

# Babel support for the Greek language

Apostolos Syropoulos, Günter Milde

March 4, 2023

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

The file `babel-greek.dtx`<sup>1</sup> is the literate source for the Babel language definition file `greek.ldf`.

## Contents

<b>1</b>	<b>Requirements</b>	<b>2</b>
<b>2</b>	<b>Usage</b>	<b>2</b>
2.1	Language attributes	3
2.2	Modifiers	3
2.3	Language hooks	3
2.4	Input of Greek text	4
2.5	Greek vs. Latin script	4
2.6	Greek numbering	5
<b>3</b>	<b>Implementation</b>	<b>6</b>
3.1	Hyphenation patterns	6
3.2	Language variants	7
3.2.1	<code>polutoniko</code>	7
3.2.2	<code>polytonic</code>	7
3.2.3	<code>ancient</code>	8
3.3	Font setup	8
3.3.1	Greek font encoding	8
3.3.2	Ensure loading of Greek font encoding definitions.	9
3.3.3	Font encoding switches	9
3.3.4	Additional commands for the LGR font encoding	10
3.3.5	LGR workarounds	12
3.4	Definitions for the Greek language	13
3.4.1	Auto-strings for Greek	13

---

<sup>1</sup>The file described in this section has version number 1.12 and was last revised on 2023/03/04. The original author is Apostolos Syropoulos, code from `kdgreeek.sty` by David Kastrup was used.

3.4.2	Auto-strings for polytonic Greek . . . . .	14
3.4.3	Auto-strings for ancient Greek . . . . .	14
3.4.4	Date specification . . . . .	15
3.4.5	Greek numerals . . . . .	17
3.5	Character codes . . . . .	21
3.6	Symbol name aliases . . . . .	23

## 1 Requirements

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

The [CB Greek fonts](#) created by CLAUDIO BECCARI<sup>2</sup> are a complete set of 8-bit T<sub>E</sub>X 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 standard `\DeclareFontFamilySubstitution` macro can be used to set up Greek supplements for other T<sub>E</sub>X font families (like Times or Palatino).

Unicode fonts (used with XeT<sub>E</sub>X or LuaT<sub>E</sub>X) provide slots for all Unicode characters in one font but commonly only a subset of the actual glyphs. **Many Unicode fonts, including the default [Latin Modern](#), do not support the Greek script!** Authors need to set up an alternative font like CM Unicode, Linux Libertine, or DejaVu with [fontspec](#) or the [babel](#) font configuration system.

With 8-bit TeX and XeT<sub>E</sub>X, hyphenation patterns must be pre-loaded in the format file. This is a limitation by TeX, common to all languages. The LuaT<sub>E</sub>X engine loads hyphenation patterns dynamically.

## 2 Usage

To activate Greek language 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 text parts in an English document.

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.

---

<sup>2</sup>Apostolos Syropoulos wishes to thank Claudio Beccari for his patience, collaboration, comments and suggestions.

## 2.1 Language attributes

The attributes `monotoniko`, `polutoniko`<sup>3</sup>, and `ancient` allow the specification of the used orthography. The language variant affects automatic hyphenation, spelling of auto-generated strings and support for multi-accented letters.

The default is modern *monotonic* Greek, while

```
\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}
```

## 2.2 Modifiers

The following modifiers cannot be set with `\languageattribute`. Misspelled modifiers are ignored without warning!

Some workarounds for the non-standard LGR font encoding may have serious side-effects. The `local-LGR-fixes` *modifier* restricts the re-definitions in section 3.3.5 to text parts using the Greek language. The `no-LGR-fixes` *modifier* disables them completely. You may try, e.g.,

```
\usepackage[greek.local-LGR-fixes,english]{babel}
```

as a last resort if the workarounds make a document uncompileable and using Xe/LuaTeX with Unicode fonts is not an option. Check for problems with enumerations in Greek text parts and with Roman and Greek numerals everywhere (especially in the ToC).

These modifiers are provisional, naming and behaviour may change.

## 2.3 Language hooks

`\extragreek`  
`\noextragreek`

The macro `\extragreek` is called by `babel` on every switch of the active language to Greek. The macro `\noextragreek` is called when switching away from Greek. Package and document authors can add setup and tear-down code to the hooks with the help of the `\addto` command provided by `babel`. The first call of `\addto<hookname>{<code>}` initializes the hook, subsequent calls append `<code>` to its definition.

---

<sup>3</sup>with the alias `polytonic`

Babel-greek uses these hooks to, e.g, select correct hyphenation patterns (cf. section 3.1) or ensure a font encoding supporting the Greek script is used for Greek text parts (cf. section 3.3).

## 2.4 Input of Greek text

There are several alternatives to write Greek text.

- Literal input using the UTF-8 encoding is the standard input method. With 8-bit TeX, this requires the package `greek-inputenc`.  
With the packages `inputenc` and `greek-inputenc`, literal Greek characters can also be input using the legacy encodings *iso-8859-7* and *macgreek*.
- 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`. The LICR macros `\textAlpha ... \textomega` are a safe but cumbersome method to input Greek characters.
- The `alphabet` package, bundled with `greek-fontenc`, makes the short macro names `\Alpha ... \omega` available in both, text and math mode.

## 2.5 Greek vs. Latin script

When switching the language to Greek, `babel-greek` ensures that the Greek script is supported. The following macros allow the use of Greek vs. Latin script without changing the active language:

<code>\greekscript</code>	The <i>TextCommand</i> <sup>4</sup> <code>\greekscript</code> switches to a font encoding supporting the
<code>\greektext</code>	Greek script. The declaration <code>\greektext</code> always switches the font encoding to
	LGR. Both declarations do not change the active language.
<code>\latintext</code>	<code>\latintext</code> (defined by the Babel core, deprecated since March 2014) can be
	used to switch back to an encoding supporting the Latin script.
<code>\ensuregreek</code>	The function <code>\ensuregreek</code> takes one argument which is typeset using a font
	encoding supporting the Greek script. It only switches the font encoding if required
	(i.e. if the current font encoding does not support Greek letters and symbols).
<code>\lgrfont</code>	The function <code>\lgrfont</code> <sup>5</sup> switches to the non-standard Greek 8-bit font encod-
	ing LGR. Hint: Use <code>\lgrfont</code> , if you want to use the <i>Latin transliteration</i> input
	method and <code>\ensuregreek</code> else.
<code>\ensureascii</code>	The Babel core defines <code>\ensureascii</code> that typesets its argument using an
	ASCII-compatible “standard text font encoding”. It is the recommended way for
	text parts requiring Latin letters but no language switch.

<sup>4</sup>For a discussion of TextCommands, see the *LaTeX font guide*, too.

<sup>5</sup>The legacy name `\textgreek` is available as alias.

## 2.6 Greek numbering

The [Greek \(Milesian\) alphabetical numbering system](#)<sup>6</sup> is still used in everyday life for short enumerations. It was used for dates and numbers in the range of several thousands in official editions up to the beginning of the 20th century and is still used by the Eastern Orthodox Church and certain scholars. Unfortunately, most Greeks don't know how to write Greek numbers bigger than 20 or 30.

`\greeknumeral`      The command `\greeknumeral` makes it possible to typeset Greek numerals for numbers up to 999 999. `\Greeknnumeral` is the “uppercase” version of this macro. 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*<sup>7</sup>, *zeta*, *eta*, *theta*, followed by a *keraia*, a mark similar to the mathematical symbol “prime”.
- Decades from 10 to 90 are denoted by letters *iota*, *kappa*, *lambda*, *mu*, *nu*, *xi*, *omikron*, *pi*, *koppa*<sup>8</sup>, again followed by the numeric mark.
- 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 an *aristeri keraia*, a prime inverted by 180 degrees and placed in front of the letter, under the baseline. When a group of letters denoting thousands is followed by a group of letters denoting a number under 1000, both marks are used.

