

# Babel support for the Greek language

Apostolos Syropoulos, Günter Milde

August 18, 2023

Babel-greek is a contributed package providing support for the Greek language and script via the `babel` system. See `babel-greek-doc` 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 attributes . . . . .	7
3.2.1	<code>polutoniko</code> . . . . .	7
3.2.2	<code>polytonic</code> . . . . .	8
3.2.3	<code>ancient</code> . . . . .	8
3.2.4	<code>keep-semicolon</code> . . . . .	8
3.3	Report unsupported modifiers . . . . .	9
3.4	Font setup . . . . .	9
3.4.1	Greek font encoding . . . . .	9
3.4.2	Ensure loading of Greek font encoding definitions. . . . .	10
3.4.3	Font encoding switches . . . . .	10
3.4.4	Additional commands for the LGR font encoding . . . . .	11
3.4.5	LGR workarounds . . . . .	12

---

<sup>1</sup>The file described in this section has version number 1.14 and was last revised on 2023/08/18. The original author is Apostolos Syropoulos, code from `kdgreek.sty` by David Kastrup was used.

3.5	Definitions for the Greek language	13
3.5.1	Auto-strings for Greek	14
3.5.2	Auto-strings for polytonic Greek	15
3.5.3	Auto-strings for ancient Greek	15
3.5.4	Date specification	16
3.5.5	Greek numerals	18
3.6	Character codes	21
3.7	MakeUppercase fixes	23
3.8	Symbol name aliases	24

## 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 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 standard `\DeclareFontFamilySubstitution` macro can be used to set up Greek supplements for other TeX font families (like Times or Palatino).

Unicode fonts (used with XeTeX or LuaTeX) 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 XeTeX, hyphenation patterns must be pre-loaded in the format file. This is a limitation by TeX, common to all languages. The LuaTeX 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.

```
\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.

---

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

## 2.1 Language attributes

The attributes `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.

The `keep-semicolon` language attribute (new in babel-greek 1.13) ensures that a SEMICOLON character (;) can be used as input for the similar looking Greek question mark (*erotimatiko*). By default, the LGR font encoding uses the QUESTION MARK (?) as input for the *erotimatiko* and maps the SEMICOLON to an *ano teleia* (·).

## 2.2 Modifiers

All language attributes may also be used as modifiers, e.g.

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

In addition, there are modifiers that cannot be set with `\languageattribute`.

The `local-LGR-fixes` modifier restricts the re-definitions in section 3.4.5 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).

The `local-MakeUppercase-fixes` modifier skips the mapping of standard accents to “capital” accents in section 3.7.

These modifiers are provisional, naming and behaviour may change.

## 2.3 Language hooks

`\extrasgreek`  
`\noextrasgreek`

The macro `\extrasgreek` is called by `babel` on every switch of the active language to Greek. The macro `\noextrasgreek` is called when switching away from Greek. Package and document authors can add setup and tear-down code to

---

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

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.

`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 3.4).

## 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` and special handling for Latin letters and some symbols (consider using the `keep-semicolon` attribute).  
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
<code>\latintext</code>	LGR. Both declarations do not change the active language. <code>\latintext</code> (defined by the Babel core, deprecated since March 2014) can be
<code>\ensuregreek</code>	used to switch back to an encoding supporting the Latin script. The function <code>\ensuregreek</code> takes one argument which is typeset using a font
<code>\lgrfont</code>	encoding supporting the Greek script. It only switches the font encoding if required
<code>\ensureascii</code>	(i.e. if the current font encoding does not support Greek letters and symbols). The function <code>\lgrfont</code> <sup>5</sup> switches to the non-standard Greek 8-bit font encoding
	LGR. Hint: Use <code>\lgrfont</code> , if you want to use the <i>Latin transliteration</i> input
	method and <code>\ensuregreek</code> else.
	The Babel core defines <code>\ensureascii</code> that typesets its argument using an

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

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

ASCII-compatible “standard text font encoding”. It is the recommended way for text parts requiring Latin letters but no language switch.

## 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. `\Greeknumeral` 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 *keraiia*, 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 keraiia*, 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 `\Greeknumeral` 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`

<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.

does the same for `\Greeknatural`. The default values are `\textsigma` and `\textStigma`.

`\greeknumeralninet`

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

`\greeknumeralNinety`

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}
```

Set up the Babel shorthands feature. It is used later to insert literal ~ characters with polytonic Greek and LGR and to prevent LGR converting a literal semicolon ; to an *ano teleia* ·.

```
4 \addto\extrasgreek{\languageshorthands{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.

```
5 \ifx\l@greek\@undefined
6 \nopatterns{greek}
7 \adddialect\l@greek 0
8 \fi
9 \ifx\l@monogreek\@undefined
10 \nopatterns{greek}
11 \adddialect\l@monogreek 0
12 \fi
13 \ifx\l@ancientgreek\@undefined
14 \nopatterns{greek}
15 \adddialect\l@ancientgreek 0
16 \fi
```

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

```

17 \newcount\bbl@monogreek \bbl@monogreek=\l@monogreek
18 \newcount\bbl@polygreek \bbl@polygreek=\l@greek
19 \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

```

```

20 \addto\extragreek{\let\l@greek=\bbl@monogreek}
21 \addto\extraspolutonikogreek{\l@greek=\bbl@polygreek}
22 \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.

```
23 \providehyphenmins{\CurrentOption}{\@ne\@ne}
```

## 3.2 Language attributes

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.

We use an auxiliary function for the setup part used with several language attributes: Add the expansion of `\extraspolutonikogreek` to `\extragreek` to set up support for multi-accented characters and hyphenation patterns for the polytonic orthography and use polytonic spelling for auto-strings (captions and month names). More code is added later (cf. section 3.6).

```

24 \def\bbl@greek@setup@polytonic{%
25   \expandafter\addto\expandafter\extragreek
26   \expandafter{\extraspolutonikogreek}%
27   \let\captionsgreek\captionspolutonikogreek
28   \let\gr@month\gr@polutoniko@month
29 }

```

Now declare the option. For backwards compatibility, modern Greek with “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.<sup>11</sup> We also define aliases to allow language switching commands using the language name `polutonikogreek`:

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

<sup>11</sup>Use of more than one Greek orthographies in one document is possible with `\babelprovide`. However, there are side-effects. See the example in `test-greek.tex`.

```

31 \bbl@greek@setup@polytonic
32 \let\l@polutonikogreek\l@greek
33 \let\datepolutonikogreek\dategreek
34 \let\extrapolutonikogreek\extrasgreek
35 \let\noextrapolutonikogreek\noextrasgreek
36 }

```

### 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.

```

37 \bbl@declare@ttribute{greek}{polytonic}{%
38 \bbl@greek@setup@polytonic
39 }

```

### 3.2.3 ancient

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

```

40 \bbl@declare@ttribute{greek}{ancient}{%
41 \bbl@greek@setup@polytonic
42 \expandafter\addto\expandafter\extrasgreek
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

```

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).

```

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

```

### 3.2.4 keep-semicolon

The LGR font encoding uses the Latin question mark as input for the Greek question mark (*erotimatiko*) and maps the semicolon to a middle dot (*ano teleia*). As a result, Unicode-encoded texts that use the semicolon (;) as *erotimatiko* end up with an *ano teleia* (·) in its place!

