

The xgreek package

Apostolos Syropoulos
Xanthi, Greece
asyropoulos@yahoo.com

2007/01/09

Abstract

The `xgreek` package provides rudimentary support for Greek language typesetting with $\text{Xe}\text{\LaTeX}$. In particular, it provides support for modern Greek (either monotonic or polytonic) and ancient Greek.

1 Introduction

The `xgreek` package provides rudimentary support for Greek language typesetting with $\text{Xe}\text{\LaTeX}$. Users will be able to typeset documents in either modern Greek (monotonic or polytonic) or ancient Greek by selecting the appropriate package option. The default ‘‘language’’ is monotonic modern greek.

2 The Source Code

We define the various strings that correspond to the standard \LaTeX captions. Since monotonic Greek is the default option, we define these strings accordingly. Options

```
1 \langle *xgreek \rangle
2 \def\prefacename{Πρόλογος}%
3 \def\refname{Αναφορές}%
4 \def\abstractname{Περίληψη}%
5 \def\bibname{Βιβλιογραφία}%
6 \def\chaptername{Κεφάλαιο}%
7 \def\appendixname{Παράρτημα}%
8 \def\contentsname{Περιεχόμενα}%
9 \def\listfigurename{Κατάλογος σχημάτων}%
10 \def\listtablename{Κατάλογος πινάκων}%
11 \def\indexname{Ευρετήριο}%
12 \def\figurename{Σχήμα}%
13 \def\tablename{Πίνακας}%
```

```

14 \def\partname{Μέρος}%
15 \def\enclname{Συνημμένα}%
16 \def\ccname{Κοινοποίηση}%
17 \def\headtoname{Προς}%
18 \def\pagename{Σελίδα}%
19 \def\seename{βλέπε}%
20 \def\alsoname{βλέπε επίσης}%
21 \def\proofname{Απόδειξη}%
22 \def\glossaryname{Γλωσσάρι}%

```

The macro `\dategreek` redefines the command `\today` to produce greek dates. The names of months are defined by the macro `\gr@month`.

```

23 \def\gr@month{%
24   \ifcase\month\or Ιανουαρίου\or Φεβρουαρίου\or Μαρτίου\or Απριλίου\or
25     Μαΐου\or Ιουνίου\or Ιουλίου\or Αυγούστου\or
26     Σεπτεμβρίου\or Οκτωβρίου\or Νοεμβρίου\or Δεκεμβρίου\fi}
27 \def\today{\number\day \space \gr@month\space \number\year}

```

Macro `\polytonicn@mes` is invoked when either polytonic Greek or ancient Greek is the main language of the document.

```

28 \def\polytonicn@mes{%
29   \def\refname{Αναφορές}%
30   \def\indexname{Εύρετήριο}%
31   \def\figurename{Σχήμα}%
32   \def\headtoname{Πρός}%
33   \def\alsoname{βλέπε επίσης}%
34   \def\proofname{Απόδειξη}%
35 }

```

When either polytonic Greek or ancient Greek is the main language of the document, then the macro `\gr@c@month` becomes active.

```

36 \def\gr@c@month{%
37   \ifcase\month\or Ιανουαρίου\or Φεβρουαρίου\or Μαρτίου\or Απριλίου\or
38     Μαΐου\or Ιουνίου\or Ιουλίου\or Αυγούστου\or Σεπτεμβρίου\or
39     Οκτωβρίου\or Νοεμβρίου\or Δεκεμβρίου\fi}

```

Now that we have defined the language dependant macros, we can safely define the various supported options.

```

40 \DeclareOption{monotonic}{%
41   \language\modernngreek
42 }
43 \DeclareOption{polytonic}{%
44   \language\modernngreek%
45   \polytonicn@mes%
46   \let\gr@month\gr@c@month%
47 }
48 \DeclareOption{ancient}{%
49   \language\ancientgreek%
50   \polytonicn@mes%
51   \let\gr@month\gr@c@month%
52 }

