Π Ο Λ Υ Γ Λ Ο Σ Σ Ι Α

Polyglossia: A Babel Replacement for XeLaTeX

François Charette

2008/07/11 v1.0
(PDF file generated on 11th July 2008)

Contents

1 Introduction

Polyglossia is a package for facilitating multilingual typesetting with XeLaTeX. 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 in function of 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 numeration system, modifying the formatting of numbers appropriately (this also includes redefining the alphabetic sequence for non-Latin alphabets).
8. Ensuring the 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 (obviously) 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 much 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 specifically for XeLaTeX, it obviously also relies on fontspec by Will Robertson. For languages written from right to left, it needs the package bidi (by the present author). 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 to set language-specific options.[1] 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,…}` .

2.2 The "Babel way"

As with babel, polyglossia also allows to load language definition files as package options. In most cases, option `⟨lang⟩` will load the file `gloss-⟨lang⟩.ldf`. Note however that the *first* language listed in

`\usepackage[lang1,lang2,…]polyglossia`

will be the default language for the document, which is the opposite convention of babel. Note also that this method may not work in some cases, and should be considered deprecated.

2.3 Supported languages

Table 2.3 lists all languages currently supported. Those in red have specific options and/or command that are explained in section 4 below.

Some options are convenient shortcuts for loading languages with specific options:

[1] More on language-specific options below.

albanian	czech	german	magyar	slovak
arabic	danish	greek	norsk	slovenian
bahasai	divehi	hebrew	nynorsk	spanish
bahasam	dutch	hindi	polish	swedish
basque	english	icelandic	portuges	syriac
brazil	esperanto	interlingua	romanian	thai
breton	estonian	irish	russian	turkish
bulgarian	farsi	italian	samin	ukrainian
catalan	finnish	latin	sanskrit	usorbian
coptic	french	latvian	scottish	welsh
croatian	galician	lsorbian	serbian	

Table 1: Languages currently supported in polyglossia

- american = english with option 'variant=american'
- USenglish = english with option 'variant=american'
- UKenglish = english with option 'variant=british'
- british = english with option 'variant=british'
- australian = english with option 'variant=australian'
- newzealand = english with option 'variant=newzealand'
- ogerman = german with option 'spelling=old'
- monogreek = greek with option 'variant=monotonic' (or 'mono')
- polygreek = greek with option 'variant=polytonic' (or 'poly')
- ancientgreek = greek with option 'variant=ancient'

Another option (turned off by default) is 'nolocalmarks', which prevents the redefinition of the internal LaTeX macros `\markboth` and `\markright`.

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 11 июля 2008 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]
τὸν δ' ἠμείβετ' ἔπειτα θεά, γλαυκῶπις Ἀθήνη: "ὦ πάτερ ἡμέτερε Κρονίδη, ὕπατε
```