The shape of the obsolete characters used for number 6 (*digamma*/*stigma*) and 90 (*koppa*) evolved over time and different characters are in use for them today. The following four macros can be re-defined to configure `\greeknumeral` and `\Greeknnumeral` respectively:

`\greeknumeralsix`      Originally, the sixth letter of the alphabet, standing for 6, was the *digamma* – just as its Latin equivalent F is the sixth letter of the Latin alphabet. As Greek script turned to uncial and then lowercase, digamma changed its shape – it became similar to the ligature for sigma-tau (*stigma*). People started using the stigma or the digraph sigma tau<sup>9</sup>. The macro `\greeknumeralsix` allows configuring the symbol for the number 6 in `\greeknumeral`, the macro `\greeknumeralSix` does the same for `\Greeknnumeral`. The default values are `\textstigma` and `\textStigma`.

`\greeknumeralninety`      Three symbols are in use for the number 90: Classicists prefer the q-like

<sup>6</sup>Attic numerals, which predate the Milesian numerals are implemented in package `athnum`.

<sup>7</sup>cf. `\greeknumeralsix`

<sup>8</sup>cf. `\greeknumeralninety`

<sup>9</sup>Mainly because the letter stigma is not always available, so people opted to write down the first two letters of its name instead.

`\greeknumeralNinety`

“archaic” *koppa* and, more rarely, its uncial form<sup>10</sup>, modern Greek uses the zig-zag shaped “modern” *koppa* exclusively. The macro `\greeknumeralninety` allows configuring the symbol for the number 90 in `\greeknumeral`, the macro `\greeknumeralNinety` does the same for `\Greeknuneral`. The default values are `\textkoppa` and `\textKoppa` for modern Greek and `\textqoppa` and `\textQoppa` for ancient Greek.

There is no such variation in the shape of the *sampi* used for the number 900.

### 3 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}
```

#### 3.1 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` and set the hyphenation to `\language0` for the missing ones.

```
4 \ifx\l@greek\@undefined
5   \@nopatterns{greek}
6   \addialect\l@greek 0
7 \fi
8 \ifx\l@monogreek\@undefined
9   \@nopatterns{greek}
10  \addialect\l@monogreek 0
11 \fi
12 \ifx\l@ancientgreek\@undefined
13   \@nopatterns{greek}
14   \addialect\l@ancientgreek 0
15 \fi
16 \newcount\bbl@monogreek \bbl@monogreek=\l@monogreek
17 \newcount\bbl@polygreek \bbl@polygreek=\l@greek
18 \newcount\bbl@ancientgreek \bbl@ancientgreek=\l@ancientgreek
```

Use the *language hooks* (cf. section 2.3) to set the correct hyphenation patterns. (We collect setup code for the language variants `polutoniko` and `ancient` in `\extraspolutonikogreek` and `\extrasancientgreek`; their content is added to `\extragreek` by the respective language attributes, cf. section 3.2).

`\extraspolutonikogreek`  
`\extrasancientgreek`

<sup>10</sup>resembling CYRILLIC LETTER KOPPA or GOTHIC LETTER NINETY

```

19 \addto\extrasgreek{\let\l@greek=\bbl@monogreek}
20 \addto\extraspolutonikogreek{\l@greek=\bbl@polygreek}
21 \addto\extrasancientgreek{\l@greek=\bbl@ancientgreek}

```

`\providehyphenmins` The macro `\providehyphenmins` is used to set the correct values of the hyphenation parameters `\lefthyphenmin` and `\righthyphenmin`. Yannis Haralambous has suggested the value 1.

```

22 \providehyphenmins{\CurrentOption}{\@ne\@ne}

```

## 3.2 Language variants

The Babel core provides the command `\bbl@declare@ttribute` for the declaration of language attributes in language definition files. It takes three arguments: the name of the language, the attribute to be defined, and the code to be executed when the attribute is to be used. If the language attribute is selected, the third argument is executed after reading the \*.ldf file.

### 3.2.1 polutoniko

The `polutoniko` language attribute selects the “polytonic” spelling. This code adds the expansion of `\extraspolutonikogreek` to `\extrasgreek` to set up support for multi-accented characters and hyphenation patterns for the polytonic orthography.

```

23 \bbl@declare@ttribute{greek}{polutoniko}{%
24   \expandafter\addto\expandafter\extrasgreek
25     \expandafter{\extraspolutonikogreek}%

```

It also uses polytonic spelling for auto-strings (captions and month names).

```

26 \let\captionsgreek\captionspolutonikogreek
27 \let\gr@month\gr@polutoniko@month

```

For backwards compatibility, “polytonic” spelling can also be selected via the dummy language `polutonikogreek`. However, it is not possible to use both options, `greek` and `polutonikogreek` in one document. We also define aliases to allow language switching commands using the language name `polutonikogreek`:

```

28 \let\l@polutonikogreek\l@greek
29 \let\datepolutonikogreek\dategreek
30 \let\extraspolutonikogreek\extrasgreek
31 \let\noextraspolutonikogreek\noextrasgreek
32 }

```

### 3.2.2 polytonic

The `polytonic` language attribute is an alias for the attribute `polutoniko` matching the spelling for this orthography variant in `polyglossia` and Babel \*.ini files.

```

33 \bbl@declare@ttribute{greek}{polytonic}{%
34   \expandafter\addto\expandafter\extrasgreek
35     \expandafter{\extraspolutonikogreek}%

```

```

36 \let\captionsgreek\captionspolutonikogreek
37 \let\gr@month\gr@polutoniko@month
38 }

```

### 3.2.3 ancient

The `ancient` language attribute is used for classical Greek. This attribute adds the expansion of `\extraspolutonikogreek` and `\extrasancientgreek` to `\extragreek` to set up support for multi-accented characters and ancient hyphenation patterns.

```

39 \bbl@declare@ttribute{greek}{ancient}{%
40   \expandafter\addto\expandafter\extragreek
41     \expandafter{\extraspolutonikogreek}% multi-accented letters
42   \expandafter\addto\expandafter\extragreek
43     \expandafter{\extrasancientgreek}%

```

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).

```

44 \let\captionsgreek\captionssancientgreek
45 \let\gr@month\gr@polutoniko@month % (modern) polytonic month names

```

Classicists tend to use the Q-like “archaic” koppa for the number 90. Thus, for classical Greek, we set the default to the “archaic” koppa (cf. section 2.6).

```

46 \renewcommand{\greeknumeralninety}{\textqoppa}%
47 \renewcommand{\greeknumeralNinety}{\textQoppa}%
48 }

```

## 3.3 Font setup

### 3.3.1 Greek font encoding

`\greekfontencoding` The macro `\greekfontencoding` holds the name of the font encoding<sup>11</sup> used to ensure support of the Greek script. The default is LGR for 8-bit TeX and TU for Xe/LuaTeX.<sup>12</sup> It can be overridden defining `\greekfontencoding` with a custom value before loading `babel`.

Also store the name of the *encoding definition file*<sup>13</sup> with the extended Greek setup for the Greek font encoding.

```

49 \ifdefined\UnicodeEncodingName % set by XeTeX/LuaTeX
50   \providecommand*\greekfontencoding{\UnicodeEncodingName}
51   \providecommand*\bbl@greek@fontencdef{tuenc-greek}
52 \else
53   \providecommand*\greekfontencoding{LGR}
54   \providecommand*\bbl@greek@fontencdef{lgrenc}

```

<sup>11</sup>cf. [encguide.pdf](#)

<sup>12</sup>Document authors must ensure that the selected font actually contains the required glyphs.

LGR-encoded fonts can be used alongside Unicode fonts with XeTeX/LuaTeX to enable the input of Greek letters via the Latin transliteration (with some limitations, see `test-greek.tex`).

<sup>13</sup>see [fntguide.pdf](#)

```

55 % TODO the more generic version fails :(
56 % \edef\bbl@greek@fontencdef{\lowercase{\greekfontencoding}enc}
57 \fi

```

### 3.3.2 Ensure loading of Greek font encoding definitions.

If the *encoding definition file* for `\greekfontencoding` is not yet loaded, do this now:

```

58 \@ifl@aded{def}{\bbl@greek@fontencdef}{}
59 {% else
60   \InputIfFileExists{\bbl@greek@fontencdef .def}{}
61   {% else
62     \bbl@error{Font support for the Greek script missing.\\
63               babel-greek can't typeset Greek.\\
64               Install the "greek-fontenc" package\\
65               or use XeTeX/LuaTeX with polyglossia.}
66     {I can't find the \bbl@greek@fontencdef .def file
67      for the Greek fonts (encoding \greekfontencoding)}
68   \@end
69 }
70 }

```

If the PU font encoding is defined (by [hyperref](#)), load extended Greek support for it. Do this in the `\AtBeginDocument` hook because documents may load `hyperref` after `babel`. We cannot rely on `@` being a letter when the hook is called and we must not use `\makeatother` in the hook ([explanation at stackexchange](#)). We use a temporary function to save and restore the previous catcode.

