMglobalfalse}
```

And here is the counter-definition for restore.

```
\RestoringDo 1292 \long\def\RestoringDo{%
1293   \gmu@SMdo@setscope
1294   \long\def\do##1{%
1295     \gmu@SMdo@scope
1296     \@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 1304 \def\n@melet#1#2{%
1305   \edef\gmu@nl@reserveda{%
1306     \let\@xa\@nx\csname#1\endcsname
1307     \@xa\@nx\csname#2\endcsname}%
1308   \gmu@nl@reserveda}
```

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

```
\gn@melet 1312 \def\gn@melet#1#2{%
```

```

1313 \edef\gmu@nl@reserveda{%
1314   \global\let\@xa\@nx\csname#1\endcsname
1315   \@xa\@nx\csname#2\endcsname}%
1316 \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.

```

\notonlypreamble 1333 \newcommand\notonlypreamble[1]{\%
1334   \def\do##1{\ifx#1##1\else\@nx\do\@nx##1\fi}%
1335   \xdef\@preamblecmds{\@preamblecmds}}
1337 \notonlypreamble\@preamblecmds
1338 \notonlypreamble\@ifpackageloaded
1339 \notonlypreamble\@ifclassloaded
1340 \notonlypreamble\@ifl@aded
1341 \notonlypreamble\@pkgextension

```

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

```

\gm@notprerr 1346 \def\gm@notprerr{\can_be_used_only_in_preamble(\on@line)}
1348 \AtBeginDocument{%
1349   \def\do#1{\@nx\do\@nx#1}%
1350   \edef\@preamblecmds{%
1351     \def\@nx\do##1{%
1352       \def##1{\@nx\PackageError{gmutils/LaTeX}%
1353         {\@nx\string##1\@nx\gm@notprerr}\@nx\@eha}}%
1354   \@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.

```

\gmu@nocite@ampulex 1371 \def\gmu@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.
1373   \ampulexset{#####1}{\#####1}}% 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.
1378   \ampulexdef\nocite\ifx
1379   {\onlypreamble\document}%
1380   \iftrue}
1382 \AtBeginDocument\gmu@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 1409 \newcounter{gm@PronounGender}
\gm@atppron 1411 \newcommand*\gm@atppron[2]{%
1412 \stepcounter{gm@PronounGender}% remember \stepcounter is global.
1413 \ifodd\value{gm@PronounGender}#1\else#2\fi}
\heshe 1415 \newcommand*\heshe{\gm@atppron{he}{she}}
\hisher 1416 \newcommand*\hisher{\gm@atppron{his}{her}}
\himher 1417 \newcommand*\himher{\gm@atppron{him}{her}}
\hishers 1418 \newcommand*\hishers{\gm@atppron{his}{hers}}
\HeShe 1420 \newcommand*\HeShe{\gm@atppron{He}{She}}
\HisHer 1421 \newcommand*\HisHer{\gm@atppron{His}{Her}}
\HimHer 1422 \newcommand*\HimHer{\gm@atppron{Him}{Her}}
\HisHers 1423 \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.

```
1480 \AtBeginDocument{% because we don't know when exactly hyperref is loaded and
maybe after this package.
NoNumSecs 1482 \@ifpackageloaded{hyperref}{\newcounter{NoNumSecs}%
1483 \setcounter{NoNumSecs}{617}% to make \refing to an unnumbered section
visible (and funny?).
\gm@hyperrefstepcounter 1485 \def\gm@hyperrefstepcounter{\refstepcounter{NoNumSecs}}%
\gm@targetheading 1486 \pdef\gm@targetheading#1{%
1487 \hypertarget{#1}{#1}}}% end of then
\gm@hyperrefstepcounter 1488 {\def\gm@hyperrefstepcounter{}%
\gm@targetheading 1489 \def\gm@targetheading#1{#1}}}% end of else
1490 }% of \AtBeginDocument
```

Auxiliary macros for the kernel sectioning macro:

```
bersectionsoutofmainmatter 1493 \def\gm@dontnumbersectionsoutofmainmatter{%
1494 \if@mainmatter\else\HeadingNumberedfalse\fi}
gm@clearpagesduetoopenright 1495 \def\gm@clearpagesduetoopenright{%
1496 \if@openright\cleardoublepage\else\clearpage\fi}
```

² A. Berg, Wozzeck.

To avoid `\def`ing 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 1509 \long\def\@ifnotmw#1#2{\@ifundefined{mw@sectionxx}{#1}{#2}}
1511 \let\gmu@def\def
\@ifnotmw 1512 \@ifnotmw{%
\gmu@def 1513 \StoreMacro\gmu@def_\def\gmu@def#1#2{\RestoreMacro\gmu@def}}{}
```

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

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

The kernel of MW's sectioning commands:

```
1532 \gmu@def\mw@sectionxx#1#2[#3]#4{%
1533 \edef\mw@HeadingLevel{\csname_\#1@level\endcsname
1534 \space}% space delimits level number!
1535 \ifHeadingNumbered
1536 \ifnum_\mw@HeadingLevel>\c@secnumdepth_\%
\HeadingNumberedfalse_\fi
```

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

```
1539 \@ifundefined{if@mainmatter}{\fi}{%
\gm@dontnumbersectionsoutofmainmatter}
1540 \fi
% \ifHeadingNumbered
% \refstepcounter{#1}%
% \protected@edef\HeadingNumber{\csname
the#1\endcsname\relax}%
% \else
% \let\HeadingNumber\@empty
% \fi
```

```
\HeadingRHeadText 1549 \def\HeadingRHeadText{#2}%
\HeadingTOCText 1550 \def\HeadingTOCText{#3}%
\HeadingText 1551 \def\HeadingText{#4}%
\mw@HeadingType 1552 \def\mw@HeadingType{#1}%
1553 \if\mw@HeadingBreakBefore
1554 \if@specialpage\else\thispagestyle{closing}\fi
1555 \@ifundefined{if@openright}{\gm@clearpagesduetoopenright}%
1556 \if\mw@HeadingBreakAfter
1557 \thispagestyle{blank}\else
1558 \thispagestyle{opening}\fi
```

```

1559         \global\@topnum\z@
1560     \fi% of \if\mw@HeadingBreakBefore
placement of \refstep suggested by me (GM):
1563     \ifHeadingNumbered
1564         \refstepcounter{#1}%
1565         \protected@edef\HeadingNumber{\csname_ the#1\endcsname\relax}%
1566     \else
1567         \let\HeadingNumber\@empty
1568         \gm@hyperrefstepcounter
1569     \fi% of \ifHeadingNumbered
1571     \if\mw@HeadingRunIn
1572         \mw@runinheading
1573     \else
1574         \if\mw@HeadingWholeWidth
1575             \if@twocolumn
1576                 \if\mw@HeadingBreakAfter
1577                     \onecolumn
1578                     \mw@normalheading
1579                     \pagebreak\relax
1580                     \if@twoside
1581                         \null
1582                         \thispagestyle{blank}%
1583                     \newpage
1584                 \fi% of \if@twoside
1585             \twocolumn
1586         \else
1587             \@topnewpage[\mw@normalheading]%
1588             \fi% of \if\mw@HeadingBreakAfter
1589         \else
1590             \mw@normalheading
1591             \if\mw@HeadingBreakAfter\pagebreak\relax\fi
1592             \fi% of \if@twocolumn
1593         \else
1594             \mw@normalheading
1595             \if\mw@HeadingBreakAfter\pagebreak\relax\fi
1596         \fi% of \if\mw@HeadingWholeWidth
1597     \fi% of \if\mw@HeadingRunIn
1598 }

```

An improvement of MW's \SetSectionFormatting

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

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

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

```

1621 \relaxen\SetSectionFormatting
\SetSectionFormatting 1622 \newcommand*\SetSectionFormatting[5][\empty]{%

```

```

\mw@HeadingRunIn 1623 \ifx\empty#1\relax\else% empty (not \empty!) #1 also launches \else.
\mw@HeadingBreakBefore 1624 \def\mw@HeadingRunIn{10}\def\mw@HeadingBreakBefore{10}%
\mw@HeadingBreakAfter 1625 \def\mw@HeadingBreakAfter{10}\def\mw@HeadingWholeWidth{10}%
\mw@HeadingWholeWidth 1626 \@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.

1629 \fi
1630 \@ifundefined{#2}{\@namedef{#2}{\mw@section{#2}}}{\%
1631 \mw@secdef{#2}{@preskip}{#3}{2\oblig.}%
1632 \mw@secdef{#2}{@head}{#4}{3\oblig.}%
1633 \mw@secdef{#2}{@postskip}{#5}{4\oblig.}%
1634 \ifx\empty#1\relax
1635 \mw@secundef{#2@flags}{1\optional)}%
1636 \else\mw@setflags{#2}%
1637 \fi}

\mw@secdef 1639 \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.
1646 \@ifempty{#3}
1647 {\mw@secundef{#1#2}{#4}}
1648 {\@namedef{#1#2}{#3}}}

\mw@secundef 1650 \def\mw@secundef#1#2{%
1651 \@ifundefined{#1}{%
1652 \ClassError{mwcls/gm}{%
1653 command\backslash#1\undefined\MessageBreak
1654 after\backslashSetSectionFormatting!!!\MessageBreak}{%
1655 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 1660 \def\addtoheading#1#2{%
1661 \n@melet{gmu@reserveda}{#1@head}%
1662 \toks\z@=\@xa{gmu@reserveda}%
1663 \toks\tw@={#2}%
1664 \edef\gmu@reserveda{\the\toks\tw@\the\toks\z@}%
1665 \n@melet{#1@head}{gmu@reserveda}%
1667 }

```

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.

```

1679 \@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 1684 \def\@afterheading{%
1685   \@nobreaktrue
1686   \xdef\gmu@prevsec{\mw@HeadingType}% added now
1687   \everypar{%
1688     \grelaxen\gmu@prevsec% added now. All the rest is original LATEX.
1689     \if@nobreak
1690     \@nobreakfalse
1691     \clubpenalty_\@M
1692     \if@afterindent_\else
1693     {\setbox\z@\lastbox}%
1694     \fi
1695     \else
1696     \clubpenalty_\@clubpenalty
1697     \everypar{}}%
1698     \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 1705 \def\gmu@checkaftersec{%
1706   \@ifundefined{gmu@prevsec}{-}{%
1707     \ifgmu@postsec% an additional switch that is true by default but may be
                        turned into an \ifdim in special cases, see line 1743.
1710     {\@xa\mw@getflags\@xa{\gmu@prevsec}%
1711      \glet\gmu@reserveda\mw@HeadingBreakAfter}%
\gmu@reserveda 1712   \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.
1715   \if\gmu@reserveda\else
1716   \penalty10000\relax
1717   \skip\z@=\csname\gmu@prevsec_\@postskip\endcsname\relax
1718   \skip\tw@=\csname\mw@HeadingType_\@preskip\endcsname\relax
1719   \@ifundefined{\mw@HeadingType_\@twoheadskip}{
1720     \ifdim\skip\z@>\skip\tw@
1721     \vskip-\skip\z@% we strip off the post-skip of previous header if it's bigger
                        than current pre-skip
1723     \else
1724     \vskip-\skip\tw@% we strip off the current pre-skip otherwise
1725     \fi}{% But if the two-header-skip is defined, we put it
1727     \penalty10000
1728     \vskip-\skip\z@
1729     \penalty10000
1730     \vskip-\skip\tw@
1731     \penalty10000
1732     \vskip\csname\mw@HeadingType_\@twoheadskip\endcsname
1733     \relax}%
1734   \penalty10000
1735   \hrule_\height\z@\relax% to hide the last (un)skip before
                        subsequent \addvspaces.
```



```

1737     \penalty10000
1738     \fi
1739     \fi
1740   }% of \@ifundefined{gmu@prevsec} 'else'.
1741 }% of \def\gmu@checkaftersec.

\ParanoidPostsec 1743 \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 1746 \def\ifgmu@postsec{% note this macro expands to an open \if.
1747     \skip\z@=\csname\gmu@prevsec\endcsname\relax
1748     \ifdim\lastskip=\skip\z@\relax% we play with the vskips only if the last
                    skip is the previous heading's postskip (a counter-example I met while
                    gmdocing).

1752   }}
1754 \let\ifgmu@postsec\iftrue

\gmu@getaddvs 1756 \def\gmu@getaddvs#1\addvspace#2\gmu@getaddvs{%
1757     \toks\z@={#1}
1758     \toks\tw@={#2}}

    And the modification of the inner macros at last:

\gmu@setheading 1761 \def\gmu@setheading#1{%
1762     \@xa\gmu@getaddvs#1\gmu@getaddvs
1763     \edef#1{%
1764         \the\toks\z@\@nx\gmu@checkaftersec
1765         \@nx\addvspace\the\toks\tw@}}
1767 \gmu@setheading\mw@normalheading
1768 \gmu@setheading\mw@runinheading

\SetTwoheadSkip 1770 \def\SetTwoheadSkip#1#2{\@namedef{#1@twoheadskip}{#2}}
1772 }% of \@ifnotmw.

```

My heading setup for mwcls

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

```

1784 \@ifnotmw{}{% We define this declaration only when in mwcls.
\WPheadings 1785 \def\WPheadings{%
1786     \SetSectionFormatting[breakbefore,wholewidth]
1787         {part}{\z@\@plusifill}{\z@\@plus3fill}%
1789     \@ifundefined{chapter}{}%
1790     \SetSectionFormatting[breakbefore,wholewidth]
1791         {chapter}
1792         {66\p@}% {67\p@} for Adventor/Schola o,95.
1793         {\FormatHangHeading{\LARGE}}
1794         {27\p@\@pluso,2\p@\@minus1\p@}%
1795   }%
1797   \SetTwoheadSkip{section}{27\p@\@pluso,5\p@}%
1798   \SetSectionFormatting{section}
1799       {24\p@\@pluso,5\p@\@minus5\p@}%
1800       {\FormatHangHeading{\Large}}

```

```

1801      {10\p@\@pluso,5\p}%ed. Krajewska of “Wiedza Powszechna”, as we un-
      derstand her, wants the skip between a heading and text to be rigid.
1805  \SetTwoheadSkip{subsection}{11\p@\@pluso,5\p@\@minus1\p}%
1806  \SetSectionFormatting{subsection}
1807      {19\p@\@pluso,4\p@\@minus6\p}
1808      {\FormatHangHeading{\large}}% 12/14 pt
1809      {6\p@\@pluso,3\p}% after-skip 6 pt due to p.12, not to squeeze the before-
      skip too much.

1812  \SetTwoheadSkip{subsubsection}{10\p@\@plus1,75\p@\@minus1\p}%
1813  \SetSectionFormatting{subsubsection}
1814      {10\p@\@pluso,2\p@\@minus1\p}
1815      {\FormatHangHeading{\normalsize}}
1816      {3\p@\@pluso,1\p}% those little skips should be smaller than you calcu-
      late out of a geometric progression, because the interline skip enlarges
      them.

