

The `paresse` package^{*}

Yvon Henel[†]

Released 2020/10/06

Résumé

Cette extension, reprenant un exemple de T. LACHAND-ROBERT dans [1], fournit un moyen de taper des lettres grecques isolées à l'aide du caractère `§` actif et redéfini. Au lieu de `\(\alpha\)` ou tape `§a` pour obtenir α .

Important : Il doit être chargé **après** `inputenc` si ce dernier est utilisé. De plus, il faut que le signe `§` soit une lettre pour `TEX`.

Depuis la version 4, on peut utiliser cette extension même dans un source codé en utf-8 avec `LATEX`, `LuaLATEX` ou `XƎLATEX`.

La documentation française pour l'utilisateur de l'extension `paresse` est disponible sous le nom de `paresse-fra`.

Abstract

This package implements an example from T. LACHAND-ROBERT in [1]. It provides a means of typing isolated greek letters with the character `§` activated and redefined. Instead of `\(\alpha\)` one types `§a` to obtain α .

Important: You have to load it **after** the `inputenc` package if the latter is used. Moreover the sign `§` must be a letter for `TEX`.

Since version 4, one can use this package even with utf8-encoded source for `LATEX`, `LuaLATEX`, or `XƎLATEX`.

The English documentation for the final user of the package `paresse` is available in the file `paresse-eng`.

1 `paresse` implementation

Three packages are coded below: the main one `paresse` loads one of the other two `paresse-old` or `paresse-utf8` depending on the `TEX`-engine used and the encoding of the source.

The prefix is `paresse` for the three packages.

```
1 <@@=paresse>
```

The main package `paresse` begins here.

```
2 \ProvidesExplPackage
```

```
3 {\ExplFileName}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}
```

^{*}This file describes v5.0, last revised 2020/10/06.

[†]E-mail: le.texnicien.de.surface@yvon-henel.fr

Header of paresse-old:

```
4 \ProvidesExplPackage
5   {\ExplFileName-old}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}
```

Header of paresse-utf8:

```
6 \ProvidesExplPackage
7   {\ExplFileName-utf8}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}
```

The three packages require xparse and l3keys2:

```
8 \RequirePackage{xparse}
9 \RequirePackage{l3keys2e}
```