```

71 \AtBeginDocument{%
72   \@ifl@aded{def}{puenc}%
73   {\@ifl@aded{def}{puenc-greek}
74    {%
75     {\edef\RestoreAtCatcode{\catcode'\@=\the\catcode'\@relax}%
76      \makeatletter
77      \InputIfFileExists{puenc-greek.def}%
78      {}%
79      {\bbl@warning{I cannot find the "puenc" Greek fixes
80                   from "greek-fontenc".}%
81      }%
82      \RestoreAtCatcode
83    }%
84   }% end "puenc.def loaded" branch
85   {% empty "puenc.def not loaded" branch
86 }

```

### 3.3.3 Font encoding switches

`\greekscript` The `\greekscript` is a declaration that switches the font encoding to `\greekfontencoding`. The extended Greek font encoding definitions

<sup>14</sup>See [fntguide.pdf](#) for more info about *TextCommands*.

from [greek-fontenc](#) define empty local variants for TU, LGR, and PU, so that the declaration does nothing if the active font encoding supports the Greek script.

```
87 \ProvideTextCommandDefault{\greekscript}{%
88 \fontencoding{\greekfontencoding}\selectfont
89 \def\encodingdefault{\greekfontencoding}}
```

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

```
90 \ProvideTextCommandDefault{\ensuregreek}[1]{%
91 \leavevmode {\greekscript #1}}
```

**\BabelGreekRestoreFontEncoding** The declaration `\BabelGreekRestoreFontEncoding` changes the font encoding to the value it had before the switch to the Greek language. It does nothing, if the language switch did not trigger a font encoding switch.

```
92 \def\BabelGreekRestoreFontEncoding{%
93 \ifx\cf@encoding\BabelGreekPreviousFontEncoding
94 \else
95 \let\encodingdefault\BabelGreekPreviousFontEncoding
96 \fontencoding{\encodingdefault}\selectfont
97 \fi
98 }
```

Add font encoding switches to the language hooks (cf. section [2.3](#)) to ensure a font encoding supporting the Greek script is used in Greek text parts:

```
99 \addto\extrasgreek{%
100 \let\BabelGreekPreviousFontEncoding\cf@encoding
101 \greekscript}
102 \addto\noextrasgreek{\BabelGreekRestoreFontEncoding}
```

### 3.3.4 Additional commands for the LGR font encoding

The actions in this section add “harmless” setup steps for the LGR font encoding that cannot be done in the `lgrenc.def` encoding definition file.

We do this only, if the LGR font encoding is defined (either by `fontenc` or `babel-greek`), but also if it is not the `\greekfontencoding`:

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

**\greektext** The declaration `\greektext` switches to LGR. Use this if you explicitly require LGR (e.g. to use the Latin transliteration or special fonts). Use `\greekscript` instead, if you want to avoid a font encoding change if the current font encoding already supports the Greek script (e.g. TU). For shorter pieces of text, the `\lgrfont` (see below) or `\ensuregreek` commands should be used. Cf. section [3.3.3](#).

```
104 \DeclareRobustCommand{\greektext}{%
105 \fontencoding{LGR}\selectfont
106 \def\encodingdefault{LGR}}
```

`\lgrfont` This command takes an argument which is typeset using the LGR font encoding. The original name `\textgreek` is deprecated because of its ambiguity: The command does not change the text *language* but only the font encoding, which allows the use of the Greek *script* but does not activate Greek hyphenation and case-changing rules.

```
107 \DeclareTextFontCommand{\lgrfont}{\greektext}
108 \let\textgreek\lgrfont
```

`\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 definition file and is only kept for backwards compatibility.)

```
109 \def\outlfamily{\usefont{LGR}{cmro}{m}{n}}
110 \DeclareTextFontCommand{\textol}{\outlfamily}
```

Add LGR-specific variants to some *TextCommands* that use Latin characters in their default definition. These definitions cannot be done in `lgrenc.def` because they rely on `\ensureascii` (defined by `babel`).

```
111 \ProvideTextCommand{\textcopyright}{LGR}{\ensureascii{\textcopyright}}
112 \ProvideTextCommand{\textregistered}{LGR}{\ensureascii{%
113                                     \textregistered}}
114 \ProvideTextCommand{\texttrademark}{LGR}{\ensureascii{\texttrademark}}
```

`\textampersand` LGR has a “middle dot” glyph at the place of the ampersand. Provide the *TextCommand* `\textampersand` and an LGR-specific version. It is used in the next section to define a version of `&` that also works in LGR.

```
115 \let\bbl@greek@original@amp&
116 \ProvideTextCommandDefault{\textampersand}{\bbl@greek@original@amp}
117 \ProvideTextCommand{\textampersand}{LGR}{%
118     \ensureascii{\bbl@greek@original@amp}}
```

`\EnsureStandardFontEncoding` The *TextCommand* `\EnsureStandardFontEncoding` can be used to make existing commands “LGR-proof”. It makes sure its argument is typeset using a [standard text font encoding](#). The default is an empty command: almost all commonly used font encodings are standard text encodings – LGR is the notable exception. The local LGR variant uses `\ensureascii` from the Babel core that comes with elaborate heuristics to select a suitable standard font encoding. A special clause for `hyperref` avoids warnings from this package.

```
119 \ProvideTextCommandDefault{\EnsureStandardFontEncoding}{\@firstofone}
120 \ProvideTextCommand{\EnsureStandardFontEncoding}{LGR}[1]{%
121                                     \ensureascii{#1}}
122 \AtBeginDocument{\@ifpackageloaded{hyperref}
123                 {\pdfstringdefDisableCommands{%
124                     \let\EnsureStandardFontEncoding\@firstofone}}
125                 {}}
```

End the LGR additions block:

```
126 }{}
```

### 3.3.5 LGR workarounds

The following redefinitions work around problems with the non-standard LGR font encoding. As they may have serious side-effects, they are only done if LGR is the default Greek font encoding (cf. section 3.3.1).

As an emergency measure, the `local-LGR-fixes` or `no-LGR-fixes` modifiers (cf. section 2.2) can be used to restrict the “roman” redefinitions to text parts using the Greek language or skip them completely.

To prevent Roman numerals being typeset with Greek letters in text parts using the LGR font encoding, they must be wrapped in `\ensureascii`. However, Roman numerals are also auto generated by LaTeX and used in moving arguments.<sup>15</sup> These “moving” Roman numbers must be LGR-proofed also if they originate from a text part using a standard font encoding. This can only be ensured by a global re-definition of the generating functions `\@roman` and `\@Roman`. On the other hand, the re-definition breaks the assumption by MakeIndex, that page numbers are plain character sequences. Hyperref assumes that `\thepage` is expandable and doesn’t contain formatting instructions (cf. Babel issue #170).

The ampersand macro `\&` is used in both, text and math mode. Let it use the new defined *TextCommand* `\textampersand` in text mode.

