Polyglossia: An Alternative to Babel for \LaTeX\ and \texttt{Lua\LaTeX}\n
François Charette
Current maintainer: Arthur Reutenauer

2019/04/04 v1.44
(pdf file generated on 4 April 2019)

Contents

1 Introduction 1

2 Loading language definition files 2
  2.1 The recommended way ........................................... 2
  2.2 The "Babel way" – obsolete .................................... 2
  2.3 Supported languages ............................................. 3

3 Language-switching commands 4
  3.1 Other commands .................................................. 4

4 Font setup 5

5 Hyphenation disabling 5

6 Language-specific options and commands 6
  6.1 arabic ............................................................... 6
  6.2 bengali ............................................................. 6
  6.3 catalan ............................................................... 6
  6.4 dutch ................................................................. 7
  6.5 english ............................................................. 7
1 Introduction

Polyglossia is a package for facilitating multilingual typesetting with \LaTeX{} and (at an early stage) Lua\LaTeX{}. Basically, it can be used as an alternative to \texttt{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 \texttt{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).
8. Ensuring proper directionality if the document contains languages that are written from right to left (via the package \texttt{bidi}, available separately).

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

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

\footnote{1For the Arabic script this is now done by the bundled package \texttt{arabicnumbers}.}
2 Loading language definition files

2.1 The recommended way

You can determine the default language by means of the command:
\setdefaultlanguage⟨options⟩{lang}
(\setdefaultlanguage{lang})

Secondary languages can be loaded with
\setotherlanguage⟨options⟩{lang}.
\setotherlanguage{lang}.

These commands have the advantage of being explicit and of allowing you to
set language-specific options.\footnote{More on language-specific options below.}
It is also possible to load a series of secondary
languages at once using
\setotherlanguages{lang1,lang2,lang3,…}.
\setotherlanguages{lang1,lang2,lang3,…}.

Language-specific options can be set or changed at any time by means of
\setkeys{⟨lang⟩}{opt1=value1,opt2=value2,…}.
\setkeys{⟨lang⟩}{opt1=value1,opt2=value2,…}.

2.2 The “Babel way” – obsolete

\footnote{Feedback is welcome.}

\textbf{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 op-
tions and/or commands that are explained in section 6 below.

\textbf{NB:} The support for Amharic \footnote{See acknowledgements at the end for due credit to the various contributors.}
should be considered an experimental
attempt to port the package ethiop.\footnote{Feedback is welcome.}
Version 1.1.1 \footnote{More on language-specific options below.}
added support for Asturian, Lithuanian, and Urdu. Version 1.2 \footnote{Feedback is welcome.}
adds support for Armenian, Occitan, Bengali, Lao, Malayalam, Marathi, Tamil, Telugu, and Turkmen.\footnote{See acknowledgements at the end for due credit to the various contributors.}

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 re-
spective languages for details.

Another option (turned off by default) is 'localmarks', which redefines the
internal \LaTeX macros \texttt{\textbackslash markboth} and \texttt{\textbackslash markright}. \footnote{Feedback is welcome.}
Note that this was formerly
turned on by default, but we now realize that it causes more problems than oth-
erwise. For backwards-compatibility the opposite option 'nolocalmarks' is still
Table 1: Languages currently supported in polyglossia available.

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

3 Language-switching commands

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 4 апреля 2019 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}
μὴν ἂν ἔτει τέλη ἡ θεᾶ Πηληϊάδες Ἀχιλῆος οὐλομένην, ἤ μυρί' ἄχαιοις ἄλγε' ἔθηκε, πολλὰς δ' ἱφτίμους ψυχὰς ἀδίδ προίαψεν ἥρων, αὐτοὺς δὲ ἐλώρα τευχεῖ κύνεσιν οἰωνοῖς τε πάσι, Διὸς δ' ἔτελείτο βουλή, ἐξ οὗ δὴ τὰ πρῶτα διστάτην ἔρισαντε ἀτρείδης τε ἀναξ ἀνδρῶν καὶ δίος Ἀχιλλεὺς.
\end{greek}
μῆνιν ἄειδε θεὰ Πηληΐαδεω Αχιλήος οὐλομένην, ἢ μυρί’ Ἀχαιοῖς ἄλγε’ έθηκε,
pολλάς δ’ ἴφθιμος ψυχὰς Ἄϊδι προί̈αψεν ἥρων, αὐτοὺς δὲ ἐλώρια τεῦχε κύ-
νεσσιν οἰωνοῖσι τε πάσι, Διός δ’ ἐτελείετο βουλή, εξ οὔ δ’ τὰ πρῶτα διαστήτην
ἐρίσαντε Ατρεί̈δης τε ἄναξ ἀνδρῶν καὶ δῖος Αχιλλεύς.