but the main paresse alone requires iftex to proceed according to the T_EX-engine.

```
10 \RequirePackage{iftex}
11 \cs_if_exist:NF \__paresse_main_package_loaded:
12 {
13   \keys_define:nn {paresse}
14   {
15     encoding_is_legacy .bool_set:N = \g__paresse_encoding_is_legacy_bool,
16     encoding_is_levi .bool_set_inverse:N =
17       \g__paresse_encoding_is_legacy_bool,
18     legacy .meta:n = {
19       encoding_is_legacy = #1
20     },
21     legacy .default:n = {true},
22     legacy .initial:n = {true},
23     levi .meta:n = {
24       encoding_is_levi = #1
25     },
26     levi .default:n = {true},
27     levi .initial:n = {false},
28
29     mood_is_wild .bool_set:N = \g__paresse_mood_is_wild_bool,
30     mood_is_tame .bool_set_inverse:N =
31       \g__paresse_mood_is_wild_bool,
32     wild .meta:n = {
33       mood_is_wild = #1
34     },
35     wild .default:n = {true},
36     wild .initial:n = {true},
37     tame .meta:n = {
38       mood_is_tame = #1
39     },
40     tame .default:n = {true},
41     tame .initial:n = {false},
42
43     letter_t_is_theta .bool_set:N = \g__paresse_letter_t_is_theta_bool,
44     letter_t_is_tau .bool_set_inverse:N =
45       \g__paresse_letter_t_is_theta_bool,
46     theta .meta:n = {
47       letter_t_is_theta = #1
48     },
49     theta .default:n = {true},
50     theta .initial:n = {true},
```

```

51 tau .meta:n = {
52     letter_t_is_tau = #1
53 },
54 tau .default:n = {true},
55 tau .initial:n = {false},
56
57 letter_theta_is_theta .bool_set:N = \g__paresse_letter_theta_is_theta_bool,
58 letter_theta_is_vartheta .bool_set_inverse:N =
59     \g__paresse_letter_theta_is_theta_bool,
60 theta .meta:n = {
61     letter_theta_is_theta = #1
62 },
63 theta .default:n = {true},
64 theta .initial:n = {true},
65 vartheta .meta:n = {
66     letter_theta_is_vartheta = #1
67 },
68 vartheta .default:n = {true},
69 vartheta .initial:n = {false},
70
71 letter_pi_is_pi .bool_set:N = \g__paresse_letter_pi_is_pi_bool,
72 letter_pi_is_varpi .bool_set_inverse:N =
73     \g__paresse_letter_pi_is_pi_bool,
74 pi .meta:n = {
75     letter_pi_is_pi = #1
76 },
77 pi .default:n = {true},
78 pi .initial:n = {true},
79 varpi .meta:n = {
80     letter_pi_is_varpi = #1
81 },
82 varpi .default:n = {true},
83 varpi .initial:n = {false},
84
85 letter_rho_is_rho .bool_set:N = \g__paresse_letter_rho_is_rho_bool,
86 letter_rho_is_varrho .bool_set_inverse:N =
87     \g__paresse_letter_rho_is_rho_bool,
88 rho .meta:n = {
89     letter_rho_is_rho = #1
90 },
91 rho .default:n = {true},
92 rho .initial:n = {true},
93 varrho .meta:n = {
94     letter_rho_is_varrho = #1
95 },
96 varrho .default:n = {true},
97 varrho .initial:n = {false},
98
99 letter_sigma_is_sigma .bool_set:N = \g__paresse_letter_sigma_is_sigma_bool,
100 letter_sigma_is_varsigma .bool_set_inverse:N =
101     \g__paresse_letter_sigma_is_sigma_bool,
102 sigma .meta:n = {
103     letter_sigma_is_sigma = #1
104 },

```

```

105   sigma .default:n = {true},
106   sigma .initial:n = {true},
107   varsigma .meta:n = {
108     letter_sigma_is_varsigma = #1
109   },
110   varsigma .default:n = {true},
111   varsigma .initial:n = {false},
112
113   letter_epsilon_is_epsilon .bool_set:N =
114     \g__paresse_letter_epsilon_is_epsilon_bool,
115   letter_epsilon_is_varepsilon .bool_set_inverse:N =
116     \g__paresse_letter_epsilon_is_epsilon_bool,
117   epsilon .meta:n = {
118     letter_epsilon_is_epsilon = #1
119   },
120   epsilon .default:n = {true},
121   epsilon .initial:n = {false},
122   varepsilon .meta:n = {
123     letter_epsilon_is_varepsilon = #1
124   },
125   varepsilon .default:n = {true},
126   varepsilon .initial:n = {true},
127
128   letter_phi_is_phi .bool_set:N = \g__paresse_letter_phi_is_phi_bool,
129   letter_phi_is_varphi .bool_set_inverse:N =
130     \g__paresse_letter_phi_is_phi_bool,
131   phi .meta:n = {
132     letter_phi_is_phi = #1
133   },
134   phi .default:n = {true},
135   phi .initial:n = {false},
136   varphi .meta:n = {
137     letter_phi_is_varphi = #1
138   },
139   varphi .default:n = {true},
140   varphi .initial:n = {true},
141 }
142 \ProcessKeysOptions {paresse}
143 \bool_if:nTF {\g__paresse_letter_epsilon_is_epsilon_bool}
144 {\cs_new:Npn \__paresse_epsilon {\epsilon}}
145 {\cs_new:Npn \__paresse_epsilon {\varepsilon}}
146
147 \bool_if:nTF {\g__paresse_letter_theta_is_theta_bool}
148 {\cs_new:Npn \__paresse_theta {\theta}}
149 {\cs_new:Npn \__paresse_theta {\vartheta}}
150
151 \bool_if:nTF {\g__paresse_letter_pi_is_pi_bool}
152 {\cs_new:Npn \__paresse_pi {\pi}}
153 {\cs_new:Npn \__paresse_pi {\varpi}}
154
155 \bool_if:nTF {\g__paresse_letter_rho_is_rho_bool}
156 {\cs_new:Npn \__paresse_rho {\rho}}
157 {\cs_new:Npn \__paresse_rho {\varrho}}