With the `keep-semicolon` language attribute, 003B SEMICOLON is made active and inserts an *erotimatiko* also with LGR encoded fonts:

```

48 \bbl@declare@ttribute{greek}{keep-semicolon}{%
49 \def\bbl@tempa{LGR}
50 \ifx\greekfontencoding\bbl@tempa
51 \ProvideTextCommandDefault{\textsemicolon}{;}

```



```

52 \ProvideTextCommand{\textsemicolon}{LGR}{\texterotimatiko}
53 \initiate@active@char{;}
54 \addto\extragreek{\bbl@activate{;}}
55 \addto\noextragreek{\bbl@deactivate{;}}
56 \declare@shorthand@greek{;}{\TextOrMath{\textsemicolon}{;}}
57 \fi
58 }

```

### 3.3 Report unsupported modifiers

Test for unsupported (or misspelled) [modifiers](#) (code contributed by Javier Bezos).

```

59 \def\bbl@greek@modifiers{,%
60 polutoniko,polytonic,ancient,keep-semicolon,%
61 local-LGR-fixes,no-LGR-fixes,local-MakeUppercase-fixes,}
62 \ifx\BabelModifiers\relax\else
63 \bbl@foreach\BabelModifiers{%
64 \expandtwoargs\in@{,#1,}{\bbl@greek@modifiers}
65 \ifin@\else
66 \bbl@warning
67 {Unknown/misspelled modifier '#1' in '\CurrentOption'.
68 See "babel-greek.pdf" for valid modifiers.}
69 \fi}%
70 \fi
71

```

### 3.4 Font setup

#### 3.4.1 Greek font encoding

`\greekfontencoding` The macro `\greekfontencoding` holds the name of the font encoding<sup>12</sup> used to ensure support of the Greek script. The default is LGR for 8-bit TeX and TU for Xe/LuaTeX.<sup>13</sup> It can be overridden defining `\greekfontencoding` with a custom value before loading `babel`.

Also store the name of the *encoding definition file*<sup>14</sup> with the extended Greek setup for the Greek font encoding.

```

72 \ifdefined\UnicodeEncodingName % set by XeTeX/LuaTeX
73 \providecommand*\greekfontencoding{\UnicodeEncodingName}
74 \providecommand*\bbl@greek@fontencdef{tuenc-greek}
75 \else
76 \providecommand*\greekfontencoding{LGR}
77 \providecommand*\bbl@greek@fontencdef{lgrenc}
78 % TODO the more generic version fails :(
79 % \edef\bbl@greek@fontencdef{\lowercase{\greekfontencoding}enc}
80 \fi

```

<sup>12</sup>cf. [encguide.pdf](#)

<sup>13</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>14</sup>see [ftguide.pdf](#)

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

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

```
81 \ifl@aded{def}{\bbl@greek@fontencdef}{}
82 {% else
83   \InputIfFileExists{\bbl@greek@fontencdef .def}{}
84   {% else
85     \bbl@error{Font support for the Greek script missing.\\
86               babel-greek can't typeset Greek.\\
87               Install the "greek-fontenc" package\\
88               or use XeTeX/LuaTeX with polyglossia.}
89     {I can't find the \bbl@greek@fontencdef .def file
90      for the Greek fonts (encoding \greekfontencoding)}
91   \@@end
92 }
93 }
```

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.

```
94 \AtBeginDocument{%
95   \ifl@aded{def}{puenc}%
96   {\@ifl@aded{def}{puenc-greek}
97    {}%
98    {\edef\RestoreAtCatcode{\catcode'@=\the\catcode'@\relax}%
99     \makeatletter
100    \InputIfFileExists{puenc-greek.def}%
101     {}%
102     {\bbl@warning{I cannot find the "puenc" Greek fixes
103                  from "greek-fontenc".}%
104     }%
105     \RestoreAtCatcode
106    }%
107   }% end "puenc.def loaded" branch
108   {}% empty "puenc.def not loaded" branch
109 }
```

### 3.4.3 Font encoding switches

`\greekscript` The `\TextCommand`<sup>15</sup> `\greekscript` is a declaration that switches the font encoding to `\greekfontencoding`. The extended Greek font encoding definitions 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.

```
110 \ProvideTextCommandDefault{\greekscript}{%
111   \fontencoding{\greekfontencoding}\selectfont
```

<sup>15</sup>See [fntguide.pdf](#) for more info about *TextCommands*.

```
112 \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.

```
113 \ProvideTextCommandDefault{\ensuregreek}[1]{%
114 \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.

```
115 \def\BabelGreekRestoreFontEncoding{%
116 \ifx\cf@encoding\BabelGreekPreviousFontEncoding
117 \else
118 \let\encodingdefault\BabelGreekPreviousFontEncoding
119 \fontencoding{\encodingdefault}\selectfont
120 \fi
121 }
```

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:

```
122 \addto\extrasgreek{%
123 \let\BabelGreekPreviousFontEncoding\cf@encoding
124 \greekscript}
125 \addto\noextrasgreek{\BabelGreekRestoreFontEncoding}
```

### 3.4.4 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`:

```
126 \@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 3.4.3.

```
127 \DeclareRobustCommand{\greektext}{%
128 \fontencoding{LGR}\selectfont
129 \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.

```
130 \DeclareTextFontCommand{\lgrfont}{\greektext}
131 \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.)

```
132 \def\outlfamily{\usefont{LGR}{cmro}{m}{n}}
133 \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`).

```
134 \ProvideTextCommand{\textcopyright}{LGR}{\ensureascii{\textcopyright}}
135 \ProvideTextCommand{\textregistered}{LGR}{\ensureascii{%
136                                     \textregistered}}
137 \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.

