Babel support for the Greek language

Apostolos Syropoulos, Günter Milde

October 21, 2014

Babel-greek is a contributed package providing support for the Greek language and script via the Babel system. See the README file for an overwiew of the `babel-greek` package and links to requirements and related packages.

The file `babel-greek.dtx` [1] is the literate source for the Babel language definition file `greek.ldf`.

1 Requirements

Typesetting Greek with Babel requires (of course) the Babel core, support for Greek font encodings (greek-fontenc) and a text font supporting the Greek script.

The CB Greek fonts created by CLAUDIO BECCARI[2] are a complete set of 8-bit TeX fonts matching KNUTH's Computer Modern. The package `cbfonts-fd` sets them up as Greek substitute for the Computer Modern and Latin Modern font families. The `substitutefont` package can be used to set up Greek supplements for other font families (like Times or Palatino).

The Latin Modern OpenType font that is the default font for XeTeX/LuaTeX does not support the Greek script. The user needs to set up an alternative font like Linux Libertine or DejaVu with `fontspec`.

2 Usage

To activate Greek support with babel, specify the option greek, either as global option or as option to the `babel` package. Remember, that the *last* language option determines the document language, e.g.

```
\usepackage[greek,english]{babel}
```

activates support for Greek in an English document. The default is modern "monotonic" Greek, while

---

[1] The file described in this section has version number v1.9c and was last revised on 2014/10/21. The original author is Apostolos Syropoulos, code from `kdgreek.sty` by David Kastrup was used.

[2] Apostolos Syropoulos wishes to thank Claudio Beccari for his patience, collaboration, comments and suggestions.

```
\usepackage[english,greek]{babel}
\languageattribute{greek}{polutoniko}
```

sets the document language to modern Greek with "polytonic" spelling and

```
\usepackage[english,greek]{babel}
\languageattribute{greek}{ancient}
```

sets the document language to ancient Greek.

Both attributes may also be used as modifiers as in

```
\usepackage[greek.polutoniko,english]{babel}
```

and similarly

```
\usepackage[greek.ancient,english]{babel}
```

Version 1.9 fixes the activation of hyphenation patterns: By default, `babel-greek` activates the modern Greek monotonic hyphenation; the attribute/modifier `polutoniko` activates the modern Greek polytonic hyphenation and the attribute/modifier `ancient` activates the ancient Greek hyphenation. 8-bit TeX and XeTeX requires the hyphenation patterns to be pre-loaded in the format file. This is a limitation by TeX, common to all languages. The LuaTeX engine loads hyphenation patterns on demand.

`\selectlanguage`
`\foreignlanguage`   The Babel core provides two commands to switch the active language: The declaration `\selectlanguage{greek}` switches to the Greek language. The macro `\foreignlanguage{greek}{<some text>}` sets its second argument in the Greek language. This is intended for short text parts. For details see the Babel documentation.

For backwards compatibility, "polytonic" spelling can also be selected via the dummy language **polutonikogreek**. In this case, the language name **polutonikogreek** must also be used for language switches with `\selectlanguage` or `\foreignlanguage`. However, it is not possible to use both options, **greek** and **polutonikogreek** in one document.

2.1   Input of Greek text

There are several alternatives to write Greek text.

- With the packages `inputenc` and `greek-inputenc`, literal Greek characters can be input using the utf-8, iso-8859-7, or macgreek encoding.

  Literal input using the utf-8 encoding is also the standard input method if compiling with the XeTeX and LuaTeX engines.

- The Latin transliteration defined by the LGR font encoding is explained in the file `usage.pdf`.

- The package `greek-fontenc` defines *LaTeX internal character representation* (LICR) macros for Greek letters and text symbols. It is required by `babel-greek`, so the `\textAlpha` ... `\textomega` macros are a safe but cumbersome method to input Greek characters.

- The `alphabeta` package bundled with greek-fontenc makes the short macro names `\Alpha` ... `\omega` available in text mode, too.

The greek-fontenc bundle has details and examples for all these input methods.

2.2 Greek vs. Latin script

When switching the language to Greek, `babel-greek` ensures that the Greek script is supported. With 8-bit LaTeX, Greek text fonts use a non-standard font encoding[3] (LGR) that misses Latin letters. Latin characters in the source are instead interpreted as a transcription for Greek characters. The following macros allow the use of Greek vs. Latin script without changing the active language. The *TextCommand*[4] `\greekscript` switches to a font encoding supporting the Greek script. The declaration `\greektext` always switches the font encoding to LGR. Both declarations do not change the active language. `\latintext` (defined by the Babel core) can be used to switch back to an encoding supporting the Latin script.

`\greekscript`
`\greektext`

`\latintext`

`\ensuregreek`
`\textgreek`

`\textlatin`

The macros `\ensuregreek` and `\textgreek` take one argument which is typeset using a font encoding supporting the Greek script. While `\ensuregreek` only switches the font encoding if required, `\textgreek` always uses the LGR font encoding. The Babel core defines a corresponding `\textlatin` macro.