1820  \SetSectionFormatting[runin]{paragraph}
1821      {7\p@\@pluso,15\p@\@minus1\p}
1822      {\FormatRunInHeading{\normalsize}}
1823      {2\p}%
1825  \SetSectionFormatting[runin]{subparagraph}
1826      {4\p@\@plus1\p@\@minuso,5\p}
1827      {\FormatRunInHeading{\normalsize}}
1828      {\z@}%
1829  }% of \WPheadings
1830  }% 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.)

```

1871 \@ifnotmw{% we are not in mwcls and want to handle mwcls-like sectionings i.e.,
      those written with two optionals.
\gm@secini 1874 \def\gm@secini{\gm@la}%
\gm@secxx 1876 \def\gm@secxx#1#2[#3]#4{%
1877 \ifx\gm@secstar\@empty
1878 \n@melet{\gm@true@#1mark}{#1mark}% a little trick to allow a special ver-
      sion of the heading just to the running head.
1880 \@namedef{#1mark}##1{% we redefine \<sec>mark to gobble its argument
      and to launch the stored true marking command on the appropriate
      argument.
1883 \csname\gm@true@#1mark\endcsname{#2}%
1884 \n@melet{#1mark}{\gm@true@#1mark}% after we've done what we
      wanted we restore original \#1mark.
1886 }%
\gm@secstar 1887 \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.
1892 \fi
1893 \@xa\@xa\csname\gm@secini#1\endcsname
1894 \gm@secstar{#4}}%
1896 }{% 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 1899 \def\gm@secini{\gm@mw}%
1901 \let\gm@secmarkh\@gobble% in mwcls there's no need to make tricks for special
      version to running headings.
\gm@secxx 1904 \def\gm@secxx#1#2[#3]#4{%
1905 \@xa\@xa\csname\gm@secini#1\endcsname
1906 \gm@secstar[#2][#3]{#4}}%
1907 }
\gm@sec 1909 \def\gm@sec#1{\@dblarg{\gm@secx{#1}}}
\gm@secx 1910 \def\gm@secx#1[#2]{%
1911 \@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 1915 \def\gm@straightensec#1{% the parameter is for the command's name.
1916 \@ifundefined{#1}{}{% we don't change the ontological status of the command
      because someone may test it.
1918 \n@melet{\gm@secini#1}{#1}%
1919 \@namedef{#1}{%
\gm@secstar 1920 \@ifstar{\def\gm@secstar{*}\gm@sec{#1}}{%
\gm@secstar 1921 \def\gm@secstar{}\gm@sec{#1}}}%
1922 }%
1924 \let\do\gm@straightensec
1925 \do{part}\do{chapter}\do{section}\do{subsection}\do{%
      subsubsection}
1926 \@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* 1942 \@namedef{enumerate*}{%
1943   \ifnum\@enumdepth>\thr@@
1944     \@toodeep
1945   \else
1946     \advance\@enumdepth\@ne
1947     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1948     \@xa\list\csname\label\@enumctr\endcsname{%
1949       \partopsep\topsep\topsep\z@\leftmargin\z@
1950       \itemindent\@parindent%\advance\itemindent\labelsep
1951       \labelwidth\@parindent
1952       \advance\labelwidth-\labelsep
1953       \listparindent\@parindent
1954       \usecounter\@enumctr
1955       \def\makelabel##1{##1\hfil}}%
1956   \fi}
1957 \@namedef{endenumerate*}{\endlist}

itemize* 1960 \@namedef{itemize*}{%
1961   \ifnum\@itemdepth>\thr@@
1962     \@toodeep
1963   \else
1964     \advance\@itemdepth\@ne
1965     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1966     \@xa\list\csname\@itemitem\endcsname{%
1967       \partopsep\topsep\topsep\z@\leftmargin\z@
1968       \itemindent\@parindent
1969       \labelwidth\@parindent
1970       \advance\labelwidth-\labelsep
1971       \listparindent\@parindent
1972       \def\makelabel##1{##1\hfil}}%
1973   \fi}
1974 \@namedef{enditemize*}{\endlist}
```

The logos

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

```
1983 \let\oldLaTeX\LaTeX
1984 \let\oldLaTeXe\LaTeXe
1986 \def\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}
\DeclareLogo 1988 \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 1996 \ifx\relax#1\def\gmu@reserveda{\@xa\@gobble\string#2}%
1997 \else
\gmu@reserveda 1998 \def\gmu@reserveda{#1}%
1999 \fi
2000 \edef\gmu@reserveda{%
2001 \@nx\DeclareTextCommand\@nx#2{PD1}{\gmu@reserveda}}
2002 \gmu@reserveda
2003 \DeclareTextCommandDefault#2{#3}%
\pdf 2004 \pdf#2{#3}}% added for XeTeX
\DeclareLogo 2007 \DeclareLogo\LaTeX{%
2008 {%
2010 L%
2011 \setbox\z@\hbox{\check@mathfonts
2012 \fontsize\sf@size\z@
2013 \math@fontsfalse\selectfont
2014 A}%
2015 \kern-.57\wd\z@
2016 \sbox\tw@T%
2017 \vbox\to\ht\tw@{\copy\z@\vss}%
2018 \kern-.2\wd\z@}% originally -, 15em for T.
2019 {%
2020 \ifdim\fontdimen1\font=\z@
2021 \else
2022 \count\z@=\fontdimen5\font
2023 \multiply\count\z@by64\relax
2024 \divide\count\z@by\p@
2025 \count\tw@=\fontdimen1\font
2026 \multiply\count\tw@by\count\z@
2027 \divide\count\tw@by64\relax
2028 \divide\count\tw@by\tw@
2029 \kern-\the\count\tw@sp\relax
2030 \fi}%
2031 \TeX}
\LaTeXe 2033 \DeclareLogo\LaTeXe{\mbox{\m@th\if
2034 b\expandafter\@car\f@series\@nil\boldmath\fi
2035 \LaTeX\kern.15em2$_{\textstyle\varepsilon}$}}
2037 \StoreMacro\LaTeX
2038 \StoreMacro*\LaTeX_
'(L)TeX' in my opinion better describes what I work with/in than just 'LTeX'.
\LaTeXpar 2044 \DeclareLogo[(La)TeX]{\LaTeXpar}{%
2045 {%
2046 \setbox\z@\hbox{({}% )
2047 \copy\z@
2048 \kern-.2\wd\z@L%
2049 \setbox\z@\hbox{\check@mathfonts
2050 \fontsize\sf@size\z@
2051 \math@fontsfalse\selectfont
2052 A}%
2053 \kern-.57\wd\z@
2054 \sbox\tw@T%

```

```

2055     \vbox\to\ht\tw@\{\box\z@%
2056     \vss}%
2057 }%
2058 \kern-.07em% originally -, 15 em for T.
2059 {%(
2060     \sbox\z@)%
2061     \kern-.2\wd\z@\copy\z@
2062     \kern-.2\wd\z@}\TeX
2063 }

```

“Here are a few definitions which can usefully be employed when documenting package files: now we can readily refer to $\mathcal{A}\mathcal{M}\mathcal{S}$ - \TeX , $\mathcal{B}\mathcal{I}\mathcal{B}$ - \TeX and $\mathcal{S}\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$, as well as the usual $\mathcal{T}\mathcal{E}\mathcal{X}$ and $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. There’s even a $\mathcal{P}\mathcal{L}\mathcal{A}\mathcal{I}\mathcal{N}$ $\mathcal{T}\mathcal{E}\mathcal{X}$ and a $\mathcal{W}\mathcal{E}\mathcal{B}$.”

```

2070 \@ifundefined{AmSTeX}
\AmSTeX 2071 {\def\AmSTeX{\leavevmode\hbox{\$ \mathcal{A}\kern-.2em%
          \lower.376ex%
          \hbox{\$ \mathcal{M}\$}\kern-.2em\mathcal{S}\$-\TeX}}\{\}
2072
\BibTeX 2074 \DeclareLogo\BibTeX{\rmfamily\B\kern-.05em%
2075     \textsc{i{\kern-.025em}b}\kern-.08em% the kern is wrapped in braces
          for my \fakescaps’ sake.
2076     \TeX}}
2077
\SlitEX 2080 \DeclareLogo\SlitEX{\rmfamily\B\kern-.06emL\kern-.18em%
          \raise.32ex\hbox
2081     {\scshape\i}\kern-.03em\TeX}}
\PlainTeX 2083 \DeclareLogo\PlainTeX{\textsc{Plain}\kern2pt\TeX}
\Web 2085 \DeclareLogo\Web{\textsc{Web}}

There’s also the ( $\mathcal{L}\mathcal{A}$ )- $\mathcal{T}\mathcal{E}\mathcal{X}$  logo got with the \LaTeXpar macro provided by gmutils. And
here The  $\mathcal{T}\mathcal{E}\mathcal{X}$ book’s logo:
\TeXbook 2088 \DeclareLogo[The\TeXbook]\TeXbook{\textsl{The\TeXbook}}
2089 \let\TB\TeXbook% TUG Boat uses this.

\TeX 2091 \DeclareLogo[e-TeX]\TeX{%
2092     \ensuremath{\varepsilon}\kern-.125em\TeX}% definition sent by Karl Berry
          from TUG Boat itself.

\pdfTeX 2095 \DeclareLogo[pdfe-TeX]\pdfTeX{pdf\TeX}
\pdfTeX 2097 \DeclareLogo\pdfTeX{pdf\TeX}

2099 \@ifundefined{XeTeX}{%
\XeTeX 2100 \DeclareLogo\XeTeX{X\kern-.125em\relax
2101     \@ifundefined{reflectbox}{%
2102         \lower.5ex\hbox{E}\kern-.1667em\relax}{%
2103         \lower.5ex\hbox{\reflectbox{E}}\kern-.1667em\relax}%
2104     \TeX}}\{\}

2106 \@ifundefined{XeLaTeX}{%
\XeLaTeX 2107 \DeclareLogo\XeLaTeX{X\kern-.125em\relax
2108     \@ifundefined{reflectbox}{%
2109         \lower.5ex\hbox{E}\kern-.1667em\relax}{%
2110         \lower.5ex\hbox{\reflectbox{E}}\kern-.1667em\relax}%
2111     \LaTeX}}

```

As you see, if $\mathcal{T}\mathcal{E}\mathcal{X}$ doesn’t recognize `\reflectbox` (graphics isn’t loaded), the first E will not be reversed. This version of the command is intended for non- $\mathcal{X}\mathcal{E}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ usage. With

X_YTeX, you can load the xltextra package (e.g. with the gmutils \XeTeXthree declaration) and then the reversed E you get as the Unicode Latin Letter Reversed E.

Expandable turning stuff all into ‘other’

While typesetting a unicode file contents with inputenc package I got a trouble with some Unicode sequences that expanded to unexpandable cses: they could’nt be used within \csname . . . \endcsname. My T_EXGuru 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 2136 \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:

```
2141 \edef\gmu@tempa{%
\gm@gobmacro 2142   \def\@nx\gm@gobmacro##1\@xa\@gobble\string\macro:##2->{}}
2143 \gmu@tempa
```

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

```
\unex@namedef 2149 \long\def\unex@namedef#1#2{%
2150   \edef@other\gmu@reserveda{#1}%
2151   \@xa\long\@xa\def\csname\gmu@reserveda\endcsname{#2}}

\unex@nameuse 2154 \long\def\unex@nameuse#1{%
2155   \edef@other\gmu@reserveda{#1}%
2156   \csname\gmu@reserveda\endcsname}
```

Brave New World of X_YTeX

```
\@ifXeTeX 2161 \newcommand\@ifXeTeX[2]{%
2162   \ifdefined\XeTeXversion
2163   \unless\ifx\XeTeXversion\relax\afterfifi{#1}\else\afterfifi{%
      #2}\fi
2164   \else\afterfi{#2}\fi}

\XeTeXthree 2167 \def\XeTeXthree{%
2169   \@ifXeTeX{%
2170     \@ifpackageloaded{gmverb}{\StoreMacro\verb}{}}%
2171     \RequirePackage{xltextra}% since v 0.4 (2008/07/29) this package rede-
      fines \verb and verbatim*, and quite elegantly provides an option to
      suppress the redefinitions, but unfortunately that option excludes also
      a nice definition of \xxt@visible space which I fancy.
2178   \@ifpackageloaded{gmverb}{\RestoreMacro\verb}{}}%
2179   \AtBeginDocument{%
2180     \RestoreMacro\LaTeX\RestoreMacro*{LaTeX_}}% my version of the
      LATEX logo has been stored just after defining, in line 2038.
2184 }{}}
```

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

```

2189 \AtBeginDocument{%
2190   \@ifpackageloaded{fontspec}{%
\udigits 2191     \pdf\udigits{%
2192       \addfontfeature{Numbers=Uppercase}}}%
2193   }{%
2194     \emptify\udigits}}

```

Fractions

```

\Xedekfrac 2199 \def\Xedekfrac{\@ifstar\gmu@xedekfracstar\gmu@xedekfracplain}

```

(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@dekfrac).

(**) But Latin Modern Sans Serif Quotation doesn't support the numerator and denominator positions so we provide the double star version for it, which takes the char from font if it exist and typesets with lowers and kerns otherwise.

```

\gmu@xedekfracstar 2214 \def\gmu@xedekfracstar{%
\gmu@xfraccdef 2215   \def\gmu@xfraccdef##1##2{%
2216     \iffontchar\font_##2
2217     \@namedef\gmu@xfracc##1}{\char##2_}%
2218     \else
2219     \n@melet\gmu@xfracc##1}{relax}%
2220     \fi}%
\gmu@dekfrac 2222   \def\gmu@dekfrac##1/##2{%
2223     {\addfontfeature{VerticalPosition=Numerator}##1}%
      \gmu@numeratorkern
2224     \char"2044_\gmu@denominatorkern
2225     {\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.

```

2229   \gmu@xfraccdef{1/4}{"BC}%
2230   \gmu@xfraccdef{1/2}{"BD}%
2231   \gmu@xfraccdef{3/4}{"BE}%
2232   \gmu@xfraccdef{1/3}{"2153}%
2233   \gmu@xfraccdef{2/3}{"2154}%
2234   \gmu@xfraccdef{1/8}{"215B}%
2235   \gmu@xfraccdef{3/8}{"215C}%
2236   \gmu@xfraccdef{5/8}{"215D}%
2237   \gmu@xfraccdef{7/8}{"215E}%
\dekfrac 2238 \def\dekfrac##1/##2{%
\gm@duppa 2239   \def\gm@duppa{##1/##2}%
2240   \@ifundefined\gmu@xfracc\all@other\gm@duppa}{%
2241     \gmu@dekfrac{##1}/{##2}}{%
2242     \csname_\gmu@xfracc\all@other\gm@duppa\endcsname}}%
2243   \@ifstar{\let\gmu@dekfrac\gmu@dekfraccsimple}{}%
2244   }
\gmu@xedekfracplain 2246 \def\gmu@xedekfracplain{% 'else' of the main \@ifstar
\dekfrac 2247   \def\dekfrac##1/##2{%
2248     \addfontfeature{Fractions=On}%

```



```

2249     ##1/##2}}}%
2250 }

```

```

\gmu@numeratorkern 2252 \def\gmu@numeratorkern{\kern-.05em\relax}
2253 \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 Unices of built-in fractions and `\dekfrac` command is defined to use them.

