Babel support for the Greek language

Apostolos Syropoulos, Günter Milde

October 13, 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 overwiew of the babel-greek package and links to requirements and related packages.

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

Contents

---

[1] The file described in this section has version number 1.15 and was last revised on 2023/10/13. The original author is Apostolos Syropoulos, code from `kdgreek.sty` by David Kastrup was used.

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[2] are a complete set of 8-bit TEX fonts matching KNUTH's Computer Modern. The package cbfonts-fd sets them up as Greek substitute for the Computer Modern and Latin Modern font families. The 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.

2.1 Language attributes

The attributes polutoniko[3], 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}
```

---

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

[3] with the alias polytonic

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 ressort if the workarounds make a document uncompilable 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 no-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 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 alphabeta 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:

`\greekscript` `\greektext` The *TextCommand*[4] `\greekscript` switches to a font encoding supporting the Greek script. The declaration `\greektext` always switches the font encoding to LGR. Both declarations do not change the active language.

`\latintext` `\latintext` (defined by the Babel core, deprecated since March 2014) can be used to switch back to an encoding supporting the Latin script.

`\ensuregreek` The function `\ensuregreek` 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).

`\lgrfont` The function `\lgrfont`[5] switches to the non-standard Greek 8-bit font encoding LGR. Hint: Use `\lgrfont`, if you want to use the *Latin transliteration* input method and `\ensuregreek` else.

`\ensureascii` The Babel core defines `\ensureascii` 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.

2.6 Greek numbering

The Greek (Milesian) alphabetical numbering system[6] 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` `\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*[7], *zeta*, *eta*, *theta*, followed by a *keraia*, a mark similar to the mathematical symbol "prime".

---

[4] For a discussion of TextCommands, see the *LaTeX font guide*.

[5] The legacy name `\textgreek` is available as alias.

[6] Attic numerals, which predate the Milesian numerals are implemented in package athnum.

[7] cf. `\greeknumeralsix`

- Decades from 10 to 90 are denoted by letters *iota*, *kappa*, *lambda*, *mu*, *nu*, *xi*, *omikron*, *pi*, *koppa*[8], 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 `\Greeknumeral` respectively:

`\greeknumeralsix` `\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[9]. The macro `\greeknumeralsix` allows configuring the symbol for the number 6 in `\greeknumeral`, the macro `\greeknumeralSix` does the same for `\Greeknumeral`. The default values are `\textstigma` and `\textStigma`.

`\greeknumeralninety` `\greeknumeralNinety`

Three symbols are in use for the number 90: Classicists prefer the q-like "archaic" *koppa* and, more rarely, its uncial form[10], 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 `\Greeknumeral`. 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.

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

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

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

---

[8] cf. `\greeknumeralninety`

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

[10] resembling CYRILLIC LETTER KOPPA or GOTHIC LETTER NINETY

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* ·.

```
\addto\extrasgreek{\languageshorthands{greek}}
```

An auxiliary macro allows to test whether a macro holds the string 'LGR' with `\ifx`:

```
\def\bbl@greek@LGR{LGR}
```

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.

```
\ifx\l@greek\@undefined
  \@nopatterns{greek}
  \adddialect\l@greek 0
\fi
\ifx\l@monogreek\@undefined
  \@nopatterns{greek}
  \adddialect\l@monogreek 0
\fi
\ifx\l@ancientgreek\@undefined
  \@nopatterns{greek}
  \adddialect\l@ancientgreek 0
\fi
\newcount\bbl@monogreek \bbl@monogreek=\l@monogreek
\newcount\bbl@polygreek \bbl@polygreek=\l@greek
\newcount\bbl@ancientgreek \bbl@ancientgreek=\l@ancientgreek
```

`\extraspolutonikogreek`
`\extrasancientgreek`

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 `\extrasgreek` by the respective language attributes, cf. section 3.2).

```
\addto\extrasgreek{\let\l@greek=\bbl@monogreek}
\addto\extraspolutonikogreek{\l@greek=\bbl@polygreek}
\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.

```
\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 `\extrasgreek` 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).

```
\def\bbl@greek@setup@polytonic{%
  \expandafter\addto\expandafter\extrasgreek
    \expandafter{\extraspolutonikogreek}%
  \let\captionsgreek\captionspolutonikogreek
  \let\gr@month\gr@polutoniko@month
}
```

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.[11] We also define aliases to allow language switching commands using the language name `polutonikogreek`:

```
\bbl@declare@ttribute{greek}{polutoniko}{%
  \bbl@greek@setup@polytonic
  \let\l@polutonikogreek\l@greek
  \let\datepolutonikogreek\dategreek
  \let\extraspolutonikogreek\extrasgreek
  \let\noextraspolutonikogreek\noextrasgreek
}
```

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.

```
\bbl@declare@ttribute{greek}{polytonic}{%
  \bbl@greek@setup@polytonic
}
```

3.2.3 ancient

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

```
\bbl@declare@ttribute{greek}{ancient}{%
  \bbl@greek@setup@polytonic
  \expandafter\addto\expandafter\extrasgreek
    \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).

```
  \let\captionsgreek\captionsancientgreek
```

---

[11] Use of more than one Greek orthograpies in one document is possible with `\babelprovide`. However, there are side-effects. See the example in `test-greek.tex`.