3 Greek numbering

The Greek alphabetical numbering system, like the Roman one, is still used in everyday life for short enumerations. Unfortunately most Greeks don't know how to write Greek numbers bigger than 20 or 30. Nevertheless, in official editions of the last century and beginning of this century this numbering system was also used for dates and numbers in the range of several thousands. Nowadays this numbering system is primary used by the Eastern Orthodox Church and by certain scholars. It is hence necessary to be able to typeset any Greek numeral up to 999 999. Here are the conventions:

- There is no Greek numeral for any number less than or equal to 0.

- Numbers from 1 to 9 are denoted by letters alpha, beta, gamma, delta, epsilon, stigma, zeta, eta, theta, followed by a mark similar to the mathematical symbol "prime". (Nowadays instead of letter stigma the digraph sigma tau is used for number 6. Mainly because the letter stigma is not always available, so people opt to write down the first two letters of its name

---

[3]LaTeX font encodings are described in the *LaTeX font guide* (fntguide.pdf).
[4]For a discussion of TextCommands, see the *LaTeX font guide* (fntguide.pdf), too.

as an alternative. In our implementation we produce the letter stigma, not the digraph sigma tau.)

- Decades from 10 to 90 are denoted by letters iota, kappa, lambda, mu, nu, xi, omikron, pi, qoppa, again followed by the numeric mark. The qoppa used for this purpose has a special zig-zag form, which doesn't resemble at all the original 'q'-like qoppa.

- Hundreds from 100 to 900 are denoted by letters rho, sigma, tau, upsilon, phi, chi, psi, omega, sampi, followed by the numeric mark.

- Any number between 1 and 999 is obtained by a group of letters denoting the hundreds decades and units, followed by a numeric mark.

- To denote thousands one uses the same method, but this time the mark is placed in front of the letter, and under the baseline (it is inverted by 180 degrees). When a group of letters denoting thousands is followed by a group of letters denoting a number under 1000, then both marks are used.

\greeknumeral   Using these conventions one obtains numbers up to 999 999. The command \greeknumeral makes it possible to typeset Greek numerals. There is also an \Greeknumeral   "uppercase" version of this macro: \Greeknumeral.

    Another system which was in wide use only in Athens, could express any positive number. This system is implemented in package athnum.

4   Implementation

The macro \LdfInit takes care of preventing that this file is loaded more than once, checking the category code of the @ sign, etc.

```
1 ⟨*code⟩
2 \LdfInit\CurrentOption{captions\CurrentOption}
```

When the option polutonikogreek was used, redefine \CurrentOption to prevent problems later on.

```
3 \gdef\CurrentOption{greek}
```

Hyphenation patterns: When this file is read as an option, i.e. by the \usepackage command, greek could be an 'unknown' language in which case we have to make it known. So we check for the existence of the three variants of the Greek language \l@greek, \l@monogreek, and l@ancientgreek to see whether we have to do something here.

```
4 \ifx\l@greek\@undefined
5   \@nopatterns{greek}
6   \adddialect\l@greek 0\fi
7 \ifx\l@monogreek\@undefined
8   \adddialect\l@monogreek 0\fi
9 \ifx\l@ancientgreek\@undefined
10   \adddialect\l@ancientgreek 0\fi
11 \newcount\bbl@monogreek \bbl@monogreek=\l@monogreek
```

```
12 \newcount\bbl@polygreek \bbl@polygreek=\l@greek
13 \newcount\bbl@ancientgreek  \bbl@ancientgreek=\l@ancientgreek
```

The `\extrasgreek` macro is called when switching the language to Greek. We collect code for polytonic and ancient greek in the macros `extraspolutonikogreek` and `extrasancientgreek` and add it later to `extrasgreek`.

```
14 \addto\extrasgreek{\let\l@greek\bbl@monogreek\relax}
15 \addto\extraspolutonikogreek{\l@greek=\bbl@polygreek}
16 \addto\extrasancientgreek{\l@greek=\bbl@ancientgreek}
```

Now we declare the `polutoniko` language attribute.

```
17 \bbl@declare@ttribute{greek}{polutoniko}{%
```

This code adds the expansion of `\extraspolutonikogreek` to `\extrasgreek` and changes the definition of `\today` for Greek to produce polytonic month names (by re-defining `\gr@month`).

```
18   \expandafter\addto\expandafter\extrasgreek
19     \expandafter{\extraspolutonikogreek}%
20   \let\captionsgreek\captionspolutonikogreek
21   \let\gr@month\gr@c@month
```

We need to take some extra precautions in order not to break older documents which still use the old `polutonikogreek` pseuso-language option: language switching commands like `\selectlanguage{polutonikogreek}`, call the `extraspolutonikogreek` macro, set the language to `\l@polutonikogreek` and use `\datepolutonikogreek` for `\today`. We define aliases:

```
22   \let\l@polutonikogreek\l@greek
23   \let\datepolutonikogreek\dategreek
24   \let\extraspolutonikogreek\extrasgreek
25   \let\noextraspolutonikogreek\noextrasgreek
26 }
```

The `ancient` language attribute is used for classical Greek.

```
27 \bbl@declare@ttribute{greek}{ancient}{%
```