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

Note that in the first argument of `\@ifstar` we wrote 8 (eight) `#s` to get the correct definition and in the second argument ‘only’ 4. (The $\text{\LaTeX} 2_{\epsilon}$ Source claims that that is changed in the ‘new implementation’ of `\@ifstar` so maybe it’s subject to change.)

A simpler version of `\dekfrac` is provided in line [2632](#).

```

\resizegraphics
2276 \@ifXeTeX{%
\resizegraphics 2277 \def\resizegraphics#1#2#3{%
2278 \setboxo=\hbox{\XeTeXpicfile_#3_}%
2279 \ifx!#1\else
2280 \dimeno=#1\relax
2281 \count2=\wdo
2282 \divide\count2_by1000\relax
2283 \counto=\dimeno\relax
2284 \divide\counto\count2
2285 \fi
2286 \ifx!#2\else
2287 \dimeno=#1\relax
2288 \count6=\hto
2289 \divide\count6_by1000\relax
2290 \count4=\dimeno\relax
2291 \divide\count4\count6
2292 \fi
2293 \ifx!#1\counto=\count4\fi
2294 \ifx!#2\count4=\counto\fi
2295 \XeTeXpicfile_#3_xscaled_\counto_yscaled_\count4
2296 }}}}%
\resizegraphics 2297 \def\resizegraphics#1#2#3{%
2298 \resizebox{#1}{#2}{%
2299 \includegraphics{#3}}}%

```

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

```

width <dimen>
height <dimen>
scaled <scalefactor>
xscaled <scalefactor>
yscaled <scalefactor>
rotated <degrees>

```

```

\GMtextsuperscript 2310 \def\GMtextsuperscript{%
2311 \@ifXeTeX{%
\textsuperscript 2312 \def\textsuperscript##1{%

```

```

2313         \addfontfeature{VerticalPosition=Numerator}##1}}%
2314     }\truetextsuperscript}}
\truetextsuperscript 2316 \def\truetextsuperscript{%
\textsuperscript      2317     \pdef\textsuperscript##1{%
2318         \@textsuperscript{\selectfont##1}}%
\@textsuperscript    2319     \def\@textsuperscript##1{%
2320         {\m@th\ensuremath{\sim\mbox{\fontsize\sf@size\z@##1}}}}}}

```

Varia

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

```

\gmu@tilde 2332 \def\gmu@tilde{%
2333     \leavevmode\lower.8ex\hbox{${\,\widetilde{\mbox{\_}}}\,,$}}

```

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

```

\* 2337 \pdef\*{\gmu@tilde}
2343 \AtBeginDocument{% to bypass redefinition of \~ as a text command with various
    encodings
\texttilde 2345 \pdef\texttilde{%
2348     \@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`.)

```

2360 \foone{\catcode\_\active}%
\gmobeyspaces 2361 {\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.

```

2368 \foone{\catcode\^~M\active}% the comment signs here are crucial.
\defobeylines 2369 {\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 2378 \def\dekssmallskip{\vskip\smallskipamount}
\undeekssmallskip 2379 \def\undeekssmallskip{\vskip-\smallskipamount}
\dekmedskip 2380 \def\dekmedskip{\vskip\medskipamount}
\dekbigskip 2381 \def\dekbigskip{\vskip\bigskipamount}
\hfillneg 2384 \def\hfillneg{\hskip\_opt\_plus\_ifill\relax}

```

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

```

\@firstofmany 2392 \long\def\@firstofmany#1#2\@@nil{#1}

```

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

```

\TODO 2396 \newcommand*{\TODO}[1] []{{%

```

```

2397 \sffamily\bfseries\huge\TODO!\if\relax#1\relax\else\space%
      \fi#1}}

```

I like twocolumn tables of contents. First I tried to provide them by writing `\begin{% multicols}{2}` and `\end{multicols}` outto the .toc file but it worked wrong in some cases. So I redefine the internal L^AT_EX macro instead.

```

\twocoltoc 2432 \newcommand*\twocoltoc{%
2433   \RequirePackage{multicol}%
\@starttoc 2434   \def\@starttoc##1{%
2435     \begin{multicols}{2}\makeatletter\@input_{\jobname_}.##1}%
2436     \if@filesw_\@xa_\newwrite_\csname_tf@##1\endcsname
2437     \immediate_\openout_\csname_tf@##1\endcsname_\jobname_
      .##1\relax
2438     \fi
2439     \@nobreakfalse\end{multicols}}}
2441 \@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 2446 \newcommand*\enoughpage[1]{%
2447   \par
2448   \dimeno=\pagegoal
2449   \advance\dimeno_by-\pagetotal
2450   \ifdim\dimeno<#1\relax\newpage\fi}

```

An equality sign properly spaced:

```

\equals 2459 \pdef\equals{\hunskip$={}$\ignorespaces}

```

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

```

\eequals 2461 \pdef\eequals{\hunskip$==$}\ignorespaces}
\cdot 2463 \pdef\cdot{\hunskip$={}\cdot}\ignorespaces}

```

While typesetting a UTF-8 ls-R result I found a difficulty that follows: UTF-8 encoding is handled by the inputenc package. It's O.K. so far. The UTF-8 sequences are managed using active chars. That's O.K. so far. While writing such sequences to a file, the active chars expand. You feel the blues? When the result of expansion is read again, it sometimes is again an active char, but now it doesn't star 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 2483 \def\freeze@actives{%
2484   \count\z@\z@
2486   \@whilenum\count\z@<\ccclvi\do{%
2487     \ifnum\catcode\count\z@=\active
\~ 2488     \uccode`~=\count\z@
2489     \uppercase{\let~\relax}%
2490     \fi
2491     \advance\count\z@\@ne}}

```

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

```

\ShowFont 2497 \newcommand*\ShowFont[1][6]{%
2498   \begin{multicols}{#1}[The_current_font_(the_f@encoding\
      encoding):]

```

```

2499 \parindent\z@
2500 \count\z@\m@ne
2501 \@whilenum\count\z@<\@ccclv\do{
2502     \advance\count\z@\@ne
2503     \_\the\count\z@:\~\char\count\z@\par}
2504 \end{multicols}}

```

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

```

\liturgiques 2512 \newcommand*\liturgiques[1][red]{% Requires the color package.
2513     \gmu@RPfor{color}\color%
\czerwo 2514 \newcommand*\czerwo{\small\color{#1}}% environment
\czer 2515 \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.
\* 2518 \def\*\czer{$*}$}
\+ 2519 \def\+\czer{$\dag$}
\nieczer 2520 \newcommand*\niezer[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 2525 \newcommand*\gmu@RPfor[3][]{%
2527     \ifx\relax#1\relax
\gmu@resa 2528     \else_\def\gmu@resa{[#1]}%
2529     \fi
2530     \@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.

```

2536 \AtBeginDocument{%
\gmu@RPfor 2537 \renewcommand*\gmu@RPfor[3][]{%
2538     \unless\ifdefined#3%
2539     \@ifpackageloaded{#2}{}{%
2540     \typeout{^^J!_Package_`#2'_not_loaded!!!(\on@line)^^J}}%
2541     \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 2547 \pprovide\continuum{\gmu@RPfor{eufrak}\mathfrak\mathfrak{c}}

```

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

```

\iteracro 2551 \def\iteracro{%
\acro 2552     \pdef\acro##1{\gmu@acrospaces##1_\gmu@acrospaces}%
2553 }
2555 \iteracro
\gmu@acrospaces 2557 \def\gmu@acrospaces#1_#2\gmu@acrospaces{%
2558     \gmu@acroinner#1\gmu@acroinner
2559     \ifx\relax#2\relax\else
2560     \space
2561     \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.
2565     \fi}

```

```

\gmu@acroinner 2568 \def\gmu@acroinner#1{%
2569   \ifx\gmu@acroinner#1\relax\else
2570     \ifcat_\a\@nx#1\relax%
2571       \ifnum`#1=\ucode`#1%
2572         {\acrocore{#1}}%
2573       \else{#1}% tu byto \smallerr
2574     \fi
2575   \else#1%
2576   \fi
2577   \afterfi\gmu@acroinner
2578 \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 2582 \def\acrocore{\scshape\lowercase}

```

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

```

\IMO 2587 \newcommand*\IMO{\acro{IMO}}
\AKA 2588 \newcommand*\AKA{\acro{AKA}}
\usc 2590 \pdef\usc#1{{\addfontfeature{Letters=UppercaseSmallCaps}#1}}
\uscacro 2592 \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 2610 \def\secondclass{%
\ifSecondClass 2611   \newif\ifSecondClass
2612   \SecondClasstrue
2613   \@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 2625 \def\gmu@dekfraccsimple#1/#2{\leavevmode\kern.1em
2626   \raise.5ex\hbox{\udigits\smaller[3]#1}\gmu@numeratorkern
2627   \dekfracslash\gmu@denominator\kern
2629   {\udigits\smaller[3]#2}}%
\dekfraccsimple 2632 \def\dekfraccsimple{%
2633   \let\dekfrac\gmu@dekfraccsimple
2634 }
\dekfracslash 2635 \@ifXeTeX{\def\dekfracslash{\char"2044_\}}{%
\dekfracslash 2636   \def\dekfracslash{/}}_\% You can define it as the fraction slash, \char"2044_\
2638 \dekfraccsimple

```

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

```
\ikern 2646 \newcommand*\ikern{\,\penalty10000\hskiposp\relax}
```

And a macro to forbid hyphenation of the next word:

```
\nohy 2650 \newcommand*\nohy{\leavevmode\kernosp\relax}
```

```
\yeshy 2651 \newcommand*\yeshy{\leavevmode\penalty10000\hskiposp\relax}
```

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

```
\@ifempty
```

```
\@ifempty 2657 \long\pdef\@ifempty#1#2#3{%
\gmu@reserveda 2658 \def\gmu@reserveda{#1}%
2659 \ifx\gmu@reserveda\@empty\afterfi{#2}%
2660 \else\afterfi{#3}\fi
2661 }
```

\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@gettext 2673 \def\gmu@gettext#1.#2\@nil{%
\gmu@filename 2674 \def\gmu@filename{#1}%
\gmu@fileext 2675 \def\gmu@fileext{#2}}

2677 \def\include#1{\relax
2678 \ifnum\@auxout=\@partaux
2679 \@latex@error{\string\include\space cannot be nested}\@eha
2680 \else\@include#1\fi}

\@include 2682 \def\@include#1{%
2683 \gmu@gettext#1.\@nil
\gmu@fileext 2684 \ifx\gmu@fileext\empty\def\gmu@fileext{tex}\fi
2685 \clearpage
2686 \if@filesw
2687 \immediate\write\@mainaux{\string\@input{\gmu@filename.aux}}%
2688 \fi
2689 \@tempswatrue
2690 \if@partsw
2691 \@tempswafalse
2692 \edef\reserved@b{#1}%
2693 \@for\reserved@a:=\@partlist\do{%
2694 \ifx\reserved@a\reserved@b\@tempswatrue\fi}%
2695 \fi
2696 \if@tempswa
2697 \let\@auxout\@partaux
2698 \if@filesw
2699 \immediate\openout\@partaux\gmu@filename.aux
2700 \immediate\write\@partaux{\relax}%
2701 \fi
2702 \@input{\gmu@filename.\gmu@fileext}%
2703 \inclasthook
2704 \clearpage
```

```

2705 \writeckpt{\gmu@filename}%
2706 \if@filesw
2707 \immediate\closeout\@partaux
2708 \fi
2709 \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.

```

2713 \deadcycles\z@
2714 \@nameuse{cp@\gmu@filename}%
2715 \fi
2716 \let\@auxout\@mainaux}

```

```

\whenonly 2719 \newcommand\whenonly[3]{%
\gmu@whonly 2720 \def\gmu@whonly{#1,}%
2721 \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 2725 \def\inclasthook{\thispagestyle{closing}}

```

Faked small caps

```

\gmu@scapLetters 2731 \def\gmu@scapLetters#1{%
2732 \ifx#1\relax\relax\else% two \relaxes to cover the case of empty #1.
2733 \ifcat_#1\relax
2734 \ifnum\the\lccode`#1=`#1\relax
2735 {\fakescapsscore\MakeUppercase{#1}}}% not Plain \uppercase because
that works bad with inputenc.
2737 \else#1%
2738 \fi
2739 \else#1%
2740 \fi%
2741 \@xa\gmu@scapLetters
2742 \fi}%
\gmu@scapSpaces 2744 \def\gmu@scapSpaces#1_#2\@@nil{%
2745 \ifx#1\relax\relax
2746 \else\gmu@scapLetters#1\relax
2747 \fi
2748 \ifx#2\relax\relax
2749 \else\afterfi{\_ \gmu@scapSpaces#2\@@nil}%
2750 \fi}
\gmu@scapss 2752 \def\gmu@scapss#1\@@nil{\def~{\nobreakspace}}%
\nobreakspace 2753 \gmu@scapSpaces#1\_ \@@nil}}% \_ \def\\{\newline}}\relax adding re-
definition of \\ caused stack overflow Note it disallows hyphenation ex-
cept at \-.
\fakescaps 2757 \DeclareRobustCommand\fakescaps[1]{%
2758 \gmu@scapss#1\@@nil}}
2760 \let\fakescapsscore\gmu@scalematchX
Experimente z akcentami patrz no3.tex.
\tinycap 2763 \def\tinycap{\tiny\AE}}% to use in \fakescaps[\tiny]{...}
2765 \RequirePackage{calc}
wg \zf@calc@scale pakietu fontspec.