```

```

158
159 \bool_if:nTF {\g__paresse_letter_sigma_is_sigma_bool}
160 {\cs_new:Npn \__paresse_sigma {\sigma}}
161 {\cs_new:Npn \__paresse_sigma {\varsigma}}
162
163 \bool_if:nTF {\g__paresse_letter_phi_is_phi_bool}
164 {\cs_new:Npn \__paresse_phi {\phi}}
165 {\cs_new:Npn \__paresse_phi {\varphi}}
166
167 \bool_new:c {g__paresse_file_encoding_is_utf8_bool}
168 }
169 \cs_new:Nn \__paresse_main_package_loaded: {}
170 \ifluatex
luatex specific code
171 \PackageInfo{paresse}{compiling~with~luatex}
172 \RequirePackageWithOptions{paresse-old}
173 \else
code for other engines
174 \ifxetex
xetex specific code
175 \PackageInfo{paresse}{compiling~with~xetex}
176 \RequirePackageWithOptions{paresse-old}
177 \else
nor luatex nor xetex
178
179 \PackageInfo{paresse}{compiling~with~tex}
180
181 \newcommand\default@encoding@message{
182   \bool_gset_true:c {g__paresse_file_encoding_is_utf8_bool}
183   \PackageWarningNoLine{paresse}{
184     The~package~"paresse"~expects~a~"utf8"~encoded~file.~
185     In~case~of~an~8~bits~encoding,~
186     please~use~inputenc~with~the~correct~option}
187 }
188
189 \newcommand\find@encoding{
190   \begingroup
191   \def\utf@encoding{utf8}
192   \def\temp@a{
193     \bool_gset_false:c {g__paresse_file_encoding_is_utf8_bool}
194     \PackageInfo{paresse}{8-bit~encoding~expected\@gobble}
195   }
196   \ifx\utf@encoding\inputencodingname
197   \def\temp@a{
198     \bool_gset_true:c {g__paresse_file_encoding_is_utf8_bool}
199     \PackageInfo{paresse}{utf8~encoding~detected\@gobble}
200   }
201   \fi
202   \expandafter\endgroup\temp@a}
203

```

```

204 \@ifpackageloaded{inputenc}
205 {
206   \PackageInfo{paresse}{inputenc~loaded}
207 }
208 {
209   \PackageInfo{paresse}{inputenc~not~loaded,~utf8~expected\@gobble}
210 }
211
212 \AtBeginDocument
213 {
214   \@ifpackageloaded{inputenc} \find@encoding \default@encoding@message
215   \ExplSyntaxOn
216   \makeatletter
217   \bool_if:nTF {\use:c{g__paresse_file_encoding_is_utf8_bool}}
218   {
219     \PackageInfo{paresse}{charge~utf8}
220     \RequirePackageWithOptions{paresse-utf8}
221   }
222   {
223     \PackageInfo{paresse}{charge~vieux}
224     \RequirePackageWithOptions{paresse-old}
225   }
226   \makeatother
227   \ExplSyntaxOff
228 }
229
230 \fi % END OF ifxetex
231 \fi % END OF ifluatex
232
233 \count@=\catcode'\^a7
234 \let\GA@LaVieilleParesse= ^^a7
235 \def\^^a7{\S}
236 \newcommand{\makeparesseletter}{\catcode '\^^a7=11\relax}
237 \newcommand{\makeparesseother}{\catcode '\^^a7=12\relax}
238 \newcommand{\GA@MakeParesseActive}{\catcode '\^^a7=\active}
239 \def\GA@ActiveLaParessexii{\catcode '\^^a7=\active}
240 \def\GA@ActiveLaParessexiii{\catcode '\^^a7=\active
241   \def^^a7##1{\ensuremath{\csname \string^^a7##1\endcsname}}}
242 \ifnum\count@=12
243   \let\GA@ActiveLaParesse \GA@ActiveLaParessexii
244 \else
245   \let\GA@ActiveLaParesse \GA@ActiveLaParessexiii
246 \fi
247 \newcommand{\ActiveLaParesse}{\GA@ActiveLaParesse}
248 \newenvironment{ParesseActive}{\GA@ActiveLaParesse}{\GA@ActiveLaParesse}{}
249
250 \makeparesseletter
251 \def\^^a7a {\alpha}
252 \def\^^a7b {\beta}
253 \def\^^a7g {\gamma}
254 \def\^^a7d {\delta}
255 \def\^^a7e {\_\_paresse_epsilon}
256 \def\^^a7z {\zeta}
257 \def\^^a7h {\eta}
258 \def\^^a7i {\iota}