```

127 \def\bbl@tempa{LGR}
128 \ifx\greekfontencoding\bbl@tempa
129   \def\bbl@greek@roman#1{\expandafter\EnsureStandardFontEncoding%
130     \expandafter{\romannumeral#1}}
131   \def\bbl@greek@Roman#1{\expandafter\EnsureStandardFontEncoding%
132     \expandafter{\expandafter\@slowromancap\romannumeral#1@}}
133   \DeclareRobustCommand{\bbl@greek@ampersand}{%
134     \ifmmode\bbl@greek@original@amp\else\textampersand\fi}
135   \bbl@xin@{,no-LGR-fixes,}{,\BabelModifiers,}%
136   \ifin@
137     % skip re-definitions
138   \else
139     \bbl@xin@{,local-LGR-fixes,}{,\BabelModifiers,}%
140     \ifin@
141       \addto\extrasgreek{%
142         \babel@save\@roman
143         \babel@save\@Roman
144         \let\@roman\bbl@greek@roman
145         \let\@Roman\bbl@greek@Roman
146         \babel@save\&%
147         \let\&\bbl@greek@ampersand%
148       }
149     \else
150       \let\@roman\bbl@greek@roman
151       \let\@Roman\bbl@greek@Roman
152       \let\&\bbl@greek@ampersand
153     \fi

```

---

<sup>15</sup>For example, Roman page numbers are generated at “unpredictable” positions and can move to the ToC, (hyper)references, or an index.

```

154 \fi
155 \fi

```

### 3.4 Definitions for the Greek language

The next step consists in defining macros for the requirements of Greek typesetting which will later be added to the language switch hooks.

#### 3.4.1 Auto-strings for Greek

`\captionsgreek` The macro `\captionsgreek` defines all strings used in the four standard document classes provided with L<sup>A</sup>T<sub>E</sub>X.

```

156 \addto\captionsgreek{%
157   \def\prefacename{\textPi\textrho\acctonos\textomicron\textlambda
158     \textomicron\textgamma\textomicron\textfinalsigma}%
159   \def\refname{\textAlpha\textnu\textalpha
160     \textphi\textomicron\textrho\acctonos\textepsilon\textfinalsigma}%
161   \def\abstractname{\textPi\textepsilon\textrho\acctonos\textiota
162     \textlambda\texteta\textpsi\texteta}%
163   \def\bibname{\textBeta\textiota\textbeta\textlambda\textiota
164     \textomicron\textgamma\textrho\textalpha\textphi\acctonos
165     \textiota\textalpha}%
166   \def\chaptername{\textKappa\textepsilon\textphi\acctonos\textalpha
167     \textlambda\textalpha\textiota\textomicron}%
168   \def\appendixname{\textPi\textalpha\textrho\acctonos\textalpha\textrho
169     \texttau\texteta\textmu\textalpha}%
170   \def\contentsname{\textPi\textepsilon\textrho\textiota
171     \textepsilon\textchi\acctonos\textomicron\textmu\textepsilon
172     \textnu\textalpha}%
173   \def\listfigurename{\textKappa\textalpha\texttau\acctonos\textalpha
174     \textlambda\textomicron\textgamma\textomicron\textfinalsigma{}
175     \textSigma\textchi\texteta\textmu\acctonos\textalpha\texttau
176     \textomega\textnu}%
177   \def\listtablename{\textKappa\textalpha\texttau\acctonos\textalpha
178     \textlambda\textomicron\textgamma\textomicron\textfinalsigma{}
179     \textPi\textiota\textnu\acctonos\textalpha\textkappa\textomega
180     \textnu}%
181   \def\indexname{\textEpsilon\textupsilon\textrho\textepsilon
182     \texttau\acctonos\texteta\textrho\textiota\textomicron}%
183   \def\figurename{\textSigma\textchi\acctonos\texteta\textmu\textalpha}%
184   \def\tablename{\textPi\acctonos\textiota\textnu\textalpha
185     \textkappa\textalpha\textfinalsigma}%
186   \def\partname{\textMu\acctonos\textepsilon\textrho\textomicron
187     \textfinalsigma}%
188   \def\enclname{\textSigma\textupsilon\textnu\texteta\textmu
189     \textmu\acctonos\textepsilon\textnu\textalpha}%
190   \def\ccname{\textKappa\textomicron\textiota\textnu\textomicron
191     \textpi\textomicron\acctonos\textiota\texteta\textsigma\texteta}%
192   \def\headtoname{\textPi\textrho\textomicron\textfinalsigma}%

```

```

193 \def\pagename{\textSigma\textepsilon\textlambda\acconos\textiota
194 \textdelta\textalpha}%
195 \def\seename{\textbeta\textlambda\acconos\textepsilon\textpi
196 \textepsilon}%
197 \def\alsoname{\textbeta\textlambda\acconos\textepsilon\textpi
198 \textepsilon{} \textepsilon\textpi\acconos\textiota\textsigma
199 \texteta\textfinalsigma}%
200 \def\proofname{\textAlpha\textpi\acconos\textomicron
201 \textdelta\textepsilon\textiota\textxi\texteta}%
202 \def\glossaryname{\textGamma\textlambda\textomega\textsigma
203 \textsigma\acconos\textalpha\textrho\textiota}%
204 }

```

### 3.4.2 Auto-strings for polytonic Greek

`\captionspolutonikogreek` For texts written in 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.

```

205 \let\captionspolutonikogreek\captionsgreek
206 \addto\captionspolutonikogreek{%
207 \def\refname{\accpsili\textAlpha\textnu\textalpha
208 \textphi\textomicron\textrho\accvaria\textepsilon\textfinalsigma}%
209 \def\indexname{\textEpsilon\accdasia\textupsilon\textrho\textepsilon
210 \texttau\acconos\texteta\textrho\textiota\textomicron}%
211 \def\figurename{\textSigma\textchi\accperispomeni\texteta\textmu
212 \textalpha}%
213 \def\headtoname{\textPi\textrho\accvaria\textomicron\textfinalsigma}%
214 \def\alsoname{\textbeta\textlambda\acconos\textepsilon\textpi
215 \textepsilon{} \accpsili\textepsilon\textpi\acconos\textiota
216 \textsigma\texteta\textfinalsigma}%
217 \def\proofname{\accpsili\textAlpha\textpi\acconos\textomicron
218 \textdelta\textepsilon\textiota\textxi\texteta}%
219 }

```

### 3.4.3 Auto-strings for ancient Greek

`\captionsansientgreek` For texts written in ancient Greek, we took the translations from Apostolos Syropoulos' [xgreek](#) package. For now we just add extra definitions to `\captionsgreek` in order to override the earlier definitions.

```

220 \let\captionsansientgreek\captionsgreek
221 \addto\captionsansientgreek{%
222 \def\prefacename{\textPi\textrho\textomicron\textomicron
223 \acconos\textiota\textmu\textiota\textomicron\textnu}%
224 \def\refname{\accpsili\textAlpha\textnu\textalpha\textphi\textomicron
225 \textrho\textalpha\accvaria\textiota}%
226 \def\abstractname{\textPi\textepsilon\textrho\acconos\textiota
227 \textlambda\texteta\textpsi\textiota\textvarsigma}%
228 \def\bibname{\textBeta\textiota\textbeta\textlambda\textiota
229 \textomicron\textgamma\textrho\textalpha\textphi

```

```

230 \acconos\textiota\textalpha}%
231 \def\chaptername{\textKappa\textepsilon\textphi\acconos\textalpha
232 \textlambda\textalpha\textiota\textomicron\textnu}%
233 \def\appendixname{\textPi\textalpha\textrho\acconos\textalpha
234 \textrho\texttau\texteta\textmu\textalpha}%
235 \def\contentsname{\textPi\textepsilon\textrho\textiota\textepsilon
236 \textchi\acconos\textomicron\textmu\textepsilon\textnu\textalpha}%
237 \def\listfigurename{\textKappa\textalpha\texttau\acconos\textalpha
238 \textlambda\textomicron\textgamma\textomicron\textvarsigma{}
239 \textsigma\textchi\texteta\textmu\acconos\textalpha\texttau
240 \textomega\textnu}%
241 \def\listtablename{\textKappa\textalpha\texttau\acconos\textalpha
242 \textlambda\textomicron\textgamma\textomicron\textvarsigma{}
243 \textpi\textiota\textnu\acconos\textalpha\textkappa
244 \textomega\textnu}%
245 \def\indexname{\textEpsilon\accasia\textupsilon\textrho\textepsilon
246 \texttau\acconos\texteta\textrho\textiota\textomicron\textnu}%
247 \def\figurename{\textSigma\textchi\accperispomeni\texteta\textmu
248 \textalpha}%
249 \def\tablename{\textPi\acconos\textiota\textnu\textalpha\textxi}%
250 \def\partname{\textMu\acconos\textepsilon\textrho\textomicron
251 \textvarsigma}%
252 \def\enclname{\textSigma\textupsilon\textnu\texteta\textmu\textmu
253 \acconos\textepsilon\textnu\textomega\textvarsigma}%
254 \def\ccname{\textKappa\textomicron\textiota\textnu\textomicron\textpi
255 \textomicron\acconos\textiota\texteta\textsigma\textiota
256 \textvarsigma}%
257 \def\headtoname{\textPi\textrho\accvaria\textomicron\textvarsigma}%
258 \def\pagename{\textSigma\textepsilon\textlambda\accvaria\textiota
259 \textvarsigma}%
260 \def\seename{\accasiaoxia\textomicron\textrho\textalpha}%
261 \def\alsoname{\accasiaoxia\textomicron\textrho\textalpha}
262 \accasia\textomega\textsigma\textalpha\acconos\textupsilon
263 \texttau\textomega\textvarsigma}%
264 \def\proofname{\accpsili\textAlpha\textpi\acconos\textomicron
265 \textdelta\textepsilon\textiota\textxi\textiota\textvarsigma}%
266 \def\glossaryname{\textGamma\textlambda\textomega\textsigma\textsigma
267 \acconos\textalpha\textrho\textiota\textomicron\textnu}%
268 }

```

### 3.4.4 Date specification

`\gr@month` The auxiliary macro `\gr@month` returns Greek month names in monotonic spelling.