This attribute adds the expansion of `\extraspolutonikogreek` and `\extrasancientgreek` to `\extrasgreek` to set up support for multi-accented characters and ancient hyphenation patterns. Auto-strings (captions) are specific to ancient Greek while `\today` uses modern polytonic month names (as there existed incompatible sets of month names and no common calendar in ancient Greece.

```
28   \expandafter\addto\expandafter\extrasgreek
29     \expandafter{\extraspolutonikogreek}% multi-accented letters
30   \expandafter\addto\expandafter\extrasgreek
31     \expandafter{\extrasancientgreek}%
32   \let\captionsgreek\captionsancientgreek
33   \let\gr@month\gr@c@month % (modern) polytonic month names
34 }
```

5   Font setup

Typesetting Greek texts requires a font with Greek characters:

For 8-bit LaTeX, this package uses fonts with the LGR font encoding. With the XeTeX or LuaTeX engines and Unicode fonts, the user must ensure that the selected font contains the required glyphs. LGR-encoded fonts can also be used alongside Unicode fonts with XeTeX/LuaTeX to enable the input of Greek letters via the Latin transcription provided by this font encoding.

`\greekfontencoding`   We test for available font encodings and set `\greekfontencoding` and the internal macro `\bbl@greek@fontencdef` with the name of the font definition file with Greek LICR macros. The fallback is LGR.

```
35 \@ifl@aded{def}{eu1enc}{
36   \providecommand*{\greekfontencoding}{EU1}
37   \providecommand*{\bbl@greek@fontencdef}{greek-euenc}
38   \renewcommand*{\LastDeclaredEncoding}{EU1}
39 }{}
40 \@ifl@aded{def}{eu2enc}{
41   \providecommand*{\greekfontencoding}{EU2}
42   \providecommand*{\bbl@greek@fontencdef}{greek-euenc}
43   \renewcommand{\LastDeclaredEncoding}{EU2}
44 }{}
45 \providecommand*{\greekfontencoding}{LGR}
46 \providecommand*{\bbl@greek@fontencdef}{lgrenc}
```

Ensure that the Greek LICR macros are available:

```
47 \@ifl@aded{def}{\bbl@greek@fontencdef}{}{
48   \InputIfFileExists{\bbl@greek@fontencdef .def}{}{
49     \errhelp{I can't find the \bbl@greek@fontencdef .def file
50       for the Greek fonts}%
51     \errmessage{Font support for the Greek script missing.^^J
52       babel-greek can't typeset Greek.^^J
53       Install the "greek-fontenc" package^^J
54       or use XeTeX/LuaTeX with polyglossia.}%
55     \@@end
56   }
57 }
```

`\greekscript`   The TextCommand `\greekscript` is a declaration that switches the font encoding to `\greekfontencoding` if the current font encoding does not provide a (typically empty) local variant.

```
58 \ProvideTextCommandDefault{\greekscript}{%
59   \fontencoding{\greekfontencoding}\selectfont
60     \def\encodingdefault{\greekfontencoding}}
```

`\ensuregreek`   The TextCommand `\ensuregreek` sets it argument in `\greekfontencoding` if the current font encoding does not provide a (typically empty) local variant.

```
61 \ProvideTextCommandDefault{\ensuregreek}[1]{%
62   \leavevmode{\greekscript #1}}
```

`\latinencoding`   The babel core defines the macro `\latinencoding` and sets it to either OT1 or (if available) T1. We extend the choice with the Unicode encodings EU1 and EU2

6

so that `\latintext` under XeTeX/LuaTeX does not prevent the use of Unicode fonts.

```
63 \AtBeginDocument{%
64     \@ifl@aded{def}{eu1enc}{\xdef\latinencoding{EU1}}{}% XeTeX
65     \@ifl@aded{def}{eu2enc}{\xdef\latinencoding{EU2}}{}% LuaTeX
66 }
```

5.1   Fixes for the LGR font encoding

```
67 \@ifl@aded{def}{lgrenc}{%
```

We redefine a few commands in the LGR encoding to work around problems because LGR is no *standard text encoding* but has Greek letters in the places of Latin ones.

```
68     \ProvideTextCommand{\textcopyright}{LGR}{\textlatin{\textcopyright}}
69     \ProvideTextCommand{\textregistered}{LGR}{\textlatin{\textregistered}}
70     \ProvideTextCommand{\texttrademark}{LGR}{\textlatin{\texttrademark}}
71     \let\ltx@amp\&
72     \DeclareTextCommandDefault{\&}{\ltx@amp}
73     \DeclareTextCommand{\&}{LGR}{\textlatin{\ltx@amp}}
74     \ProvideTextCommand{\SS}{LGR}{\textlatin{\SS}}
```

`\greek@roman`
`\greek@Roman` To prevent roman numerals being typeset in greek letters we need to adopt the internal LaTeX commands `\@roman` and `\@Roman`. *Up to version 1.5, the definition caused errors where* **roman** *ends up in a situation where the argument needs to be expanded (e.g. Roman page numbers in the ToC).*

```
75     \def\@roman#1{\expandafter\textlatin\expandafter{\romannumeral#1}}
76     \def\@Roman#1{\expandafter\textlatin\expandafter{%
77                 \expandafter\@slowromancap\romannumeral#1@}}
```

Now we define two commands that switch to the LGR font encoding. The Babel core defines corresponding commands to switch to a Latin font encoding.