```
138 \let\bbl@greek@original@amp&
139 \ProvideTextCommandDefault{\textampersand}{\bbl@greek@original@amp}
140 \ProvideTextCommand{\textampersand}{LGR}{%
141     \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.

```
142 \ProvideTextCommandDefault{\EnsureStandardFontEncoding}{\@firstofone}
143 \ProvideTextCommand{\EnsureStandardFontEncoding}{LGR}[1]{%
144                                     \ensureascii{#1}}
145 \AtBeginDocument{\@ifpackageloaded{hyperref}
146                 {\pdfstringdefDisableCommands{%
147                 \let\EnsureStandardFontEncoding\@firstofone}}
148                 {}}
```

End the LGR additions block:

```
149 }{}
```

### 3.4.5 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 [3.4.1](#)).

As an emergency measure, the local-LGR-fixes or no-LGR-fixes [modifiers](#) 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>16</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.

```

150 \def\bbl@tempa{LGR}
151 \ifx\greekfontencoding\bbl@tempa
152   \def\bbl@greek@roman#1{\expandafter\EnsureStandardFontEncoding%
153     \expandafter{\romannumeral#1}}
154   \def\bbl@greek@Roman#1{\expandafter\EnsureStandardFontEncoding%
155     \expandafter{\expandafter\@slowromancap\romannumeral#1@}}
156   \DeclareRobustCommand{\bbl@greek@ampersand}{%
157     \ifmode\bbl@greek@original@amp\else\textampersand\fi}
158   \bbl@xin@{,no-LGR-fixes,}{,\BabelModifiers,}%
159   \ifin@
160     % skip re-definitions
161   \else
162     \bbl@xin@{,local-LGR-fixes,}{,\BabelModifiers,}%
163     \ifin@
164       \addto\extrasgreek{%
165         \babel@save\@roman
166         \babel@save\@Roman
167         \let\@roman\bbl@greek@roman
168         \let\@Roman\bbl@greek@Roman
169         \babel@save\&%
170         \let\&\bbl@greek@ampersand%
171       }
172     \else
173       \let\@roman\bbl@greek@roman
174       \let\@Roman\bbl@greek@Roman
175       \let\&\bbl@greek@ampersand
176     \fi
177   \fi
178 \fi

```

### 3.5 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.

---

<sup>16</sup>For example, Roman page numbers are generated at “unpredictable” positions and can move to the ToC, (hyper)references, or an index.

### 3.5.1 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.

```

179 \addto\captionsgreek{%
180   \def\prefacename{\textPi\extrho\acctonos\textomicron\textlambda
181     \textomicron\textgamma\textomicron\textfinalsigma}%
182   \def\refname{\textAlpha\textnu\textalpha
183     \textphi\textomicron\extrho\acctonos\textepsilon\textfinalsigma}%
184   \def\abstractname{\textPi\textepsilon\extrho\acctonos\textiota
185     \textlambda\texteta\textpsi\texteta}%
186   \def\bibname{\textBeta\textiota\textbeta\textlambda\textiota
187     \textomicron\textgamma\extrho\textalpha\textphi\acctonos
188     \textiota\textalpha}%
189   \def\chaptername{\textKappa\textepsilon\textphi\acctonos\textalpha
190     \textlambda\textalpha\textiota\textomicron}%
191   \def\appendixname{\textPi\textalpha\extrho\acctonos\textalpha\extrho
192     \texttau\texteta\textmu\textalpha}%
193   \def\contentsname{\textPi\textepsilon\extrho\textiota
194     \textepsilon\textchi\acctonos\textomicron\textmu\textepsilon
195     \textnu\textalpha}%
196   \def\listfigurename{\textKappa\textalpha\texttau\acctonos\textalpha
197     \textlambda\textomicron\textgamma\textomicron\textfinalsigma}
198     \textSigma\textchi\texteta\textmu\acctonos\textalpha\texttau
199     \textomega\textnu}%
200   \def\listtablename{\textKappa\textalpha\texttau\acctonos\textalpha
201     \textlambda\textomicron\textgamma\textomicron\textfinalsigma}
202     \textPi\textiota\textnu\acctonos\textalpha\textkappa\textomega
203     \textnu}%
204   \def\indexname{\textEpsilon\textupsilon\extrho\textepsilon
205     \texttau\acctonos\texteta\extrho\textiota\textomicron}%
206   \def\figurename{\textSigma\textchi\acctonos\texteta\textmu\textalpha}%
207   \def\tablename{\textPi\acctonos\textiota\textnu\textalpha
208     \textkappa\textalpha\textfinalsigma}%
209   \def\partname{\textMu\acctonos\textepsilon\extrho\textomicron
210     \textfinalsigma}%
211   \def\enclname{\textSigma\textupsilon\textnu\texteta\textmu
212     \textmu\acctonos\textepsilon\textnu\textalpha}%
213   \def\ccname{\textKappa\textomicron\textiota\textnu\textomicron
214     \textpi\textomicron\acctonos\textiota\texteta\textsigma\texteta}%
215   \def\headtoname{\textPi\extrho\textomicron\textfinalsigma}%
216   \def\pagename{\textSigma\textepsilon\textlambda\acctonos\textiota
217     \textdelta\textalpha}%
218   \def\seename{\textbeta\textlambda\acctonos\textepsilon\textpi
219     \textepsilon}%
220   \def\alsoname{\textbeta\textlambda\acctonos\textepsilon\textpi
221     \textepsilon} \textepsilon\textpi\acctonos\textiota\textsigma
222     \texteta\textfinalsigma}%
223   \def\proofname{\textAlpha\textpi\acctonos\textomicron
224     \textdelta\textepsilon\textiota\textxi\texteta}%

```

```

225 \def\glossaryname{\textGamma\textlambda\textomega\textsigma
226 \textsigma\acconos\textalpha\textrho\textiota}%
227 }

```

### 3.5.2 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.

```

228 \let\captionspolutonikogreek\captionsgreek
229 \addto\captionspolutonikogreek{%
230 \def\refname{\accpsili\textAlpha\textnu\textalpha
231 \textphi\textomicron\textrho\accvaria\textepsilon\textfinalsigma}%
232 \def\indexname{\textEpsilon\accdasia\textupsilon\textrho\textepsilon
233 \texttau\acconos\texteta\textrho\textiota\textomicron}%
234 \def\figurename{\textSigma\textchi\accperispomeni\texteta\textmu
235 \textalpha}%
236 \def\headtoname{\textPi\textrho\accvaria\textomicron\textfinalsigma}%
237 \def\alsoname{\textbeta\textlambda\acconos\textepsilon\textpi
238 \textepsilon} \accpsili\textepsilon\textpi\acconos\textiota
239 \textsigma\texteta\textfinalsigma}%
240 \def\proofname{\accpsili\textAlpha\textpi\acconos\textomicron
241 \textdelta\textepsilon\textiota\textxi\texteta}%
242 }