```

269 \def\gr@month{%
270 \ifcase\month\or
271 \textIota\textalpha\textnu\textomicron\textupsilon\textalpha
272 \textrho\acconos\textiota\textomicron\textupsilon \or
273 \textPhi\textepsilon\textbeta\textrho\textomicron\textupsilon

```

```

274 \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
275 \textMu\textalpha\textrho\texttau\acctonos\textiota\textomicron
276 \textupsilon \or
277 \textAlpha\textpi\textrho\textiota\textlambda\acctonos\textiota
278 \textomicron\textupsilon \or
279 \textMu\textalpha\'"\textiota\textomicron\textupsilon \or
280 \textIota\textomicron\textupsilon\textnu\acctonos\textiota
281 \textomicron\textupsilon \or
282 \textIota\textomicron\textupsilon\textlambda\acctonos\textiota
283 \textomicron\textupsilon \or
284 \textAlpha\textupsilon\textgamma\textomicron\acctonos\textupsilon
285 \textsigma\texttau\textomicron\textupsilon \or
286 \textSigma\textepsilon\textpi\texttau\textepsilon\textmu
287 \textbeta\textrho\acctonos\textiota\textomicron\textupsilon \or
288 \textOmicron\textkappa\texttau\textomega\textbeta
289 \textrho\acctonos\textiota\textomicron\textupsilon \or
290 \textNu\textomicron\textepsilon\textmu\textbeta
291 \textrho\acctonos\textiota\textomicron\textupsilon \or
292 \textDelta\textepsilon\textkappa\textepsilon\textmu\textbeta
293 \textrho\acctonos\textiota\textomicron\textupsilon
294 \fi
295 }

```

`\gr@polutoniko@month` The auxiliary macro `\gr@polutoniko@month` returns Greek month names in polytonic spelling. It is activated by the `polutoniko` language option.

```

296 \def\gr@polutoniko@month{%
297 \ifcase\month\or
298 \accpsili\textIota\textalpha\textnu\textomicron\textupsilon
299 \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
300 \textPhi\textepsilon\textbeta\textrho\textomicron\textupsilon
301 \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
302 \textMu\textalpha\textrho\texttau\acctonos\textiota\textomicron
303 \textupsilon \or
304 \accpsili\textAlpha\textpi\textrho\textiota\textlambda
305 \acctonos\textiota\textomicron\textupsilon \or
306 \textMu\textalpha\accdialytikatonos\textiota\textomicron
307 \textupsilon \or
308 \accpsili\textIota\textomicron\textupsilon\textnu
309 \acctonos\textiota\textomicron\textupsilon \or
310 \accpsili\textIota\textomicron\textupsilon\textlambda
311 \acctonos\textiota\textomicron\textupsilon \or
312 \textAlpha\accpsili\textupsilon\textgamma\textomicron\acctonos
313 \textupsilon\textsigma\texttau\textomicron\textupsilon \or
314 \textSigma\textepsilon\textpi\texttau\textepsilon\textmu\textbeta
315 \textrho\acctonos\textiota\textomicron\textupsilon \or
316 \accpsili\textOmicron\textkappa\texttau\textomega\textbeta
317 \textrho\acctonos\textiota\textomicron\textupsilon \or
318 \textNu\textomicron\textepsilon\textmu\textbeta
319 \textrho\acctonos\textiota\textomicron\textupsilon \or
320 \textDelta\textepsilon\textkappa\textepsilon\textmu

```

```

321      \textbeta\textrho\acctonos\textiota\textomicron\textupsilon
322  \fi
323 }

```

**\dategreek** The macro `\dategreek` redefines the command `\today` to produce greek dates. The name of the month is produced by the macro `\gr@month` since it is also needed in the definition of the macro `\Grtoday`.

```

324 \def\dategreek{%
325   \def\today{\number\day \space \gr@month\space \number\year}}

```

**\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.

```

326 \def\Grtoday{%
327   \expandafter\Greeknatural\expandafter{\the\day}\space
328   \gr@polutoniko@month \space
329   \expandafter\Greeknatural\expandafter{\the\year}}

```

### 3.4.5 Greek numerals

**\greeknumeralsix** The shape of the obsolete characters used for number 6 (digamma/stigma) and **\greeknumeralSix** 90 (koppa) evolved over time and different characters are in use for them today.

**\greeknumeralninety** We define placeholders that allow configuration by the user or a package.

```

\greeknumeralNinety 330 \providecommand*\greeknumeralsix{\textstigma}
331 \providecommand*\greeknumeralSix{\textStigma}
332 \providecommand*\greeknumeralninety{\textkoppa}
333 \providecommand*\greeknumeralNinety{\textKoppa}

```

**\greeknumeral** The commands `\greeknumeral` and `\Greeknatural` produce the lowercase and uppercase [Greek numerals](#) respectively.

The command `\greeknumeral` needs to be *fully* expandable in order to get the right information in auxiliary files. It should also be usable in PDF-strings. Therefore we use the implementation from the `\HyPsd@GreekPatch` in [hyperref](#) (version 7.00e 2020-05-15).

```

334 \def\greeknumeral#1{%
335   {\greekscript
336     \bbl@greek@GreekNum\@firstoftwo{#1}}}%
337 }

```

**\Greeknatural** The command `\Greeknatural` prints uppercase greek numerals.

```

338 \def\Greeknatural#1{%
339   {\greekscript
340     \bbl@greek@GreekNum\@secondoftwo{#1}}}%
341 }

```

**\bbl@greek@ill@value** When the argument of `\greeknumeral` has a value outside of the acceptable bounds ( $0 < x < 999999$ ) a warning will be issued (and the argument be printed).