```
κρειόντων, καὶ λίην κεῖνός γε ἐοικότι κεῖται ὀλέθρῳ: ὡς ἀπόλοιτο καὶ ἄλλος,
ὅτις τοιαῦτά γε ῥέζοι: ἀλλά μοι ἀμφ' Ὀδυσῆι δαίφρονι δαίεται ἦτορ, δυσμόρῳ, ὃς
δὴ δηθὰ φίλων ἄπο πήματα πάσχει νήσῳ ἐν ἀμφιρύτῃ, ὅθι τ' ὀμφαλός ἐστι θαλάσσης …"
\end{greek}
```

τὸν δ' ἠμείβετ' ἔπειτα θεά, γλαυκῶπις Ἀθήνη: "ὦ πάτερ ἡμέτερε Κρονίδη, ὔπατε κρειόντων, καὶ λίην κεῖνός γε ἐοικότι κεῖται ὀλέθρῳ: ὡς ἀπόλοιτο καὶ ἄλλος, ὅτις τοιαῦτά γε ῥέζοι: ἀλλά μοι ἀμφ' Ὀδυσῆι δαίφρονι δαίεται ἦτορ, δυσμόρῳ, ὃς δὴ δηθὰ φίλων ἄπο πήματα πάσχει νήσῳ ἐν ἀμφιρύτῃ, ὅθι τ' ὀμφαλός ἐστι θαλάσσης …"

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 numeration definitions for the default language.

`\resetdefaultlanguage`
- `\resetdefaultlanguage` (experimental): completely switches the default language to another one in the middle of a document: *this may have adverse effects*!

`\normalfontlatin`
`\rmfamilylatin`
`\sffamilylatin`
`\ttfamilylatin`
- `\normalfontlatin`: in an environment where `\normalfont` has been redefined to a non-latin script, this will call the font defined with `\setromanfont` etc. Likewise it is possible to use `\rmfamilylatin`, `\sffamilylatin`, and `\ttfamilylatin`.
- Also some macros defined in babel 's `hyphen. cfg` (and thus usually compiled into the xelatex format) are redefined, but keep a similar behaviour, namely `\selectlanguage`, `\foreignlanguage`, and the environment `otherlanguage`.

`\selectlanguage`
`\foreignlanguage`
`otherlanguage`

Since the XeLaTeX format incorporates babel's `hyphen. cfg`, the low-level commands for hyphenation and language switching defined there are also accessible.[2]

4 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.

[2] The file `hyphen_cfg` (available on the XeTeX subversion repository) is meant to eventually replace babel 's `hyphen. cfg`. If you want to experiment with it, rename it into `hyphen. cfg`, copy it to `…/tex/xelatex/polyglossia/` and rebuild the xelatex format.

4.1 arabic

Options:

- **calendar** = *islamic* (= hijri) or gregorian
- **locale** = *default*,[3] mashriq,[4] 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)

Commands:

`\abjad` `\abjadmaghribi`

- `\abjad` and `\abjadmaghribi` (see section 5)

4.2 english

Options:

- **variant** = *american* (= us), british (= uk), australian or newzealand
- **ordinalmonthday** = true/*false* (true by default only when variant = british)

4.3 esperanto

Commands:

`\hodiau` `\hodiaun`

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

4.4 farsi

Options:

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

Commands:

`\abjad`

- `\abjad` (see section 5)

4.5 german

Options:

- **spelling** = *new* or old

[3] For Egypt, Sudan, Yemen and the Golf states.
[4] For Iraq, Syria, Jordan, Lebanon and Palestine.

4.6 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 5).
- 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).[5]

4.7 hebrew

Options:

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

Commands:

\hebrewnumeral
\hebrewalph

- `\hebrewnumeral` (= `\hebrewalph`) (see section 5).

4.8 lsorbian and usorbian

Commands:

\oldtoday

- `\oldtoday`: see the babel documentation.

4.9 magyar

Commands:

\ontoday
\ondatemagyar

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

4.10 russian

Options:

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

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

4.11 serbian

Options:

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

4.12 syriac

Options:

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

Commands:

`\abjadsyriac`

- `\abjadsyriac` (see section 5)

4.13 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.

5 Alphabetic numeration in Greek, Arabic, Hebrew, Syriac and Farsi

In certain languages, numbers can be represented by a special alphanumerical notation.[6] Note that the Hebrew implementation in polyglossia is currently less sophisticated than the one in babel, where various special cases are taken into account.

`\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 غضسجـ. 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 شظصجـ.

`\abjadmaghribi`

`\hebrewnumeral`

Hebrew numerals are generated with the command `\hebrewnumeral`. Example: `\hebrewnumeral{1863}` yields אׇ׳חׇסׇג.

`\abjadsyriac`

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

[6] 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.

6 Calendars

6.1 Hebrew calendar (hebrewcal.sty)

The package `hebrewcal.sty` is almost a verbatim copy of `hebcal.sty` that comes with babel. The command `\Hebrewtoday`

`\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).

6.2 Islamic calendar (hijrical.sty)

This new package computes dates in the Islamic (Hijra) calendar, which is lunar.[7] 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 (depending of the current writing direction this is set either in Arabic or in roman transliteration). It also accepts an optional argument to add or subtract a correction (in days) to the date computed by the arithmetical algorithm.[8] For instance if `\Hijritoday` yields the date "7 Rajab 1429" (which is the one for 11th July 2008 indicated on the front page of aljazeera.net), `\Hijritoday[1]` would rather print "8 Rajab 1429" (the date given on the site gulfnews.com).

6.3 Farsi (jalālī) calendar (farsical.sty)

Again this is taken almost verbatim from `Arabiftoday.sty` (in the Arabi package), itself a slight modification of the file `ftoday.sty` in FarsiTEX.[9] Here we have renamed the command `\ftoday` to `\Jalalitoday`. Example: today is 21 Tīr 1387.

`\Jalalitoday`

7 Acknowledgements

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

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

[8] 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 their can be differences between different localities, as well as between civil and religious authorities.

[9] I intend to rewrite farsical from scratch using the algorithm in Reingold & Gershowitz (ref. n. 7).

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). I should also thank other individuals for their assistance in supporting specific languages: Yves Codet (Sanskrit), Zdenek Wagner (Hindi), and other members of the xetex user community. And of course my gratitude also goes to Jonathan Kew, the formidable author of XeTeX!