```

### 3.5.3 Auto-strings for ancient Greek

`\captionسانcientgreek` 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.

```

243 \let\captionسانcientgreek\captionsgreek
244 \addto\captionسانcientgreek{%
245 \def\prefacename{\textPi\textrho\textomicron\textomicron
246 \acconos\textiota\textmu\textiota\textomicron\textnu}%
247 \def\refname{\accpsili\textAlpha\textnu\textalpha\textphi\textomicron
248 \textrho\textalpha\accvaria\textiota}%
249 \def\abstractname{\textPi\textepsilon\textrho\acconos\textiota
250 \textlambda\texteta\textpsi\textiota\textvarsigma}%
251 \def\bibname{\textBeta\textiota\textbeta\textlambda\textiota
252 \textomicron\textgamma\textrho\textalpha\textphi
253 \acconos\textiota\textalpha}%
254 \def\chaptername{\textKappa\textepsilon\textphi\acconos\textalpha
255 \textlambda\textalpha\textiota\textomicron\textnu}%
256 \def\appendixname{\textPi\textalpha\textrho\acconos\textalpha
257 \textrho\texttau\texteta\textmu\textalpha}%
258 \def\contentsname{\textPi\textepsilon\textrho\textiota\textepsilon
259 \textchi\acconos\textomicron\textmu\textepsilon\textnu\textalpha}%
260 \def\listfigurename{\textKappa\textalpha\texttau\acconos\textalpha
261 \textlambda\textomicron\textgamma\textomicron\textvarsigma}

```

```

262 \textsigma\textchi\texteta\textmu\acconos\textalpha\texttau
263 \textomega\textnu}%
264 \def\listtablename{\textKappa\textalpha\texttau\acconos\textalpha
265 \textlambda\textomicron\textgamma\textomicron\textvarsigma{}}
266 \textpi\textiota\textnu\acconos\textalpha\textkappa
267 \textomega\textnu}%
268 \def\indexname{\textEpsilon\accdasia\textupsilon\textrho\textepsilon
269 \texttau\acconos\texteta\textrho\textiota\textomicron\textnu}%
270 \def\figurename{\textSigma\textchi\accperispomeni\texteta\textmu
271 \textalpha}%
272 \def\tablename{\textPi\acconos\textiota\textnu\textalpha\textxi}%
273 \def\partname{\textMu\acconos\textepsilon\textrho\textomicron
274 \textvarsigma}%
275 \def\enclname{\textSigma\textupsilon\textnu\texteta\textmu\textmu
276 \acconos\textepsilon\textnu\textomega\textvarsigma}%
277 \def\ccname{\textKappa\textomicron\textiota\textnu\textomicron\textpi
278 \textomicron\acconos\textiota\texteta\textsigma\textiota
279 \textvarsigma}%
280 \def\headtoname{\textPi\textrho\accvaria\textomicron\textvarsigma}%
281 \def\pagename{\textSigma\textepsilon\textlambda\accvaria\textiota
282 \textvarsigma}%
283 \def\seename{\accdasiaoxia\textomicron\textrho\textalpha}%
284 \def\alsoname{\accdasiaoxia\textomicron\textrho\textalpha{}}
285 \accdasia\textomega\textsigma\textalpha\acconos\textupsilon
286 \texttau\textomega\textvarsigma}%
287 \def\proofname{\accpsili\textAlpha\textpi\acconos\textomicron
288 \textdelta\textepsilon\textiota\textxi\textiota\textvarsigma}%
289 \def\glossaryname{\textGamma\textlambda\textomega\textsigma\textsigma
290 \acconos\textalpha\textrho\textiota\textomicron\textnu}%
291 }

```

### 3.5.4 Date specification

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

```

292 \def\gr@month{%
293 \ifcase\month\or
294 \textIota\textalpha\textnu\textomicron\textupsilon\textalpha
295 \textrho\acconos\textiota\textomicron\textupsilon \or
296 \textPhi\textepsilon\textbeta\textrho\textomicron\textupsilon
297 \textalpha\textrho\acconos\textiota\textomicron\textupsilon \or
298 \textMu\textalpha\textrho\texttau\acconos\textiota\textomicron
299 \textupsilon \or
300 \textAlpha\textpi\textrho\textiota\textlambda\acconos\textiota
301 \textomicron\textupsilon \or
302 \textMu\textalpha\' "\textiota\textomicron\textupsilon \or
303 \textIota\textomicron\textupsilon\textnu\acconos\textiota
304 \textomicron\textupsilon \or
305 \textIota\textomicron\textupsilon\textlambda\acconos\textiota

```



```

306     \textomicron\textupsilon \or
307     \textAlpha\textupsilon\textgamma\textomicron\acctonos\textupsilon
308     \textsigma\texttau\textomicron\textupsilon \or
309     \textSigma\textepsilon\textpi\texttau\textepsilon\textmu
310     \textbeta\textrho\acctonos\textiota\textomicron\textupsilon \or
311     \textOmicron\textkappa\texttau\textomega\textbeta
312     \textrho\acctonos\textiota\textomicron\textupsilon \or
313     \textNu\textomicron\textepsilon\textmu\textbeta
314     \textrho\acctonos\textiota\textomicron\textupsilon \or
315     \textDelta\textepsilon\textkappa\textepsilon\textmu\textbeta
316     \textrho\acctonos\textiota\textomicron\textupsilon
317 \fi
318 }

```

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

```

319 \def\gr@polutoniko@month{%
320   \ifcase\month\or
321     \accpsili\textIota\textalpha\textnu\textomicron\textupsilon
322     \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
323     \textPhi\textepsilon\textbeta\textrho\textomicron\textupsilon
324     \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
325     \textMu\textalpha\textrho\texttau\acctonos\textiota\textomicron
326     \textupsilon \or
327     \accpsili\textAlpha\textpi\textrho\textiota\textlambda
328     \acctonos\textiota\textomicron\textupsilon \or
329     \textMu\textalpha\accdialytikatonos\textiota\textomicron
330     \textupsilon \or
331     \accpsili\textIota\textomicron\textupsilon\textnu
332     \acctonos\textiota\textomicron\textupsilon \or
333     \accpsili\textIota\textomicron\textupsilon\textlambda
334     \acctonos\textiota\textomicron\textupsilon \or
335     \textAlpha\accpsili\textupsilon\textgamma\textomicron\acctonos
336     \textupsilon\textsigma\texttau\textomicron\textupsilon \or
337     \textSigma\textepsilon\textpi\texttau\textepsilon\textmu\textbeta
338     \textrho\acctonos\textiota\textomicron\textupsilon \or
339     \accpsili\textOmicron\textkappa\texttau\textomega\textbeta
340     \textrho\acctonos\textiota\textomicron\textupsilon \or
341     \textNu\textomicron\textepsilon\textmu\textbeta
342     \textrho\acctonos\textiota\textomicron\textupsilon \or
343     \textDelta\textepsilon\textkappa\textepsilon\textmu
344     \textbeta\textrho\acctonos\textiota\textomicron\textupsilon
345 \fi
346 }

```

`\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`.

```

347 \def\dategreek{%
348   \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. (The [teubner](#) package defines a matching lowercase version `\grtoday`.)

```
349 \def\Grtoday{%
350   \expandafter\Greeknatural\expandafter{\the\day}\space
351   \gr@polutoniko@month \space
352   \expandafter\Greeknatural\expandafter{\the\year}}
```

### 3.5.5 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.  
`\greeknumeraln ninety` We define placeholders that allow configuration by the user or a package.