`\greektext` The declaration `\greektext` switches to LGR. For shorter pieces of text the `\textgreek` macro should be used.

```
78     \DeclareRobustCommand{\greektext}{%
79       \fontencoding{LGR}\selectfont
80       \def\encodingdefault{LGR}}
```

`\textgreek` This command takes an argument which is then typeset using the LGR font encoding. In order to avoid many encoding switches it operates in a local scope.

```
81     \DeclareRobustCommand{\textgreek}[1]{\leavevmode{\greektext #1}}
```

`\textol` The CB Greek fonts contain an outline family. In order to make it available, we define the command `\textol`. (This font-specific macro does not fit in a language defintion file is and only kept for backwards compatibility.)

```
82     \def\outlfamily{\usefont{LGR}{cmro}{m}{n}}
83     \DeclareTextFontCommand{\textol}{\outlfamily}
84 }{} % End of LGR fixes.
```

6 Definitions for the Greek language

The next step consists in defining commands to switch to (and from) the Greek language.

\greekhyphenmins This macro is used to store the correct values of the hyphenation parameters \lefthyphenmin and \righthyphenmin.

```
85 % Yannis Haralambous has suggested this value
86 \providehyphenmins{\CurrentOption}{\@ne\@ne}
```

\captionsgreek The macro \captionsgreek defines all strings used in the four standard document classes provided with LaTeX.

```
87 \addto\captionsgreek{%
88   \def\prefacename{\textPi\textrho\acctonos\textomicron\textlambda
89     \textomicron\textgamma\textomicron\textfinalsigma}%
90   \def\refname{\textAlpha\textnu\textalpha
91     \textphi\textomicron\textrho\acctonos\textepsilon\textfinalsigma}%
92   \def\abstractname{\textPi\textepsilon\textrho\acctonos\textiota
93     \textlambda\texteta\textpsi\texteta}%
94   \def\bibname{\textBeta\textiota\textbeta\textlambda\textiota\textomicron
95     \textgamma\textrho\textalpha\textphi\acctonos\textiota\textalpha}%
96   \def\chaptername{\textKappa\textepsilon\textphi\acctonos\textalpha
97     \textlambda\textalpha\textiota\textomicron}%
98   \def\appendixname{\textPi\textalpha\textrho\acctonos\textalpha\textrho
99     \texttau\texteta\textmu\textalpha}%
100  \def\contentsname{\textPi\textepsilon\textrho\textiota
101    \textepsilon\textchi\acctonos\textomicron\textmu\textepsilon
102    \textnu\textalpha}%
103  \def\listfigurename{\textKappa\textalpha\texttau\acctonos\textalpha
104    \textlambda\textomicron\textgamma\textomicron\textfinalsigma{}
105    \textSigma\textchi\texteta\textmu\acctonos\textalpha\texttau
106    \textomega\textnu}%
107  \def\listtablename{\textKappa\textalpha\texttau\acctonos\textalpha
108    \textlambda\textomicron\textgamma\textomicron\textfinalsigma{}
109    \textPi\textiota\textnu\acctonos\textalpha\textkappa\textomega\textnu}%
110  \def\indexname{\textEpsilon\textupsilon\textrho\textepsilon
111    \texttau\acctonos\texteta\textrho\textiota\textomicron}%
112  \def\figurename{\textSigma\textchi\acctonos\texteta\textmu\textalpha}%
113  \def\tablename{\textPi\acctonos\textiota\textnu\textalpha
114    \textkappa\textalpha\textfinalsigma}%
115  \def\partname{\textMu\acctonos\textepsilon\textrho\textomicron
116    \textfinalsigma}%
117  \def\enclname{\textSigma\textupsilon\textnu\texteta\textmu
118    \textmu\acctonos\textepsilon\textnu\textalpha}%
119  \def\ccname{\textKappa\textomicron\textiota\textnu\textomicron
120    \textpi\textomicron\acctonos\textiota\texteta\textsigma\texteta}%
121  \def\headtoname{\textPi\textrho\textomicron\textfinalsigma}%
122  \def\pagename{\textSigma\textepsilon\textlambda\acctonos\textiota
123    \textdelta\textalpha}%
```

8

```
124  \def\seename{\textbeta\textlambda\acctonos\textepsilon\textpi\textepsilon}%
125  \def\alsoname{\textbeta\textlambda\acctonos\textepsilon\textpi\textepsilon{}
126    \textepsilon\textpi\acctonos\textiota\textsigma\texteta\textfinalsigma}%
127  \def\proofname{\textAlpha\textpi\acctonos\textomicron
128    \textdelta\textepsilon\textiota\textxi\texteta}%
129  \def\glossaryname{\textGamma\textlambda\textomega\textsigma
130    \textsigma\acctonos\textalpha\textrho\textiota}%
131  }
```