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).

```
  \renewcommand{\greeknumeralninety}{\textqoppa}%
  \renewcommand{\greeknumeralNinety}{\textQoppa}%
}
```

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:

```
\bbl@declare@ttribute{greek}{keep-semicolon}{%
    \ifx\greekfontencoding\bbl@greek@LGR
    \ProvideTextCommandDefault{\textsemicolon}{;}
    \ProvideTextCommand{\textsemicolon}{LGR}{\texterotimatiko}
    \initiate@active@char{;}
    \addto\extrasgreek{\bbl@activate{;}}
    \addto\noextrasgreek{\bbl@deactivate{;}}
    \declare@shorthand{greek}{;}{\TextOrMath{\textsemicolon}{;}}
  \fi
}
```

3.3 Report unsupported modifiers

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

```
\def\bbl@greek@modifiers{,%
  polutoniko,polytonic,ancient,keep-semicolon,%
  local-LGR-fixes,no-LGR-fixes,no-MakeUppercase-fixes,}
\ifx\BabelModifiers\relax\else
  \bbl@foreach\BabelModifiers{%
    \@expandtwoargs\in@{,#1,}{\bbl@greek@modifiers}
    \ifin@\else
      \bbl@warning
        {Unknown/misspelled modifier ’#1’ in ’\CurrentOption’.
         See "babel-greek.pdf" for valid modifiers.}
    \fi}%
\fi

```

3.4 Font setup

3.4.1 Greek font encoding

`\greekfontencoding`

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

---

[12] cf. encguide.pdf

[13] Document authors must ensure that the selected font actually contains the required glyphs.

Also store the name of the *encoding definition file*[14] with the extended Greek setup for the Greek font encoding.

```
\ifdefined\UnicodeEncodingName % set by XeTeX/LuaTeX
  \providecommand*{\greekfontencoding}{\UnicodeEncodingName}
  \providecommand*{\bbl@greek@fontencdef}{tuenc-greek}
\else
  \providecommand*{\greekfontencoding}{LGR}
  \providecommand*{\bbl@greek@fontencdef}{lgrenc}
\fi
```

TODO: Why does the more generic version
`\edef\bbl@greek@fontencdef{\lowercase{\greekfontencoding}.enc}`
fail when used in `\ifl@aded`?

3.4.2 Ensure loading of Greek font encoding definitions.

If the *encoding definition file* for `\greekfontencoding` is not yet loaded, do this now. (Using `\RequirePackage` fails as we are in an “Options Section”.) TODO: set with `\AtEndOfPackage`? (All definitions requiring the font encoding must be delayed as well!)

```
\@ifl@aded{def}{\bbl@greek@fontencdef}
  {}
  {\InputIfFileExists{\bbl@greek@fontencdef .def}
     {}
     {\bbl@error{Font support for the Greek script missing.\\
                 babel-greek can’t typeset Greek.\\
                 Install the "greek-fontenc" package\\
                 or use XeTeX/LuaTeX with polyglossia.}
                {I can’t find the \bbl@greek@fontencdef .def file
                 for the Greek fonts (encoding \greekfontencoding)}
      \@@end
     }
  }
```

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.

```
\AtBeginDocument{%
  \@ifl@aded{def}{puenc}%
    {\@ifl@aded{def}{puenc-greek}
       {}%
       {\edef\RestoreAtCatcode{\catcode‘@=\the\catcode‘@\relax}%
        \makeatletter
        \InputIfFileExists{puenc-greek.def}%
          {}%
          {\bbl@warning{I cannot find the Greek fixes for PDF strings
                        ("punec-greek.def" from "greek-fontenc").}%
          }%
```

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`).

[14] see fntguide.pdf

```
        \RestoreAtCatcode
       }%
    }% end "puenc.def loaded" branch
    {}% empty "puenc.def not loaded" branch
}
```

3.4.3 Font encoding switches

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

```
\addto\extrasgreek{%
  \let\BabelGreekPreviousFontEncoding\encodingdefault
  \greekscript}
\addto\noextrasgreek{\BabelGreekRestoreFontEncoding}
```

`\greekscript` The TextCommand[15] `\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.

```
\ProvideTextCommandDefault{\greekscript}{%
  \fontencoding{\greekfontencoding}\selectfont
  \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.

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

`\BabelGreekRestoreFontEncoding` The declaration `\BabelGreekRestoreFontEncoding` changes the font encoding to the value of `\encodingdefault` before the switch to the Greek language. It does nothing, if there was no font encoding change when entering Greek.

```
\def\BabelGreekRestoreFontEncoding{%
  \ifx\encodingdefault\BabelGreekPreviousFontEncoding
  \else
    \let\encodingdefault\BabelGreekPreviousFontEncoding
    \fontencoding{\encodingdefault}\selectfont
  \fi
}
```

Exception: don't keep LGR if it was the initial encoding as it is clearly unsuited for non-Greek texts. If Greek is the main language, `\extrasgreek` is called before `\begin{document}` — we can check the saved value of the previous font encoding and replace LGR with the default generic text font encoding.