```
\greeknumeralN ninety 353 \providecommand*\greeknumeralsix{\textstigma}
354 \providecommand*\greeknumeralSix{\textStigma}
355 \providecommand*\greeknumeraln ninety{\textkoppa}
356 \providecommand*\greeknumeralN ninety{\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).

```
357 \def\greeknumeral#1{%
358   {\greekscript
359   \bbl@greek@Greeknatural@firstoftwo{#1}}%
360 }
```

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

```
361 \def\Greeknatural#1{%
362   {\greekscript
363   \bbl@greek@Greeknatural@secondoftwo{#1}}%
364 }
```

`\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).

```
365 \def\bbl@greek@ill@value#1{%
366   \PackageWarningNoLine{babel}{Illegal value (#1) for greeknumeral}%
367   \@arabic{#1}%
368 }
```

`\bbl@greek@Greeknatural` The auxiliary macros provide the actual conversion. They are taken from [hyperref](#) as well.

```
\bbl@greek@GreeknaturalI 369 \def\bbl@greek@Greeknatural#1#2{%
\bbl@greek@GreeknaturalII 370   \ifnum#2<\@ne
\bbl@greek@GreeknaturalIII 371   \bbl@greek@ill@value{#2}%
\bbl@greek@GreeknaturalIV
\bbl@greek@GreeknaturalV
\bbl@greek@GreeknaturalVI
```

```

372     \else
373         \ifnum#2<1000000 %
374             \bbl@greek@@GreekNum#1{#2}%
375         \else
376             \bbl@greek@i11@value{#2}%
377         \fi
378     \fi
379 }
380 \def\bbl@greek@@GreekNum#1#2{%
381     \ifnum#2<\@m
382         \ifnum#2<10 %
383             \expandafter\bbl@greek@GreekNumI
384             \expandafter\@gobble\expandafter#1\number#2%
385         \else
386             \ifnum#2<100 %
387                 \expandafter\bbl@greek@GreekNumII
388                 \expandafter\@gobble\expandafter#1\number#2%
389             \else
390                 \expandafter\bbl@greek@GreekNumIII
391                 \expandafter\@gobble\expandafter#1\number#2%
392             \fi
393         \fi
394         \ifnum#2>\z@
395             \textnumeralsigngreek
396         \fi
397     \else
398         \ifnum#2<\@M
399             \expandafter\bbl@greek@GreekNumIV\expandafter#1\number#2%
400         \else
401             \ifnum#2<100000 %
402                 \expandafter\bbl@greek@GreekNumV\expandafter#1\number#2%
403             \else
404                 \expandafter\bbl@greek@GreekNumVI\expandafter#1\number#2%
405             \fi
406         \fi
407     \fi
408 }
409 \def\bbl@greek@GreekNumI#1#2#3{%
410     #1{%
411         \ifnum#3>\z@
412             \textnumeralsignlowergreek
413         \fi
414     }%
415     \expandafter#2%
416     \ifcase#3 %
417         {}{}%
418         \or\textalpha\textAlpha
419         \or\textbeta\textBeta
420         \or\textgamma\textGamma
421         \or\textdelta\textDelta

```

```

422 \or\textepsilon\textEpsilon
423 \or\greeknumeralsix\greeknumeralSix % stigma or digamma
424 \or\textzeta\textZeta
425 \or\texteta\textEta
426 \or\texttheta\textTheta
427 \else
428   {}{}%
429 \fi
430 }
431 \def\bbl@greek@GreekNumII#1#2#3#4{%
432   #1{%
433     \ifnum#3>\z@
434       \textnumeralsignlowergreek
435       \fi
436   }%
437 \expandafter#2%
438 \ifcase#3 %
439   {}{}%
440 \or\textiota\textIota
441 \or\textkappa\textKappa
442 \or\textlambda\textLambda
443 \or\textmugreek\textMu
444 \or\textnu\textNu
445 \or\textxi\textXi
446 \or\textomicron\textOmicron
447 \or\textpi\textPi
448 \or\greeknumeralninty\greeknumeralNinty % koppa or qoppa
449 \else
450   {}{}%
451 \fi
452 \bbl@greek@GreekNumI#1#2#4%
453 }
454 \def\bbl@greek@GreekNumIII#1#2#3#4#5{%
455   #1{%
456     \ifnum#3>\z@
457       \textnumeralsignlowergreek
458       \fi
459   }%
460 \expandafter#2%
461 \ifcase#3 %
462   {}{}%
463 \or\textrho\textRho
464 \or\textsigma\textSigma
465 \or\texttau\textTau
466 \or\textupsilon\textUpsilon
467 \or\textphi\textPhi
468 \or\textchi\textChi
469 \or\textpsi\textPsi
470 \or\textomega\textOmega
471 \or\textsampigreek\textSampigreek

```

```

472 \else
473   {}{}%
474 \fi
475 \bbl@greek@GreekNumII#1#2#4#5%
476 }
477 \def\bbl@greek@GreekNumIV#1#2#3#4#5{%
478 \bbl@greek@GreekNumI\@firstofone#1#2%
479 \bbl@greek@@GreekNum#1{#3#4#5}%
480 }
481 \def\bbl@greek@GreekNumV#1#2#3#4#5#6{%
482 \bbl@greek@GreekNumII\@firstofone#1#2#3%
483 \bbl@greek@@GreekNum#1{#4#5#6}%
484 }
485 \def\bbl@greek@GreekNumVI#1#2#3#4#5#6#7{%
486 \bbl@greek@GreekNumIII\@firstofone#1#2#3#4%
487 \bbl@greek@@GreekNum#1{#5#6#7}%
488 }

```

`\greek@alph` and `\greek@Alph` In the previous release of this language definition file the commands `\greek@aplh` 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 `\expandafters` are needed in order to make sure the table of contents will be correct, e.g., when we have appendixes.

```

489 \def\greek@alph#1{\expandafter\greeknumeral\expandafter{\the#1}}
490 \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>17</sup> instead of the Latin alphabet<sup>18</sup> in Greek text parts.

```

491 \addto\extrasgreek{%
492 \babel@save\@alph
493 \babel@save\@Alph
494 \let\@alph\greek@alph
495 \let\@Alph\greek@Alph
496 }

```

### 3.6 Character codes

In order to get correct hyphenation we need to set the lower case code for all characters that can be part of a word.

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 `"`.

<sup>17</sup>cf. section 3.5.5

<sup>18</sup>Eventually interpreted as Latin transliteration and converted to Greek letters in a “strange” order.

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.

