

# Π Ο Λ Υ Γ Λ Ω Σ Σ Ι Α

## Polyglossia: Modern multilingual typesetting with Xe<sub>Λ</sub>TeX and Lua<sub>Λ</sub>TeX

FRANÇOIS CHARETTE      ARTHUR REUTENAUER\*  
BASTIEN ROUCARIÈS      JÜRGEN SPITZMÜLLER

2019/10/27      v1.45  
(PDF file generated on 27 October 2019)

### Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>                      | <b>1</b> |
| <b>2</b> | <b>Setting up multilingual documents</b> | <b>2</b> |
| 2.1      | Activating languages . . . . .           | 2        |
| 2.2      | Supported languages . . . . .            | 3        |
| 2.3      | Relation to Babel languages . . . . .    | 4        |
| 2.4      | Global options . . . . .                 | 4        |
| <b>3</b> | <b>Language-switching commands</b>       | <b>4</b> |
| 3.1      | Recommended commands . . . . .           | 4        |
| 3.2      | Babel commands . . . . .                 | 6        |
| 3.3      | Other commands . . . . .                 | 7        |
| <b>4</b> | <b>Font setup</b>                        | <b>7</b> |
| <b>5</b> | <b>Adapting hyphenation</b>              | <b>8</b> |
| 5.1      | Hyphenation exceptions . . . . .         | 8        |
| 5.2      | Hyphenation disabling . . . . .          | 8        |

---

\*Current maintainer

|          |   |          |
|----------|---|----------|
| <b>6</b> | <b>Language-specific options and commands</b> | <b>9</b> |
| 6.1      | arabic . . . . .                              | 9        |
| 6.2      | armenian . . . . .                            | 9        |
| 6.3      | bengali . . . . .                             | 9        |
| 6.4      | catalan . . . . .                             | 10       |
| 6.5      | czech . . . . .                               | 10       |
| 6.6      | dutch . . . . .                               | 11       |
| 6.7      | english . . . . .                             | 11       |
| 6.8      | esperanto . . . . .                           | 11       |
| 6.9      | finnish . . . . .                             | 12       |
| 6.10     | french . . . . .                              | 12       |
| 6.11     | gaelic . . . . .                              | 13       |
| 6.12     | german . . . . .                              | 13       |
| 6.13     | greek . . . . .                               | 14       |
| 6.14     | hebrew . . . . .                              | 15       |
| 6.15     | hindi . . . . .                               | 15       |
| 6.16     | hungarian . . . . .                           | 15       |
| 6.17     | italian . . . . .                             | 15       |
| 6.18     | korean . . . . .                              | 16       |
| 6.19     | kurdish . . . . .                             | 16       |
| 6.20     | lao . . . . .                                 | 16       |
| 6.21     | latin . . . . .                               | 17       |
| 6.22     | malay . . . . .                               | 17       |
| 6.23     | mongolian . . . . .                           | 17       |
| 6.24     | norwegian . . . . .                           | 18       |
| 6.25     | persian . . . . .                             | 18       |
| 6.26     | portuguese . . . . .                          | 18       |
| 6.27     | russian . . . . .                             | 18       |
| 6.28     | sami . . . . .                                | 19       |
| 6.29     | sanskrit . . . . .                            | 19       |
| 6.30     | serbian . . . . .                             | 19       |
| 6.31     | slovenian . . . . .                           | 20       |
| 6.32     | sorbian . . . . .                             | 20       |
| 6.33     | syriac . . . . .                              | 20       |
| 6.34     | thai . . . . .                                | 21       |
| 6.35     | tibetan . . . . .                             | 21       |
| 6.36     | ukrainian . . . . .                           | 21       |
| 6.37     | welsh . . . . .                               | 21       |