```

```

258 \def^^a7k {\kappa}
259 \def^^a7l {\lambda}
260 \def^^a7m {\mu}
261 \def^^a7n {\nu}
262 \def^^a7x {\xi}
263 \def^^a7p {\_paresse_pi}
264 \def^^a7r {\_paresse_rho}
265 \def^^a7s {\_paresse_sigma}
266 \def^^a7u {\upsilon}
267 \def^^a7f {\_paresse_phi}
268 \def^^a7w {\omega}
269
270 \bool_if:nTF {\g__paresse_encoding_is_legacy_bool}
271 {
272   \def^^a7j {\varsigma}
273   \def^^a7v {\_paresse_theta}
274   \def^^a7y {\tau}
275   \def^^a7c {\chi}
276   \def^^a7q {\psi}
277   \bool_if:nTF {\g__paresse_letter_t_is_theta_bool}
278   {
279     \def^^a7t {\_paresse_theta}
280   }
281   {
282     \def^^a7t {\tau}
283   }
284 }
285 {
286   \def^^a7j {\_paresse_theta}
287   \def^^a7t {\tau}
288   \def^^a7q {\chi}
289   \def^^a7y {\psi}
290   \def^^a7c {\varsigma}
291 }
292 \def^^a7G {\Gamma}
293 \def^^a7D {\Delta}
294 \def^^a7L {\Lambda}
295 \def^^a7P {\Pi}
296 \def^^a7S {\Sigma}
297 \def^^a7U {\Upsilon}
298 \def^^a7F {\Phi}
299 \def^^a7X {\Xi}
300 \def^^a7W {\Omega}
301 \bool_if:nTF {\g__paresse_encoding_is_legacy_bool}
302 {
303   \def^^a7V {\Theta}
304   \def^^a7T {\Theta}
305   \def^^a7Q {\Psi}
306 }
307 {
308   \def^^a7J {\Theta}
309   \def^^a7Y {\Psi}
310 }
311 \def^^a7Z {\S}