For texts written in the $\pi o\lambda v\tau ov\kappa\acute{o}$ (polytonic greek) the translations are the same as above, but some words are spelled differently. For now we just add extra definitions to \captionsgreek in order to override the earlier definitions.

```
132  \let\captionspolutonikogreek\captionsgreek
133  \addto\captionspolutonikogreek{%
134    \def\refname{\accpsili\textAlpha\textnu\textalpha
135      \textphi\textomicron\textrho\accvaria\textepsilon\textfinalsigma}%
136    \def\indexname{\textEpsilon\accdasia\textupsilon\textrho\textepsilon
137      \texttau\acctonos\texteta\textrho\textiota\textomicron}%
138    \def\figurename{\textSigma\textchi\accperispomeni\texteta\textmu\textalpha}%
139    \def\headtoname{\textPi\textrho\accvaria\textomicron\textfinalsigma}%
140    \def\alsoname{\textbeta\textlambda\acctonos\textepsilon\textpi\textepsilon{}
141      \accpsili\textepsilon\textpi\acctonos\textiota\textsigma\texteta\textfinalsigma}%
142    \def\proofname{\accpsili\textAlpha\textpi\acctonos\textomicron
143      \textdelta\textepsilon\textiota\textxi\texteta}%
144  }
```

For texts written in ancient Greek, we took the translations from Aposto-los Syropoulos' xgreek package. For now we just add extra definitions to \captionsgreek in order to override the earlier definitions.

```
145  \let\captionsancientgreek\captionsgreek
146  \addto\captionsancientgreek{%
147    \def\prefacename{\textPi\textrho\textomicron\textomicron\acctonos\textiota\textmu
148      \textiota\textomicron\textnu}%
149    \def\refname{\accpsili\textAlpha\textnu\textalpha\textphi\textomicron\textrho
150      \textalpha\accvaria\textiota}%
151    \def\abstractname{\textPi\textepsilon\textrho\acctonos\textiota\textlambda\texteta
152      \textpsi\textiota\textvarsigma}%
153    \def\bibname{\textBeta\textiota\textbeta\textlambda\textiota\textomicron
154      \textgamma\textrho\textalpha\textphi\acctonos\textiota\textalpha}%
155    \def\chaptername{\textKappa\textepsilon\textphi\acctonos\textalpha\textlambda
156      \textalpha\textiota\textomicron\textnu}%
157    \def\appendixname{\textPi\textalpha\textrho\acctonos\textalpha\textrho\texttau
158      \texteta\textmu\textalpha}%
159    \def\contentsname{\textPi\textepsilon\textrho\textiota\textepsilon\textchi
160      \acctonos\textomicron\textmu\textepsilon\textnu\textalpha}%
161    \def\listfigurename{\textKappa\textalpha\texttau\acctonos\textalpha\textlambda
162      \textomicron\textgamma\textomicron\textvarsigma{}
163      \textsigma\textchi\texteta\textmu\acctonos\textalpha\texttau\textomega\textnu}%
164    \def\listtablename{\textKappa\textalpha\texttau\acctonos\textalpha\textlambda
```

```
165      \textomicron\textgamma\textomicron\textvarsigma{}
166      \textpi\textiota\textnu\acctonos\textalpha\textkappa\textomega\textnu}%
167    \def\indexname{\textEpsilon\accdasia\textupsilon\textrho\textepsilon\texttau
168      \acctonos\texteta\textrho\textiota\textomicron\textnu}%
169    \def\figurename{\textSigma\textchi\accperispomeni\texteta\textmu\textalpha}%
170    \def\tablename{\textPi\acctonos\textiota\textnu\textalpha\textxi}%
171    \def\partname{\textMu\acctonos\textepsilon\textrho\textomicron\textvarsigma}%
172    \def\enclname{\textSigma\textupsilon\textnu\texteta\textmu\textmu
173      \acctonos\textepsilon\textnu\textomega\textvarsigma}%
174    \def\ccname{\textKappa\textomicron\textiota\textnu\textomicron\textpi
175      \textomicron\acctonos\textiota\texteta\textsigma\textiota\textvarsigma}%
176    \def\headtoname{\textPi\textrho\accvaria\textomicron\textvarsigma}%
177    \def\pagename{\textSigma\textepsilon\textlambda\accvaria\textiota\textvarsigma}%
178    \def\seename{\accdasiavaria\textomicron\textrho\textalpha}%
179    \def\alsoname{\accdasiavaria\textomicron\textrho\textalpha{} \accdasia\textomega\textsigma
180      \textalpha\acctonos\textupsilon\texttau\textomega\textvarsigma}%
181    \def\proofname{\accpsili\textAlpha\textpi\acctonos\textomicron\textdelta\textepsilon
182      \textiota\textxi\textiota\textvarsigma}%
183    \def\glossaryname{\textGamma\textlambda\textomega\textsigma\textsigma
184      \acctonos\textalpha\textrho\textiota\textomicron\textnu}%
185  }
```

```
186 \def\gr@month{%
187   \ifcase\month\or
188     \textIota\textalpha\textnu\textomicron\textupsilon\textalpha
189       \textrho\acctonos\textiota\textomicron\textupsilon \or
190     \textPhi\textepsilon\textbeta\textrho\textomicron\textupsilon
191       \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
192     \textMu\textalpha\textrho\texttau\acctonos\textiota\textomicron\textupsilon
193     \or \textAlpha\textpi\textrho\textiota\textlambda\acctonos\textiota
194       \textomicron\textupsilon \or
195     \textMu\textalpha\'"\textiota\textomicron\textupsilon \or
196     \textIota\textomicron\textupsilon\textnu\acctonos\textiota
197       \textomicron\textupsilon \or
198     \textIota\textomicron\textupsilon\textlambda\acctonos\textiota
199       \textomicron\textupsilon \or
200     \textAlpha\textupsilon\textgamma\textomicron\acctonos\textupsilon
201       \textsigma\texttau\textomicron\textupsilon \or
202     \textSigma\textepsilon\textpi\texttau\textepsilon\textmu
203       \textbeta\textrho\acctonos\textiota\textomicron\textupsilon \or
204     \textOmicron\textkappa\texttau\textomega\textbeta
205       \textrho\acctonos\textiota\textomicron\textupsilon \or
206     \textNu\textomicron\textepsilon\textmu\textbeta
207       \textrho\acctonos\textiota\textomicron\textupsilon \or
208     \textDelta\textepsilon\textkappa\textepsilon\textmu\textbeta
209       \textrho\acctonos\textiota\textomicron\textupsilon
210   \fi
```

```
211 }
212 \def\dategreek{%
213   \def\today{\number\day \space \gr@month\space \number\year}}
```

```
214 \def\gr@c@month{%
215   \ifcase\month\or
216     \accpsili\textIota\textalpha\textnu\textomicron\textupsilon\textalpha
217       \textrho\acctonos\textiota\textomicron\textupsilon \or
218     \textPhi\textepsilon\textbeta\textrho\textomicron\textupsilon
219       \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
220     \textMu\textalpha\textrho\texttau\acctonos\textiota\textomicron
221       \textupsilon \or
222     \accpsili\textAlpha\textpi\textrho\textiota\textlambda
223       \acctonos\textiota\textomicron\textupsilon \or
224     \textMu\textalpha\"'\textiota\textomicron\textupsilon \or
225     \accpsili\textIota\textomicron\textupsilon\textnu
226       \acctonos\textiota\textomicron\textupsilon \or
227     \accpsili\textIota\textomicron\textupsilon\textlambda
228       \acctonos\textiota\textomicron\textupsilon \or
229     \textAlpha\accpsili\textupsilon\textgamma\textomicron\acctonos
230       \textupsilon\textsigma\texttau\textomicron\textupsilon \or
231     \textSigma\textepsilon\textpi\texttau\textepsilon\textmu\textbeta
232       \textrho\acctonos\textiota\textomicron\textupsilon \or
233     \accpsili\textOmicron\textkappa\texttau\textomega\textbeta
234       \textrho\acctonos\textiota\textomicron\textupsilon \or
235     \textNu\textomicron\textepsilon\textmu\textbeta
236       \textrho\acctonos\textiota\textomicron\textupsilon \or
237     \textDelta\textepsilon\textkappa\textepsilon\textmu
238       \textbeta\textrho\acctonos\textiota\textomicron\textupsilon
239   \fi
240 }
```