```

497 \def\bb1@tempa{LGR}
498 \ifx\greekfontencoding\bb1@tempa
499   \addto\extrasgreek{%
500     \babel@savevariable{\lccode'v}\lccode'v='v%
501     \babel@savevariable{\lccode'\'}\lccode'\}'='\'%
502     \babel@savevariable{\lccode'\'}\lccode'\}'='\'%
503   }
504   \addto\extraspolutonikogreek{%
505     % \l@greek=\bb1@polygreek
506     \babel@savevariable{\lccode'\<'}\lccode'\<='<%
507     \babel@savevariable{\lccode'\>'}\lccode'\>='>%
508     \babel@savevariable{\lccode'\~'}\lccode'\~='~%
509     \babel@savevariable{\lccode'\|'}\lccode'\|='|%
510     \babel@savevariable{\lccode'\'}\lccode'\}'='\'%
511   }

```

Also set the lc code for the precomposed characters in the upper half of the code table. 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 also in monotonic spelling.

```

512   \addto\extrasgreek{%
513     % ‘high bit characters’: set in a loop and correct exceptions
514     \@tempcnta=128%
515     \@whilenum\@tempcnta<253\do{%
516       \expandafter\babel@savevariable\expandafter{%
517         \expandafter\lccode\the\@tempcnta}%
518       \lccode\@tempcnta=\@tempcnta
519       \advance\@tempcnta\@ne
520     }%
521     % Fix non-word characters:
522     \lccode151=0%
523     \lccode155=0%
524     \lccode159=0%
525     \lccode199=0%
526     % Fix capital letters:
527     \lccode195=147% GREEK LETTER DIGAMMA
528     \lccode219=240% GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
529     \lccode223=244% GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
530   }

```

`\bb1@greek@tilde` By default, the tilde produces an unbreakable space in text mode. For the variants “polutoniko” and “ancient”, we change its meaning to allow using `~` in the Latin transliteration of characters with perispomeni and in composite diacritics.

```

531   \DeclareTextSymbol{\bb1@greek@tilde}{LGR}{126}

```

```

532 \addto\bb1@greek@setup@polytonic{
533   \declare@shorthand@greek}{~}{\bb1@greek@tilde}
534 }
535 \fi % End of LGR-specific code.

```

### 3.7 MakeUppercase fixes

In Greek typographical praxis, letters drop accents (except dialytika) and breathings in UPPERCASE. This is not cared for by the Unicode standard. For Unicode literals, `\MakeUppercase` implements locale-specific corrections.<sup>19</sup>

To fix the behaviour of the 2022 `\MakeUppercase` implementation with standard accent macros, we define and use “capital” accent macros. Font-encoding specific definitions for the “capital” accent macros in `greek-fontenc`  $\geq 2.4$  suppress them on Greek letters. The `local-MakeUppercase-fixes` modifier can be used to skip this step (cf. section 2.2).

```

536 \providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion} % new in 2020
537 \bb1@xin@{,local-MakeUppercase-fixes,}{,\BabelModifiers,}%
538 \ifin@
539 \else
540   \IfFormatAtLeastTF{2022/06/01}
541   {\ProvideTextCommandDefault{\accACUTE}{\@tabacckludge'}}
542   \ProvideTextCommandDefault{\accGRAVE}{\@tabacckludge'}
543   \ProvideTextCommandDefault{\accTILDE}{\@tabacckludge~}
544   \ProvideTextCommandDefault{\accDIAERESIS}{\@tabacckludge"}
545   \addto\@uclclist{\'\accACUTE \'\accGRAVE \'\accTILDE \'\accDIAERESIS}%
546   }
547   {}
548 \fi

```

Drop diacritics also with “input ligatures” defined in LGR fonts:

Since 2023/06, we can set up character mappings to be used on the argument of `\MakeUppercase`.<sup>20</sup> The optional “locale” argument [e1] restricts the mapping to Greek text parts.

```

549 \def\bb1@tempa{LGR}
550 \ifx\greekfontencoding\bb1@tempa
551   \ifdefined\DeclareUppercaseMapping % new in 2023
552     % \DeclareUppercaseMapping[e1]{\1FBE}{\prosgegrammeni}%
553     \DeclareUppercaseMapping[e1]{\0027}{\}% '
554     \addto\bb1@greek@setup@polytonic{
555       \DeclareUppercaseMapping[e1]{\003C}{\}% <
556       \DeclareUppercaseMapping[e1]{\003E}{\}% >
557       \DeclareUppercaseMapping[e1]{\0060}{\}% `
558       \DeclareUppercaseMapping[e1]{\007E}{\}% ~
559     }
560 \fi

```

<sup>19</sup>The pre-2022 implementation is corrected by character code definitions in `tuenc-greek.def` from `greek-fontenc`, cf. section 3.4.2.

<sup>20</sup>cf. [LaTeX News 37](#)

If LaTeX is older than 2022/06, we set the `\uccode` of the relevant characters to a dummy character (`\uccode` changes are ignored by `\MakeUppercase` since 2022). To minimize side-effects, the re-definition is limited to Greek text parts.

```

561 \IfFormatAtLeastTF{2022/06/01}{-}
562   {% else (LaTeX format older than 2022/06/01)
563     \addto\extrasgreek{%
564       \babel@savevariable{\uccode`\~}\uccode`~="`~%
565       \babel@savevariable{\uccode`\'}\uccode`\'=159% 159 == ^^9f
566     }
567     \addto\extraspolutonikogreek{%
568       \babel@savevariable{\uccode`\~}\uccode`\~=159%
569       \babel@savevariable{\uccode`\>}\uccode`\>=159%
570       \babel@savevariable{\uccode`\<}\uccode`\<=159%
571       \babel@savevariable{\uccode`\'}\uccode`\'=159%
572     }

```

To avoid 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>21</sup> To be independent of `inputenc`, we do not use `\DeclareInputText` but code modelled after its definition to declare an empty expansion.

```

573   \bgroup
574     \uccode`\~159%
575     \uppercase{%
576   \egroup
577     \def~{}%
578   }

```

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 [greek-fontenc](#) package for details).

```

579   \DeclareTextCompositeCommand{\~}{LGR}{^^9f}{\accdialytika}
580   \DeclareTextCompositeCommand{\'}{LGR}{^^9f}{\LGR@hiatus}
581   \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.

```

582   \ifdefined\UnicodeEncodingName % set by XeTeX/LuaTeX
583     \DeclareTextCompositeCommand{\"}{TU}{^^9f}{\accdialytika}
584   \fi
585 }% end of the \IfFormatAtLeastTF else block
586 \fi % End of LGR-specific code.