```
\AtBeginDocument{
  \ifx\BabelGreekPreviousFontEncoding\bbl@greek@LGR
    \let\BabelGreekPreviousFontEncoding\latinencoding
  \fi
}
```

---

[15] See fntguide.pdf for more info about *TextCommands*.

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`:

```
\@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.

```
  \DeclareRobustCommand{\greektext}{%
    \fontencoding{LGR}\selectfont
    \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 ambiguitiy: 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.

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

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

```
  \ProvideTextCommand{\textcopyright}{LGR}{\ensureascii{\textcopyright}}
  \ProvideTextCommand{\textregistered}{LGR}{\ensureascii{%
                                                      \textregistered}}
  \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.

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

```
  \ProvideTextCommandDefault{\EnsureStandardFontEncoding}{\@firstofone}
  \ProvideTextCommand{\EnsureStandardFontEncoding}{LGR}[1]{%
    \ensureascii{#1}
  }
  \AtBeginDocument{%
    \@ifpackageloaded{hyperref}
      {\pdfstringdefDisableCommands{%
         \let\EnsureStandardFontEncoding\@firstofone}
      }
      {}
  }
```

End the LGR additions block:

```
}{}
```

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.[16] 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.

```
\ifx\greekfontencoding\bbl@greek@LGR
  \def\bbl@greek@roman#1{\expandafter\EnsureStandardFontEncoding%
                         \expandafter{\romannumeral#1}}
  \def\bbl@greek@Roman#1{\expandafter\EnsureStandardFontEncoding%
           \expandafter{\expandafter\@slowromancap\romannumeral#1@}}
  \DeclareRobustCommand{\bbl@greek@ampersand}{%
           \ifmmode\bbl@greek@original@amp\else\textampersand\fi}
  \bbl@xin@{,no-LGR-fixes,}{,\BabelModifiers,}%
  \ifin@
    % skip re-definitions
  \else
    \bbl@xin@{,local-LGR-fixes,}{,\BabelModifiers,}%
    \ifin@
```

---

[16] For example, Roman page numbers are generated at "unpredictable" positions and can move to the ToC, (hyper)references, or an index.

```
   \addto\extrasgreek{%
    \babel@save\@roman
    \babel@save\@Roman
    \let\@roman\bbl@greek@roman
    \let\@Roman\bbl@greek@Roman
    \babel@save\&%
    \let\&\bbl@greek@ampersand%
    }
  \else
   \let\@roman\bbl@greek@roman
   \let\@Roman\bbl@greek@Roman
   \let\&\bbl@greek@ampersand
  \fi
 \fi
\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.

3.5.1 Auto-strings for Greek

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

```
\addto\captionsgreek{%
 \def\prefacename{\textPi\textrho\acctonos\textomicron\textlambda
  \textomicron\textgamma\textomicron\textfinalsigma}%
 \def\refname{\textAlpha\textnu\textalpha
  \textphi\textomicron\textrho\acctonos\textepsilon\textfinalsigma}%
 \def\abstractname{\textPi\textepsilon\textrho\acctonos\textiota
  \textlambda\texteta\textpsi\texteta}%
 \def\bibname{\textBeta\textiota\textbeta\textlambda\textiota
  \textomicron\textgamma\textrho\textalpha\textphi\acctonos
  \textiota\textalpha}%
 \def\chaptername{\textKappa\textepsilon\textphi\acctonos\textalpha
  \textlambda\textalpha\textiota\textomicron}%
 \def\appendixname{\textPi\textalpha\textrho\acctonos\textalpha\textrho
  \texttau\texteta\textmu\textalpha}%
 \def\contentsname{\textPi\textepsilon\textrho\textiota
  \textepsilon\textchi\acctonos\textomicron\textmu\textepsilon
  \textnu\textalpha}%
 \def\listfigurename{\textKappa\textalpha\texttau\acctonos\textalpha
  \textlambda\textomicron\textgamma\textomicron\textfinalsigma{}
  \textSigma\textchi\texteta\textmu\acctonos\textalpha\texttau
  \textomega\textnu}%
 \def\listtablename{\textKappa\textalpha\texttau\acctonos\textalpha
  \textlambda\textomicron\textgamma\textomicron\textfinalsigma{}
  \textPi\textiota\textnu\acctonos\textalpha\textkappa\textomega
  \textnu}%
 \def\indexname{\textEpsilon\textupsilon\textrho\textepsilon
  \texttau\acctonos\texteta\textrho\textiota\textomicron}%
 \def\figurename{\textSigma\textchi\acctonos\texteta\textmu\textalpha}%
```