```

```

312 \catcode '\^a7=\active
313 \def^^a7#1{\ensuremath{\csname \string^^a7#1\endcsname}}
314
315 \bool_if:nF {\g__paresse_mood_is_wild_bool}
316 {
317   \catcode'\^a7=\count@
318   \ifnum\count@=13 \let ^^a7=\GA@LaVieilleParesse \fi
319 }
320 \begingroup
321 \catcode'\ "=12
322 \catcode'\ <=12
323 \catcode'\ .=12
324 \catcode'\ ,=12
325 \catcode'\ ;=12
326 \catcode'\ !=12
327 \catcode'\ -=13
328 \global\let\GA@parse@UTFviii@a=\parse@UTFviii@a
329 \global\let\GA@parse@UTFviii@b=\parse@UTFviii@b
330 \gdef\declareunicodecharacter#1#2{
331   \count@"#1\relax
332   \wlog{\space\space Redefining~Unicode~char~U+#1~(decimal~\the\count@)}
333   \begingroup
334     \GA@parse@XML@charref
335     \def\UTFviii@two@octets##1##2{\csname u8:##1\string##2\endcsname}
336     \def\UTFviii@three@octets##1##2##3{\csname u8:##1
337       \string##2\string##3\endcsname}
338     \def\UTFviii@four@octets##1##2##3##4{\csname u8:##1
339       \string##2\string##3\string##4\endcsname}
340     \expandafter\expandafter\expandafter
341     \expandafter\expandafter\expandafter
342     \expandafter
343     \gdef\UTFviii@tmp{\IeC{#2}}
344     \endgroup}
345 \gdef\GA@parse@XML@charref{
346   \ifnum\count@<"A0\relax
347     \PackageError{inputenc}{Cannot\space define\space Unicode\space
348       char\space value\space <\space 00A0}\@eha
349   \else\ifnum\count@<"800\relax
350     \GA@parse@UTFviii@a,
351     \GA@parse@UTFviii@b C\UTFviii@two@octets.,
352   \else\ifnum\count@<"10000\relax
353     \GA@parse@UTFviii@a;
354     \GA@parse@UTFviii@a,
355     \GA@parse@UTFviii@b E\UTFviii@three@octets.{,;;}
356   \else
357     \GA@parse@UTFviii@a;
358     \GA@parse@UTFviii@a,
359     \GA@parse@UTFviii@a!
360     \GA@parse@UTFviii@b F\UTFviii@four@octets.{!,;;}
361   \fi
362   \fi
363   \fi}
364 \endgroup
365 \newcommand{\makeparesseletter}{\DeclareUnicodeCharacter{00A7}\S}

```



```

366 \newcommand{\makeparesseother}{
367   \PackageWarning{paresse}{
368     Command \string\makeparesseother disabled}{
369     The command \string\makeparesseother would have been meaningless
370     in the current context}}
371 \newcommand{\ActiveLaParesse}{\declareunicodecharacter{00A7}\@paresse}
372 \newenvironment{ParesseActive}{\ActiveLaParesse}{
373 \ActiveLaParesse
374 \def\@paresse{\futurelet\@toks\@i@paresse}
375 \def\@i@paresse{
376   \csname@ifcat a\@toks @ii@paresse\else S\fi\endcsname}
377 \def\@ii@paresse#1{
378   \ifcsname @paresse@#1\endcsname
379   \csname @paresse@#1\endcsname\expandafter\@gobble
380   \else
381     \expandafter\@firstofone
382     \fi{\S#1}}
383 \def\define@paresse@car#1#2{
384   \expandafter\def\csname @paresse@#1\endcsname{\ensuremath{#2}}}
385
386 \define@paresse@car Z\S
387 \define@paresse@car a\alpha
388 \define@paresse@car b\beta
389 \define@paresse@car g\gamma
390 \define@paresse@car d\delta
391 \define@paresse@car e\_\paresse_epsilon
392 \define@paresse@car z\zeta
393 \define@paresse@car h\eta
394 \define@paresse@car i\iota
395 \define@paresse@car k\kappa
396 \define@paresse@car l\lambda
397 \define@paresse@car m\mu
398 \define@paresse@car n\nu
399 \define@paresse@car x\xi
400 \define@paresse@car p\_\paresse_pi
401 \define@paresse@car r\_\paresse_rho
402 \define@paresse@car s\_\paresse_sigma
403 \define@paresse@car u\upsilon
404 \define@paresse@car f\_\paresse_phi
405 \define@paresse@car w\omega
406
407 \bool_if:nTF {\g__paresse_encoding_is_legacy_bool}
408 {
409   \define@paresse@car j\varsigma
410   \define@paresse@car v\_\paresse_theta
411   \define@paresse@car y\tau
412   \define@paresse@car c\chi
413   \define@paresse@car q\psi
414
415   \bool_if:nTF {\g__paresse_letter_t_is_theta_bool}
416   {
417     \define@paresse@car t\_\paresse_theta
418   }
419 }