```

342 \def\bbl@greek@ill@value#1{%
343   \PackageWarningNoLine{babel}{Illegal value (#1) for greeknumeral}}%

```

```

344 \@arabic{#1}%
345 }

\bb1@greek@GreekNum The auxiliary macros provide the actual conversion. They are taken from hyperref
\bb1@greek@@GreekNum as well.
\bb1@greek@GreekNumI 346 \def\bb1@greek@GreekNum#1#2{%
\bb1@greek@GreekNumII 347 \ifnum#2<\@ne
\bb1@greek@GreekNumIII 348 \bb1@greek@i11@value{#2}%
\bb1@greek@GreekNumIV 349 \else
\bb1@greek@GreekNumV 350 \ifnum#2<1000000 %
\bb1@greek@GreekNumVI 351 \bb1@greek@@GreekNum#1{#2}%
352 \else
353 \bb1@greek@i11@value{#2}%
354 \fi
355 \fi
356 }
357 \def\bb1@greek@@GreekNum#1#2{%
358 \ifnum#2<\@m
359 \ifnum#2<10 %
360 \expandafter\bb1@greek@GreekNumI
361 \expandafter\@gobble\expandafter#1\number#2%
362 \else
363 \ifnum#2<100 %
364 \expandafter\bb1@greek@GreekNumII
365 \expandafter\@gobble\expandafter#1\number#2%
366 \else
367 \expandafter\bb1@greek@GreekNumIII
368 \expandafter\@gobble\expandafter#1\number#2%
369 \fi
370 \fi
371 \ifnum#2>\z@
372 \textnumeralsigngreek
373 \fi
374 \else
375 \ifnum#2<\@M
376 \expandafter\bb1@greek@GreekNumIV\expandafter#1\number#2%
377 \else
378 \ifnum#2<100000 %
379 \expandafter\bb1@greek@GreekNumV\expandafter#1\number#2%
380 \else
381 \expandafter\bb1@greek@GreekNumVI\expandafter#1\number#2%
382 \fi
383 \fi
384 \fi
385 }
386 \def\bb1@greek@GreekNumI#1#2#3{%
387 #1{%
388 \ifnum#3>\z@
389 \textnumeralsignlowergreek
390 \fi

```

```

391 }%
392 \expandafter#2%
393 \ifcase#3 %
394   {}{}%
395 \or\textalpha\textAlpha
396 \or\textbeta\textBeta
397 \or\textgamma\textGamma
398 \or\textdelta\textDelta
399 \or\textepsilon\textEpsilon
400 \or\greeknumeralsix\greeknumeralSix % stigma or digamma
401 \or\textzeta\textZeta
402 \or\texteta\textEta
403 \or\texttheta\textTheta
404 \else
405   {}{}%
406 \fi
407 }
408 \def\bbl@greek@GreekNumII#1#2#3#4{%
409   #1{%
410     \ifnum#3>\z@
411       \textnumeralsignlowergreek
412       \fi
413   }%
414 \expandafter#2%
415 \ifcase#3 %
416   {}{}%
417 \or\textiota\textIota
418 \or\textkappa\textKappa
419 \or\textlambda\textLambda
420 \or\textmugreek\textMu
421 \or\textnu\textNu
422 \or\textxi\textXi
423 \or\textomicron\textOmicron
424 \or\textpi\textPi
425 \or\greeknumeralninety\greeknumeralNinety % koppa or qoppa
426 \else
427   {}{}%
428 \fi
429 \bbl@greek@GreekNumI#1#2#4%
430 }
431 \def\bbl@greek@GreekNumIII#1#2#3#4#5{%
432   #1{%
433     \ifnum#3>\z@
434       \textnumeralsignlowergreek
435       \fi
436   }%
437 \expandafter#2%
438 \ifcase#3 %
439   {}{}%
440 \or\textrho\textRho

```

```

441 \or\textsigma\textSigma
442 \or\texttau\textTau
443 \or\textupsilon\textUpsilon
444 \or\textphi\textPhi
445 \or\textchi\textChi
446 \or\textpsi\textPsi
447 \or\textomega\textOmega
448 \or\textsamplegreek\textSamplegreek
449 \else
450   {}{}%
451 \fi
452 \bbl@greek@GreekNumII#1#2#4#5%
453 }
454 \def\bbl@greek@GreekNumIV#1#2#3#4#5{%
455   \bbl@greek@GreekNumI\@firstofone#1#2%
456   \bbl@greek@@GreekNum#1{#3#4#5}%
457 }
458 \def\bbl@greek@GreekNumV#1#2#3#4#5#6{%
459   \bbl@greek@GreekNumII\@firstofone#1#2#3%
460   \bbl@greek@@GreekNum#1{#4#5#6}%
461 }
462 \def\bbl@greek@GreekNumVI#1#2#3#4#5#6#7{%
463   \bbl@greek@GreekNumIII\@firstofone#1#2#3#4%
464   \bbl@greek@@GreekNum#1{#5#6#7}%
465 }

```

`\greek@alph` In the previous release of this language definition file the commands `\greek@aplh`  
`\greek@Alph` and `\greek@Alph` were kept just for reasons of compatibility. Here again they become meaningful macros. They are defined in a way that even page numbering with greek numerals is possible.

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

```

466 \def\greek@alph#1{\expandafter\greeknumeral\expandafter{\the#1}}
467 \def\greek@Alph#1{\expandafter\Greeknatural\expandafter{\the#1}}

```

Redefine the internal macros `\@alph` and `\@Alph` in the language hook, so that we use Greek numerals<sup>16</sup> instead of the Latin alphabet<sup>17</sup> in Greek text parts.

```

468 \addto\extrasgreek{%
469   \babel@save\@alph
470   \babel@save\@Alph
471   \let\@alph\greek@alph
472   \let\@Alph\greek@Alph
473 }

```

<sup>16</sup>cf. section 3.4.5

<sup>17</sup>Eventually interpreted as Latin transliteration and converted to Greek letters in a “strange” order.

### 3.5 Character codes

Greek letters drop diacritics (except dialytika and sub-iota) in UPPERCASE. This is not cared for by the Unicode standard. The file `greek-euenc.def` from `greek-fontenc` contains the required `\lccode` and `\uccode` corrections from the `xgreek` package by Apostolos Syropoulos. It is loaded if the Greek font encoding is TU (i.e. with XeTeX/LuaTeX), see section 3.3.2.

If the Greek font encoding is LGR, character code changes are done here because they must be restricted to text parts using the LGR encoding.

```
474 \def\bbl@tempa{LGR}
475 \ifx\greekfontencoding\bbl@tempa
```

In order to get correct hyphenation we need to set the lower case code of a number of characters.

In LGR encoded fonts, diacritics can be obtained using Knuth's ligature mechanism (see `usage.pdf`). This means that the characters `<`, `>`, `~`, `'`, `,`, `"`, and `|` may be part of a word. Therefore, their `\lccode` is changed when polytonic Greek is in effect. For monotonic Greek, we only need `'` and `"`.

The `'v`' character has a special usage in LGR-encoded fonts: The LGR ligature mechanism detects the end of a word and assures that a final sigma ( $\varsigma$ ) is used. The `'v`' after an `'s`' overrides this ligature mechanism so that it is possible to typeset an isolated  $\sigma$  without it becoming a  $\varsigma$ . Because of this we make sure its lowercase code is not changed.

```
476 \addto\extrasgreek{%
477   \babel@savevariable{\lccode'v}\lccode'v='v%
478   \babel@savevariable{\lccode'\'}\lccode'\']='\'%
479   \babel@savevariable{\lccode'\'}\lccode'\']='\'%
480 }
481 \addto\extrapolutoniumgreek{%
482   % \l@greek=\bbl@polygreek
483   \babel@savevariable{\lccode'<}\lccode'<='<%
484   \babel@savevariable{\lccode'>}\lccode'>='>%
485   \babel@savevariable{\lccode'~}\lccode'~='~%
486   \babel@savevariable{\lccode'|}\lccode'|='|%
487   \babel@savevariable{\lccode'\'}\lccode'\']='\'%
488 }
```

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 (used 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.

```
489 \addto\extrasgreek{%
490   % 'high bit characters': set in a loop and correct exceptions
491   \@tempcnta=128%
492   \@whilenum\@tempcnta<253\do{%
493     \expandafter\babel@savevariable\expandafter{%
494       \expandafter\lccode\the\@tempcnta}%
495   }
```

```

495     \lccode\@tempcnta=\@tempcnta
496     \advance\@tempcnta\@ne
497   }%
498   % Fix non-word characters:
499   \lccode151=0%
500   \lccode155=0%
501   \lccode159=0%
502   \lccode199=0%
503   % Fix capital letters:
504   \lccode195=147% GREEK LETTER DIGAMMA
505   \lccode219=240% GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
506   \lccode223=244% GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
507 }

```

In order to drop diacritics (except dialytika and sub-iota) in UPPERCASE also with the “input ligatures” the `\uccode` of the relevant characters is set to a dummy character. This is only done, if LaTeX is older than 2022/06/01 because the `\MakeUppercase` implementation introduced in this version ignores uccodes and fails with the “dummy” character 0x9f.