```
  \def\tablename{\textPi\acctonos\textiota\textnu\textalpha
    \textkappa\textalpha\textfinalsigma}%
  \def\partname{\textMu\acctonos\textepsilon\textrho\textomicron
    \textfinalsigma}%
  \def\enclname{\textSigma\textupsilon\textnu\texteta\textmu
    \textmu\acctonos\textepsilon\textnu\textalpha}%
  \def\ccname{\textKappa\textomicron\textiota\textnu\textomicron
    \textpi\textomicron\acctonos\textiota\texteta\textsigma\texteta}%
  \def\headtoname{\textPi\textrho\textomicron\textfinalsigma}%
  \def\pagename{\textSigma\textepsilon\textlambda\acctonos\textiota
    \textdelta\textalpha}%
  \def\seename{\textbeta\textlambda\acctonos\textepsilon\textpi
    \textepsilon}%
  \def\alsoname{\textbeta\textlambda\acctonos\textepsilon\textpi
    \textepsilon{} \textepsilon\textpi\acctonos\textiota\textsigma
    \texteta\textfinalsigma}%
  \def\proofname{\textAlpha\textpi\acctonos\textomicron
    \textdelta\textepsilon\textiota\textxi\texteta}%
  \def\glossaryname{\textGamma\textlambda\textomega\textsigma
    \textsigma\acctonos\textalpha\textrho\textiota}%
}
```

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.

```
\let\captionspolutonikogreek\captionsgreek
\addto\captionspolutonikogreek{%
  \def\refname{\accpsili\textAlpha\textnu\textalpha
    \textphi\textomicron\textrho\accvaria\textepsilon\textfinalsigma}%
  \def\indexname{\textEpsilon\accdasia\textupsilon\textrho\textepsilon
    \texttau\acctonos\texteta\textrho\textiota\textomicron}%
  \def\figurename{\textSigma\textchi\accperispomeni\texteta\textmu
    \textalpha}%
  \def\headtoname{\textPi\textrho\accvaria\textomicron\textfinalsigma}%
  \def\alsoname{\textbeta\textlambda\acctonos\textepsilon\textpi
    \textepsilon{} \accpsili\textepsilon\textpi\acctonos\textiota
    \textsigma\texteta\textfinalsigma}%
  \def\proofname{\accpsili\textAlpha\textpi\acctonos\textomicron
    \textdelta\textepsilon\textiota\textxi\texteta}%
}
```

3.5.3 Auto-strings for ancient Greek

`\captionsancientgreek` 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.

```
\let\captionsancientgreek\captionsgreek
\addto\captionsancientgreek{%
  \def\prefacename{\textPi\textrho\textomicron\textomicron
    \acctonos\textiota\textmu\textiota\textomicron\textnu}%
  \def\refname{\accpsili\textAlpha\textnu\textalpha\textphi\textomicron
```

```
    \textrho\textalpha\accvaria\textiota}%
  \def\abstractname{\textPi\textepsilon\textrho\acctonos\textiota
    \textlambda\texteta\textpsi\textiota\textvarsigma}%
  \def\bibname{\textBeta\textiota\textbeta\textlambda\textiota
    \textomicron\textgamma\textrho\textalpha\textphi
    \acctonos\textiota\textalpha}%
  \def\chaptername{\textKappa\textepsilon\textphi\acctonos\textalpha
    \textlambda\textalpha\textiota\textomicron\textnu}%
  \def\appendixname{\textPi\textalpha\textrho\acctonos\textalpha
    \textrho\texttau\texteta\textmu\textalpha}%
  \def\contentsname{\textPi\textepsilon\textrho\textiota\textepsilon
    \textchi\acctonos\textomicron\textmu\textepsilon\textnu\textalpha}%
  \def\listfigurename{\textKappa\textalpha\texttau\acctonos\textalpha
    \textlambda\textomicron\textgamma\textomicron\textvarsigma{}
    \textsigma\textchi\texteta\textmu\acctonos\textalpha\texttau
    \textomega\textnu}%
  \def\listtablename{\textKappa\textalpha\texttau\acctonos\textalpha
    \textlambda\textomicron\textgamma\textomicron\textvarsigma{}
    \textpi\textiota\textnu\acctonos\textalpha\textkappa
    \textomega\textnu}%
  \def\indexname{\textEpsilon\accdasia\textupsilon\textrho\textepsilon
    \texttau\acctonos\texteta\textrho\textiota\textomicron\textnu}%
  \def\figurename{\textSigma\textchi\accperispomeni\texteta\textmu
    \textalpha}%
  \def\tablename{\textPi\acctonos\textiota\textnu\textalpha\textxi}%
  \def\partname{\textMu\acctonos\textepsilon\textrho\textomicron
    \textvarsigma}%
  \def\enclname{\textSigma\textupsilon\textnu\texteta\textmu\textmu
    \acctonos\textepsilon\textnu\textomega\textvarsigma}%
  \def\ccname{\textKappa\textomicron\textiota\textnu\textomicron\textpi
    \textomicron\acctonos\textiota\texteta\textsigma\textiota
    \textvarsigma}%
  \def\headtoname{\textPi\textrho\accvaria\textomicron\textvarsigma}%
  \def\pagename{\textSigma\textepsilon\textlambda\accvaria\textiota
    \textvarsigma}%
  \def\seename{\accdasiaoxia\textomicron\textrho\textalpha}%
  \def\alsoname{\accdasiaoxia\textomicron\textrho\textalpha{}
    \accdasia\textomega\textsigma\textalpha\acctonos\textupsilon
    \texttau\textomega\textvarsigma}%
  \def\proofname{\accpsili\textAlpha\textpi\acctonos\textomicron
    \textdelta\textepsilon\textiota\textxi\textiota\textvarsigma}%
  \def\glossaryname{\textGamma\textlambda\textomega\textsigma\textsigma
    \acctonos\textalpha\textrho\textiota\textomicron\textnu}%
}
```

3.5.4 Date specification

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