```

```

2769 \ifXeTeX{%
\gmu@scalar 2770 \def\gmu@scalar{1.0}%
\zf@scale 2771 \def\zf@scale{}%
\gmu@scalematchX 2772 \def\gmu@scalematchX{%
2773 \begingroup
\gmu@scalar 2774 \ifx\zf@scale\empty\def\gmu@scalar{1.0}%
2775 \else\let\gmu@scalar\zf@scale\fi
2776 \setlength\@tempdima{\fontdimen5\font}% 5—ex height
2777 \setlength\@tempdimb{\fontdimen8\font}% 8—XYTEX synthesized up-
percase height.
2779 \divide\@tempdimb\by1000\relax
2780 \divide\@tempdima\by\@tempdimb
2781 \setlength{\@tempdima}{\@tempdima*\real{\gmu@scalar}}%
2782 \@ifundefined{fakesc@extrascalse}{}{%
2783 \setlength{\@tempdima}{\@tempdima*\real{%
\fakesc@extrascalse}}}%
2784 \@tempcnta=\@tempdima
2785 \divide\@tempcnta\by1000\relax
2786 \@tempcntb=-1000\relax
2787 \multiply\@tempcntb\by\@tempcnta
2788 \advance\@tempcntb\by\@tempdima
2789 \xdef\gmu@scscale{\the\@tempcnta.%
2790 \ifnum\@tempcntb<1000\fi
2791 \ifnum\@tempcntb<100\fi
2792 \the\@tempcntb}%
2794 \endgroup
2795 \addfontfeature{Scale=\gmu@scscale}%
2796 }{}{\let\gmu@scalematchX\smallerr}
\fakecsextrascalse 2798 \def\fakecsextrascalse#1{\def\fakecsextrascalse{#1}}
\fakecsextrascalse

```

See above/see below

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

```

\wyzejnizej 2804 \newcommand*\wyzejnizej[1]{%
2805 \edef\gmu@tempa{\@ifundefined{r@#1}{\arabic{page}}}%
2806 \@xa\@xa\@xa\@secondoftwo\csname_r@#1\endcsname}%
2807 \ifnum\gmu@tempa<\arabic{page}\relax\wy\zej\fi
2808 \ifnum\gmu@tempa>\arabic{page}\relax\ni\zej\fi
2809 \ifnum\gmu@tempa=\arabic{page}\relax\@xa\ignorespaces\fi
2810 }

```

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 2820 \def\napapierkistretch{0,3pt}% It's quite much for 11/13pt typesetting
\napapierkicore 2822 \def\napapierkicore{\advance\baselineskip%
2823 \by\optplus\napapierkistretch\relax}
napapierki 2825 \newenvironment*{napapierki}{%
2826 \par\global\napapierkicore}%
2827 \par\dimen\z@=\baselineskip

```


2828 \global\baselineskip=\dimen\z@}% so that you can use \endnapapierki in
interlacing environments.

\gmu@luzniej 2832 \newcount\gmu@luzniej

\luzniejcore 2834 \newcommand*\luzniejcore[1][1]{%

2835 \advance\gmu@luzniej\@ne% We use this count to check whether we open the
environment or just set \looseness inside it again.

2837 \ifnum\gmu@luzniej=\@ne\multiply\tolerance\by2\fi

2838 \looseness=#1\relax}

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

luzniej 2842 \newenvironment*{luzniej*}{\par\luzniejcore}{\par}

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

luzniej* 2847 \newenvironment*{luzniej*}[1][1]{%

2848 \multiply\tolerance\by2\relax

2849 \everypar{\looseness=#1\relax}}{\par}

\nawj 2851 \newcommand*\nawj{\kerno,1em\relax}% to put between parentheses and letters with lower ... such as *j* or *y* in certain fonts.

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

2858 \ifdefined\XeTeXversion

\pauza@skipcore 2859 \def\pauza@skipcore{\hskipo.2em\pluso.1em\relax

\pauzacore 2860 \pauzacore\hskip.2em\pluso.1em\relax\ignorespaces}%

\ppauza@skipcore 2862 \def\ppauza@skipcore{\unskip\penalty10000\hskipo.2em\pluso.1em%\relax

2863 -\hskip.2em\pluso.1em\ignorespaces}

2865 \AtBeginDocument{% to be independent of moment of loading of polski.

\- 2866 \pdef\--{%

2867 \ifhmode

2868 \unskip\penalty10000

2869 \afterfi{%

2870 \@ifnextspace{\pauza@skipcore}%

2871 {\@ifnextMac\pauza@skipcore}%

2872 \pauzacore\penalty\hyphenpenalty\hskip\z@}}}%

2873 \else

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

2877 \leavevmode\pauzacore\penalty10000\hskipo,5em\ignorespaces

2878 \fi}%

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

\pauza 2881 \pdef\pauza{%

2882 \ifhmode

2883 \unskip\penalty10000

2884 \hskipo.2em\pluso.1em\relax

2885 \pauzacore\hskip.2em\pluso.1em\relax\ignorespaces%

2886 \else

2887 \pauzadial

2888 \fi}%

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

```
\pauzadial 2893 \pdef\pauzadial{%
2894 \leavevmode\pauzacore\penalty10000\hskip0,5em\ignorespaces}
```

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

```
\lpauza 2898 \pdef\lpauza{%
2899 \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...

```
\- 2907 \pdef\-%
2908 \ifvmode\PackageError{gmutils}{%
2909 command_\backslash_ppauza_(en_dash)_not_intended_for_vmode.}{%
2910 Use_\backslash_ppauza_(en_dash)_only_in_number_and_numerical_
ranges.}%
2911 \else
2912 \afterfi{%
2913 \@ifnextspace{\ppauza@skipcore}{%
2914 \@ifnextMac\ppauza@skipcore{\unskip\discretionary{-}{-}{-}}}%
2915 \fi}%
\ppauza 2917 \pdef\ppauza{%
2918 \ifvmode\PackageError{gmutils}{%
2919 command_\backslash_ppauza_(en_dash)_not_intended_for_vmode.}{%
2920 Use_\backslash_ppauza_(en_dash)_only_in_number_and_numerical_
ranges.}%
2921 \else
2922 \unskip\discretionary{-}{-}{-}%
2923 \fi}%
\emdash 2925 \def\emdash{\char`-}
2926 }% of at begin document
\longpauza 2928 \def\longpauza{\def\pauzacore{-}}
\pauzacore 2929 \longpauza
\shortpauza 2930 \def\shortpauza{%
\pauzacore 2931 \def\pauzacore{-\kern,23em\relax\llap{-}}
2932 \fi% of if XeTeX.
```

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

```
2938 \ifdefined\XeTeXversion
2939 \foone{\catcode`\active\catcode`\active\catcode`\active}{%
\adashes 2940 \def\adashes{\AtBeginDocument\adashes}% because \pauza is defined at
begin document.
\adashes 2942 \AtBeginDocument{\def\adashes{%
2943 \catcode`\active\let-=-%
2944 \catcode`\active\let-=-%
2946 }}}
2947 \else
```

```

2948 \relaxen\adashes
2949 \fi

```

The hyphen shouldn't be active `\no` because it's used in `TEX` control such as `\hskip-2pt`. Therefore we provide the `\ahyphen` declaration reluctantly, because sometimes we need it and always use it with caution. Note that my active hyphen in vertical and math modes expands to `-12`.

```

\gmu@dywiz 2958 \def\gmu@dywiz{\ifmmode-\else
2959 \ifvmode-\else\afterfifi\dywiz\fi\fi}%
2961 \foone{\catcode`\- \active}{%
\ahyphen 2962 \def\ahyphen{\let-\gmu@dywiz\catcode`\- \active}}

To get current time. Works in  $\varepsilon$ -TeXs, including XYTeX. \czas typesets 21.18 and
\czas[:] typesets 21:18.

\czas 2967 \newcommand*\czas[1][.]{%
2968 \the\numexpr(\time-30)/60\relax#1%
2969 \@tempcnta=\numexpr\time-(\time-30)/60*60\relax
2970 \ifnum\@tempcnta<10\o\fi\the\@tempcnta}

\textbullet 2981 \@ifXeTeX{\chardef\textbullet="2022\o}{\def\textbullet{\$ \bullet \$}}

tytulowa 2983 \newenvironment*{tytulowa}{\newpage}{\par\thispagestyle{empty}%
\newpage}

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

\nazwired 2986 \def\nazwired{\quad\textsc}

```

Settings for mathematics in main font

`\gmath` I used these terrible macros while typesetting E. Szarzyński's *Letters* in 2008. The `\gmath` declaration defines one-letter an one-digt cses etc., the `\garamath` declaration redefines the quantifiers and is more Garamond Premier Pro-specific.

```

\gmath 2994 \pdef\gmath{%
2995 \everymath{%
2996 \def\do##1{\edef##1{\@nx\mathit{\@xa@gobble\string##1}}}%
2997 \do\A_\do\A_\do\B_\do\B_\do\C_\do\C_\do\D_\do\D_\do\E%
\do\F
2998 \do\F\do\G\do\G_\do\I_\do\I_\do\J_\do\J_\do\K_\do\K_\do\L_\do\L_\do\m
\do\M_\do\N_\do\N_\do\P_\do\P_\do\Q_\do\Q_\do\R_\do\R
2999 \let\sectionsign\S_\do\S_\do\S_\do\T_\do\T_\do\U_\do\U_\do\V%
3000 \do\V
3001 \do\W_\do\W_\do\X_\do\X_\do\Y_\do\Y_\do\Z_\do\Z
3002 \def\do##1{\edef##1{\@nx\mathrm{\@xa@gobble\string##1}}}%
3003 \do\o\do\1\do\2\do\3\do\4\do\5\do\6\do\7\do\8\do\9%
3004 \relaxen\do
3005 \newcommand*\do[4][\mathit]{\def##2{##3{##1{\char"##4}}}}%
3006 \do\alpha{}{03B1}%
3007 \do[\mathrm]\Delta{}{0394}%
3008 \do\varepsilon{}{03B5}%
3009 \do\vartheta{}{03D1}%
3010 \do\nu{}{03BD}%
3011 \do\pi{}{03C0}%

```

```

3015 \do\phi{}\{o3D5}%
3016 \do[\mathrm]\Phi{}\{o424}%
3017 \do\sigma{}\{o3C3}%
3018 \do\varsigma{}\{o3DA}%
3019 \do\psi{}\{o3C8}%
3020 \do\omega{}\{o3C9}%
3021 \do\infty{}\{221E}%
3022 \do[\mathrm]\neg{\mathbin}{ooAC}%
3023 \do[\mathrm]\neq{\mathrel}{226o}%
3024 \do\partial{}\{2202}%
3025 \do[\mathrm]\pm{}\{ooB1}%
3026 \do[\mathrm]\pm{\mathbin}{ooB1}%
3027 \do[\mathrm]\sim{\mathrel}{oo7E}%
3029 \def\do##1##2##3{\def##1{%
\mathop{\mathchoice{\hbox{%
3030 \rm
3031 \edef\gma@tempa{\the\fontdimen8\font}%
3032 \larger[3]%
3033 \lower\dimexpr(\fontdimen8\font-\gma@tempa)/2_%
3034 \hbox{##2}}}\{\hbox{%
3035 \rm
3036 \edef\gma@tempa{\the\fontdimen8\font}%
3037 \larger[2]%
3038 \lower\dimexpr(\fontdimen8\font-\gma@tempa)/2_%
3039 \hbox{##2}}}%
3040 {\mathrm{##2}}\{\mathrm{##2}}\}##3}}%
3041 \do\sum{\char"2211}{}%
3042 \do\forall{\gma@quantifierhook_\rotatebox[origin=c]{180}{A}%
3043 \setboxo=\hbox{A}\setbox2=\hbox{\scriptsize_x}%
3044 \kern\dimexpr\ht2/3*2_-wdo/2\relax}\{nolimits}%
3045 \do\exists{\rotatebox[origin=c]{180}{\gma@quantifierhook_E}}%
3046 \nolimits%
3048 \def\do##1##2##3{\def##1{##3{%
\mathchoice 3049 \mathchoice{\hbox{\rm##2}}{\hbox{\rm##2}}%
3050 {\hbox{\rm\scriptsize##2}}{\hbox{\rm\tiny##2}}}}}%
3051 \do\vee{\rotatebox[origin=c]{90}{<}}\mathbin
3052 \do\wedge{\rotatebox[origin=c]{-90}{<}}\mathbin
3053 \do\leftarrow{\char"2190}\mathrel
3054 \do\rightrightarrows{\char"2192}\mathrel
3055 \do\leftrightharpoons{\char"2190\kern-0,1em\char"2192}\mathrel
3057 \gmu@storespecialchars[\do\`do\`\do\=]%
3058 \gmu@septify[\do\`12\do\"12\do\=12]%
3059 \def\do##1##2##3{%
3060 \catcode`##1=12\relax
3061 \scantokens{\mathcode`##1="8ooo\relax
3062 \foone{\catcode`##1=\active}\{\def##1}{##3{%
\mathchoice 3063 \mathchoice{\hbox{\rm##2}}{\hbox{\rm##2}}%
3064 {\hbox{\rm\scriptsize##2}}{\hbox{\rm\tiny##2}}}}}%
3065 \ignorespaces}}% to eat the lineend (scantokens acts as \read icluding
line end).
3067 \do..\mathpunct_\do,,\mathpunct_\do.....\mathpunct
3068 \do((\mathopen%
3069 \@ifundefined{resetMathstrut@}{\,%an error ocured `bad mathchar etc.`

```

because amsmath.sty doesn't take account of a possibility of '(' being math-active.

```

\resetMathstrut@ 3073 \def\resetMathstrut@{%
3074 \setbox\z@\hbox{%
%%\mathchardef\@tempa\mathcode`\(\relax
%%\def\@tempb##1"##2##3{\the\textfont"##3\char"}%
%%\expandafter\@tempb\meaning\@tempa\relax
3078 (%)
3079 }%
3080 \ht\Mathstrutbox@\ht\z@\dp\Mathstrutbox@\dp\z@
3081 }}%
3082 \do)\mathclose
3083 \do[[\mathopen\do]]\mathclose
3084 \do-\{\char"2212\}\mathbin\do++\mathbin\do==\mathrel\do%
\do\times\mathbin
3085 \do::\mathbin\do\cdot\mathbin\do//\mathbin\do<<\mathrel
3086 \do>>\mathrel
3087 \gmu@restorespecials
3088 \def\do##1##2##3{\def##1####1{##2{\hbox{%
3090 \rm
3091 \setboxo=\hbox{####1}%
3092 \edef\gma@tempa{\the\hto}%
3093 \edef\gma@tempb{\the\dpo}%
3094 ##3%
3095 \setboxo=\hbox{####1}%
3096 \lower\dimexpr(\hto+\dpo)/2-\dpo-(\gma@tempa+
\gma@tempb)/2-\gma@tempb)\boxo}}}%
3097 \do\bigl\mathopen\larger
3098 \do\bigr\mathclose\larger
3099 \do\Bigl\mathopen\largerr
3100 \do\Bigr\mathclose\largerr
3101 \do\biggl\mathopen{\larger[3]}%
3102 \do\biggr\mathclose{\larger[3]}%
3103 \do\Biggl\mathopen{\larger[4]}%
3104 \do\Biggr\mathclose{\larger[4]}%
3105 \def\do##1##2{\def##1{\ifmmode##2{\mathchoice
3107 {\hbox{\rm\char`##1}}{\hbox{\rm\char`##1}}%
3108 {\hbox{\rm\scriptsize\char`##1}}{\hbox{\rm\tiny%
3109 \char`##1}}}%
3110 \else\char`##1\fi}}%
3111 \do\{\mathopen
3112 \do\}\mathclose
3113 \def\={\mathbin{=}}%
\neqb 3114 \def\neqb{\mathbin{\neq}}%
3115 \def\do##1{\edef\gma@tempa{%
3116 \def\@xa\@nx\cname\@xa\gobble\string##1r\endcname{%
3117 \@nx\mathrel{\@nx##1}}}%
3118 \gma@tempa}%
3119 \do\vee\do\wedge\do\neg
\fake 3120 \def\fake{\mkern-3mu}%
3121 \thickmuskip=8mu\plus4mu\relax
3122 \gma@gmathhook
3123