```

```

420 \define@paresse@car t\tau
421 }
422 }
423 {
424 \define@paresse@car j\__paresse_theta
425 \define@paresse@car t\tau
426 \define@paresse@car q\chi
427 \define@paresse@car y\psi
428 \define@paresse@car c\varsigma
429 }
430 \define@paresse@car G\Gamma
431 \define@paresse@car D\Delta
432 \define@paresse@car L\Lambda
433 \define@paresse@car P\Pi
434 \define@paresse@car S\Sigma
435 \define@paresse@car U\Upsilon
436 \define@paresse@car F\Phi
437 \define@paresse@car X\Xi
438 \define@paresse@car W\Omega
439
440 \bool_if:nTF {\g__paresse_encoding_is_legacy_bool}
441 {
442 \define@paresse@car V\Theta
443 \define@paresse@car T\Theta
444 \define@paresse@car Q\Psi
445 }
446 {
447 \define@paresse@car J\Theta
448 \define@paresse@car Y\Psi
449 }
450
451 \bool_if:nF {\g__paresse_mood_is_wild_bool}{\makeparesseletter}

```

References

- [1] T. LACHAND-ROBERT. *La maîtrise de T_EX et L^AT_EX*. Masson, Paris, Milan, Barcelone, 1995. ISBN : 2-225-84832-7.

Change History

v0.0		v2
General: New name, some redefinitions, first public version . . .	1	General: A useless macro is deleted . . . 1 Documentation reorganised 1 New wrapping to provide a tds.zip. . . 1
v1		v2.1
General: Documentation updated	1	General: Correction of a bug in Infofile, once again 1
v1a		v3
General: Correction of a bug in Infofile which prevented the loading of the package	1	General: Option for Sylvio LEVI's encoding 1

v4		(moving forward) skeyval	1
	General: Documentation of the code in	v5.0	
	English	General: <code>\varsigma</code> is added as <code>\$j</code> to	
	parese goes utf8	legacy encoding.	1
v4.1		Two sub-packages.	1
	General: Using skeyval-bc instead of	Uses expl3 to replace skeyval.	1

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols		\cs_new:Nn 169
\! 326	\cs_new:Npn 144, 145, 148, 149,
\" 321	152, 153, 156, 157, 160, 161, 164, 165
\, 324	\csname 240, 313, 335, 336, 338, 376, 379, 384
\. 323	
\; 325	D
\< 322	\DeclareUnicodeCharacter 365
\^ 232, 234, 235, 236, 237,	\declareunicodecharacter 330, 371
	238, 239, 250, 251, 252, 253, 254,	\def 191, 192, 197, 234, 238,
	255, 256, 257, 258, 259, 260, 261,	239, 240, 250, 251, 252, 253, 254,
	262, 263, 264, 265, 266, 267, 268,	255, 256, 257, 258, 259, 260, 261,
	272, 273, 274, 275, 276, 279, 282,	262, 263, 264, 265, 266, 267, 268,
	286, 287, 288, 289, 290, 292, 293,	272, 273, 274, 275, 276, 279, 282,
	294, 295, 296, 297, 298, 299, 300,	286, 287, 288, 289, 290, 292, 293,
	303, 304, 305, 308, 309, 311, 312, 317	294, 295, 296, 297, 298, 299, 300,
\~ 327	303, 304, 305, 308, 309, 311, 313,
		335, 336, 338, 374, 375, 377, 383, 384
A		\Delta 293, 431
\active 237, 238, 239, 312		\delta 253, 390
\ActiveLaParsesse 246, 371, 372, 373		E
\alpha 250, 387		\else . 173, 177, 243, 349, 352, 356, 376, 380
\AtBeginDocument 212		\endcsname 240,
		313, 335, 337, 339, 376, 378, 379, 384
B		\endgroup 202, 344, 364
\begingroup 190, 320, 333		\ensuremath 240, 313, 384
\beta 251, 388		\epsilon 144
bool commands:		epsilon internal commands:
\bool_gset_false:N 193		__paresse_epsilon . 144, 145, 254, 391
\bool_gset_true:N 182, 198		\eta 256, 393
\bool_if:nTF		\expandafter 202, 340, 341, 342, 379, 381, 384
. 143, 147, 151, 155, 159, 163, 217,		\ExplFileDate 3, 5, 7
270, 277, 301, 315, 407, 415, 440, 451		\ExplFileDescription 3, 5, 7
\bool_new:N 167		\ExplFileName 3, 5, 7
		\ExplFileVersion 3, 5, 7
C		\ExplSyntaxOff 227
\catcode 232, 235, 236, 237, 238, 239, 312,		\ExplSyntaxOn 215
317, 321, 322, 323, 324, 325, 326, 327		
\chi 275, 288, 412, 426		F
cs commands:		\fi 201, 230,
\cs if exist:NTF 11		231, 245, 318, 361, 362, 363, 376, 382