```

```

53 \ExecuteOptions{monotonic}
54 \frenchspacing
55 \ProcessOptions

```

Before we proceed let us define a few macros, which allow one to access characters that are not usually easily accessible such as the sampi or koppa symbols.

```

56 \def\anwtonos{\char"0374\relax}
57 \def\katwtonos{\char"0375\relax}
58 \def\koppa{\char"03DF\relax}
59 \def\stigma{\char"03DA\relax}
60 \def\sampi{\char"03E1\relax}
61 \def\Digamma{\char"03DC\relax}
62 \def\ddigamma{\char"03DD\relax}
63 \def\euro{\char"20AC\relax}
64 \def\permill{\char"2030\relax}

```

Now we are going to define the macros that typeset Greek numerals. The code is borrowed from the greek option for the babel package.

`\gr@ill@value` When the argument of `\greeknumeral` has a value outside of the acceptable bounds ($0 < x < 999999$) a warning will be issued (and nothing will be printed).

```

65 \def\gr@ill@value#1{%
66   \PackageWarning{xgreek}{Illegal value (#1) for greeknumeral}}

```

`\anw@true` When a large number with three *trailing* zero's is to be printed those
`\anw@false` zeros *and* the numeric mark need to be discarded. As each 'digit' is
`\anw@print` processed by a separate macro *and* because the processing needs to be
expandable we need some helper macros that help remember to *not* print
the numeric mark (`\anwtonos`).

The command `\anw@false` switches the printing of the numeric mark off by making `\anw@print` expand to nothing. The command `\anw@true` (re)enables the printing of the numeric mark. These macro's need to be robust in order to prevent improper expansion during writing to files or during `\uppercase`.

```

67 \DeclareRobustCommand\anw@false{%
68   \DeclareRobustCommand\anw@print{}}
69 \DeclareRobustCommand\anw@true{%
70   \DeclareRobustCommand\anw@print{\anwtonos}}
71 \anw@true

```

`\greeknumeral` The command `\greeknumeral` needs to be *fully* expandable in order to get the right information in auxiliary files. Therefore we use a big `\if`-construction to check the value of the argument and start the parsing at the right level.

```

72 \def\greeknumeral#1{%

```

If the value is negative or zero nothing is printed and a warning is issued.

```

73 \ifnum#1<\@ne\space\gr@ill@value{#1}%
74 \else
75 \ifnum#1<10\expandafter\gr@num@i\number#1%
76 \else
77 \ifnum#1<100\expandafter\gr@num@ii\number#1%
78 \else

```

We use the available shorthands for 1.000 (\@m) and 10.000 (\@M) to save a few tokens.

```

79 \ifnum#1<\@m\expandafter\gr@num@iii\number#1%
80 \else
81 \ifnum#1<\@M\expandafter\gr@num@iv\number#1%
82 \else
83 \ifnum#1<100000\expandafter\gr@num@v\number#1%
84 \else
85 \ifnum#1<1000000\expandafter\gr@num@vi\number#1%
86 \else

```

If the value is too large, nothing is printed and a warning is issued.

```

87 \space\gr@ill@value{#1}%
88 \fi
89 \fi
90 \fi
91 \fi
92 \fi
93 \fi
94 \fi
95 }

```

`\Greeknatural` The command `\Greeknatural` prints uppercase greek numerals. The parsing is performed by the macro `\greeknumeral`.

```

96 \def\Greeknatural#1{%
97 \expandafter\MakeUppercase\expandafter{\greeknumeral{#1}}

```

`\greek@alph` Here we redefine the macros `\@alph` and `\@Alph`. First we define some placeholders

```

98 \let\latin@alph\@alph
99 \let\latin@Alph\@Alph

```

Then we define the Greek versions; the additional `\expandafters` are needed in order to make sure the table of contents will be correct, e.g., when we have appendixes.

```

100 \def\greek@alph#1{\expandafter\greeknumeral\expandafter{\the#1}}
101 \def\greek@Alph#1{\expandafter\Greeknatural\expandafter{\the#1}}