```

```

3126     }% of \everymath.
3127 }% of \def\gmath.

\quantifierhook 3129 \emptify\gma@quantifierhook
\gma@quantifierhook 3130 \def\quantifierhook#1{%
3131   \def\gma@quantifierhook{#1}}

\gmathhook 3133 \emptify\gma@gmathhook
3134 \def\gmathhook#1{\addtomacro\gma@gmathhook{#1}}

\gma@dollar 3137 \def\gma@dollar$#1${{\gmath$#1$}}%
\gma@bare 3138 \def\gma@bare#1{\gma@dollar$#1$}%
\gma@checkbracket 3139 \def\gma@checkbracket{\@ifnextchar\[%
3140   \gma@bracket\gma@bare}
\gma@bracket 3141 \def\gma@bracket[#1\]{{\gmath[#1\]}\@ifnextchar\par}{%
\noindent}}

\gma 3142 \def\gma{\@ifnextchar$%
3143   \gma@dollar\gma@checkbracket}

\garamath 3149 \def\garamath{%
3150   \quantifierhook{\addfontfeature{OpticalSize=800}}%
\gma@arrowdash 3152 \def\gma@arrowdash{%
3153   \setboxo=\hbox{\char"2192}\copyo\kern-0,6\wdo
3154   \bgcolor\rule[-\dpo]{0,6\wdo}{\dimexpr\hto+\dpo}\kern-0,6%
\wdo}}%

\gma@gmathhook 3156 \def\gma@gmathhook{%
3157   \def\do####1####2####3{\def####1{####3}%
\mathchoice 3158   \mathchoice{\hbox{\rm####2}}{\hbox{\rm####2}}%
3159   {\hbox{\rm\scriptsize####2}}{\hbox{\rm\tiny####2}}}}%
3160   \do\mapsto{\rule[0,4ex]{0,1ex}{0,4ex}\kern-0,05em%
3161   \gma@arrowdash\kern-0,05em\char"2192}\mathrel
3162   \do\cup{\scshape\mathbin
3163   \do\varnothing{\setboxo=\hbox{\gma@quantifierhook%
\addfontfeature{Scale=1.272727}0}%
3164   \setbox2=\hbox{\char"2044}%
3165   \copyo\kern-0,5\wdo\kern-0,5\wd2\lowero,125\wdo\copy2
3166   \kerno,5\wdo\kern-0,5\wd2}{%
3167   \do\leftarrow{\char"2190\kern-0,05em\gma@arrowdash}\mathrel
3168   \do\rightarrow{\gma@arrowdash\kern-0,05em\char"2192}\mathrel
3169   \do\in{\gma@quantifierhook\char"0454}\mathbin
3170 }}

```

Typesetting dates in my memoirs

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

```

\polskadata 3183 \newcommand*\polskadata{%
\datef 3184   \def\datef##1-##2-##3##4{%
3185     \if\relax##2\relax##3##4%
3186     \else
3187     \ifnum##3##4=0\relax

```

```

3188 \else
3189 \ifnum##3=0\relax
3190 \else##3%
3191 \fi##4%
3192 \fi
3193 \ifcase##2\relax\or\_stycznia\or\_lutego%
3194 \or\_marca\or\_kwietnia\or\_maja\or\_czerwca\or\_lipca\or\_
sierpnia%
3195 \or\_września\or\_października\or\_listopada\or\_grudnia%
\else
3196 {}%
3197 \fi
3198 \if\relax##1\relax\else\_fi\_##1%
3199 \fi}%
\datefsl 3202 \def\datefsl##1/##2/##3##4{%
3203 \if\relax##2\relax##3##4%
3204 \else
3205 \ifnum##3##4=0\relax
3206 \else
3207 \ifnum##3=0\relax
3208 \else##3%
3209 \fi##4%
3210 \fi
3211 \ifcase##2\relax\or\_stycznia\or\_lutego%
3212 \or\_marca\or\_kwietnia\or\_maja\or\_czerwca\or\_lipca\or\_
sierpnia%
3213 \or\_września\or\_października\or\_listopada\or\_grudnia%
\else
3214 {}%
3215 \fi
3216 \if\relax##1\relax\else\_fi\_##1%
3217 \fi}%
3218 }% of \polskadata
3220 \polskadata
For documentation in English:
\englishdate 3223 \newcommand*\englishdate{%
\datef 3224 \def\datef##1-##2-##3##4{%
3225 \if\relax##2\relax##3##4%
3226 \else
3227 \ifcase##2\relax\or\_January\or\_February%
3228 \or\_March\or\_April\or\_May\or\_June\or\_July\or\_August%
3229 \or\_September\or\_October\or\_November\or\_December\else
3230 {}%
3231 \fi
3232 \ifnum##3##4=0\relax
3233 \else
3234 \_ %
3235 \ifnum##3=0\relax
3236 \else##3%
3237 \fi##4%
3238 \ifcase##3##4\relax\or\_st\or\_nd\or\_rd\else\_th\fi
3239 \fi

```

```

3240     \if\relax##1\relax\else,\_\fi_##1%
3241     \fi
3242 }%
\datefsl 3243 \def\datefsl##1/##2/##3##4{%
3244     \if\relax##2\relax##3##4%
3245     \else
3246         \ifcase##2\relax\or_January\or_February%
3247         \or_March\or_April\or_May\or_June\or_July\or_August%
3248         \or_September\or_October\or_November\or_December\else
3249         {}%
3250     \fi
3251     \ifnum##3##4=0\relax
3252     \else
3253         \_ %
3254         \ifnum##3=0\relax
3255         \else##3%
3256         \fi##4%
3257         \ifcase##3##4\relax\or_st\or_nd\or_rd\else_th\fi
3258     \fi
3259     \if\relax##1\relax\else,\_\fi_##1%
3260     \fi
3261 }%
3262 }
\ifgmu@dash 3264 \newif\ifgmu@dash
\gmu@ifnodash 3266 \def\gmu@ifnodash#1-#2\@@nil{%
3267     \def\@tempa{#2}%
3268     \ifx\@tempa\@empty}
\gmu@testdash 3270 \def\gmu@testdash#1\ifgmu@dash{%
3271     \gmu@ifnodash#1-\@@nil
3272     \gmu@dashfalse
3273     \else
3274     \gmu@dashtrue
3275     \fi
3276     \ifgmu@dash}

```

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

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

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

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

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

```

\ifdate 3307 \newif\ifdate
\data 3309 \newcommand*{\data}[1]{%
3310 \ifdate\gmu@testdash#1\ifgmu@dash\datef#1\else\datefsl#1\fi\fi}

\linedate 3312 \newcommand*{\linedate}[1]{\par\ifdate\addvspace{\dateskip}%
3313 \date@line{\footnotesize\itshape_\date@biway{#1}}%
3314 \nopagebreak\else%%\ifnum\arabic{dateinsection}>0\dekbigskip\fi
3315 \addvspace{\bigskipamount}%
3316 \fi}% end of \linedate.

3318 \let\dateskip\medskipamount

\date@biway 3320 \def\date@biway#1{%
3321 \gmu@testdash#1\ifgmu@dash\datef#1\else\datefsl#1\fi}

\rdate 3323 \newcommand*\rdate[1]{\let\date@line\rightline_\linedate{#1}}
\ldate 3324 \newcommand*\ldate[1]{\let\date@line\leftline_\linedate{#1}}
\runindate 3325 \newcommand*\runindate[1]{%
3326 \paragraph{\footnotesize\itshape_\datef#1\@@nil}\stepcounter{%
dateinsection}}

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

3329 \let\thedata\ldate

\zwrobcy 3332 \pdef\zwrobcy#1{\emph{#1}}_\% ostinato, allegro con moto, garden party etc.,
także komplement

\tytul 3335 \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 3341 \newenvironment{maszynopis}[1][\ttfamily
3342 \hyphenchar\font=45\relax% this assignment is global for the font.
3343 \@tempskipa=\glueexpr\rightskip+\leftskip\relax
3344 \ifdim\gluestretch\@tempskipa=\z@
3345 \tolerance900
it worked well with tolerance = 900.
3347 \advance\rightskip_\by\z@_pluso,5em\relax\fi
3348 \fontdimen3\font=\z@% we forbid stretching spaces...
%\fontdimen4\font=\z@ but allow shrinking them.
3350 \hyphenpenalty0_\% 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.

3353 \StoreMacro\pauzacore
\pauzacore 3354 \def\pauzacore{-\rlap{\kern-0,3em-}}%
3355 }\par}

\justified 3359 \newcommand*\justified{%
3360 \leftskip=1\leftskip% to preserve the natural length and discard stretch and
shrink.
3362 \rightskip=1\rightskip
3363 \parfillskip=1\parfillskip
3364 \advance\parfillskip_\by_osp_plus_1fil\relax
3365 \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 3370 \newcommand*\fullpar{%
3371   \hunskip
3372   \bgroup\parfillskip\z@skip\par\egroup}
```

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

```
\twoparinit 3381 \newcommand*\twoparinit{% the name stands for 'last paragraph line's length
               minimum two \parindent.
3383   \edef\twopar{%
3384     \hunskip% it's \protected, remember?
3385     \bgroup
3386     \parfillskip=\the\glueexpr
3387     \dimexpr\textwidth-2\parindent\relax
3388     minus\dimexpr\textwidth-2\parindent\relax
3389     \relax% to delimit \glueexpr.
3390     \relax% to delimit the assignment.
3391     \par\egroup
3392   }% of \gmu@twoparfill
3397 }% of \twoparinit.
```

For dati under poems.

```
\wherncore 3404 \newcommand\wherncore[1]{%
3405   \rightline{%
3406     \parbox{0,7666\textwidth}{
3407       \leftskip\pplus\textwidth
3408       \parfillskip\relax
3409       \let\\\linebreak
3410       \footnotesize\#1}}
3412 \def\whern{%
3413   \@ifstar\wherncore{\vskip\whernskip\wherncore}}
\whernskip 3416 \newskip\whernskip
3417 \whernskip2\baselineskip\minus2\baselineskip\relax
\whernup 3419 \newcommand\whernup[1]{\par\wherncore{\#1}}
```

Minion and Garamond Premier kerning and ligature fixes

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

```
\gmu@tempa 3427 \def\gmu@tempa{\kern-0,08em\penalty10000\hskip\relax
3428   s\penalty10000\hskip\relax}
3430 \protected\edef\Vs{V\gmu@tempa}
3432 \protected\edef\Ws{W\gmu@tempa}
\Wz 3434 \pdef\Wz{W\kern-0,05em\penalty10000\hskip\relax_z}
```

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 3455 \def\leftslanting{%
\litshape      3456 \pdef\litshape{%
3458     \itshape
3459     \@tempdima=-2\fontdimen1\font
3460     \advance\leftskip_\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.
3464     \edef\gmu@tempa{%
3465     \@nx\addfontfeature{RawFeature={slant=\strip@pt%
                    \@tempdima}}}% when not \edefed, it caused an error, which is
                    perfectly understandable.
3468     \gmu@tempa}%
\textlit      3471 \pdef\textlit##1{%
3472     {\litshape##1}}%
\lit          3474 \pdef\lit{\rm\litshape}%
\lsl          3477 \pdef\lsl{\it
3480     \@tempdima=-\fontdimen1\font
3481     \xdef\gmu@tempa{%
3482     \@nx\addfontfeature{RawFeature={slant=\strip@pt%
                    \@tempdima}}}%}%
3483     \rm_\% Note in this declaration we left-slant the basic Roman font not the it-
                    shape of the current family.
3485     \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          3493 \pdef\em{%
3494     \ifdim\fontdimen1\font=\z@_\itshape
3495     \else
3496     \ifdim\fontdimen1\font>\z@_\litshape
3497     \else_\upshape
3498     \fi
3499     \fi}%
3502 \pdef\emph##1{%
3503     {\em##1}}%
3504 }% of \leftslanting.
```

Thousand separator

`\thousep` 3508 `\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.

```
3512 \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.
3515 \gmu@thou@fiver{#1}{% if more than five digits:
3516 \emptify\gmu@thou@put
3517 \relaxen\gmu@thou@o\relaxen\gmu@thou@i\relaxen\gmu@thou@ii
3518 \@tempcnta\z@
3519 \gmu@thou@putter#1\gmu@thou@putter
3520 \gmu@thou@put
3521 }}
```

```
\gmu@thou@fiver 3523 \def\gmu@thou@fiver#1#2#3#4#5\gmu@thou@fiver#6#7{%
3524 \ifx\relax#5\relax\afterfi{#6}\else\afterfi{#7}\fi}
```

```
\gmu@thou@putter 3526 \def\gmu@thou@putter#1#2{% we are sure to have at least five tokens before the
      guardian \gmu@thou@putter.
3528 \advance\@tempcnta\@ne
3529 \@tempcntb\@tempcnta
3530 \divide\@tempcntb3\relax
3531 \@tempcnta=\numexpr\@tempcnta-\@tempcntb*3
3532 \edef\gmu@thou@put{\gmu@thou@put#1%
3533 \ifx\gmu@thou@putter#2\else
3534 \ifcase\@tempcnta
3535 \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.
3538 \fi
3539 \fi}% of \edef
3540 \ifx\gmu@thou@putter#2% if we are at end of the digits...
3541 \edef\gmu@tempa{%
3542 \ifcase\@tempcnta
3543 \gmu@thou@o\or\gmu@thou@i\or\gmu@thou@ii
3544 \fi}%
3545 \@xa\let\gmu@tempa\gmu@thousep% ... we set the proper cs...
3546 \else% or ...
3547 \afterfi{% iterate.
3548 \gmu@thou@putter#2}% of \afterfi
3549 \fi% of if end of digits.
3550 }% of \gmu@thou@putter.
\gmu@thousep 3552 \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 `\expandafthers` and `\the`. Let's do it only once for all:

```
\xathousep 3560 \pdef\xathousep#1{\@xa\thousep\@xa{\the#1}}
```

