Π Ο Λ Υ Γ Λ Ω Σ Σ Ι Α

Polyglossia: A Babel Replacement for XeLaTeX and LuaLaTeX

François Charette
Current maintainer: Arthur Reutenauer

2013/05/15 v1.32.0
(pdf file generated on 15 May 2013)

Contents

1 Introduction

Polyglossia is a package for facilitating multilingual typesetting with XeLaTeX and (at an early stage) LuaLaTeX. Basically, it can be used as a replacement of babel for performing the following tasks automatically:

1. Loading the appropriate hyphenation patterns.
2. Setting the script and language tags of the current font (if possible and available), via the package fontspec.
3. Switching to a font assigned by the user to a particular script or language.
4. Adjusting some typographical conventions according to the current language (such as afterindent, frenchindent, spaces before or after punctuation marks, etc.).

5. Redefining all document strings (like “chapter”, “figure”, “bibliography”).
6. Adapting the formatting of dates (for non-Gregorian calendars via external packages bundled with polyglossia: currently the Hebrew, Islamic and Farsi calendars are supported).
7. For languages that have their own numbering system, modifying the formatting of numbers appropriately (this also includes redefining the alphabetic sequence for non-Latin alphabets).[1]
8. Ensuring proper directionality if the document contains languages that are written from right to left (via the package bidi, available separately).

Several features of babel that do not make sense in the XeTeX world (like font encodings, shorthands, etc.) are not supported. Generally speaking, polyglossia aims to remain as compatible as possible with the fundamental features of babel while being cleaner, light-weight, and modern. The package antomega has been very beneficial in our attempt to reach this objective.

Requirements: The current version of polyglossia makes use of some convenient macros defined in the etoolbox package by Philipp Lehmann. Being designed for XeLaTeX and LuaLaTeX, it obviously also relies on fontspec by Will Robertson. For languages written from right to left, it needs the package bidi by Vafa Khalighi (وفا خليقي). Polyglossia also bundles three packages for calendaric computations (hebrewcal, hijrical, and farsical).

2 Loading language definition files

2.1 The recommended way

You can determine the default language by means of the command:

\setdefaultlanguage `\setdefaultlanguage[⟨options⟩]{lang}`

\setmainlanguage (or equivalently `\setmainlanguage`). Secondary languages can be loaded with

\setotherlanguage `\setotherlanguage[⟨options⟩]{lang}.`

These commands have the advantage of being explicit and of allowing you to set language-specific options.[2] It is also possible to load a series of secondary languages at once using

\setotherlanguages `\setotherlanguages{lang1,lang2,lang3,…}.`

Language-specific options can be set or changed at any time by means of

\setkeys `\setkeys{⟨lang⟩}{opt1=value1,opt2=value2,…}.`

[1] For the Arabic script this is now done by the bundled package arabicnumbers.

[2] More on language-specific options below.

2.2 The "Babel way"

v1.2.0 ← **Warning**: polyglossia no longer supports loading language definition files as package options!

2.3 Supported languages

Table 2.3 lists all languages currently supported. Those in red have specific options and/or commands that are explained in section 5 below.

albanian	croatian	hebrew	norsk	swedish
amharic	czech	hindi	nynorsk	syriac
arabic	danish	icelandic	occitan	tamil
armenian	divehi	interlingua	polish	telugu
asturian	dutch	irish	portuges	tibetan
bahasai	english	italian	romanian	thai
bahasam	esperanto	lao	russian	turkish
basque	estonian	latin	samin	turkmen
bengali	farsi	latvian	sanskrit	ukrainian
brazil[ian]	finnish	lithuanian	scottish	urdu
breton	french	lsorbian	serbian	usorbian
bulgarian	galician	magyar	slovak	vietnamese
catalan	german	malayalam	slovenian	welsh
coptic	greek	marathi	spanish	

Table 1: Languages currently supported in polyglossia

v1.0.1 v1.1.1 v1.2.0 *NB:* The support for Amharic ← should be considered an experimental attempt to port the package ethiop.[3] Version 1.1.1 ← addded support for Asturian, Lithuanian, and Urdu. Version 1.2 ← adds support for Armenian, Occitan, Bengali, Lao, Malayalam, Marathi, Tamil, Telugu, and Turkmen.[4]

v1.1.1 Polyglossia can also be loaded with the option 'babelshorthands' ←, which globally activates babel shorthands whenever available. Currently shorthands are implemented for Catalan, Dutch, German, Italian, and Russian: see these respective languages for details.

v1.2.0 Another option (turned off by default) is 'localmarks', which redefines the internal LaTeX macros `\markboth` and `\markright`. ←Note that this was formerly

[3] Feedback is welcome.