```

By default use Greek enumerals instead of Latin numerals to enumerate items in an enumeration environment.

```

102 \let\@alph\greek@alph
103 \let\@Alph\greek@Alph

```

If for some reason, we need to have the Latin numerals, then we just have to invoke command `\nogreekalph`. And if we want to switch back, then we have to use the `\greekalph` command:

```
104 \def\nogreekalph{%
105   \let\@alph\latin@alph
106   \let\@Alph\latin@Alph}
107 \def\greekalph{%
108   \let\@alph\greek@alph
109   \let\@Alph\greek@Alph}
```

What is left now is the definition of a set of macros to produce the various digits.

`\gr@num@i` As there is no representation for 0 in this system the zeros are simply
`\gr@num@ii` discarded. When we have a large number with three *trailing* zero's also
`\gr@num@iii` the numeric mark is discarded. Therefore these macros need to pass the information to each other about the (non-)translation of a zero.

```
110 \def\gr@num@i#1{%
111   \ifcase#1\or \alpha\or \beta\or \gamma\or \delta\or \epsilon\or \stigma\or \zeta\or \eta\or \theta\fi
112   \ifnum#1=\z@\else\anw@true\fi\anw@print}
113 \def\gr@num@ii#1{%
114   \ifcase#1\or \iota\or \kappa\or \lambda\or \mu\or \nu\or \xi\or \omicron\or \pi\or \koppa\fi
115   \ifnum#1=\z@\else\anw@true\fi\gr@num@i}
116 \def\gr@num@iii#1{%
117   \ifcase#1\or \rho\or \sigma\or \tau\or \upsilon\or \phi\or \chi\or \psi\or \omega\or \sampi\fi
118   \ifnum#1=\z@\anw@false\else\anw@true\fi\gr@num@ii}
```

`\gr@num@iv` The first three 'digits' always have the numeric mark, except when one
`\gr@num@v` is discarded because it's value is zero.

```
\gr@num@vi 119 \def\gr@num@iv#1{%
120   \ifnum#1=\z@\else\katwtonos\fi
121   \ifcase#1\or \alpha\or \beta\or \gamma\or \delta\or \epsilon\or \stigma\or \zeta\or \eta\or \theta\fi
122   \gr@num@iii}
123 \def\gr@num@v#1{%
124   \ifnum#1=\z@\else\katwtonos\fi
125   \ifcase#1\or \iota\or \kappa\or \lambda\or \mu\or \nu\or \xi\or \omicron\or \pi\or \koppa\fi
126   \gr@num@iv}
127 \def\gr@num@vi#1{%
128   \katwtonos
129   \ifcase#1\or \rho\or \sigma\or \tau\or \upsilon\or \phi\or \chi\or \psi\or \omega\or \sampi\fi
130   \gr@num@v}
```

`\setlanguage` We provide the `\setlanguage` command which activates the hyphenation patterns of some other language. It is similar to babel's `\selectlanguage`, but we opted to use a new name to avoid any name conflicts.

```
131 \def\setlanguage#1{%
132   \expandafter\ifx\csname #1\endcsname\relax%
133   \typeout{^^J Error: No hyphenation pattern for language #1 loaded,}%
```

```

134 \typeout{ default hyphenation patterns are used.^^J}%
135 \language=0%
136 \else\language=\csname #1\endcsname\fi}

```

The macros `\grtoday` and `\Grtoday` produces the current date, only that the month and the day are shown as greek numerals instead of arabic as it is usually the case. In addition, the two commands differ in that the later produces the Greek numerals in uppercase.

```

137 \def\grtoday{%
138   \expandafter\greeknumeral\expandafter{\the\day}\space
139   \gr@c@month \space
140   \expandafter\greeknumeral\expandafter{\the\year}}
141 \def\Grtoday{%
142   \expandafter\Greeknatural\expandafter{\the\day}\space
143   \gr@c@month \space
144   \expandafter\Greeknatural\expandafter{\the\year}}
145 \</xgreek>

```