Note that for Arabic one cannot use the environment \arabic, as \arabic is
defined internally by \TeX. 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
\resetdefaultlanguage
\resetdefaultlanguage (experimental): completely switches the default
language to another one in the middle of a document: this may have adverse
effects!
\normalfontlatin
\normalfontlatin: in an environment where \normalfont has been re-
defined to a non-latin script, this will call the font defined with \setmainfont
\rmfamilylatin
\rmfamilylatin, \sffamilylatin, and
\ttfamilylatin
\selectlanguage
\foreignlanguage
\selectlanguage
\foreignlanguage
otherlanguage
Since the \TeX and Lua\TeX 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
\langle script\rangle font or \langle language\rangle font. For instance, if the default font defined by
\setmainfont does not support Greek, then one can define the font used to dis-
play Greek with:

\newfontfamily\greekfont[Script=Greek,\{\ldots\}]{\langle font\rangle}.
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 \langle script \rangle fontsf or \langle language \rangle fontsf and \langle script \rangle fonttt or \langle language \rangle fonttt, respectively.

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 Hyphenation disabling

In some very specific contexts (such as music score creation), \TeX hyphenation is something to avoid as it may cause troubles. polyglossia provides two functions: \disablehyphenation and \enablehyphenation. Note that when you select a new language, hyphenation will be in the same state (enabled or disabled) as before. When you reenable it, it will take the last selected language.

6 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.

6.1 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 overridden by the following option).

---

\(^5\) For Egypt, Sudan, Yemen and the Gulf states.

\(^6\) For Iraq, Syria, Jordan, Lebanon and Palestine.
• **numerals** = *mashriq* or *maghrib* (the latter is the default when locale = algeria, tunisia or morocco)

• **abjadimnotail** = *false* or *true*. ⬤ Set this to true if you want the *abjad* form of the number three to be ج – as in the manuscript tradition – instead of the modern usage ج.

**Commands:**

\abjad
\abjadmaghribi
\aemph

yields عبد. This command is also available for Farsi, Urdu, etc.

6.2 **bengali**

← **Options:**

• **numerals** = Western, Bengali or *Devanagari*

• **changecounternumbering** = *true* or *false* (use specified numerals for headings and page numbers)

6.3 **catalan**

Options:

• **babelshorthands** = *false* or *true*. ← Activates the shorthands "l and "L to type geminated l’s.

**Commands:**

\ll \LL

\ll \LL behave as in babel to type a geminated l, as in *collaborar*. ←

In polyglossia the same can also be achieved with \l\l and \L\L.

6.4 **dutch**

Options:

• **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)

\begin{footnote}
7NB: · is the glyph U+00B7 MIDDLE DOT.
\end{footnote}
- "/" 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.

6.5 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)

6.6 esperanto

Commands:
\hodiau and \hodiaun are special forms of \today (see the babel documentation)

6.7 farsi

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

Commands:
\abjad (see section 9)
\æmph (see section 6.1).

6.8 french

Options:
- automaticspacesaroundguillemets = true or false (default value = true. Adds space after the opening guillemets and before the closing guillemets. Such space is usually not typed in source code, and you should let polyglossia add it. However, if your source code contains such space, you can set this option to false.)
- frenchfootnote = true or false (default value = true. Determines whether the footnote mark starting the footnote is normal script followed by a dot (default) or superscript without a dot.)
6.9 german

Options:

- **variant** = german, austrian or swiss. ← Setting variant=austrian or variant=swiss uses some lexical variants. With spelling=old, variant=swiss furthermore loads specific hyphenation patterns.
- **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.
  - ”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.
  - ”/ a slash that allows for a line break and maintains hyphenation points.