```

508 % fallback for for LaTeX versions older than 2020-10-01
509 \providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
510 \IfFormatAtLeastTF{2022/06/01}%
511 {}
512 {% else
513   \addto\extragreek{%
514     \babel@savevariable{\uccode'\}\uccode\'="'\%
515     \babel@savevariable{\uccode'\'}\uccode\'=159% 159 == ^9f
516   }
517   \addto\extraspolutonikogreek{%
518     \babel@savevariable{\uccode'\~}\uccode'\~=159%
519     \babel@savevariable{\uccode'\>}\uccode'\>=159%
520     \babel@savevariable{\uccode'\<}\uccode'\<=159%
521     \babel@savevariable{\uccode'\|}\uccode'\|='\|%
522     \babel@savevariable{\uccode'\'}\uccode\'=159%
523   }

```

To avoid inputenc errors if the tilde is used as perispomeni (in polytonic or ancient Greek), we need to declare an expansion for the “dummy” character 0x9f = 159.<sup>18</sup> To be independent of inputenc, we do not use `\DeclareInputText` but code modelled after its definition to declare an empty expansion.

```

524   \bgroup
525     \uccode'\~159%
526     \uppercase{%
527   \egroup
528     \def~{}%
529   }

```

Add composite commands, so that the dialytika is kept or put on the following

---

<sup>18</sup>Since UTF-8 became the default encoding (cf. [LaTeX News 28](#)), an “inputenc” error is also thrown if the inputenc package is not loaded.

character of a diphthong with `\MakeUppercase` (see `lgrdef.enc` from the `greek-fontenc` package for details).

```
530 \DeclareTextCompositeCommand{"}{LGR}{^~9f}{\accdialytika}
531 \DeclareTextCompositeCommand{'}{LGR}{^~9f}{\LGR@hiatus}
532 \DeclareTextCompositeCommand{'}{LGR}{^~9f}{\LGR@accdropped}
```

If Unicode fonts are loaded together with LGR, we must also care for `\''` and `\"'` in TU, because the is kept when upcasing.

```
533 \ifdefined\UnicodeEncodingName % set by XeTeX/LuaTeX
534 \DeclareTextCompositeCommand{"}{TU}{^~9f}{\accdialytika}
535 \fi
536 }% end of the \IfFormatAtLeastTF else block
```

`\greek@tilde` By default, the tilde produces an unbreakable space in text mode. In polytonic and ancient Greek, we change its meaning to allow using `~` in the Latin transliteration of characters with perispomeni. As the perispomeni is not required with monotonic Greek, this is only done for the variants “polutoniko” and “ancient” (in `\extraspolutonikogreek`).

Let the tilde character expand to a tilde with category code 12.

```
537 \begingroup
538 \@ifundefined{active@char\string!}{\catcode'\!=12\relax}
539 \catcode'\~=12%
540 \lccode'\!='\~%
541 \lowercase{\def\x{\endgroup
542 \def\greek@tilde{!}}\x}
543 \addto\extraspolutonikogreek{\languageshorthands{greek}}
544 \declare@shorthand{greek}{~}{\greek@tilde}
```

```
545 \fi % End of LGR-specific code.
```

### 3.6 Symbol name aliases

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

```
546 \providecommand*\anwtonos{\textdexiakeraia}
547 \providecommand*\katwtonos{\textaristerikeraia}
548 \providecommand*\qoppa{\textkoppa}
549 \providecommand*\varqoppa{\textqoppa}
550 \providecommand*\stigma{\textstigma}
551 \providecommand*\sampi{\textsampi}
552 \providecommand*\Digamma{\textDigamma}
553 \providecommand*\ddigamma{\textdigamma}
554 \providecommand*\vardigamma{\textvardigamma}
555 \providecommand*\euro{\texteuro}
556 \providecommand*\permill{\textperthousand}
557 \ProvideTextCommand{\textmugreek}{\greekfontencoding}{\textmu}
```

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.