<code>\futurelet</code>	374	<code>\g__paresse_letter_epsilon_is_-</code>	
		<code>epsilon_bool</code>	114, 116, 143
		<code>\g__paresse_letter_phi_is_phi_-</code>	
		<code>bool</code>	128, 130, 163
		<code>\g__paresse_letter_pi_is_pi_bool</code>	
		71, 73, 151
		<code>\g__paresse_letter_rho_is_rho_-</code>	
		<code>bool</code>	85, 87, 155
		<code>\g__paresse_letter_sigma_is_-</code>	
		<code>sigma_bool</code>	99, 101, 159
		<code>\g__paresse_letter_t_is_theta_-</code>	
		<code>bool</code>	43, 45, 277, 415
		<code>\g__paresse_letter_theta_is_-</code>	
		<code>theta_bool</code>	57, 59, 147
		<code>__paresse_main_package_loaded:</code>	
		11, 169
		<code>\g__paresse_mood_is_wild_bool</code>	
		29, 31, 315, 451
		<code>\Phi</code>	298, 436
		<code>\phi</code>	164
		<code>phi</code> internal commands:	
		<code>__paresse_phi</code>	164, 165, 267, 404
		<code>\Pi</code>	295, 433
		<code>\pi</code>	152
		<code>pi</code> internal commands:	
		<code>__paresse_pi</code>	152, 153, 263, 400
		<code>\ProcessKeysOptions</code>	142
		<code>\ProvidesExplPackage</code>	2, 4, 6
		<code>\Psi</code>	305, 309, 444, 448
		<code>\psi</code>	276, 289, 413, 427
		R	
		<code>\relax</code>	235, 236, 331, 346, 349, 352
		<code>\RequirePackage</code>	8, 9, 10
		<code>\RequirePackageWithOptions</code>	
		172, 176, 220, 224
		<code>\rho</code>	156
		<code>rho</code> internal commands:	
		<code>__paresse_rho</code>	156, 157, 264, 401
		S	
		<code>\S</code>	234, 311, 365, 382, 386
		<code>\Sigma</code>	296, 434
		<code>\sigma</code>	160
		<code>sigma</code> internal commands:	
		<code>__paresse_sigma</code> ...	160, 161, 265, 402
		<code>\space</code>	332, 347, 348
		<code>\string</code> ..	240, 313, 335, 337, 339, 368, 369
		T	
		<code>\tau</code>	274, 282, 287, 411, 420, 425
		<code>T_EX</code> and <code>L^AT_EX 2_ε</code> commands:	
		<code>\@eha</code>	348
		<code>\@firstofone</code>	381
<code>\Gamma</code>	292, 430		
<code>\gamma</code>	252, 389		
<code>\gdef</code>	330, 343, 345		
<code>\global</code>	328, 329		
		I	
<code>\IeC</code>	343		
<code>\ifcat</code>	376		
<code>\ifcsname</code>	378		
<code>\ifluatex</code>	170		
<code>\ifnum</code>	241, 318, 346, 349, 352		
<code>\ifx</code>	196		
<code>\ifxetex</code>	174		
<code>\inputencodingname</code>	196		
<code>\iota</code>	257, 394		
		K	
<code>\kappa</code>	258, 395		
<code>keys</code> commands:			
<code>\keys_define:nn</code>	13		
		L	
<code>\Lambda</code>	294, 432		
<code>\lambda</code>	259, 396		
<code>\let</code>	233, 242, 244, 318, 328, 329		
		M	
<code>\makeatletter</code>	216		
<code>\makeatother</code>	226		
<code>\makeparesseletter</code>	235, 249, 365, 451		
<code>\makeparesseother</code>	236, 366, 368, 369		
<code>\mu</code>	260, 397		
		N	
<code>\newcommand</code>	181,		
	189, 235, 236, 237, 246, 365, 366, 371		
<code>\newenvironment</code>	247, 372		
<code>\nu</code>	261, 398		
		O	
<code>\Omega</code>	300, 438		
<code>\omega</code>	268, 405		
		P	
<code>\PackageError</code>	347		
<code>\PackageInfo</code>	171,		
	175, 179, 194, 199, 206, 209, 219, 223		
<code>\PackageWarning</code>	367		
<code>\PackageWarningNoLine</code>	183		
<code>paresse</code> internal commands:			
<code>\g__paresse_encoding_is_legacy_-</code>			
<code>bool</code>	15, 17, 270, 301, 407, 440		