```
\def\gr@month{%
  \ifcase\month\or
    \textIota\textalpha\textnu\textomicron\textupsilon\textalpha
      \textrho\acctonos\textiota\textomicron\textupsilon \or
    \textPhi\textepsilon\textbeta\textrho\textomicron\textupsilon
```

```
      \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
    \textMu\textalpha\textrho\texttau\acctonos\textiota\textomicron
      \textupsilon \or
    \textAlpha\textpi\textrho\textiota\textlambda\acctonos\textiota
      \textomicron\textupsilon \or
    \textMu\textalpha\’"\textiota\textomicron\textupsilon \or
    \textIota\textomicron\textupsilon\textnu\acctonos\textiota
      \textomicron\textupsilon \or
    \textIota\textomicron\textupsilon\textlambda\acctonos\textiota
      \textomicron\textupsilon \or
    \textAlpha\textupsilon\textgamma\textomicron\acctonos\textupsilon
      \textsigma\texttau\textomicron\textupsilon \or
    \textSigma\textepsilon\textpi\texttau\textepsilon\textmu
      \textbeta\textrho\acctonos\textiota\textomicron\textupsilon \or
    \textOmicron\textkappa\texttau\textomega\textbeta
      \textrho\acctonos\textiota\textomicron\textupsilon \or
    \textNu\textomicron\textepsilon\textmu\textbeta
      \textrho\acctonos\textiota\textomicron\textupsilon \or
    \textDelta\textepsilon\textkappa\textepsilon\textmu\textbeta
      \textrho\acctonos\textiota\textomicron\textupsilon
  \fi
}
```

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

```
\def\gr@polutoniko@month{%
  \ifcase\month\or
    \accpsili\textIota\textalpha\textnu\textomicron\textupsilon
      \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
    \textPhi\textepsilon\textbeta\textrho\textomicron\textupsilon
      \textalpha\textrho\acctonos\textiota\textomicron\textupsilon \or
    \textMu\textalpha\textrho\texttau\acctonos\textiota\textomicron
      \textupsilon \or
    \accpsili\textAlpha\textpi\textrho\textiota\textlambda
      \acctonos\textiota\textomicron\textupsilon \or
    \textMu\textalpha\accdialytikatonos\textiota\textomicron
      \textupsilon \or
    \accpsili\textIota\textomicron\textupsilon\textnu
      \acctonos\textiota\textomicron\textupsilon \or
    \accpsili\textIota\textomicron\textupsilon\textlambda
      \acctonos\textiota\textomicron\textupsilon \or
    \textAlpha\accpsili\textupsilon\textgamma\textomicron\acctonos
      \textupsilon\textsigma\texttau\textomicron\textupsilon \or
    \textSigma\textepsilon\textpi\texttau\textepsilon\textmu\textbeta
      \textrho\acctonos\textiota\textomicron\textupsilon \or
    \accpsili\textOmicron\textkappa\texttau\textomega\textbeta
      \textrho\acctonos\textiota\textomicron\textupsilon \or
    \textNu\textomicron\textepsilon\textmu\textbeta
      \textrho\acctonos\textiota\textomicron\textupsilon \or
    \textDelta\textepsilon\textkappa\textepsilon\textmu
      \textbeta\textrho\acctonos\textiota\textomicron\textupsilon
  \fi
}
```

`\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`.

```
\def\dategreek{%
  \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`.)

```
\def\Grtoday{%
  \expandafter\Greeknumeral\expandafter{\the\day}\space
  \gr@polutoniko@month \space
  \expandafter\Greeknumeral\expandafter{\the\year}}
```

3.5.5 Greek numerals

`\greeknumeralsix`
`\greeknumeralSix`
`\greeknumeralninety`
`\greeknumeralNinety`

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

```
\providecommand*{\greeknumeralsix}{\textstigma}
\providecommand*{\greeknumeralSix}{\textStigma}
\providecommand*{\greeknumeralninety}{\textkoppa}
\providecommand*{\greeknumeralNinety}{\textKoppa}
```

`\greeknumeral` The commands `\greeknumeral` and `\Greeknumeral` 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).

```
\def\greeknumeral#1{%
  {\greekscript
   \bbl@greek@GreekNum\@firstoftwo{#1}}%
}
```

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

```
\def\Greeknumeral#1{%
  {\greekscript
   \bbl@greek@GreekNum\@secondoftwo{#1}}%
}
```

`\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).