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

Storing and restoring the catcodes of specials

```
\gmu@storespecialchars 3567 \newcommand*\gmu@storespecialchars[1] [] {% we provide a possibility of adding
                        stuff. For usage see line 3057.
3569 \def\do##1{\catcode`\@nx##1=\the\catcode`##1\relax}%
3570 \edef\gmu@restorespecials{\dospecials\do\^~M#1}}

\gmu@septify 3572 \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
3575 \def\do{\relax\catcode`}%
3576 \do\_\do\1\do\0\do\{1\do\}2\do\$3\do\&4%
3577 \do\#6\do\^7\do\_8\do\%14\do\~13\do\^~M5#1\relax}
```

hyperref’s \nolinkurl into \url*

```
\urladdstar 3581 \def\urladdstar{%
3582 \AtBeginDocument{%
3583 \ifpackageloaded{hyperref}{%
3584 \StoreMacro\url
\url 3585 \def\url{\@ifstar{\nolinkurl}{\storedcsname{url}}}%
3586 }{}}
3588 \@onlypreamble\urladdstar
3591 \endinput
```

Change History

- | | |
|---|---|
| <p>vo.74</p> <p>\@begnamedgroup@ifcs:</p> <p>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, 727</p> <p>General:</p> <p>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, 3591</p> <p>vo.75</p> <p>\@ifnextcat:</p> <p>\let for #1 changed to \def to allow things like \noexpand~ , 517</p> <p>\@ifnextif:</p> <p>\let for #1 changed to \def to allow things like \noexpand~, 553</p> <p>\@ifnif:</p> | <p>added, 578</p> <p>vo.76</p> <p>General:</p> <p>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, 3591</p> <p>\freeze@actives:</p> <p>added, 2463</p> <p>vo.77</p> <p>General:</p> <p>\afterfi & pals made two-argument as the Marcin Woliński’s analogoi are. At this occasion some redundant macros of that family are deleted, 3591</p> <p>vo.78</p> <p>General:</p> <p>\@namelet renamed to \n@melet to solve a conflict with the beamer class. The package contents regrouped, 3591</p> <p>vo.79</p> <p>\not@onlypreamble:</p> <p>All the actions are done in a group and therefore \xdef used instead of \edef because this command has to</p> |
|---|---|

	use <code>\do</code> (which is contained in the <code>\@preamblecmds</code> list) and <code>\not@onlypreamble</code> itself should be able to be let to <code>\do</code> , 1316	General: CheckSum 4035, 0 <code>\XeTeXthree</code> : adjusted to the redefinition of <code>\verb</code> in <code>xlxtra</code> 2008/07/29, 2167
vo.80	General: CheckSum 1689, 0 <code>\hfillneg</code> : added, 2384	vo.91 General: CheckSum 4055, 0 removed <code>\jobname</code> since <code>\jobname</code> is always without extension. <code>\xiispace</code> forked to <code>\visiblespace</code> <code>\let</code> to <code>\xxt@visiblespace</code> of <code>xltxtra</code> if available. The documentation driver integrated with the <code>.sty</code> file, 3591
vo.81	<code>\dekfrac</code> slash: moved here from <code>pmlectionis.cls</code> , 2638 <code>\ifSecondClass</code> : moved here from <code>pmlectionis.cls</code> , 2613	vo.92 <code>\@checkend</code> : shortened thanks to <code>\@ifenvir</code> , 759 <code>\@gif</code> : added redefinition so that now switches defined with it are <code>\protected</code> so they won't expand to a further expanding or unbalanced <code>\iftrue/false</code> in an <code>\edef</code> , 272 <code>\@ifenvir</code> : added, 751 General: CheckSum 4133, 0
vo.82	<code>\ikern</code> : added, 2646	vo.93 <code>\@nameedef</code> : added, 220 General: A couple of <code>\DeclareRobustCommand*</code> changed to <code>\pdef</code> , 3591 CheckSum 4140, 0 CheckSum 4501, 0 The numerical macros commented out as obsolete and never really used, 3591
vo.83	<code>\~</code> : postponed to <code>\begin{document}</code> to avoid overwriting by a text command and made sensible to a subsequent <code>/</code> , 2337	<code>\ampulexdef</code> : added, 411 <code>\em</code> : added, 3493, 3502 <code>\gmu@RPfor</code> : renamed from <code>\gmu@RPif</code> and #3 changed from a <code>csname</code> to <code>cs</code> , 2525 <code>\litshape</code> : copied here from E. Szarzyński's <i>The Letters</i> , 3456 <code>\lsl</code> : copied here from E. Szarzyński's <i>The Letters</i> , 3477 <code>\nocite</code> : a bug fixed: with <code>natbib</code> an 'extra <code>}</code> ' error. Now it fixes only the standard version of <code>\nocite</code> , 1354 <code>\pdef</code> :
vo.84	General: CheckSum 2684, 0	
vo.85	General: CheckSum 2795, 0 fixed behaviour of too clever headings with <code>gmdoc</code> by adding an <code>\ifdim</code> test, 3591	
vo.86	<code>\texttilde</code> : renamed from <code>\~</code> since the latter is one of L ^A T _E X accents, 2345	
vo.87	General: CheckSum 4027, 0 the package goes ε -T _E X even more, making use of <code>\ifdefined</code> and the code using UTF-8 chars is wrapped in a X _E L ^A T _E X-condition, 3591	
vo.88	General: CheckSum 4040, 0 <code>\RestoreEnvironment</code> : added, 1255 <code>\storedcsname</code> : added, 1246 <code>\StoreEnvironment</code> : added, 1251	
vo.89	General: removed obsolete adjustment of <code>pgf</code> for X _E L ^A T _E X, 3591	
vo.90		

added, 178
\pprovide:
added, 207
\provide:
added, 192

\textlit:
added, 3471
\thousep:
added, 3508

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.

*, 606, <u>607</u> , 608, 610, 614, 615, 616, 620, <u>2337</u> , <u>2518</u>	\@ifl@aded, 1340	\@tempdimb, 2777, 2779, 2780
\+, <u>2519</u>	\@ifncat, 524, <u>527</u> , 542	\@textsuperscript, 2318, 2319
\-, <u>1044</u> , 2962	\@ifnextMac, <u>630</u> , 2871, 2914	\@toodeep, 1944, 1962
\<...>, <u>975</u>	\@ifnextac, <u>589</u>	\@topnewpage, 1587
\@@nil, 1001, 1003, 1005, 2392, 2673, 2683, 2744, 2749, 2752, 2753, 2758, 3266, 3271, 3326	\@ifnextcat, <u>517</u> , 590	\@undefined, 157
\@M, 1691	\@ifnexttif, <u>553</u>	\@whilenum, 2486, 2501
\@afterheading, <u>1684</u>	\@ifnextspace, <u>613</u> , 2870, 2913	\@xa, <u>174</u>
\@badend, 759	\@ifnif, 560, <u>563</u> , 578	\@xifncat, 529, <u>542</u>
\@begnamedgroup, <u>696</u> , 712, 720, 725	\@ifnotmw, <u>1509</u> , <u>1512</u> , 1679, 1784, 1871, 1926	\@xifnif, 565, 578
\@begnamedgroup@ifcs, 721, <u>724</u>	\@include, 2680, <u>2682</u>	\acro, 2552, 2587, 2588, 2592
\@car, 2034	\@itemdepth, 1961, 1964, 1965	\acrocore, 2572, <u>2582</u>
\@ccclv, 2501	\@itemitem, 1965, 1966	\adashes, 2940, 2940, <u>2942</u> , 2948
\@ccclvi, 2486	\@minus, 1794, 1799, 1805, 1807, 1812, 1814, 1821, 1826	\addfontfeature, 2192, 2223, 2225, 2248, 2313, 2590, 2795, 3150, 3163, 3465, 3482
\@checkend, <u>759</u>	\@nameedef, <u>220</u>	\addto@macro, <u>359</u> , 366
\@clsextension, 2613	\@nobreakfalse, 1690, 2439	\addtoheading, <u>1660</u>
\@clubpenalty, 1696	\@nobreaktrue, 1685	\addtomacro, <u>366</u> , 3134
\@currenvir, 699, 753	\@normalcr, 3365	\addtotoks, <u>374</u>
\@currenvline, 703	\@nx, <u>175</u>	\AE, 2763
\@currsize, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812	\@oarg, 1079, 1081, 1082	\afterfi, 197, 593, 617, 618, <u>656</u> , 725, 726, 755, <u>756</u> , 993, 1024, 1095, 1138, 1215, 2164, 2561, 2577, 2659, 2660, 2721, 2749, 2869, 2912, 3524, 3547
\@emptify, <u>378</u> , 379, 383	\@oargsq, 1079, <u>1082</u> , 1097	\afterfifi, 194, 195, <u>658</u> , 1194, 1199, 2163, 2959
\@enumctr, 1947, 1948, 1954	\@onlypreamble, 1379, 2441, 3588	\afterfififi, <u>665</u>
\@enumdepth, 1943, 1946, 1947	\@parg, 1086, <u>1088</u> , 1089	\afteriffifi, <u>659</u>
\@fileswithoptions, 2613	\@pargp, 1086, <u>1089</u> , 1098	\afteriffififi, <u>664</u>
\@firstofmany, <u>2392</u>	\@parindent, 1950, 1951, 1953, 1968, 1969, 1971	\afteriffiffifi, <u>663</u>
\@gif, 258, 259, <u>266</u>	\@pkgextension, 1341	\ahyphen, 2962
\@if, 281, 282, <u>285</u>	\@preamblecmds, 1335, 1337, 1350, 1354	\AKA, <u>2588</u>
\@ifXeTeX, <u>2161</u> , 2169, 2276, 2311, 2635, 2769, 2981	\@relaxen, <u>387</u> , 388, 392	
\@ifempty, 429, 1626, 1646, 2657	\@starttoc, 2434	
\@ifenvir, <u>751</u> , 759	\@tempcntb*, 3531	
	\@tempdima, 2776, 2780, 2781, 2783, 2784, 2788, 3459, 3465, 3480, 3482	
	\@tempdima*, 2781, 2783	

<code>\all@other</code> , 426, 427, 436, 437, 438, <u>2136</u> , 2240, 2242	2291, 2293, 2294, 2295, 2484, 2486, 2487, 2488, 2491, 2500, 2501, 2502, 2503	<code>\dywiz</code> , 2959
<code>\alpha</code> , 3009	<code>\cs</code> , <u>1043</u> , 1049, 1053, 1068	<code>\edef@other</code> , 2150, 2155
<code>\ampulexdef</code> , <u>411</u> , 447, 460, 1378	<code>\cup</code> , <u>3162</u>	<code>\eequals</code> , <u>2461</u>
<code>\ampulexset</code> , <u>465</u> , 471, 1373	<code>\czas</code> , <u>2967</u>	<code>\egRestore@Macro</code> , 1208, <u>1210</u>
<code>\AmSTeX</code> , <u>2071</u>	<code>\czer</code> , <u>2515</u> , 2518, 2519	<code>\egRestore@MacroSt</code> , 1208, <u>1211</u>
<code>\arg</code> , <u>1094</u> , <u>1095</u>	<code>\czerwo</code> , <u>2514</u> , 2515	<code>\egroupfirstofone</code> , 237, 239
<code>\AtBeginDocument</code> , 913, 1093, 1348, 1382, 1480, 2179, 2189, 2343, 2536, 2865, 2940, 2942, 3582	<code>\dag</code> , 2519	<code>\egStore@Macro</code> , 1128, <u>1133</u>
<code>\begin</code> , <u>720</u>	<code>\data</code> , <u>3309</u>	<code>\egStore@MacroSt</code> , 1128, <u>1134</u>
<code>\begin*</code> , <u>720</u>	<code>\date@biway</code> , 3313, <u>3320</u>	<code>\em</code> , <u>3493</u> , 3503
<code>\bgcolor</code> , 3154	<code>\date@line</code> , 3313, 3323, 3324	<code>\emdash</code> , <u>2925</u>
<code>\BibTeX</code> , <u>2074</u>	<code>\datef</code> , <u>3184</u> , <u>3224</u> , 3310, 3321, 3326	<code>\emptyify</code> , <u>379</u> , 379, 2194, 3129, 3133, 3516
<code>\Biggl</code> , 3104	<code>\datefsl</code> , <u>3202</u> , <u>3243</u> , 3310, 3321	<code>\enamegroup</code> , <u>714</u>
<code>\biggl</code> , 3102	<code>\dateskip</code> , 3312, 3318	<code>\endlist</code> , 1957, 1974
<code>\Biggr</code> , 3105	<code>\deadcycles</code> , 2713	<code>\englishdate</code> , <u>3223</u>
<code>\biggr</code> , 3103	<code>\DeclareLogo</code> , <u>1988</u> , <u>2007</u> , 2033, 2044, 2074, 2080, 2083, 2085, 2088, 2091, 2095, 2097, 2100, 2107	<code>\enoughpage</code> , <u>2446</u>
<code>\Bigl</code> , 3100	<code>\DeclareRobustCommand</code> , 2757	<code>\ensuremath</code> , 947, 960, 2092, 2320
<code>\bigl</code> , 3098	<code>\DeclareRobustCommand*</code> , 834, 835, 836, 837, 1043	<code>enumerate*</code> , <u>1942</u>
<code>\Bigr</code> , 3101	<code>\DeclareTextCommand</code> , 2001	<code>\env</code> , 1049
<code>\bigr</code> , 3099	<code>\DeclareTextCommandDefault</code> , 2003	<code>\equals</code> , <u>2459</u>
<code>\bigskipamount</code> , 2381, 3315	<code>\defobeylines</code> , <u>2369</u>	<code>\eTeX</code> , <u>2091</u> , 2095
<code>\bnamegroup</code> , <u>712</u>	<code>\dekbigskip</code> , <u>2381</u>	<code>\everymath</code> , 2995
<code>\boldmath</code> , 2034	<code>\dekfracc</code> , <u>2238</u> , <u>2247</u> , 2633	<code>\everypar</code> , 1687, 1697, 2849
<code>\box</code> , 2055, 3097	<code>\dekfraccsimple</code> , <u>2632</u> , 2638	<code>\exists</code> , 3046
<code>\bslash</code> , <u>892</u> , 1043, 1146, 1223, 1247, 1653, 1654, 1655, 2909, 2910, 2919, 2920	<code>\dekfraccslash</code> , 2627, 2635, 2636	<code>\f@encoding</code> , 2498
<code>\bullet</code> , 2981	<code>\dekmedskip</code> , <u>2380</u>	<code>\f@series</code> , 2034
<code>\c@gm@PronounGender</code> , <u>1409</u>	<code>\deksmallskip</code> , <u>2378</u>	<code>\fakern</code> , <u>3122</u>
<code>\c@NoNumSecs</code> , 1482	<code>\Delta</code> , 3010	<code>\fakesc@extrascap</code> , 2783, <u>2798</u>
<code>\c@secnumdepth</code> , 1536	<code>\dimen</code> , 2280, 2283, 2287, 2290, 2448, 2449, 2450, 2827, 2828	<code>\fakescaps</code> , <u>2757</u>
<code>\chardef</code> , 2981	<code>\dimexpr</code> , 3034, 3039, 3045, 3096, 3154, 3387, 3388	<code>\fakescapscore</code> , 2735, 2760
<code>\ClassError</code> , 1652	<code>\discre</code> , <u>986</u> , 995, 1012	<code>\fakescextrascap</code> , <u>2798</u>
<code>\cleardoublepage</code> , 1496	<code>\discret</code> , <u>988</u> , 993	<code>\file</code> , <u>1017</u>
<code>\clubpenalty</code> , 1691, 1696	<code>\divide</code> , 2024, 2027, 2028, 2282, 2284, 2289, 2291, 2779, 2780, 2785, 3530	<code>\fooatletter</code> , <u>241</u>
<code>\cmd</code> , <u>1068</u>	<code>\document</code> , 1379	<code>\foone</code> , <u>237</u> , 241, 629, 849, 852, 860, 876, 896, 900, 903, 1052, 2360, 2368, 2939, 2961, 3062
<code>\cmd@to@cs</code> , 1068, <u>1070</u>	<code>\dp</code> , 3080, 3093, 3096, 3154	<code>\forall</code> , 3043
<code>\color</code> , 2513, 2514		<code>\FormatHangHeading</code> , 1793, 1800, 1808, 1815
<code>\continuum</code> , <u>2547</u>		<code>\FormatRunInHeading</code> , 1822, 1827
<code>\copy</code> , 2017, 2047, 2061, 3153, 3165		<code>\freeze@actives</code> , <u>2483</u>
<code>\count</code> , 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2281, 2282, 2283, 2284, 2288, 2289, 2290,		<code>\fullpar</code> , <u>3370</u>
		<code>\g@emptyify</code> , 383, 384
		<code>\g@relaxen</code> , <u>392</u> , 393