|           |  |           |
|-----------|--|-----------|
| <b>7</b>  | <b>Modifying or extending captions, date formats and language settings</b> | <b>21</b> |
| <b>8</b>  | <b>Script-specific numbering</b>   | <b>22</b> |
| 8.1       | General localization of numbering . . . . .                                | 23        |
| 8.2       | Non-Western decimal digits . . . . .                                       | 23        |
| 8.3       | Non-Latin alphabetic numbering . . . . .                                   | 24        |
| <b>9</b>  | <b>Footnotes in right-to-left context</b>                                  | <b>25</b> |
| 9.1       | Horizontal footnote position . . . . .                                     | 25        |
| 9.2       | Footnote rule length and position . . . . .                                | 26        |
| <b>10</b> | <b>Calendars</b>   | <b>26</b> |
| 10.1      | Hebrew calendar (hebrewcal.sty) . . . . .                                  | 26        |
| 10.2      | Islamic calendar (hijrical.sty) . . . . .                                  | 27        |
| 10.3      | Farsi (jalālī) calendar (farsical.sty) . . . . .                           | 27        |
| <b>11</b> | <b>Accessing language information</b>                                      | <b>27</b> |
| <b>12</b> | <b>Acknowledgements (by François Charette)</b>                             | <b>28</b> |
| <b>13</b> | <b>More acknowledgements (by Arthur Reutenauer)</b>                        | <b>29</b> |

## 1 Introduction

`Polyglossia` is a package for facilitating multilingual typesetting with  $\text{Xe}_{\text{L}}\text{TeX}$  and (with some exceptions)  $\text{Lua}_{\text{L}}\text{TeX}$ . Basically, it can be used as an alternative to `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).<sup>1</sup>
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 Xe<sub>La</sub>TeX 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 and JOSEPH WRIGHT. Being designed for Xe<sub>La</sub>TeX and Lua<sub>La</sub>TeX, it obviously also relies on `fontspec` by WILL ROBERTSON. For languages written from right to left, it needs the package `bidi` (for Xe<sub>La</sub>TeX) or `luabidi` (for Lua<sub>La</sub>TeX) by VAFA KHALIGHI (وفا خليقي) and the `bidi-tex` GitHub Organisation. `Polyglossia` also bundles three packages for calendaric computations (`hebrewcal`, `hijrical`, and `farsical`).

## 2 Setting up multilingual documents

### 2.1 Activating languages

The default language of a document is specified by means of the command

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

(or, equivalently, `\setmainlanguage`). Secondary languages are specified with

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

All these commands allow you to set language-specific options.<sup>2</sup> It is also possible to load a series of secondary languages at once (but without options) using

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

All language-specific options can be modified locally by means of the language-switching commands described in section 3.

**Note** In general, it is advisable to activate the languages *after* all packages have been loaded. This is particularly important if you use right-to-left scripts or languages with `babel` shorthands.

<sup>1</sup>For the Arabic script this is now done by the bundled package `arabicnumbers`.

<sup>2</sup>Section 6 documents these options for the respective languages.

## 2.2 Supported languages

Table 2.2 lists all languages currently supported. Those in red have specific options and/or commands that are explained in section 6 below.

|           |           |             |             |            |
|-----------|-----------|-------------|-------------|------------|
| albanian  | dutch     | interlingua | mongolian   | slovenian  |
| amharic   | english   | italian     | nko         | sorbian    |
| arabic    | esperanto | japanese    | norwegian   | spanish    |
| armenian  | estonian  | kannada     | occitan     | swedish    |
| asturian  | finnish   | khmer       | persian     | syriac     |
| basque    | french    | korean      | piedmontese | tamil      |
| bengali   | friulian  | kurdish     | polish      | telugu     |
| breton    | gaelic    | lao         | portuguese  | thai       |
| bulgarian | galician  | latin       | romanian    | tibetan    |
| catalan   | german    | latvian     | romansh     | turkish    |
| coptic    | greek     | lithuanian  | russian     | turkmen    |
| croatian  | hebrew    | macedonian  | sami        | ukrainian  |
| czech     | hindi     | malay       | sanskrit    | urdu       |
| danish    | hungarian | malayalam   | serbian     | vietnamese |
| divehi    | icelandic | marathi     | slovak      | welsh      |