[4] See acknowledgements at the end for due credit to the various contributors.

turned on by default, but ẃe now realize that it causes more problems than otherwise. For backwards-compatibility the opposite option 'nolocalmarks' is still available.

There is also the option 'quiet' which turns off most info messages and some of the warnings issued by LaTeX, fontspec and polyglossia.

3 Language-switching commands

\text⟨lang⟩

⟨lang⟩

Whenever a language definition file `gloss-⟨lang⟩.ldf` is loaded, the command `\text⟨lang⟩[⟨options⟩]{…}` becomes available for short insertions of text in that language. For example `\textrussian{\today}` yields 15 мая 2013 г. Longer passages are better put between the environment `⟨lang⟩` (again with the possibility of setting language options locally. For instance the following allows us to quote the beginning of Homer's *Iliad*:

```
\begin{greek}[variant=ancient]
μῆνιν ἄειδε θεὰ Πηληϊάδεω Ἀχιλῆος οὐλομένην, ἣ μυρί' Ἀχαιοῖς ἄλγε'
ἔθηκε, πολλὰς δ' ἰφθίμους ψυχὰς Ἄϊδι προΐαψεν ἡρώων, αὐτοὺς δὲ ἑλώρια
τεῦχε κύνεσσιν οἰωνοῖσί τε πᾶσι, Διὸς δ' ἐτελείετο βουλή, ἐξ οὗ δὴ τὰ
πρῶτα διαστήτην ἐρίσαντε Ἀτρείδης τε ἄναξ ἀνδρῶν καὶ δῖος Ἀχιλλεύς.
\end{greek}
```

μῆνιν ἄειδε θεὰ Πηληϊάδεω Ἀχιλῆος οὐλομένην, ἣ μυρί' Ἀχαιοῖς ἄλγε' ἔθηκε, πολλὰς δ' ἰφθίμους ψυχὰς Ἄϊδι προΐαψεν ἡρώων, αὐτοὺς δὲ ἑλώρια τεῦχε κύνεσσιν οἰωνοῖσί τε πᾶσι, Διὸς δ' ἐτελείετο βουλή, ἐξ οὗ δὴ τὰ πρῶτα διαστήτην ἐρίσαντε Ἀτρείδης τε ἄναξ ἀνδρῶν καὶ δῖος Ἀχιλλεύς.

Arabic

Note that for Arabic one cannot use the environment `arabic`, as `\arabic` is defined internally by LaTeX. In this case we need to use the environment `Arabic` instead.

3.1 Other commands

The following commands are probably of lesser interest to the end user, but ought to be mentioned here.

\selectbackgroundlanguage

- `\selectbackgroundlanguage`: this selects the global font setup and the numbering definitions for the default language.

\resetdefaultlanguage
\normalfontlatin
\rmfamilylatin
\sffamilylatin
\ttfamilylatin
\selectlanguage
\foreignlanguage
otherlanguage

- `\resetdefaultlanguage` (experimental): completely switches the default language to another one in the middle of a document: *this may have adverse effects*!
- `\normalfontlatin`: in an environment where `\normalfont` has been redefined to a non-latin script, this will call the font defined with `\setmainfont` etc. Likewise it is possible to use `\rmfamilylatin`, `\sffamilylatin`, and `\ttfamilylatin`.
- Some macros defined in babel's `hyphen.cfg` (and thus usually compiled into the XeLaTeX and LuaLaTeX format) are redefined, but keep a similar behaviour, namely `\selectlanguage`, `\foreignlanguage`, and the environment `otherlanguage`.

Since the XeLaTeX and LuaLaTeX format incorporate babel's `hyphen.cfg`, the low-level commands for hyphenation and language switching defined there are also accessible.