<code>\gaddtomacro</code> , 354	<code>\gmathhook</code> , 3134	<code>\gmu@reserveda</code> , 606 ,
<code>\garamath</code> , <i>p.</i> 43 , 3149	<code>\gml@StoreCS</code> , 1173 , 1196 ,	608 , 614 , 616 , 753 ,
<code>\gemptyify</code> , 384 , 384	1233	755 , 1186 , 1189 , 1192 ,
<code>\glet</code> , 346 , 1711	<code>\gml@storemacros</code> , 1174 ,	1662 , 1664 , 1711 ,
<code>\glueexpr</code> , 3343 , 3386	1185 , 1194 , 1199 , 1236	1712 , 1715 , 1996 ,
<code>\gluestretch</code> , 3344	<code>\gmobeyspaces</code> , 2361	1998 , 2000 , 2001 ,
<code>\gm@atppron</code> , 1411 , 1415 ,	<code>\gmshowlists</code> , 480	2002 , 2150 , 2151 ,
1416 , 1417 , 1418 ,	<code>\GMtextsuperscript</code> , 2310	2155 , 2156 , 2658 , 2659
1420 , 1421 , 1422 , 1423	<code>\gmu@acroinner</code> , 2558 ,	<code>\gmu@reservedb</code> , 754 , 755
<code>\gm@clearpagesduetoopenright</code> ,	2568 , 2569 , 2577	<code>\gmu@restorespecials</code> ,
1495 , 1555	<code>\gmu@acrospace</code> , 2552 ,	3087 , 3570
<code>\gm@dontnumbersectionsoutofmainmatter</code> ,	2557 , 2557 , 2561	<code>\gmu@RPfor</code> , 2513 , 2525 ,
1493 , 1539	<code>\gmu@ampulexpa</code> , 458 , 466	2537 , 2547
<code>\gm@duppa</code> , 2239 , 2240 , 2242	<code>\gmu@ampulexpb</code> , 459 , 467	<code>\gmu@scalar</code> , 2770 , 2774 ,
<code>\gm@gobmacro</code> , 2136 , 2142	<code>\gmu@checkaftersec</code> ,	2775 , 2781
<code>\gm@hyperrefstepcounter</code> ,	1705 , 1764	<code>\gmu@scalematchX</code> , 2760 ,
1485 , 1488 , 1568	<code>\gmu@dashfalse</code> , 3272	2772 , 2796
<code>\gm@ifnac</code> , 590 , 592	<code>\gmu@dashtrue</code> , 3274	<code>\gmu@scapLetters</code> , 2731 ,
<code>\gm@letspace</code> , 609 , 617	<code>\gmu@def</code> , 1511 , 1513 , 1513 ,	2741 , 2746
<code>\gm@notprerr</code> , 1346 , 1353	1532	<code>\gmu@scapSpaces</code> , 2744 ,
<code>gm@PronounGender</code> , 1409	<code>\gmu@dekfracc</code> , 2222 ,	2749 , 2753
<code>\gm@pswords</code> , 1001 , 1003 , 1005	2241 , 2243	<code>\gmu@scapss</code> , 2752 , 2758
<code>\gm@sec</code> , 1909 , 1920 , 1921	<code>\gmu@dekfraccsimple</code> ,	<code>\gmu@scscale</code> , 2789 , 2795
<code>\gm@secini</code> , 1874 , 1893 ,	2243 , 2625 , 2633	<code>\gmu@septify</code> , 3058 , 3572
1899 , 1905 , 1918	<code>\gmu@denominatorkern</code> ,	<code>\gmu@setheading</code> , 1761 ,
<code>\gm@secmarkh</code> , 1901	2224 , 2253 , 2627	1767 , 1768
<code>\gm@secstar</code> , 1877 , 1887 ,	<code>\gmu@discretionaryslash</code> ,	<code>\gmu@setsetSMglobal</code> ,
1894 , 1906 , 1920 , 1921	1012 , 1023	1172 , 1177 , 1232
<code>\gm@secx</code> , 1909 , 1910	<code>\gmu@dywiz</code> , 2958 , 2962	<code>\gmu@setSMglobal</code> , 1179 ,
<code>\gm@secxx</code> , 1876 , 1904 , 1911	<code>\gmu@fileext</code> , 2675 , 2684 ,	1181 , 1199
<code>\gm@straightensec</code> ,	2684 , 2702	<code>\gmu@SMdo@scope</code> , 1275 ,
1915 , 1924	<code>\gmu@filename</code> , 2674 ,	1277 , 1280 , 1281 , 1295
<code>\gm@targetheading</code> ,	2687 , 2699 , 2702 ,	<code>\gmu@SMdo@setscope</code> ,
1486 , 1489	2705 , 2714	1273 , 1279 , 1293
<code>\gma</code> , 3142	<code>\gmu@getaddvs</code> , 1756 ,	<code>\gmu@SMglobalfalse</code> ,
<code>\gma@arrowdash</code> , 3152 ,	1756 , 1762	1140 , 1154 , 1181 ,
3161 , 3167 , 3168	<code>\gmu@getext</code> , 2673 , 2683	1190 , 1217 , 1226 , 1283
<code>\gma@bare</code> , 3138 , 3140	<code>\gmu@gobdef</code> , 195 , 201	<code>\gmu@SMglobaltrue</code> , 1116 ,
<code>\gma@bracket</code> , 3140 , 3141	<code>\gmu@ifnodash</code> , 3266 , 3271	1179
<code>\gma@checkbracket</code> ,	<code>\gmu@luzniej</code> , 2832 , 2835 ,	<code>\gmu@smtempa</code> , 1144 , 1153 ,
3139 , 3143	2837	1220 , 1225
<code>\gma@dollar</code> , 3137 , 3138 , 3143	<code>\gmu@nl@reserveda</code> ,	<code>\gmu@storespecials</code> ,
<code>\gma@gmathhook</code> , 3125 ,	1305 , 1308 , 1313 , 1316	3057 , 3567
3133 , 3134 , 3156	<code>\gmu@nocite@ampulex</code> ,	<code>\gmu@tempa</code> , 202 , 420 , 426 ,
<code>\gma@quantifierhook</code> ,	1371 , 1382	437 , 445 , 2141 , 2143 ,
3043 , 3046 , 3129 ,	<code>\gmu@numeratorkern</code> ,	2805 , 2807 , 2808 ,
3131 , 3163 , 3169	2223 , 2252 , 2253 , 2626	2809 , 3427 , 3430 ,
<code>\gma@tempa</code> , 3032 , 3034 ,	<code>\gmu@prevsec</code> , 1686 , 1688 ,	3432 , 3464 , 3468 ,
3037 , 3039 , 3092 ,	1710 , 1717 , 1747	3481 , 3485 , 3541 , 3545
3096 , 3117 , 3120	<code>\gmu@printslashes</code> ,	<code>\gmu@tempb</code> , 421 , 427 , 438 ,
<code>\gma@tempb</code> , 3093 , 3096	1017 , 1019 , 1019 ,	446
<code>\gmath</code> , <i>p.</i> 43 , 2994 , 3137 , 3141	1021 , 1024	<code>\gmu@tempc</code> , 422 , 452
	<code>\gmu@resa</code> , 2528 , 2530	<code>\gmu@tempd</code> , 424 , 425 , 428 ,
		431 , 436 , 438 , 443 ,
		444 , 455 , 459

<code>\gmu@tempe</code> , 435, 441, 457, 461	<code>\hyphenpenalty</code> , 1005, 2872, 3350	<code>\linebreak</code> , 3409
<code>\gmu@testdash</code> , <u>3270</u> , 3310, 3321	<code>\if@afterindent</code> , 1692	<code>\linedate</code> , <u>3312</u> , 3323, 3324
<code>\gmu@thou@fiver</code> , 3512, 3515, <u>3523</u> , 3523	<code>\if@filesw</code> , 2436, 2686, 2698, 2706	<code>\list</code> , 1948, 1966
<code>\gmu@thou@i</code> , 3517, 3535, 3543	<code>\if@mainmatter</code> , 1494	<code>\listparindent</code> , 1953, 1971
<code>\gmu@thou@ii</code> , 3517, 3535, 3543	<code>\if@nobreak</code> , 1689	<code>\lit</code> , <u>3474</u>
<code>\gmu@thou@o</code> , 3517, 3535, 3543	<code>\if@openright</code> , 1496	<code>\litshape</code> , 3456, 3472, 3474, 3496
<code>\gmu@thou@put</code> , 3516, 3520, 3532	<code>\if@specialpage</code> , 1554	<code>\liturgiques</code> , <u>2512</u>
<code>\gmu@thou@putter</code> , 3519, 3526, 3533, 3540, 3548	<code>\if@twoside</code> , 1580	<code>\longpauza</code> , <u>2928</u> , 2929
<code>\gmu@thousep</code> , 3545, <u>3552</u>	<code>\ifcsname</code> , 725	<code>\looseness</code> , 2838, 2849
<code>\gmu@tilde</code> , <u>2332</u> , 2337, 2348	<code>\ifdate</code> , <u>3307</u> , 3310, 3312	<code>\lpauza</code> , <u>2898</u>
<code>\gmu@whonly</code> , 2720, 2721	<code>\ifdefined</code> , 166, 193, 855, 914, 2162, 2538, 2858, 2938	<code>\lsl</code> , <u>3477</u>
<code>\gmu@xedekfracplain</code> , 2199, <u>2246</u>	<code>\iffontchar</code> , 2216	<code>luzniej</code> , <u>2842</u>
<code>\gmu@xedekfracstar</code> , 2199, <u>2214</u>	<code>\ifgmu@dash</code> , <u>3264</u> , 3270, 3276, 3310, 3321	<code>luzniej*</code> , <u>2847</u>
<code>\gmu@xefraccddef</code> , <u>2215</u> , 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237	<code>\ifgmu@postsec</code> , 1707, <u>1746</u> , 1754	<code>\luzniejcore</code> , <u>2834</u> , 2842
<code>\gn@melet</code> , 1312	<code>\ifgmu@SMglobal</code> , <u>1114</u> , 1138, 1145, 1178, 1215, 1221, 1280	<code>\macro</code> , 2142
<code>\gobble</code> , <u>185</u> , 187, 3118	<code>\ifHeadingNumbered</code> , 1535, 1563	<code>\MakeUppercase</code> , 2735
<code>\gobbletwo</code> , <u>188</u>	<code>\ifodd</code> , 1413	<code>\mapsto</code> , 3160
<code>\grefstepcounter</code> , <u>313</u> , 329	<code>\ifSecondClass</code> , <u>2611</u>	<code>\marg</code> , <u>1074</u> , 1098
<code>\grelaxen</code> , <u>393</u> , 393, 1688	<code>\ikern</code> , <u>2646</u>	<code>maszynopis</code> , <u>3341</u>
<code>\hathat</code> , <u>1053</u>	<code>\IMO</code> , <u>2587</u>	<code>\math@arg</code> , 1094, 1095
<code>\HeadingNumber</code> , 1565, 1567	<code>\in</code> , 3169	<code>\mathbin</code> , 3022, 3026, 3051, 3052, 3084, 3085, 3115, 3116, 3162, 3169
<code>\HeadingNumberedfalse</code> , 1494, 1536	<code>\inclasthook</code> , 2703, <u>2725</u>	<code>\mathchoice</code> , 3030, <u>3049</u> , <u>3063</u> , 3107, <u>3158</u>
<code>\HeadingRHeadText</code> , <u>1549</u>	<code>\includegraphics</code> , 2299	<code>\mathclose</code> , 3082, 3083, 3099, 3101, 3103, 3105, 3113
<code>\HeadingText</code> , <u>1551</u>	<code>\infy</code> , 3021	<code>\mathfrak</code> , 2547
<code>\HeadingTOCText</code> , <u>1550</u>	<code>\itemindent</code> , 1950, 1968	<code>\mathit</code> , 2996, 3008
<code>\HeShe</code> , <u>1420</u>	<code>itemize*</code> , <u>1960</u>	<code>\mathop</code> , <u>3030</u>
<code>\heshe</code> , <u>1415</u>	<code>\iteracro</code> , <u>2551</u> , 2555	<code>\mathopen</code> , 3068, 3083, 3098, 3100, 3102, 3104, 3112
<code>\hfillneg</code> , <u>2384</u>	<code>\justified</code> , <u>3359</u>	<code>\mathpunct</code> , 3067
<code>\hgrefstepcounter</code> , <u>328</u>	<code>\labelsep</code> , 1952, 1970	<code>\mathrel</code> , 3023, 3027, 3053, 3054, 3055, 3084, 3085, 3086, 3119, 3161, 3167, 3168
<code>\hidden@iffalse</code> , <u>291</u> , 429	<code>\labelwidth</code> , 1951, 1952, 1969, 1970	<code>\mathrm</code> , 3003, 3010, 3016, 3022, 3023, 3025, 3026, 3027, 3041
<code>\hidden@iftrue</code> , <u>292</u> , 429	<code>\larger</code> , <i>p. 11</i> , <u>834</u> , 3033, 3038, 3098, 3099, 3102, 3103, 3104, 3105	<code>\Mathstrutbox@</code> , 3080
<code>\HimHer</code> , <u>1422</u>	<code>\largerr</code> , <i>p. 11</i> , <u>838</u> , 3100, 3101	<code>\medmuskip</code> , 993
<code>\himher</code> , <u>1417</u>	<code>\LaTeXe</code> , 1984, <u>2033</u>	<code>\meta</code> , <u>939</u> , 975, 1074, 1081, 1088
<code>\HisHer</code> , <u>1421</u>	<code>\LaTeXpar</code> , <u>2044</u>	<code>\meta@font@select</code> , 950, <u>969</u>
<code>\hisher</code> , 1416	<code>\ldate</code> , <u>3324</u> , 3329	<code>\mkern</code> , 3122
<code>\HisHers</code> , <u>1423</u>	<code>\leftarrow</code> , 3053, 3167	<code>\mskip</code> , 993
<code>\hishers</code> , <u>1418</u>	<code>\leftline</code> , 3324	<code>\multiply</code> , 2023, 2026, 2787, 2837, 2848
<code>\hrule</code> , 1735	<code>\leftmargin</code> , 1949, 1967	<code>\mw@getflags</code> , 1710
<code>\hunskip</code> , 336, 2459, 2461, 2463, 3371, 3384	<code>\leftrightharpoonrightarrow</code> , 3055	
	<code>\leftslanting</code> , <u>3455</u>	