\Grtoday The macro \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.

```
241 \def\Grtoday{%
242   \expandafter\Greeknumeral\expandafter{\the\day}\space
243   \gr@c@month \space
244   \expandafter\Greeknumeral\expandafter{\the\year}}
```

\extrasgreek   The macro \extrasgreek will perform all the extra definitions needed for the
\noextrasgreek Greek language. The macro \noextrasgreek is used to cancel the actions of
\extrasgreek. For the moment these macros switch the fontencoding (with 8-bit
TeX) and the definition of the internal macros \@alph and \@Alph because in
Greek we do use the Greek numerals.

```
245 \addto\extrasgreek{\greekscript}
246 \addto\noextrasgreek{\latintext}
247 % \addto\noextrasgreek{\RestoreFontEncoding}
```

11

\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).

```
248 \def\gr@ill@value#1{%
249   \PackageWarningNoLine{babel}{Illegal value (#1) for greeknumeral}}
```

\anw@true   When a a large number with three *trailing* zero's is to be printed those zeros *and*
\anw@false the numeric mark need to be discarded. As each 'digit' is processed by a separate
\anw@print 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 marc. These macro's need to be robust in order to prevent
improper expansion during writing to files or during \uppercase.

```
250 \DeclareRobustCommand\anw@false{%
251   \DeclareRobustCommand\anw@print{}}
252 \DeclareRobustCommand\anw@true{%
253   \DeclareRobustCommand\anw@print{\textdexiakeraia}} % \anwtonos
254 \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.

```
255 \def\greeknumeral#1{%
```

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

```
256   \ifnum#1<\@ne\space\gr@ill@value{#1}%
257   \else
258     \ifnum#1<10\expandafter\gr@num@i\number#1%
259     \else
260       \ifnum#1<100\expandafter\gr@num@ii\number#1%
261       \else
```

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

```
262         \ifnum#1<\@m\expandafter\gr@num@iii\number#1%
263         \else
264           \ifnum#1<\@M\expandafter\gr@num@iv\number#1%
265           \else
266             \ifnum#1<100000\expandafter\gr@num@v\number#1%
267             \else
268               \ifnum#1<1000000\expandafter\gr@num@vi\number#1%
269               \else
```

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

```
270                 \space\gr@ill@value{#1}%
271               \fi
272             \fi
273           \fi
274         \fi
```

```
275        \fi
276      \fi
277    \fi
278 }
```

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