Table 1: Languages currently supported in `polyglossia`

v1.0.1  
v1.1.1  
v1.2.0

v1.43  
v1.45

**Version Notes** The support for Amharic ← should be considered an experimental attempt to port the package `ethiop`.<sup>3</sup> Version 1.1.1 ← added support for Asturian, Lithuanian, and Urdu. Version 1.2 ← adds support for Armenian, Occitan, Bengali, Lao, Malayalam, Marathi, Tamil, Telugu, and Turkmen.<sup>4</sup> Version 1.43 ← silently introduced basic support for Japanese. This is considered experimental, and feedback is appreciated. In version 1.45 ←, support for Kurdish and Mongolian as well as some new variants (Canadian French and English) have been added. Furthermore, for consistency reasons, some language have been renamed (*farsi*→*persian*, *friulan*→*friulian*, *magyar*→*hungarian*, *portuges*→*portuguese*, *samin*→*sami*) or merged (*bahasai/bahasam*→*malay*, *brazil/portuges*→*portuguese*, *lsorbian/usorbian*→*sorbian*, *irish/scottish*→*gaelic*, *norsk/nynorsk*→*norwegian*). The old names are still supported for backwards compatibility reasons, but they might not give access to newer language features.

<sup>3</sup>Feedback is welcome.

<sup>4</sup>See acknowledgements at the end for due credit to the various contributors.

## 2.3 Relation to Babel languages

If you are familiar with the `babel` package, you will note that `polyglossia`'s language naming slightly differs. Whereas `babel` has a unique name for each language variety (e.g., *american* and *british*), `polyglossia` differentiates language varieties via language options.

Furthermore, `babel` uses sometimes abbreviations for language names (e.g., *bahasam* for Bahasa Malayu) as well as endonyms, i.e., language names coming from the designated languages (such as *bahasa*, *canadien* or *magyar*). As opposed to this, `polyglossia` always uses spelled-out (lower-cased) English language names.

Table 2.3 lists the language names that differ in both packages. `Babel` names marked in red can also be used in `polyglossia` as an alias.<sup>5</sup>

## 2.4 Global options

`Polyglossia` can be loaded with the following global package options:

v1.1.1

- ▶ `babelshorthands` ← globally activates `babel` shorthands whenever available. Currently shorthands are implemented for Catalan, Czech, Dutch, Finnish, German, Italian, Mongolian, and Russian. Please refer to the respective language descriptions (sec. 6) for details.

v1.2.0

- ▶ `localmarks` redefines the internal  $\LaTeX$  macros `\markboth` and `\markright`. In earlier versions of `polyglossia`, ← this option was set by default, but we now realize that it causes more problems than it helps, so it is now off by default. For backwards-compatibility, the option `nolocalmarks` which used to switch off the previous default, and now does nothing, is still available.
- ▶ `quiet` turns off most info messages and some of the warnings issued by  $\LaTeX$ , `fontspec` and `polyglossia`.

## 3 Language-switching commands

### 3.1 Recommended commands

`\text{lang}` For each activated language the command `\text{lang}[(options)]{...}` becomes available for short insertions of text in that language. For example `\textrussian{\today}` yields 27 октября 2019 г. This command switches to the

---

<sup>5</sup>This is for historical reasons, since earlier versions of `polyglossia` used those names as well. Note, however, that you need to use the matching language switching commands, then, as well, e.g., `\textportuges` with *portuges* (rather than `\textportuguese`).

| Babel name      | Polyglossia name | Polyglossia options            |
|-----------------|------------------|--------------------------------|
| acadien         | french           | variant=acadian                |
| american        | english          | <i>(default)</i>               |
| australian      | english          | variant=australian             |
| austrian        | german           | variant=austrian, spelling=old |
| bahasa          | malay            | <i>(default)</i>               |
| <b>bahasam</b>  | malay            | variant=malaysian              |
| <b>brazil</b>   | portuguese       | variant=brazilian              |
| british         | english          | variant=british                |
| canadian        | english          | variant=canadian               |
| canadien        | french           | variant=canadian               |
| <b>farsi</b>    | persian          |                                |
| <b>friulan</b>  | friulian         |                                |
| german          | german           | spelling=old                   |
| <b>irish</b>    | gaelic           | variant=irish                  |
| kurmanji        | kurdish          | variant=kurmanji               |
| lowersorbian    | sorbian          | variant=lower                  |
| <b>magyar</b>   | hungarian        |                                |
| naustrian       | german           | variant=austrian               |
| newzealand      | english          | variant=newzealand             |
| ngerman         | german           | <i>(default)</i>               |
| <b>norsk</b>    | norwegian        | variant=bokmal                 |
| nswissgerman    | german           | variant=swiss                  |
| <b>nynorsk</b>  | norwegian        | <i>(default)</i>               |
| polutonikogreek | greek            | variant=polytonic              |
| <b>portuges</b> | portuguese       | <i>(default)</i>               |
| <b>samin</b>    | sami             |                                |
| <b>scottish</b> | gaelic           | variant=scottish               |
| serbianc        | serbian          | script=Cyrillic                |
| slovene         | slovenian        |                                |
| swissgerman     | german           | variant=swiss, spelling=old    |
| upporsorbian    | sorbian          | variant=upper                  |