4 Font setup

With polyglossia it is possible to associate a specific font with any script or language that occurs in the document. That font should always be defined as `\⟨script⟩font` or `\⟨language⟩font`. For instance, if the default font defined by `\setmainfont` does not support Greek, then one can define the font used to display Greek with:

`\newfontfamily\greekfont[Script=Greek,⟨…⟩]{⟨font⟩}`.

Note that polyglossia will use the font thus defined as is. for instance if `\arabicfont` is explicitly defined, then one should take care of including the option `Script=Arabic` in that definition. See the fontspec documentation for more information. If a specific sans or monospace font is needed for a particular script or language, it can be defined by means of ← `\⟨script⟩fontsf` or `\⟨language⟩fontsf` and `\⟨script⟩fonttt` or `\⟨language⟩fonttt`, respectively.

v1.2.0

Whenever a new language is activated, polyglossia will first check whether a font has been defined for that language or – for languages in non-Latin scripts – for the script it uses. If it is not defined, it will use the currently active font and – in the case of OpenType fonts – will attempt to turn on the appropriate OpenType tags for the script and language used, in case these are available in the font, by means of fontspec's `\addfontfeature`. If the current font does not appear to support the script of that language, an error message is displayed.

5 Language-specific options and commands

This section gives a list of all languages for which options and end-user commands are defined. The default value of each option is given in italic.

5.1 arabic

Options:

- **calendar** = *gregorian* or islamic (= hijri)
- **locale** = *default*,[5] mashriq,[6] libya, algeria, tunisia, morocco, or mauritania. This setting influences the spelling of the month names for the Gregorian calendar, as well as the form of the numerals (unless overriden by the following option).
- **numerals** = *mashriq* or maghrib (the latter is the default when locale = algeria, tunisia or morocco)
- v1.0.3 **abjadjimnotail** = *false* or true. ← Set this to true if you want the *abjad* form of the number three to be ـﺠ – as in the manuscript tradition – instead of the modern usage ج.

Commands:

- `\abjad` and `\abjadmaghribi` (see section 8)
- `\aemph` to emphasize text with `\overline`. ← `\textarabic{\aemph{ا ب}}` yields ا̅ب̅. This command is also available for Farsi, Urdu, etc.

\abjad
\abjadmaghribi
\aemph
v1.2.0

5.2 bengali

v1.2.0 ← **Options**:

- **numerals** = Western or *Devanagari*

5.3 catalan

Options:

- v1.1.1 **babelshorthands** = *false* or true. ← Activates the shorthands `"l` and `"L` to type geminated l's.

Commands:

- `\l.l` and `\L.L` behave as in babel to type a geminated l, as in *col·laborar*. ← In polyglossia the same can also be achieved with `\l·l` and `\L·L`.[7]

\l.l
\L.L
v1.1.1
\l·l
\L·L

[5] For Egypt, Sudan, Yemen and the Gulf states.
[6] For Iraq, Syria, Jordan, Lebanon and Palestine.
[7] NB: · is the glyph U+00B7 MIDDLE DOT.

5.4 dutch

Options:

v1.1.1

- **babelshorthands** = *false* or true. ← if this is turned on, all shorthands defined in babel for fine-tuning the hyphenation of Dutch words are activated.
  - `"-` for an explicit hyphen sign, allowing hyphenation in the rest of the word
  - `"~` for a compound word mark without a breakpoint
  - `"|` disables the ligature at this position
  - `""` is like `"-`, but produces no hyphen sign (for compound words with a hyphen, e.g., `foo-""bar`)
  - `"/` to enable hyphenation in two words written together but separated by a slash.

`\-`

- In addition, the macro `\-` is redefined to allow hyphens in the rest of the word.

5.5 english

Options:

- **variant** = *american* (= us), usmax (same as 'american' but with additional hyphenation patterns), british (= uk), australian or newzealand
- **ordinalmonthday** = true/*false* (true by default only when variant = british)

5.6 esperanto

Commands:

`\hodiau`
`\hodiaun`

- `\hodiau` and `\hodiaun` are special forms of `\today` (see the babel documentation)

5.7 farsi

Options:

- **numerals** = western or *eastern*
- **locale** (not yet implemented)
- **calendar** (not yet implemented)

Commands:

`\abjad`
- `\abjad` (see section 8)

`\aemph`
- `\aemph` (see section 5.1).