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

`\bbl@greek@GreekNum`
`\bbl@greek@@GreekNum`
`\bbl@greek@GreekNumI`
`\bbl@greek@GreekNumII`
`\bbl@greek@GreekNumIII`
`\bbl@greek@GreekNumIV`
`\bbl@greek@GreekNumV`
`\bbl@greek@GreekNumVI`

The auxiliary macros provide the actual conversion. They are taken from hyperref as well.

```
\def\bbl@greek@GreekNum#1#2{%
```

```
    \ifnum#2<\@ne
      \bbl@greek@ill@value{#2}%
    \else
      \ifnum#2<1000000 %
        \bbl@greek@@GreekNum#1{#2}%
      \else
        \bbl@greek@ill@value{#2}%
      \fi
    \fi
}
\def\bbl@greek@@GreekNum#1#2{%
  \ifnum#2<\@m
    \ifnum#2<10 %
      \expandafter\bbl@greek@GreekNumI
          \expandafter\@gobble\expandafter#1\number#2%
    \else
      \ifnum#2<100 %
        \expandafter\bbl@greek@GreekNumII
            \expandafter\@gobble\expandafter#1\number#2%
      \else
        \expandafter\bbl@greek@GreekNumIII
            \expandafter\@gobble\expandafter#1\number#2%
      \fi
    \fi
    \ifnum#2>\z@
      \textnumeralsigngreek
    \fi
  \else
    \ifnum#2<\@M
      \expandafter\bbl@greek@GreekNumIV\expandafter#1\number#2%
    \else
      \ifnum#2<100000 %
        \expandafter\bbl@greek@GreekNumV\expandafter#1\number#2%
      \else
        \expandafter\bbl@greek@GreekNumVI\expandafter#1\number#2%
      \fi
    \fi
  \fi
}
\def\bbl@greek@GreekNumI#1#2#3{%
  #1{%
    \ifnum#3>\z@
      \textnumeralsignlowergreek
    \fi
  }%
  \expandafter#2%
  \ifcase#3 %
    {}{}%
  \or\textalpha\textAlpha
  \or\textbeta\textBeta
  \or\textgamma\textGamma
  \or\textdelta\textDelta
  \or\textepsilon\textEpsilon
  \or\greeknumeralsix\greeknumeralSix % stigma or digamma
  \or\textzeta\textZeta
  \or\texteta\textEta
  \or\texttheta\textTheta
  \else
    {}{}%
  \fi
}
\def\bbl@greek@GreekNumII#1#2#3#4{%
  #1{%
    \ifnum#3>\z@
      \textnumeralsignlowergreek
    \fi
  }%
  \expandafter#2%
  \ifcase#3 %
    {}{}%
  \or\textiota\textIota
  \or\textkappa\textKappa
  \or\textlambda\textLambda
  \or\textmugreek\textMu
  \or\textnu\textNu
  \or\textxi\textXi
  \or\textomicron\textOmicron
  \or\textpi\textPi
  \or\greeknumeralninety\greeknumeralNinety % koppa or qoppa
  \else
    {}{}%
  \fi
  \bbl@greek@GreekNumI#1#2#4%
}
\def\bbl@greek@GreekNumIII#1#2#3#4#5{%
  #1{%
    \ifnum#3>\z@
      \textnumeralsignlowergreek
    \fi
  }%
  \expandafter#2%
  \ifcase#3 %
    {}{}%
  \or\textrho\textRho
  \or\textsigma\textSigma
  \or\texttau\textTau
  \or\textupsilon\textUpsilon
  \or\textphi\textPhi
  \or\textchi\textChi
  \or\textpsi\textPsi
  \or\textomega\textOmega
  \or\textsampigreek\textSampigreek
  \else
    {}{}%
  \fi
  \bbl@greek@GreekNumII#1#2#4#5%
}
\def\bbl@greek@GreekNumIV#1#2#3#4#5{%
```

```
  \bbl@greek@GreekNumI\@firstofone#1#2%
  \bbl@greek@@GreekNum#1{#3#4#5}%
}
\def\bbl@greek@GreekNumV#1#2#3#4#5#6{%
  \bbl@greek@GreekNumII\@firstofone#1#2#3%
  \bbl@greek@@GreekNum#1{#4#5#6}%
}
\def\bbl@greek@GreekNumVI#1#2#3#4#5#6#7{%
  \bbl@greek@GreekNumIII\@firstofone#1#2#3#4%
  \bbl@greek@@GreekNum#1{#5#6#7}%
}
```

`\greek@alph`
`\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 `\expandafter`s are needed in order to make sure the table of contents will be correct, e.g., when we have appendixes.

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

Redefine the internal macros `\@alph` and `\@Alph` in the language hook, so that we use Greek numerals[17] instead of the Latin alphabet[18] in Greek text parts.

```
\addto\extrasgreek{%
  \babel@save\@alph
  \babel@save\@Alph
  \let\@alph\greek@alph
  \let\@Alph\greek@Alph
}
```

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 `"`.