<code>\mw@HeadingBreakAfter,</code> 1556, 1576, 1591, 1595, <u>1625</u> , 1711	<code>\not@onlypreamble,</code> <u>1333</u> , 1337, 1338, 1339, 1340, 1341	<code>\polskadata,</code> <u>3183</u> , 3220
<code>\mw@HeadingBreakBefore,</code> 1553, <u>1624</u> , 1712	<code>\nu,</code> 3013	<code>\possfil,</code> <u>1058</u>
<code>\mw@HeadingLevel,</code> 1533, 1536	<code>\numexpr,</code> 2968, 2969, 3531	<code>\ppauza,</code> <u>2917</u>
<code>\mw@HeadingRunIn,</code> 1571, <u>1624</u>	<code>\oarg,</code> 1079	<code>\ppauza@skipcore,</code> <u>2862</u> , 2913, 2914
<code>\mw@HeadingType,</code> <u>1552</u> , 1686, 1718, 1719, 1732	<code>\oldLaTeX,</code> 1983	<code>\pprovide,</code> <u>207</u> , 2547
<code>\mw@HeadingWholeWidth,</code> 1574, <u>1625</u>	<code>\oldLaTeXe,</code> 1984	<code>\prependtomacro,</code> <u>369</u>
<code>\mw@normalheading,</code> 1578, 1587, 1590, 1594, 1767	<code>\omega,</code> 3020	<code>\printspaces,</code> <u>1001</u> , 1010
<code>\mw@processflags,</code> 1626	<code>\PackageError,</code> 1352, 2908, 2918	<code>\protected,</code> 178, 207, 267, 286, 3430, 3432
<code>\mw@runinheading,</code> 1572, 1768	<code>\PackageWarning,</code> 826, 829	<code>\provide,</code> <u>192</u> , 207
<code>\mw@secdef,</code> 1631, 1632, 1633, <u>1639</u>	<code>\pagebreak,</code> 1579, 1591, 1595	<code>\psi,</code> 3019
<code>\mw@section,</code> 1630	<code>\pagegoal,</code> 2448	<code>\quad,</code> 2986
<code>\mw@sectionxxx,</code> 1532	<code>\pagetotal,</code> 2449	<code>\quantifierhook,</code> <u>3130</u> , 3150
<code>\mw@secundef,</code> 1635, 1647, <u>1650</u>	<code>\paragraph,</code> 3326	
<code>\mw@setflags,</code> 1636	<code>\ParanoidPostsec,</code> <u>1743</u>	<code>\rdate,</code> <u>3323</u>
	<code>\parg,</code> <u>1086</u>	<code>\real,</code> 2781, 2783
<code>\n@melet,</code> <u>1304</u> , 1661, 1665, 1878, 1884, 1918, 2219	<code>\partial,</code> 3024	<code>\reflectbox,</code> 2103, 2110
<code>\nameshow,</code> <u>483</u>	<code>\partopsep,</code> 1949, 1967	<code>\relaxen,</code> <u>388</u> , 388, 1621, 2948, 3007, 3517
<code>\nameshowthe,</code> <u>484</u>	<code>\pauza,</code> <u>2881</u>	<code>\relsize,</code> <i>p. 11</i> , 799, 800, 834, 835, <u>836</u> , 837, 838, 839
<code>napapierki,</code> <u>2825</u>	<code>\pauza@skipcore,</code> <u>2859</u> , 2870, 2871	<code>\renewcommand*,</code> 2537
<code>\napapierkicore,</code> <u>2822</u> , 2826	<code>\pauzacore,</code> <u>2860</u> , 2872, 2877, 2885, 2894, 2899, <u>2928</u> , <u>2931</u> , 3353, <u>3354</u>	<code>\RequirePackage,</code> 2171, 2433, 2530, 2765
<code>\napapierkistretch,</code> <u>2820</u> , 2823	<code>\pauzadial,</code> <u>2887</u> , <u>2893</u>	<code>\resetMathstrut@,</code> <u>3073</u>
<code>\nawj,</code> <u>2851</u>	<code>\pdef,</code> 178, 192, 255, 272, 291, 292, 313, 328, 336, 553, 589, 630, 799, 838, 839, 939, 1010, 1017, 1032, 1049, 1053, 1116, 1127, 1170, 1207, 1229, 1251, 1255, 1486, 2004, 2191, 2317, 2337, 2345, 2459, 2461, 2463, 2552, 2590, 2657, 2866, 2881, 2893, 2898, 2907, 2917, 2994, 3332, 3335, 3434, 3456, 3471, 3474, 3477, 3493, 3502, 3508, 3560	<code>\resizebox,</code> 2298
<code>\nazwired,</code> <u>2986</u>	<code>\pdfTeX,</code> <u>2095</u>	<code>\resizegraphics,</code> <u>2277</u> , <u>2297</u>
<code>\neg,</code> 3022, 3121	<code>\pdfTeX,</code> <u>2097</u>	<code>\Restore@Macro,</code> 1210, <u>1213</u> , 1233, 1243
<code>\neq,</code> 3023, 3116	<code>\Phi,</code> 3016	<code>\Restore@Macros,</code> 1229, <u>1231</u>
<code>\neqb,</code> <u>3116</u>	<code>\phi,</code> 3015	<code>\Restore@MacroSt,</code> 1211, <u>1219</u>
<code>\newcount,</code> 2832	<code>\pi,</code> 3014	<code>\RestoreEnvironment,</code> <u>1255</u>
<code>\newcounter,</code> 1409, 1482	<code>\pk,</code> <u>1032</u>	<code>\RestoreMacro,</code> 1207, 1513, 2178, 2180
<code>\newgif,</code> <u>255</u>	<code>\PlainTeX,</code> <u>2083</u>	<code>\RestoreMacro*,</code> 1257, 2180
<code>\newskip,</code> 3416	<code>\pm,</code> 3025, 3026	<code>\RestoreMacros,</code> <u>1229</u>
<code>\newwrite,</code> 2436		<code>\RestoringDo,</code> <u>1292</u>
<code>\nfss@text,</code> 948		<code>\rightarrow,</code> 3054, 3168
<code>\nieczer,</code> <u>2520</u>		<code>\rightline,</code> 3323, 3405
<code>\nobreakspace,</code> <u>2752</u>		<code>\romannumeral,</code> 1947, 1965
<code>\nocite,</code> 1378		<code>\rotatebox,</code> 3043, 3046, 3051, 3052
<code>\nohy,</code> <u>2650</u>		<code>\rs@size@warning,</code> 818, 823, <u>826</u>
<code>\nolimits,</code> 3045, 3046		<code>\rs@unknown@warning,</code> 813, <u>829</u>
<code>\nolinkurl,</code> 3585		<code>\runindate,</code> <u>3325</u>
<code>NoNumSecs,</code> <u>1482</u>		<code>\scantokens,</code> 3061
		<code>\scshape,</code> 2081, 2582, 3162

<code>\secondclass</code> , 2610	<code>\textsl</code> , 2088	<code>\Vs</code> , 3430
<code>\SecondClasstrue</code> , 2612	<code>\textsmaller</code> , <u>837</u>	<code>\vs</code> , <u>995</u> , 1001, 1005
<code>\sectionsign</code> , 3000	<code>\textstyle</code> , 2035	
<code>\SetSectionFormatting</code> , 1621, <u>1622</u> , 1786, 1790, 1798, 1806, 1813, 1820, 1825	<code>\textsuperscript</code> , <u>2312</u> , <u>2317</u>	<code>\wd</code> , 2015, 2018, 2048, 2053, 2061, 2062, 2281, 3045, 3153, 3154, 3165, 3166
<code>\SetTwoheadSkip</code> , <u>1770</u> , 1797, 1805, 1812	<code>\texttilde</code> , 2345	<code>\Web</code> , <u>2085</u>
<code>\sfname</code> , <u>1010</u> , 1020	<code>\textwidth</code> , 3387, 3388, 3406, 3407	<code>\wedge</code> , 3052, 3121
<code>\shortpauza</code> , <u>2930</u>	<code>\thedate</code> , 3329	<code>\whenonly</code> , <u>2719</u>
<code>\showboxbreadth</code> , 480	<code>\thickmuskip</code> , 3123	<code>\whern</code> , <u>3412</u>
<code>\showboxdepth</code> , 480	<code>\thousep</code> , <u>3508</u> , 3560	<code>\wherncore</code> , <u>3404</u> , 3413, 3419
<code>\ShowFont</code> , <u>2497</u>	<code>\thr@@</code> , 1943, 1961	<code>\whernskip</code> , 3413, <u>3416</u> , 3417
<code>\showlists</code> , 480	<code>\time</code> , 2968, 2969	<code>\whernup</code> , <u>3419</u>
<code>\showthe</code> , 484	<code>\tinycae</code> , <u>2763</u>	<code>\WPheadings</code> , <u>1785</u>
<code>\sigma</code> , 3017	<code>\TODO</code> , <u>2396</u>	<code>\Ws</code> , 3432
<code>\sim</code> , 3027	<code>\toks</code> , 1662, 1663, 1664, 1757, 1758, 1764, 1765	<code>\wyzejnizej</code> , <u>2804</u>
<code>\SliTeX</code> , <u>2080</u>	<code>\tolerance</code> , 2837, 2848, 3345	<code>\Wz</code> , <u>3434</u>
<code>\smaller</code> , <i>p. 11</i> , <u>835</u> , 2626, 2629	<code>\truetextsuperscript</code> , 2314, <u>2316</u>	<code>\xathousep</code> , <u>3560</u>
<code>\smallerr</code> , <i>p. 11</i> , <u>839</u> , 2796	<code>\twocoltoc</code> , <u>2432</u> , 2441	<code>\Xedekfrac</code> , <u>2199</u>
<code>\smallskipamount</code> , 2378, 2379	<code>\twopar</code> , 3383	<code>\XeLaTeX</code> , <u>2107</u>
<code>\smartunder</code> , <u>878</u>	<code>\twoparinit</code> , <u>3381</u>	<code>\XeTeX</code> , <u>2100</u>
<code>\SMglobal</code> , <u>1116</u>	<code>\tytul</code> , <u>3335</u>	<code>\XeTeXinputencoding</code> , 167
<code>\Store@Macro</code> , 1133, <u>1136</u> , 1173	tytulowa, <u>2983</u>	<code>\XeTeXpicfile</code> , 2278, 2295
<code>\Store@Macros</code> , 1170, <u>1171</u>	<code>\udigits</code> , <u>2191</u> , 2194, 2626, 2629	<code>\XeTeXthree</code> , <u>2167</u>
<code>\Store@MacroSt</code> , 1134, <u>1143</u>	<code>\undeksmallskip</code> , <u>2379</u>	<code>\XeTeXversion</code> , 156, 157, 166, 855, 2162, 2163, 2858, 2938
<code>\Stored@Macro</code> , 1242, <u>1243</u>	<code>\unex@namedef</code> , <u>2149</u>	<code>\xiand</code> , 901
<code>\storedcsname</code> , <u>1246</u> , 3585	<code>\unex@nameuse</code> , <u>2154</u>	<code>\xiibackslash</code> , <u>888</u> , 892
<code>\StoredMacro</code> , <u>1242</u>	<code>\unexpanded</code> , 370, 445, 446, 450, 451, 454	<code>\xiilbrace</code> , <u>863</u>
<code>\StoreEnvironment</code> , <u>1251</u>	<code>\unless</code> , 2163, 2538	<code>\xiipercen</code> , <u>897</u>
<code>\StoreMacro</code> , <u>1127</u> , 1513, 2037, 2170, 3353, 3584	<code>\upshape</code> , 3497	<code>\xiirbrace</code> , <u>864</u>
<code>\StoreMacro*</code> , 1253, 2038	<code>\url</code> , 3584, <u>3585</u>	<code>\xiispace</code> , 496, 497, <u>904</u> , 917
<code>\StoreMacros</code> , <u>1170</u>	<code>\urladdstar</code> , <u>3581</u> , 3588	<code>\xiistring</code> , <u>495</u>
<code>\StoringAndRelaxingDo</code> , <u>1272</u>	<code>\usc</code> , <u>2590</u> , 2592	<code>\xiiunder</code> , <u>853</u> , 856, 857
<code>\strip@pt</code> , 3460, 3465, 3482	<code>\uscacro</code> , <u>2592</u>	<code>\xxt@visible</code> , 914, 915
<code>\subs</code> , <u>850</u> , 880	<code>\usecounter</code> , 1954	<code>\yeshy</code> , <u>2651</u>
<code>\sum</code> , 3042		
<code>\TB</code> , 2089	<code>\value</code> , 1413	<code>\z@skip</code> , 3372
<code>\TeXbook</code> , <u>2088</u> , 2089	<code>\varepsilon</code> , 2035, 2092, 3011	<code>\zf@scale</code> , <u>2771</u> , 2774, 2775
<code>\textbullet</code> , <u>2981</u> , 2981	<code>\varnothing</code> , 3163	<code>\zwrobcy</code> , <u>3332</u>
<code>\textcolor</code> , 2520	<code>\varsigma</code> , 3018	
<code>\textlarger</code> , <u>836</u>	<code>\vartheta</code> , 3012	<code>\cdot</code> , <u>2463</u>
<code>\textlit</code> , <u>3471</u>	<code>\vee</code> , 3051, 3121	
	<code>\verb</code> , 2170, 2178	<code>\-</code> , <u>2907</u> , 2944
	<code>\visible</code> , 915, 917, 995	<code>\-</code> , <u>2866</u> , 2925, 2943