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.
6.10 greek

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

Commands:
- \Greeknumber and \greeknumber (see section 9).
- 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).\(^8\)

6.11 hebrew

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

Commands:
- \hebrewnumberal (= \hebrewalph) (see section 9).
- \aemph (see section 6.1).

6.12 hindi

← Options:
- **numerals** = Western or Devanagari

6.13 italian

Option:
- **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 allowing line breaks in the second compound word fragment.

\(^8\)See the documentation of the xgreek package for more details.
6.14 korean

The language definition file includes U. S. hyphenation patterns in order to enable hyphenation when writing English within Korean text.

6.15 lao

Options:

- **numerals** = lao or arabic

6.16 latin

Options:

- **variant** = classic, medieval or modern

6.17 lsorbian and usorbian

Commands:

- \oldtoday: see the babel documentation.

6.18 magyar

Commands:

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

6.19 russian

Options:

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

Commands:

- \Asbuk: produces the uppercase Russian alphabet, for environments such as enumerate
- \asbuk: same in lowercase

6.20 sanskrit

Options:
• **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.

`polyglossia` currently supports the typesetting of Sanskrit in the following writing systems: Devanagari, Gujarati, Malayalam, Bengali, Kannada, Telugu, and Latin. Use the `Script=` option to select the writing system you want, and enter your input in that script.

### 6.21 serbian

**Options:**

- `script = cyrillic` or `latin`

### 6.22 slovenian

**Options:**

- `localaph = true` or `false`

### 6.23 syriac

**Options:**

- `numerals = western` (i.e., 1234567890), `eastern` (for which the Oriental Arabic numerals are used: ٠١٢٣٤٥٦٧٨٩٠), or `abjad`.

**Commands:**

- `\abjadsyriac` (see section 9)
- `\aemph` (see section 6.1).

### 6.24 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.

### 6.25 ukrainian

**Commands:**

- `\Asbuk`: produces the uppercase Ukrainian alphabet, for environments such as `enumerate`
- `\asbuk`: same in lowercase
6.26 welsh

Options:

- **date** = long or short

7 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}}`

8 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 font-mappings. If needed they can be activated with the fontspec \texttt{Mapping} option, using \texttt{arabicdigits}, \texttt{farsidigits} or \texttt{thaidigits}. For instance if `\arabicfont` is defined with the option `\texttt{Mapping=arabicdigits}`, then by typing \texttt{\textarabic{2010}} one will obtain ٠١٠٢.

With version v1.1.1 ← the same conversion is achieved directly by simple \LaTeX{} macros. This prevents some problems that occur when the value of a counter has to be written and read from auxiliary files. For instance in an Arabic environment `\arabicdigits{9182/738543-X}` yields ٩١٨٢/٧٣٨٥٤٣-\texttt{X}.

\footnote{For instance the package \texttt{lastpage} did not work with `polyglossia` in situations where the display of counters was redefined to include a font-switching command.}
9 Alphabetic numbering in Greek, Arabic, Hebrew, Syriac and Farsi


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

\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 ﺯظصج.

\abjadmaghribi 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 א׳תשנ, \Hebrewnumeral{1750} yields א׳תש״נ, and \Hebrewnumeralfinal{1750} yields א׳תש״ן.

Support is also provided for Syriac abjad numerals, which can be generated with \abjadsyriac.\footnote{A fine guide to numerals in Syriac can be found at http://www.garzo.co.uk/documents/syriac-numerals.pdf.} Example: \abjadsyriac{463} yields ٨٣.

10 Calendars

10.1 Hebrew calendar (hebrewcal.sty)

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).
10.2 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{(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 \(\leftarrow\): 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

\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).

10.3 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 Farsi\TeX.\(^{14}\) Here we have renamed the command \ftoday to \Jalalitoday. Example: today is 15 Farvardin 1398.

11 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-


\(^{13}\)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 comparison: 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 Xe\TeX 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 Xe\TeX!

12 More acknowledgements (by Arthur Reutenauer)

Many thanks to all the people who have contributed bugfixes and new features to Polyglossia since I took over. Most of them can be identified from the version control log on GitHub and I won’t try to name them all (maybe, one day ...); among the ones who sent contributions directly to me I would like to especially thank Claudio Beccari, the indefatigable champion of Romance languages, and beyond!