```

558 \ldf@finish{\CurrentOption}
559 \code>

```

## Change History

babel-greek-1.08	definition (report Eike Schmidt). . . . .	21
\greekscript: New		
TextCommands “greekscript”		
and “ensuregreek”. . . . .		10
General: greek.dtx renamed to		
babel-greek.dtx (but still		
generates greek.ldf). . . . .		1
Check for EU1/EU2 font		
encoding instead of engine . . . .		8
Load euenc.def if EU1 or EU2		
font encoding is detected. . . . .		8
Restore compatibility with		
Xe/LuaTeX in 8-bit and		
Unicode mode. . . . .		1
Use EU1 or EU2 for Latin script		
if available . . . . .		10
Use font-encoding specific		
TextCommands. . . . .		11
Remove redefinition of		
\fnum@figure and		
\fnum@table. . . . .		23
babel-greek-1.08a		
\greekscript: Set		
‘encodingdefault’ to fix Greek		
in footnotes etc. with		
document language Greek. . . .		10
babel-greek-1.09		
General: Load correct hyphenation		
patterns (patch by Claudio		
Beccari). . . . .		6
Add support for ancient Greek. . .		8
Added caption names for		
\ancientgreek . . . . .		14
Added lc codes for chars 128 to		
255 . . . . .		21
babel-greek-1.09b		
\captionspolutonikogreek: Use		
named macros instead of		
non-standard short accent		
macros for psili and dasia. . . .		14
General: Remove spurious		
whitespace from ‘extragreek’		
	General: Fix dummy hyphenation	
	language names (patch Ulrike	
	Fischer). . . . .	6
	babel-greek-1.09d	
	General: uc-/lccode corrections	
	from xgreek are now in	
	greek-euenc.def (the	
	polyglossia version has bugs). . . .	21
	babel-greek-1.09e	
	General: Fix bug in lccode-setting	
	loop (patch by Enrico	
	Gregorio). . . . .	21
	babel-greek-1.09f	
	General: Check also for standard	
	Unicode text encoding “TU”	
	(new in fontspec v2.5a). . . . .	8
	babel-greek-1.09g	
	General: Babel 3.9i deprecated	
	\textlatin and fixed	
	\latinencoding. . . . .	1
	babel-greek-1.09h	
	General: Move breathing	
	composite commands to	
	textalpha. . . . .	23
	babel-greek-1.09i	
	\captionsgreek: Fix accent in	
	seename and alsoname. . . . .	13
	General: Fix accent in \seename	
	and \alsiname. . . . .	1
	Update check for Unicode fonts. . .	8
	babel-greek-1.09j	
	\textampersand: Fix ampersand	
	in math. . . . .	11
	babel-greek-1.10	
	\greeknumeral: PDF-string secure	
	implementation taken from	
	“hyperref” (thanks to Ulrike	
	Fischer). . . . .	17
	\greeknumeralNinety: Use	
	zig-zagy \textkoppa. This is	

what it looks in current Greek typography. . . . .	17	to keep encoding change local	11
General: Load <code>puenc-greek.def</code> from <code>greek-fontenc</code> if used with <code>hyperref</code> . . . . .	9	General: Use <code>\LdfInit</code> to perform initial checks. . . . .	6
Use TU with Xe/LuaTeX. . . . .	8	Moved the definition of <code>\atcatcode</code> right to the beginning . . . . .	1
babel-greek-1.11		Now use <code>\ldf@finish</code> to wrap up . . . . .	23
<code>\greeknumeral</code> : Configurable shapes for 6 and 90. 90 defaults to <code>\textqoppa</code> for ancient Greek. . . . .	17	Replaced <code>\undefined</code> with <code>\@undefined</code> and <code>\empty</code> with <code>\@empty</code> for consistency with L <sup>A</sup> T <sub>E</sub> X . . . . .	1
General: Save/restore previous font encoding instead of switching to <code>\latinencoding</code> when leaving Greek. . . . .	10	greek-1.0c	
babel-greek-1.12		<code>\greek@tilde</code> : Added command .	23
<code>\BabelGreekRestoreFontEncoding</code> : New macro. . . . .	10	greek-1.1	
<code>\EnsureStandardFontEncoding</code> : New TextCommand. . . . .	11	<code>\Grtoday</code> : Added macro <code>\Grtoday</code>	17
<code>\greek@Alph</code> : Save/restore previous expansion of <code>\alph</code> and <code>\Alph</code> with every switch to/from Greek. . . . .	20	greek-1.1a	
General: Declare char 159 expansion similar to the way it is done in <code>inputenc</code> to avoid “inputenc error”. . . . .	22	<code>\dategreek</code> : Fixed typo, <code>\Oktwbr'iou</code> instead of <code>\Oktobr'iou</code> . . . . .	17
Don't change uccodes if LaTeX is from 2022/06/01 or newer. .	22	<code>\greek@Alph</code> : removed two superfluous @'s which made <code>\@alph</code> undefined . . . . .	20
Don't use <code>\makeatother</code> in <code>\AtBeginDocument</code> . . . . .	9	greek-1.1b	
New language attribute <code>polytonic</code> (alias for <code>polutoniko</code> ). .	7	<code>\greek@tilde</code> : Made tilde expand to a tilde with <code>\catcode 12</code> . .	23
New modifiers <code>local-LGR-fixes</code> and <code>no-LGR-fixes</code> . . . . .	3	General: Added shorthand for <code>\char255</code> . . . . .	22
Only change uc/lccodes if LGR is the <code>\greekfontencoding</code> . . .	21	Added setting of <code>\uccodes</code> (after <code>\kdgreek.sty</code> ) . . . . .	22
Update and restructure documentation. . . . .	1	greek-1.1c	
Drop definition for <code>\SS</code> . . . . .	11	General: Added a couple of symbols, needed for <code>\greeknumeral</code> . . . . .	23
Remove <code>\textKoppa</code> and <code>\textmu</code> (in <code>greek-fontenc</code> since version 1.0). . . . .	23	fixed two typos . . . . .	21
Save previous font encoding in <code>\BabelGreekPreviousFontEncoding</code> . . . . . .	10	greek-1.1d	
		<code>\dategreek</code> : Macro <code>\gr@month</code> now produces the name of the month . . . . .	17
greek-1.0b		greek-1.1e	
<code>\lgrfont</code> : Added a level of braces		<code>\gr@month</code> : Macro added . . . . .	15
		General: Shorthand is changed. Active character is now <code>\char159</code> . . . . .	22
		Added caption name for proof	13
		Added lowercase code for <code>v</code> . . .	21
		Added uppercase code for special letter “ <code>v</code> ”. Uppercase code for accents is now <code>9f</code> , instead of <code>ff</code> . . . . .	22

Most symbols are removed and are now defined in package		
grsymb .....	23	
greek-1.2		
\gr@polutoniko@month: Added		
macro \datepolutonikogreek	16	
Added macro \gr@cl@month ..	16	
General: Added caption names for		
\polutonikogreek .....	14	
Added lowercase codes for		
“modern” greek .....	21	
Added uppercase codes for		
“modern” Greek. The old		
codes are now for “Polutoniko”		
Greek .....	22	
Classical Greek is now a dialect	1	
Definitions for “modern” Greek		
are now the definitions of		
“polutoniko” Greek .....	21	
greek-1.2a		
\dategreek: Use \edef to define		
\today .....	17	
General: Need shorthand to exist		
for monotonic Greek, not		
polytonik Greek .....	22	
filename lgrenc.def now		
lowercase .....	8	
greek-1.2b		
\dategreek: use \def instead of		
\edef .....	17	
General: Classical Greek is now		
called “Polutoniko” Greek.		
The previous name was at least		
misleading .....	1	
greek-1.2c		
General: Package grsymb has been		
eliminated because the CB		
fonts v2.0 do not include		
certain symbols and so the		
remaining symbol definitions		
have been moved here .....	23	
This version conforms to version		
2.0 of the CB fonts and		
consequently we added a few		
new symbol-producing		
commands .....	1	
greek-1.2e		
General: Moved redefinition of		
\@roman back to the language		
specific file .....	12	
greek-1.3a		
\gr@polutoniko@month: removed		
macro \datepolutonikogreek	16	
General: polutoniko is now an		
attribute to Greek, no longer a		
‘dialect’ .....	1	
greek-1.3d		
General: \@roman and \@Roman		
need to be added to		
\extrapolutonikogreek ...	12	
Fixed typo, bl’epe ep’ishc		
instead of bl’pe ep’ishc ...	13	
greek-1.3e		
General: \@roman and \@Roman		
need <i>not</i> be in		
\extrapolutonikogreek		
when they are already in		
\extrasgreek .....	12	
\extrasgreek and		
\extrapolutonikogreek		
should be complementary	21, 22	
greek-1.3f		
General: Added some code to make		
older documents work. ....	7	
greek-1.3g		
General:		
\noextrapolutonikogreek		
was missing. ....	7	
greek-1.3h		
\captionsgreek: Added		
\glossaryname .....	13	
\providehyphenmins: Now use		
\providehyphenmins to		
provide a default value .....	7	
greek-1.3i		
\captionsgreek: The final sigma		
in all names appears as ‘s’		
instead of ‘c’. ....	13	
General: uc code of ‘v’ is switched		
to V so that mixed text		
appears correctly in headers. .	22	
greek-1.3j		
General: Use the tilde as an alias		
for character 159 .....	22	
Don’t use the double caret		
notation here, because other		
languages might make the		
caret active. ....	22	

greek-1.3k	Support XeTeX/LuaTeX. . . . .	21
<code>\greek@tilde</code> : Make sure the character “ is not active during the definition of <code>\greek@tilde</code>	greek-1.5a	General: “extraspolutonikogreek” definition was missing with Xe/LuaTeX. Replaced non-printable literal character with mnotation. . . . .
<code>\lgrfont</code> : Added <code>\leavevmode</code> as was done with <code>\latintext</code> . . .		1
greek-1.4		6
<code>\greek@tilde</code> : Do not re-define the tilde accent macro: it works as expected with <code>lgrenc.def</code> from <code>greek-fontenc</code> .		23
General: <code>lgrenc.def</code> moved to the separate package ‘greek-fontenc’ . . . . .		8
Add <code>TextCompositeCommands</code> for “uppercase diacritics”. . . .		23
moved here from <code>lgrenc.def</code> because the definitions require the <code>\latintext</code> macro defined by Babel. . . . .		11
new maintainer . . . . .		1
greek-1.5		
<code>\textampersand</code> : Make <code>\&amp;</code> a <code>TextCommand</code> . . . . .		11
General: <code>\@roman</code> and <code>\@Roman</code> as <code>TextCommands</code> (BUG: this extended the expansion problem to all languages) . . .		12
bugfixes, change some symbol macros to aliases, LGR fixes via <code>DeclareTextCommand</code> instead of <code>extragreek/noextragreek</code> definitions, LICR macros in string definitions, LGR font encoding not used with XeTeX/LuaTeX. . . . .		1
change symbol macros to aliases enable use of “textcomp” characters for “textcopyright” and “textregistered” macros . .		11
LGR not used with XeTeX/LuaTeX. . . . .		10
LGR setup skipped with XeTeX/LuaTeX . . . . .		8
	greek-1.6	General: Apply a patch by Enrico Gregorio. Thanks to Claudio Beccari for testing and reporting. . . . .
		12
		fix <code>\@roman</code> and <code>\@Roman</code> redefinition (thanks to Enrico Gregorio and Claudio Beccari), load LICR macro definitions for Xe/LuaTeX. . . . .
		1
	greek-1.7	General: Do not load <code>euenc.def</code> with XeTeX/LuaTeX (too complicated to get it right). . . .
		8
		Do not load <code>euenc.def</code> with XeTeX/LuaTeX. Prevent re-loading <code>lgrenc.def</code> . . . . .
		1
	greek-1.7a	General: Remove spurious “fi”. . . .
		1
	greek-1.7b	General: Correct upcasing of babel strings with Xe/LuaTeX. . . . .
		1
	greek-1.8	General: Renamed to ‘babel-greek’. .
		1
	greekfdd-2.2c	General: Fixed typos, <code>\textrademark</code> misses a ‘t’, <code>\copyright</code> should be <code>\textcopyright</code> . . . . .
		11
	greekfdd-2.2d	General: removed redefinition of <code>\&amp;</code>
		11