```
279 \def\Greeknumeral#1{%
280    \expandafter\MakeUppercase\expandafter{\greeknumeral{#1}}}
```

\greek@alph
\greek@Alph   In the previous release of this language definition the commands `\greek@aplh` and `\greek@Alph` were kept just for reasons of compatibility. Here again they become meaningful macros. They are definited in a way that even page numbering with greek numerals is possible. Since the macros `\@alph` and `\@Alph` will lose their original meaning while the Greek option is active, we must save their original value. macros `\@alph`

```
281 \let\latin@alph\@alph
282 \let\latin@Alph\@Alph
```

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

```
283 \def\greek@alph#1{\expandafter\greeknumeral\expandafter{\the#1}}
284 \def\greek@Alph#1{\expandafter\Greeknumeral\expandafter{\the#1}}
```

Now we can set up the switching.

```
285 \addto\extrasgreek{%
286    \let\@alph\greek@alph
287    \let\@Alph\greek@Alph}
288 \addto\noextrasgreek{%
289    \let\@alph\latin@alph
290    \let\@Alph\latin@Alph}
```

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

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

```
291 \def\gr@num@i#1{%
292    \ifcase#1\or \textalpha\or \textbeta\or \textgamma\or \textdelta\or
293    \textepsilon\or \textstigma\or \textzeta\or \texteta\or \texttheta\fi
294    \ifnum#1=\z@\else\anw@true\fi\anw@print}
295 \def\gr@num@ii#1{%
296    \ifcase#1\or \textiota\or \textkappa\or \textlambda\or \textmu\or
297    \textnu\or \textxi\or \textomicron\or \textpi\or \textqoppa\fi
298    \ifnum#1=\z@\else\anw@true\fi\gr@num@i}
299 \def\gr@num@iii#1{%
```

```
300    \ifcase#1\or \textrho\or \textsigma\or \texttau\or \textupsilon\or
301    \textphi\or \textchi\or \textpsi\or \textomega\or \textsampi\fi
302    \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 is discarded
\gr@num@v   because it's value is zero.
\gr@num@vi

```
303 \def\gr@num@iv#1{%
304    \ifnum#1=\z@\else\textaristerikeraia\fi
305    \ifcase#1\or \textalpha\or \textbeta\or \textgamma\or \textdelta\or
306    \textepsilon\or \stigma\or \textzeta\or \texteta\or \texttheta\fi
307    \gr@num@iii}
308 \def\gr@num@v#1{%
309    \ifnum#1=\z@\else\textaristerikeraia\fi
310    \ifcase#1\or \textiota\or \textkappa\or \textlambda\or \textmu\or
311    \textnu\or \textxi\or \textomicron\or \textpi\or \textqoppa\fi
312    \gr@num@iv}
313 \def\gr@num@vi#1{%
314    \textaristerikeraia
315    \ifcase#1\or \textrho\or \textsigma\or \texttau\or \textupsilon\or
316    \textphi\or \textchi\or \textpsi\or \textomega\or \textsampi\fi
317    \gr@num@v}
```

7   Character codes

Greek letters drop diacritics (eccept dialytika and sub-iota) in UPPERCASE.
This is not cared for by the Unicode standard. The file `xgreek-fixes.def` from
`polyglossia` contains the required `\lccode` and `\uccode` corrections from the
`xgreek` package by Apostolos Syropoulos.

```
318 \@ifl@aded{def}{greek-euenc}{
319    \InputIfFileExists{xgreek-fixes.def}{}{\PackageWarningNoLine{babel}{%
320      Can't find the file "xgreek-fixes.def".\MessageBreak
321      For correct up/down-casing of Greek text,\MessageBreak
322      install the "polyglossia" package}%
323    }
324 }{}
```

The LGR encoded fonts that are used to typeset Greek with 8-bit TeX define
ligatures for characters with diacritics. In order for this to work, some characters
need to be considered as letters. These characters are <, >, ~, ', ', " and |.
Therefore, their `\lccode` is changed when Greek is in effect. In order to let
`\uppercase` give correct results, the `\uccode` of these characters is set to a non-
existing character to make them disappear. Of course not all characters are needed
when typesetting "modern" μονοτονικό. In that case we only need the ' and "
symbols which are treated in the proper way.

\greek@tilde The Greek script uses a number of characters with more than one accent. In
LGR encoded fonts combined diacritics can be obtained using Knuth's ligature
mechanism (see usage.pdf). Characters we need to have ligatures with are the

tilde, the acute and grave accent characters, the rough and smooth breathings, the subscript, and the double quote character. In text input the ~ is normally used to produce an unbreakable space.

```
325 \@ifl@aded{def}{lgrenc}{%
326    \begingroup
327      \@ifundefined{active@char\string!}{}{\catcode`!=12\relax}
328      \catcode`\~=12%
329      \lccode`\!=`\~%
330      \lowercase{\def\x{\endgroup
331          \def\greek@tilde{!}}}\x}
```