5.8 german

Options:

- **spelling** = *new* (= 1996) or old (= 1901): indicates whether hyphenation patterns for traditional (1901) or reformed (1996) orthography should be used. The latter is the default.
- **latesthyphen** = *false* or true: if this option is set to true, the latest (experimental) hyphenation patterns '(n)german-x-latest' will be loaded instead of 'german' or 'ngerman'. NB: This is based on the file `language.dat` that comes with TEXLive 2008 and later.
- **babelshorthands** = *false* or true: ← if this is turned on, all shorthands defined in babel for fine-tuning the hyphenation of German words are activated. v1.0.3
  - `"ck` for `ck` to be hyphenated as `k-k`
  - `"ff` for `ff` to be hyphenated as `ff-f`; this is also available for the letters l, m, n, p, r and t
  - `"|` disables the ligature at this position
  - `"-` for an explicit hyphen sign, allowing hyphenation in the rest of the word
  - `""` is like `"-`, but produces no hyphen sign (for compound words with a hyphen, e.g., `foo-""bar`)
  - `"~` for a compound word mark without a breakpoint
  - `"=` for a compound word mark with a breakpoint, allowing hyphenation in the composing words.

  There are also four shorthands for quotation signs:
  - `"`` for German left double quotes („)
  - `"'` for German right double quotes (“)
  - `"<` for French left double quotes («)
  - `">` for French right double quotes (»).
- **script** = *latin* or fraktur. ← Setting script=fraktur modifies the captions for typesetting German in Fraktur. v1.2.0

5.9 greek

Options:

- **variant** = *monotonic* (= mono), polytonic (= poly), or ancient
- **numerals** = *greek* or arabic
- **attic** = *false*/true

Commands:

\Greeknumber
\greeknumber
\atticnumeral
\atticnum

- `\Greeknumber` and `\greeknumber` (see section 8).
- The command `\atticnumeral` (= `\atticnum`) (activated with the option `attic=true`), displays numbers using the acrophonic numbering system (defined in the Unicode range U+10140–U+10174).[8]

5.10 hebrew

Options:

- **numerals** = hebrew or *arabic*
- **calendar** = hebrew or *gregorian*

Commands:

\hebrewnumeral
\hebrewalph
\aemph

- `\hebrewnumeral` (= `\hebrewalph`) (see section 8).
- `\aemph` (see section 5.1).

5.11 hindi

v1.2.0

← **Options**:

- **numerals** = Western or *Devanagari*

5.12 italian

Option:

v1.2.0cc

- **babelshorthands** = *false* or true. ←Activates the `"` character as a switch to perform etymological hyphenation when followed by a letter, or other tasks when followed by certain analphabetic characters; in particular `""` is used to enter double raised open quotes (the Italian keyboard misses the backtick), and `"<` and `">` to insert open and closed guillemets without any spacing after the open or before the closed sign. `"/` is made equivalent to /allowing a linebreak after the slash without any hyphen sign; `"-` produces a short rule/hyphen and a discretional line break alowing line breaks in the second compound word fragment.

5.13 lao

v1.2.0

← **Options**:

- **numerals** = lao or *arabic*

[8] See the documentation of the xgreek package for more details.

5.14 lsorbian and usorbian

Commands:

\oldtoday
- `\oldtoday`: see the babel documentation.

5.15 magyar

Commands:

\ontoday
\ondatemagyar
- `\ontoday` (= `\ondatemagyar`): special forms of `\today` (see the babel documentation).

5.16 russian

Options:

- **babelshorthands** = *false* or true.
- **spelling** = *modern* or old (for captions and date only, not for hyphenation)

5.17 sanskrit

Options:

v1.0.2
- **Script** (default = Devanagari). ← The value is passed to fontspec in cases where `\sanskritfont` or `\devanagarifont` are not defined. This can be useful if you typeset Sanskrit texts in scripts other than Devanagari.