```

### 3.8 Symbol name aliases

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

```

587 \providecommand*\anwtonos{\textdexiakeraia}
588 \providecommand*\katwtonos{\textaristerikeraia}

```

<sup>21</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.



```

589 \providecommand*\qoppa{\textkoppa}
590 \providecommand*\varqoppa{\textqoppa}
591 \providecommand*\stigma{\textstigma}
592 \providecommand*\sampi{\textsampi}
593 \providecommand*\Digamma{\textDigamma}
594 \providecommand*\ddigamma{\textdigamma}
595 \providecommand*\vardigamma{\textvardigamma}
596 \providecommand*\euro{\texteuro}
597 \providecommand*\permill{\textperthousand}
598 \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.

```

599 \ldf@finish{\CurrentOption}
600 \endcode

```

## Change History

babel-greek-1.08	Beccari). . . . .	6
<code>\greekscript</code> : New	Add support for ancient Greek. . .	8
TextCommands “greekscript”	Added caption names for	
and “ensuregreek”. . . . .	<code>\ancientgreek</code> . . . . .	15
General: <code>greek.dtx</code> renamed to	Added lc codes for chars 128 to	
<b>babel-greek.dtx</b> (but still	255 . . . . .	21
generates <code>greek.ldf</code> ). . . . .	The $\hat{\text{}}$ -notation seems to require	
Check for EU1/EU2 font	lower case letters. . . . .	24
encoding instead of engine . . . .	babel-greek-1.09b	
Load <code>euenc.def</code> if EU1 or EU2	<code>\captionspolutonikogreek</code> : Use	
font encoding is detected. . . . .	named macros instead of	
Restore compatibility with	non-standard short accent	
Xe/LuaTeX in 8-bit and	macros for psili and dasia. . . .	15
Unicode mode. . . . .	General: Remove spurious	
Use EU1 or EU2 for Latin script	whitespace from ‘extrasgreek’	
if available . . . . .	definition (report Eike	
Use font-encoding specific	Schmidt). . . . .	21
TextCommands. . . . .	babel-greek-1.09c	
Remove redefinition of	General: Fix dummy hyphenation	
<code>\fnum@figure</code> and	language names (patch Ulrike	
<code>\fnum@table</code> . . . . .	Fischer). . . . .	6
babel-greek-1.08a	babel-greek-1.09d	
<code>\greekscript</code> : Set	General: uc-/lcode corrections	
‘encodingdefault’ to fix Greek	from <code>xgreek</code> are now in	
in footnotes etc. with	<code>greek-euenc.def</code> (the	
document language Greek. . . .	<code>polyglossia</code> version has bugs). . .	23
babel-greek-1.09	babel-greek-1.09e	
General: Load correct hyphenation	General: Fix bug in <code>lcode-setting</code>	
patterns (patch by Claudio	loop (patch by Enrico	

Gregorio). . . . .	21	<code>\greek@Alph</code> : Save/restore expansion of <code>\alph</code> and <code>\Alph</code> with every switch to/from Greek. . . . .	21
babel-greek-1.09f		General: Declare char 159 expansion similar to <code>inputenc</code> to avoid “inputenc error”. . . . .	24
General: Check also for standard Unicode text encoding “TU” (new in fontspec v2.5a). . . . .	9	Don’t use <code>\makeatother</code> in <code>\AtBeginDocument</code> . . . . .	10
babel-greek-1.09g		New language attribute <code>polytonic</code> (alias for <code>polutoniko</code> ). . . . .	8
General: Babel 3.9i deprecated <code>\textlatin</code> and fixed <code>\latinencoding</code> . . . . .	1	New modifiers <code>local-LGR-fixes</code> and <code>no-LGR-fixes</code> . . . . .	3
babel-greek-1.09h		Only change <code>uc/lccodes</code> if <code>\greekfontencoding</code> is LGR. . . . .	21
General: Move breathing composite commands to <code>textalpha</code> . . . . .	24	Only change <code>uccodes</code> if LaTeX is older than 2022/06/01. . . . .	23
babel-greek-1.09i		Drop definition for <code>\SS</code> . . . . .	12
<code>\captionsgreek</code> : Fix accent in <code>seename</code> and <code>alsename</code> . . . . .	14	Remove <code>\textKoppa</code> and <code>\textmu</code> (in <code>greek-fontenc</code> since version 1.0). . . . .	24
General: Fix accent in <code>\seename</code> and <code>\alsename</code> . . . . .	1	Save previous font encoding in <code>\BabelGreekPreviousFontEncoding</code> . . . . .	11
Update check for Unicode fonts. . . . .	9	babel-greek-1.13	
babel-greek-1.09j		<code>\bbl@greek@tilde</code> : Renamed from <code>\greek@tilde</code> . Simplified definition. . . . .	22
<code>\textampersand</code> : Fix ampersand in math. . . . .	12	General: Don’t use <code>text</code> command in math mode. . . . .	8
babel-greek-1.10		New language attribute <code>keep-semicolon</code> . . . . .	8
<code>\greeknumeral</code> : PDF-string secure implementation taken from “hyperref” (thanks to Ulrike Fischer). . . . .	18	Setup <code>\languageshorthands</code> for all language variants. . . . .	6
<code>\greeknumeralNinety</code> : Use zig-zagy <code>\textkoppa</code> . This is what it looks in current Greek typography. . . . .	18	babel-greek-1.13.2	
General: Load <code>puenc-greek.def</code> from <code>greek-fontenc</code> if used with <code>hyperref</code> . . . . .	10	General: <code>\MakeUppercase</code> fix for transliteration input. . . . .	23
Use TU with Xe/LuaTeX. . . . .	9	Warn of unsupported modifiers. . . . .	9
babel-greek-1.11		babel-greek-1.14	
<code>\greeknumeral</code> : Configurable shapes for 6 and 90. 90 defaults to <code>\textqoppa</code> for ancient Greek. . . . .	18	General: <code>\MakeUppercase</code> fix for standard accent macros. . . . .	23
General: Save/restore previous font encoding instead of switching to <code>\latinencoding</code> when leaving Greek. . . . .	11	New modifier <code>local-MakeUppercase-fixes</code> . . . . .	3
babel-greek-1.12		greek-1.0b	
<code>\BabelGreekRestoreFontEncoding</code> : New macro. . . . .	11	<code>\lgrfont</code> : Added a level of braces to keep encoding change local . . . . .	11
<code>\EnsureStandardFontEncoding</code> : New <code>TextCommand</code> . . . . .	12	General: Use <code>\LdfInit</code> to perform initial checks. . . . .	6

Moved the definition of <code>\atcatcode</code> right to the beginning . . . . .	1	Most symbols are removed and are now defined in package <code>grsymb</code> . . . . .	24
Now use <code>\ldf@finish</code> to wrap up . . . . .	25	<b>greek-1.2</b> <code>\gr@polutoniko@month</code> : Added macro <code>\datepolutonikogreek</code> . . . . .	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	Added macro <code>\gr@cl@month</code> . . . . .	17