In order to get correct hyphenation we need to set the lower case code of a number of characters. The 'v' character has a special usage for the `cb` fonts: in fact this ligature mechanism detects the end of a word and assures that a final sigma is typeset with the proper sign wich is different from that of an initial or medial sigma; the 'v 'after an *isolated* sigma fools the ligature mechanism in order to typeset σ in place of ς. Because of this we make sure its lowercase code is not changed. For "modern" greek we have to deal only with ' and " and so things are easy. In order to process the suitable characters and in such a way that hyphenation patterns work also with precomposed characters, it is necessary to declare the lc code for all characters, that can be part of a word. We do this in `\extrasgreek` because this is a feature of the LGR font encoding (which is the same in all language variants). This means that multi-accented characters are regarded parts of a word (and not non-word characters) also in monotonic spelling.

```
332    \addto\extrasgreek{%
333      \babel@savevariable{\lccode`v}\lccode`v=`v%
334      \babel@savevariable{\lccode`\'}\lccode`\'=`\'%
335      \babel@savevariable{\lccode`\"}\lccode`\"=`\"%
336      % ``high bit characters'': set in a loop and correct exceptions
337      \@tempcnta=128%
338      \@whilenum\@tempcnta<253\do{%
339        \babel@savevariable{\lccode\@tempcnta}\lccode\@tempcnta=\@tempcnta
340        \advance\@tempcnta\@ne
341        }%
342      % Fix non-word characters:
343      \lccode151=0%
344      \lccode155=0%
345      \lccode159=0%
346      \lccode199=0%
347      % Fix capital letters:
348      \lccode195=147% GREEK LETTER DIGAMMA
349      \lccode219=240% GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
350      \lccode223=244% GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
351      }
352
353    \addto\extraspolutonikogreek{%
354      % \l@greek=\bbl@polygreek
355      \babel@savevariable{\lccode`\<}\lccode`\<=`\<%
```

```
356     \babel@savevariable{\lccode'\>}\lccode'\>='\>%
357     \babel@savevariable{\lccode'\~}\lccode'\~='\~%
358     \babel@savevariable{\lccode'\|}\lccode'\|='\|%
359     \babel@savevariable{\lccode'\'}\lccode'\'='\'%
360     }
```

And in order to get rid of all accents and breathings when a string is \uppercased
we also change a number of uppercase codes.

```
361   \addto\extrasgreek{%
362     \babel@savevariable{\uccode'\"}\uccode'\"='\"%
363     \babel@savevariable{\uccode'\'}\uccode'\'=159% 159 == ^^9f
364     }
365   \addto\extraspolutonikogreek{%
366     \babel@savevariable{\uccode'\~}\uccode'\~=159%
367     \babel@savevariable{\uccode'\>}\uccode'\>=159%
368     \babel@savevariable{\uccode'\<}\uccode'\<=159%
369     \babel@savevariable{\uccode'\|}\uccode'\|='\|%
370     \babel@savevariable{\uccode'\'}\uccode'\'=159%
371     }
```

For this to work we make the character ^^9f a shorthand that expands to nothing.
In order for this to work we need to make a character look like ^^9f in TEX's eyes.
The trick is to have another character and assign it a different lowercase code.
The execute the macros needed in a \lowercase environment. Usually the tilde
~ character is used for such purposes. Before we do this we save it's original
lowercase code to restore it once we're done.

```
372   \@tempcnta=\lccode'\~
373   \lccode'\~=159
374   \lowercase{%
375     \initiate@active@char{~}%
376     \declare@shorthand{greek}{~}{}}
377   \lccode'\~=\@tempcnta
```

Add composite commands, so that the dialytika is kept or put on the following
character of a diphthong with \MakeUppercase (see lgrdef.enc from the the greek-
fontenc package for details).

```
378   \DeclareTextCompositeCommand{\"}{LGR}{^^9f}{\accdialytika}
379   \DeclareTextCompositeCommand{\'}{LGR}{^^9f}{\LGR@hiatus}
380   \DeclareTextCompositeCommand{\'}{LGR}{^^9f}{\LGR@hiatus}
381   \DeclareTextCompositeCommand{\<}{LGR}{^^9f}{\LGR@hiatus}
382   \DeclareTextCompositeCommand{\>}{LGR}{^^9f}{\LGR@hiatus}
```

We can also make the tilde character itself expand to a tilde with category
code 12 to make the typing of texts easier.

```
383   \addto\extraspolutonikogreek{\languageshorthands{greek}}%
384   \declare@shorthand{greek}{~}{\greek@tilde}
385 }{} % End of LGR-specific code.
```

8   symbol name aliases

For backwards compatibility, we keep aliases for a few symbols.

```
386 \providecommand*{\anwtonos}{\textdexiakeraia}
387 \providecommand*{\katwtonos}{\textaristerikeraia}
388 \providecommand*{\qoppa}{\textqoppa}
389 \providecommand*{\stigma}{\textstigma}
390 \providecommand*{\sampi}{\textsampi}
391 \providecommand*{\Digamma}{\textDigamma}
392 \providecommand*{\ddigamma}{\textdigamma}
393 \providecommand*{\vardigamma}{\textvardigamma}
394 \providecommand*{\euro}{\texteuro}
395 \providecommand*{\permill}{\textperthousand}
```

The macro `\ldf@finish` takes care of looking for a configuration file, setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value.

```
396 \ldf@finish{\CurrentOption}
397 ⟨/code⟩
```