5.18 serbian

Options:

- **script** = *cyrillic* or latin

5.19 syriac

Options:

v1.0.1
- **numerals** = *western* (i.e., 1234567890), eastern (for which the Oriental Arabic numerals are used: ١٢٣٤٥٦٧٨٩٠), or abjad. ←.

Commands:

\abjadsyriac
- `\abjadsyriac` (see section 8)

\aemph
- `\aemph` (see section 5.1).

5.20 thai

Options:

- **numerals** = thai or *arabic*

To insert the word breaks, you need to use an external processor. See the documentation to thai-latex and the file `testthai.tex` that comes with this package.

6 Modifying or extending captions and date formats

To redefine internal macros, you can use the command `\gappto` from the package etoolbox. For compatibility with babel the command `\addto` is also available with the same effect. For instance, to change the `\chaptername` for language `lingua`, you can do this:

```
\gappto\captionslingua{\renewcommand{\chaptername}{Caput}}
```

7 Non-Western decimal digits

Several scripts have their own versions of the decimal digits commonly called 'Arabic numerals'. With the appropriate language option set, polyglossia will automatically convert the output of internal LaTeX counters to their localized forms, for instance to display page, chapter and section numbers.

In previous versions this conversion was achieved my means of TECKit fontmappings. If needed they can be activated with the fontspec `Mapping` option, using `arabicdigits`, `farsidigits` or `thaidigits`. For instance if `\arabicfont` is defined with the option `Mapping=arabicdigits`, then by typing `\textarabic{2010}` one will obtain ٢٠١٠.

v1.1.1

With version v1.1.1 ← the same conversion is achieved directly by simple TeX macros. This prevents some problems that occur when the value of a counter has to be written and read from auxiliary files.[9] These macros (currently `\arabicdigits`, `\farsidigits` and `\thaidigits` are provided) are also available to the users. For instance in an Arabic environment `\arabicdigits{9182/738543-X}` yields ٩١٨٢/٧٣٨٥٤٣-X.

`\arabicdigits`
`\farsidigits`
`\thaidigits`

[9] For instance the package lastpage did not work with polyglossia in situations where the display of counters was redefined to include a font-switching command.

8 Alphabetic numbering in Greek, Arabic, Hebrew, Syriac and Farsi

In certain languages, numbers can be represented by a special alphanumerical notation.[10]

\greeknumeral
\Greeknumeral

The Greek numerals are obtained with `\greeknumeral` (or `\Greeknumeral` in uppercase). Example: `\greeknumeral{1863}` yields ,αωξγ΄.

\abjad

The Arabic *abjad* numbers can be generated with the command `\abjad`. Example: `\abjad{1863}` yields غضسج. In the Maghrib the conventions are somewhat different, and the maghribi forms of the *abjad* numerals are obtained with the `\abjadmaghribi` command. Example: `\abjadmaghribi{1863}` yields شظصج.

\abjadmaghribi

v1.1.1
\hebrewnumeral
\Hebrewnumeral
\Hebrewnumeralfinal

The code for Hebrew numerals, which was incorrect in previous versions, was ported from the implementation in babel with v1.1.1 ←, and the user interface is identical to the one in babel. The commands `\hebrewnumeral`, `\Hebrewnumeral` and `\Hebrewnumeralfinal` behave exactly as they do in babel: the second command prints the number with *gereshayim* before the last letter, and the latter uses in addition the final forms of Hebrew letters. Examples: `\hebrewnumeral{1750}` yields אתשנ, `\Hebrewnumeral{1750}` yields א״תשנ, and `\Hebrewnumeralfinal{1750}` yields א״תשן.

\abjadsyriac

Support is also provided for Syriac abjad numerals, which can be generated with `\abjadsyriac`.[11] Example: `\abjadsyriac{463}` yields ܬܟܓ.

9 Calendars

9.1 Hebrew calendar (hebrewcal.sty)

\Hebrewtoday

The package `hebrewcal.sty` is almost a verbatim copy of `hebcal.sty` that comes with babel. The command `\Hebrewtoday` formats the current date in the Hebrew calendar (depending of the current writing direction this will automatically set either in Hebrew script or in roman transliteration).

[10] See, e.g., http://en.wikipedia.org/wiki/Greek_numerals, http://en.wikipedia.org/wiki/Abjad_numerals, and http://en.wikipedia.org/wiki/Hebrew_numerals.