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 (ς) is used. The 'v' after an 's' overrides this ligature mechanism so that it is possible to typeset an isolated σ without it becoming a ς. Because of this we make sure its lowercase code is not changed.

```
\ifx\greekfontencoding\bbl@greek@LGR
  \addto\extrasgreek{%
    \babel@savevariable{\lccode`v}\lccode`v=`v%
    \babel@savevariable{\lccode`\'}\lccode`\'=`\'%
    \babel@savevariable{\lccode`\"}\lccode`\"=`\"%
```

---

[17] cf. section 3.5.5

[18] Eventually interpreted as Latin transliteration and converted to Greek letters in a "strange" order.

```
  }
  \addto\extraspolutonikogreek{%
    % \l@greek=\bbl@polygreek
    \babel@savevariable{\lccode`\<}\lccode`\<=`\<%
    \babel@savevariable{\lccode`\>}\lccode`\>=`\>%
    \babel@savevariable{\lccode`\~}\lccode`\~=`\~%
    \babel@savevariable{\lccode`\|}\lccode`\|=`\|%
    \babel@savevariable{\lccode`\`}\lccode`\`=`\`%
  }
```

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.

```
  \addto\extrasgreek{%
    % ``high bit characters'': set in a loop and correct exceptions
    \@tempcnta=128%
    \@whilenum\@tempcnta<253\do{%
      \expandafter\babel@savevariable\expandafter{%
                         \expandafter\lccode\the\@tempcnta}%
      \lccode\@tempcnta=\@tempcnta
      \advance\@tempcnta\@ne
    }%
    % Fix non-word characters:
    \lccode151=0%
    \lccode155=0%
    \lccode159=0%
    \lccode199=0%
    % Fix capital letters:
    \lccode195=147% GREEK LETTER DIGAMMA
    \lccode219=240% GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
    \lccode223=244% GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
  }
```

`\bbl@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.

```
  \DeclareTextSymbol{\bbl@greek@tilde}{LGR}{126}
  \addto\bbl@greek@setup@polytonic{
    \declare@shorthand{greek}{~}{\bbl@greek@tilde}
  }
\fi % End of LGR-specific code.
```

3.7 MakeUppercase fixes

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

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 ≥ 2.4 suppress

---

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

them on Greek letters. The no-MakeUppercase-fixes modifier can be used to skip this step (cf. section 2.2).

```
\bbl@xin@{,no-MakeUppercase-fixes,}{,\BabelModifiers,}%
\ifin@
\else
  \ProvideTextCommandDefault{\accACUTE}{\@tabacckludge'}
  \ProvideTextCommandDefault{\accGRAVE}{\@tabacckludge`}
  \ProvideTextCommandDefault{\accTILDE}{\@tabacckludge~}
  \ProvideTextCommandDefault{\accDIAERESIS}{\@tabacckludge"}
  \addto\@uclclist{\'\accACUTE \`\accGRAVE \~\accTILDE \"\accDIAERESIS}%
\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`.[20] The optional "locale" argument `[el]` restricts the mapping to Greek text parts.

```
\ifx\greekfontencoding\bbl@greek@LGR
  \ifdefined\DeclareUppercaseMapping % new in 2023
    % \DeclareUppercaseMapping[el]{"1FBE}{\prosgegrammeni}%
    \DeclareUppercaseMapping[el]{"0027}{}% '
    \addto\bbl@greek@setup@polytonic{
      \DeclareUppercaseMapping[el]{"003C}{}% <
      \DeclareUppercaseMapping[el]{"003E}{}% >
      \DeclareUppercaseMapping[el]{"0060}{}% `
      \DeclareUppercaseMapping[el]{"007E}{}% ~
    }
  \fi
```

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.

```
  \IfFormatAtLeastTF{2022/06/01}{}
    {% else (LaTeX format older than 2022/06/01)
     \addto\extrasgreek{%
       \babel@savevariable{\uccode`\"}\uccode`\"=`\"%
       \babel@savevariable{\uccode`\'}\uccode`\'=159% 159 == ^^9f
     }
     \addto\extraspolutonikogreek{%
       \babel@savevariable{\uccode`\~}\uccode`\~=159%
       \babel@savevariable{\uccode`\>}\uccode`\>=159%
       \babel@savevariable{\uccode`\<}\uccode`\<=159%
       \babel@savevariable{\uccode`\`}\uccode`\`=159%
     }
```

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.[21] To be independent of inputenc, we do not use `\DeclareInputText` but code modelled after its definition to declare an empty expansion.

```
     \bgroup
        \uccode`\~159%
        \uppercase{%
```

---

[20] cf. LaTeX News 37

[21] Since UTF-8 became the default encoding (cf. LaTeX News 28), an "inputenc" error is also thrown if the inputenc package is not loaded.

```
     \egroup
        \def~{}%
     }
```

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).

```
     \DeclareTextCompositeCommand{\"}{LGR}{^^9f}{\accdialytika}
     \DeclareTextCompositeCommand{\'}{LGR}{^^9f}{\LGR@hiatus}
     \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.