<code>\@gobble</code>	194, 199, 209, 379	<code>\UTFviii@three@octets</code>	336, 355
<code>\@i@paresse</code>	374, 375	<code>\UTFviii@tmp</code>	343
<code>\@ifpackageloaded</code>	204, 214	<code>\UTFviii@two@octets</code>	335, 351
<code>\@ii@paresse</code>	377	<code>\the</code>	332
<code>\@paresse</code>	371, 374	<code>\Theta</code>	303, 304, 308, 442, 443, 447
<code>\@toks</code>	374, 376	<code>\theta</code>	148
<code>\count@</code>	232,	theta internal commands:	
241, 317, 318, 331, 332, 346, 349, 352		<code>__paresse_theta</code>	148, 149, 273, 279, 286, 410, 417, 424
<code>\default@encoding@message</code>	181, 214		
<code>\define@paresse@car</code>			
383, 386, 387, 388,			
389, 390, 391, 392, 393, 394, 395,			
396, 397, 398, 399, 400, 401, 402,			
403, 404, 405, 409, 410, 411, 412,			
413, 417, 420, 424, 425, 426, 427,			
428, 430, 431, 432, 433, 434, 435,			
436, 437, 438, 442, 443, 444, 447, 448			
<code>\find@encoding</code>	189, 214		
<code>\GA@ActiveLaParesse</code>	242, 244, 246, 247		
<code>\GA@ActiveLaParessexii</code>	238, 242		
<code>\GA@ActiveLaParessexiii</code>	239, 244		
<code>\GA@LaVieilleParesse</code>	233, 318		
<code>\GA@MakeParesseActive</code>	237		
<code>\GA@parse@UTFviii@a</code>			
328, 350, 353, 354, 357, 358, 359			
<code>\GA@parse@UTFviii@b</code>	329, 351, 355, 360		
<code>\GA@parse@XML@charref</code>	334, 345		
<code>\parse@UTFviii@a</code>	328		
<code>\parse@UTFviii@b</code>	329		
<code>\temp@a</code>	192, 197, 202		
<code>\utf@encoding</code>	191, 196		
<code>\UTFviii@four@octets</code>	338, 360		