[11] A fine guide to numerals in Syriac can be found at http://www.garzo.co.uk/documents/syriac-numerals.pdf.

9.2 Islamic calendar (hijrical.sty)

This package computes dates in the lunar Islamic (Hijra) calendar.[12] It provides two macros for the end-user. The command

\HijriFromGregorian `\HijriFromGregorian{⟨year⟩}{⟨month⟩}{⟨day⟩}`

\Hijritoday sets the counters `Hijriday`, `Hijrimonth` and `Hijriyear`. `\Hijritoday` formats the Hijri date for the current day. This command is now locale-aware ⟵: its output will differ depending on the currently active language. Presently polyglossia's language definition files for Arabic, Farsi, Urdu, Turkish, Bahasa Indonesia and Bahasa Melayu provide a localized version of `\Hijritoday`. If the formatting macro for the current language is undefined, the Hijri date will be formatted in Arabic or in roman transliteration, depending of the current writing direction. You can define a new format or redefine one with the command

v1.1.1

\DefineHijriDateFormat `\DefineHijriDateFormat{<lang>}{<code>}.`

The command `\Hijritoday` also accepts an optional argument to add or subtract a correction (in days) to the date computed by the arithmetical algorithm.[13] For instance if `\Hijritoday` yields the date "7 Rajab 1429" (which is the date that was displayed on the front page of aljazeera.net on 11th July 2008), `\Hijritoday[1]` would rather print "8 Rajab 1429" (the date indicated the same day on the site gulfnews.com).

9.3 Farsi (jalālī) calendar (farsical.sty)

This package is an almost verbatim copy of `Arabiftoday.sty` (in the Arabi package), itself a slight modification of `ftoday.sty` in FarsiTEX.[14] Here we have renamed the command `\ftoday` to `\Jalalitoday`. Example: today is 25 Ordībehesht 1392.

\Jalalitoday

10 Acknowledgements (by François Charette)

Polyglossia is notable for being a recycle box of previous contributions by other people. I take this opportunity to thank the following individuals, whose splen-

[12] It makes use of the arithmetical algorithm in chapter 6 of Reingold & Gershowitz, *Calendrical calculation: the Millenium edition* (Cambridge University Press, 2001).

[13] The Islamic calendar is indeed a purely lunar calendar based on the observation of the first visibility of the lunar crescent at the beginning of the lunar month, so there can be differences between different localities, as well as between civil and religious authorities.

[14] One day I may rewrite farsical from scratch using the algorithm in Reingold & Gershowitz (ref. n. 12).

did work has made my task almost trivial in comparision: Johannes Braams and the numerous contributors to the babel package (in particular Boris Lavva and others for its Hebrew support), Alexej Kryukov (antomega), Will Robertson (fontspec), Apostolos Syropoulos (xgreek), Youssef Jabri (arabi), and Vafa Khalighi (xepersian and bidi). The work of Mojca Miklavec and Arthur Reutenauer on hyphenation patterns with their package hyph-utf8 is of course invaluable. I should also thank other individuals for their assistance in supporting specific languages: Yves Codet (Sanskrit), Zdenek Wagner (Hindi), Mikhal Oren (Hebrew), Sergey Astanin (Russian), Khaled Hosny (Arabic), Sertaç Ö. Yıldız (Turkish), Kamal Abdali (Urdu), and several other members of the XɜTEX user community, notably Enrico Gregorio, who has sent me many useful suggestions and corrections and contributed the `\newXeTeXintercharclass` mechanism in xelatex.ini which is now used by polyglossia. More recently, Kevin Godby of the Ubuntu Manual project has contributed very useful feedback, bug hunting and, with the help of translators, new language definition files for Asturian, Lithuanian, Occitan, Bengali, Malayalam, Marathi, Tamil, and Telugu. It is particularly heartening to realize that this package is used to typeset a widely-read document in dozens of different languages! Support for Lao was also added thanks to Brian Wilson. I also thank Alan Munn for kindly proof-reading the penultimate version of this documentation. And of course my gratitude also goes to Jonathan Kew, the formidable author of XɜTEX!