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

3.8 Symbol name aliases

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

```
\providecommand*{\anwtonos}{\textdexiakeraia}
\providecommand*{\katwtonos}{\textaristerikeraia}
\providecommand*{\qoppa}{\textkoppa}
\providecommand*{\varqoppa}{\textqoppa}
\providecommand*{\stigma}{\textstigma}
\providecommand*{\sampi}{\textsampi}
\providecommand*{\Digamma}{\textDigamma}
\providecommand*{\ddigamma}{\textdigamma}
\providecommand*{\vardigamma}{\textvardigamma}
\providecommand*{\euro}{\texteuro}
\providecommand*{\permill}{\textperthousand}
\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.

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

Change History

babel-greek-1.08
- 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

- `\greek@Alph`: Save/restore expansion of `\alph` and `\Alph` with every switch to/from Greek. 20

babel-greek-1.13
- General: Don't use text command in math mode. 8
  - New language attribute keep-semicolon. 8
  - Setup `\languageshorthands` for all language variants. 6
- `\bbl@greek@tilde`: Renamed from `\greek@tilde`. Simplified definition. 21

babel-greek-1.13.2
- General: `\MakeUppercase` fix for transliteration input. 21
  - Warn of unsupported modifiers. 8

babel-greek-1.14
- General: `\MakeUppercase` fix for standard accent macros. 21
  - New modifier local-MakeUppercase-fixes. 3

babel-greek-1.15
- General: Fix standard accent macros also with pre-2022 `\MakeUppercase`. 21
  - Rename modifier local-MakeUppercase-fixes to no-MakeUppercase-fixes. 3
- `\BabelGreekRestoreFontEncoding`: Do not "restore" LGR when leaving Greek. 10

greek-1.0b
- General: Use `\LdfInit` to perform initial checks. 5
  - Moved the definition of `\atcatcode` right to the beginning 1
  - Now use `\ldf@finish` to wrap up 23
  - Replaced `\undefined` with `\@undefined` and `\empty` with `\@empty` for consistency with LaTeX 1
- `\lgrfont`: Added a level of braces to keep encoding change local 11

greek-1.0c
- `\bbl@greek@tilde`: Added command 21

greek-1.1
- `\Grtoday`: Added macro `\Grtoday` 17

greek-1.10
- General: Fix: `\qoppa` is the legacy name of `\textkoppa` not `\textqoppahdpindex{}\pfill23`

greek-1.1a
- `\dategreek`: Fixed typo, `Oktwbr'iou` instead of `Oktobr'iou` 17
- `\greek@Alph`: removed two superfluous @'s which made `\@alph` undefined 20

greek-1.1b
- General: Added shorthand for `\char255` 22
  - Added setting of `\uccodes` (after `kdgreek.sty`) 21
- `\bbl@greek@tilde`: Made tilde expand to a tilde with `\catcode 12`. 21

greek-1.1c
- General: Added a couple of symbols, needed for `\greeknumeral` 23
  - fixed two typos 20

greek-1.1d
- `\dategreek`: Macro `\gr@month` now produces the name of the month 17

greek-1.1e
- General: Shorthand is changed. Active character is now `\char159` 22
  - Added caption name for proof 13
  - Added lowercase code for v . . . 20
  - Added uppercase code for special letter "v". Uppercase code for accents is now `9f`, instead of `ff` 21
  - Most symbols are removed and are now defined in package grsymb 23
- `\gr@month`: Macro added 15

greek-1.2
- General: Added caption names for `\polutonikogreek` 14
  - Added lowercase codes for "modern" greek 20
  - Added uppercase codes for "modern" Greek. The old codes are now for "Polutoniko" Greek 21
  - Classical Greek is now a dialect 1

greek-1.5
- General: `\@roman` and `\@Roman` as TextCommands (BUG: this extended the expansion problem to all languages) . . . 12
  - bugfixes, change some symbol macros to aliases, LGR fixes via `\DeclareTextCommand` instead of extrasgreek/noextrasgreek definitions, LICR macros in string definitions, LGR font encoding not used with XeTeX/LuaTeX. 1
  - change symbol macros to aliases 23
  - 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
  - Support XeTeX/LuaTeX. 21
- `\textampersand`: Make `\&` a TextCommand 11

greek-1.5a
- General: provide `\extraspolutonikogreek` also for Xe/LuaTeX. 6
  - Replaced non-printable literal character with ˆ-notation (tip by Heiko Oberdiek). 23

greek-1.6
- General: Apply a patch by Enrico Gregorio. Thanks to Claudio Beccari for testing and reporting. 12
  - fix `\@roman` and `\@Roman` redefinition (thanks to Enrico Gregorio and Claudio Beccari), load LICR macro definitions for Xe/LuaTeX. 1

greek-1.7
- General: Do not load euenc.def with XeTeX/LuaTeX (too complicated to get it right). . . . 8
  - Do not load euenc.def with XeTeX/LuaTeX. Prevent re-loading lgrenc.def. 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, `\textrademark` misses a ‘t’, `\copyright` should be `\textcopyright` 11

greekfdd-2.2d
- General: removed redefinition of `\&` 11