<b>greek-1.0c</b> <code>\bbl@greek@tilde</code> : Added command . . . . .	22	General: Added caption names for <code>\polutonikogreek</code> . . . . .	15
<b>greek-1.1</b> <code>\Grtoday</code> : Added macro <code>\Grtoday</code> . . . . .	18	Added lowercase codes for “modern” greek . . . . .	21
<b>greek-1.1a</b> <code>\dategreek</code> : Fixed typo, <code>\Oktwbr’iou</code> instead of <code>\Oktobr’iou</code> . . . . .	17	Added uppercase codes for “modern” Greek. The old codes are now for “Polutoniko” Greek . . . . .	23
<code>\greek@Alph</code> : removed two superfluous @’s which made <code>\@alph</code> undefined . . . . .	21	Classical Greek is now a dialect . . . . .	1
<b>greek-1.1b</b> <code>\bbl@greek@tilde</code> : Made tilde expand to a tilde with <code>\catcode 12</code> . . . . .	22	Definitions for “modern” Greek are now the definitions of “polutoniko” Greek . . . . .	21
General: Added shorthand for <code>\char255</code> . . . . .	24	<b>greek-1.2a</b> <code>\dategreek</code> : Use <code>\edef</code> to define <code>\today</code> . . . . .	17
Added setting of <code>\uccodes</code> (after <code>kdgreek.sty</code> ) . . . . .	23	General: Need shorthand to exist for monotonic Greek, not polytonik Greek . . . . .	24
<b>greek-1.1c</b> General: Added a couple of symbols, needed for <code>\greeknumeral</code> . . . . .	24	filename <code>lgrenc.def</code> now lowercase . . . . .	9
fixed two typos . . . . .	21	<b>greek-1.2b</b> <code>\dategreek</code> : use <code>\def</code> instead of <code>\edef</code> . . . . .	17
<b>greek-1.1d</b> <code>\dategreek</code> : Macro <code>\gr@month</code> now produces the name of the month . . . . .	17	General: Classical Greek is now called “Polutoniko” Greek. The previous name was at least misleading . . . . .	1
<b>greek-1.1e</b> <code>\gr@month</code> : Macro added . . . . .	16	<b>greek-1.2c</b> General: Package <code>grsymb</code> has been eliminated because the CB fonts v2.0 do not include certain symbols and so the remaining symbol definitions have been moved here . . . . .	24
General: Shorthand is changed. Active character is now <code>\char159</code> . . . . .	24	This version conforms to version 2.0 of the CB fonts and consequently we added a few new symbol-producing commands . . . . .	1
Added caption name for proof . . . . .	14	<b>greek-1.2e</b> General: Moved redefinition of <code>\@roman</code> back to the language specific file . . . . .	13
Added lowercase code for v . . . . .	21		
Added uppercase code for special letter “v”. Uppercase code for accents is now <code>9f</code> , instead of <code>ff</code> . . . . .	23		

greek-1.3a		greek-1.3k	
<code>\gr@polutoniko@month</code> : removed		<code>\bbl@greek@tilde</code> : Make sure the	
macro <code>\datepolutonikogreek</code>	17	character “ is not active during	
General: polutoniko is now an		the definition of <code>\greek@tilde</code>	22
attribute to Greek, no longer a		<code>\lgrfont</code> : Added <code>\leavevmode</code> as	
‘dialect’	1	was done with <code>\latintext</code>	11
greek-1.3d		greek-1.4	
General: <code>\@roman</code> and <code>\@Roman</code>		<code>\bbl@greek@tilde</code> : Do not	
need to be added to		re-define the tilde accent	
<code>\extrapolutonikogreek</code>	13	macro: it works as expected	
Fixed typo, <code>bl’epe ep’ishc</code>		with <code>lgrenc.def</code> from	
instead of <code>bl’pe ep’ishc</code>	14	<code>greek-fontenc</code>	22
greek-1.3e		General: <code>lgrenc.def</code> moved to the	
General: <code>\@roman</code> and <code>\@Roman</code>		separate package	
need <i>not</i> be in		‘greek-fontenc’	9
<code>\extrapolutonikogreek</code>		Add <code>TextCompositeCommands</code>	
when they are already in		for “uppercase diacritics”	24
<code>\extragreek</code>	13	moved here from <code>lgrenc.def</code>	
<code>\extragreek</code> and		because the definitions require	
<code>\extrapolutonikogreek</code>		the <code>\latintext</code> macro defined	
should be complementary	21, 23	by Babel	12
greek-1.3f		new maintainer	1
General: Added some code to make		greek-1.5	
older documents work	7	<code>\textampersand</code> : Make <code>&amp;</code> a	
greek-1.3g		<code>TextCommand</code>	12
General:		General: <code>\@roman</code> and <code>\@Roman</code> as	
<code>\noextrapolutonikogreek</code>		<code>TextCommands</code> (BUG: this	
was missing	7	extended the expansion	
greek-1.3h		problem to all languages)	13
<code>\captionsgreek</code> : Added		bugfixes, change some symbol	
<code>\glossaryname</code>	14	macros to aliases, LGR fixes	
<code>\providehyphenmins</code> : Now use		via <code>\DeclareTextCommand</code>	
<code>\providehyphenmins</code> to		instead of	
provide a default value	7	<code>extragreek/noextragreek</code>	
greek-1.3i		definitions, LICR macros in	
<code>\captionsgreek</code> : The final sigma		string definitions, LGR font	
in all names appears as ‘s’		encoding not used with	
instead of ‘c’	14	XeTeX/LuaTeX	1
General: uc code of ‘v’ is switched		change symbol macros to aliases	24
to V so that mixed text		enable use of “textcomp”	
appears correctly in headers	23	characters for “textcopyright”	
greek-1.3j		and “textregistered” macros	12
General: Use the tilde as an alias		LGR not used with	
for character 159	24	XeTeX/LuaTeX	11
Don’t use the double caret		LGR setup skipped with	
notation here, because other		XeTeX/LuaTeX	9
languages might make the		Support XeTeX/LuaTeX	23
caret active	23	greek-1.5a	
		General: provide	
		<code>\extrapolutonikogreek</code> also	

for Xe/LuaTeX. . . . .	7	Do not load euenc.def with XeTeX/LuaTeX. Prevent re-loading lgrenc.def. . . . .	1
Replaced non-printable literal character with ^-notation (tip by Heiko Oberdiek). . . . .	24		
greek-1.6		greek-1.7a	
General: Apply a patch by Enrico Gregorio. Thanks to Claudio Beccari for testing and reporting. . . . .	13	General: Remove spurious “fi”. . . . .	1
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-1.7b	
greek-1.7		General: Correct upcasing of babel strings with Xe/LuaTeX. . . . .	1
General: Do not load euenc.def with XeTeX/LuaTeX (too complicated to get it right). . . . .	9	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> . . . . .	12
		greekfdd-2.2d	
		General: removed redefinition of <code>\&amp;</code> . . . . .	12