Table 2: Babel-polyglossia language name matching

correct hyphenation patterns, it activates some extra features for the selected language (such as extra spacing before punctuation in French), and it translates the date when using `\today`. It does not, however, translate so-called *caption strings*, i. e. “chapter”, “figure” etc., to the local language (these remain in the main language).

`\lang` The environment `\lang`, which is also available for any activated language, is meant for longer passages of text. It behaves slightly different than the `\text{\lang}{...}` command: It does everything the latter does, but additionally, the caption strings are translated as well, and the language is also passed to auxiliary files, the table of contents and the lists of figures/tables. Like the command, the environment provides 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]
μη̄νιν ἄειδε θεὰ Πηληϊάδεω Ἀχιλῆος οὐλομένην, ἣ μυρί' Ἀχαιοῖς ἄλγε'
ἔθηκε, πολλὰς δ' ἰφθίμους ψυχὰς Ἄϊδι προΐαψεν ἠρώων, αὐτοὺς δὲ ἐλώρια
τεῦχε κύνεσσιν οἰωνοῖσὶ τε πάσι, Διὸς δ' ἐτελείετο βουλή, ἐξ οὗ δὴ τὰ
πρῶτα διαστήτην ἐρίσαντε Ἀτρεΐδης τε ἄναξ ἀνδρῶν καὶ δῖος Ἀχιλλεύς.
\end{greek}
```

μη̄νιν ἄειδε θεὰ Πηληϊάδεω Ἀχιλῆος οὐλομένην, ἣ μυρί' Ἀχαιοῖς ἄλγε' ἔθηκε, πολλὰς δ' ἰφθίμους ψυχὰς Ἄϊδι προΐαψεν ἠρώων, αὐτοὺς δὲ ἐλώρια τεῦχε κύνεσσιν οἰωνοῖσὶ τε πάσι, Διὸς δ' ἐτελείετο βουλή, ἐξ οὗ δὴ τὰ πρῶτα διαστήτην ἐρίσαντε Ἀτρεΐδης 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.

### 3.2 Babel commands

Some macros defined in `babel`'s `hyphen.cfg` (and thus usually compiled into the  $\TeX$  and  $\LaTeX$  format) are redefined, but keep a similar behaviour.

```
\selectlanguage    ▶ \selectlanguage[(options)]{\lang}
\foreignlanguage  ▶ \foreignlanguage[(options)]{\lang}{...}
  otherlanguage    ▶ \begin{otherlanguage}[(options)]{\lang} ... \end{otherlanguage}
  otherlanguage*   ▶ \begin{otherlanguage*}[(options)]{\lang} ... \end{otherlanguage*}
```

`\selectlanguage{\lang}` and the `otherlanguage` environment are identical to the `\lang` environment, except that `\selectlanguage{\lang}` does not



need to be explicitly closed. The command `\foreignlanguage{<lang>}{...}` and the `otherlanguage*` environment are identical with the use of the `\text{<lang>}` command, with the one notable exception that they do not translate the date with `\today`.

Since the X<sub>Y</sub>TeX and LuaTeX format incorporate babel's `hyphen.cfg`, the low-level commands for hyphenation and language switching defined there are also accessible.

### 3.3 Other commands

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

|  |  |
|--|--|
| <code>\selectbackgroundlanguage</code> | ▶ <code>\selectbackgroundlanguage</code> : this selects the global font setup and the numbering definitions for the default language.  |
| <code>\resetdefaultlanguage</code>     | ▶ <code>\resetdefaultlanguage</code> (experimental): completely switches the default language to another one in the middle of a document: <i>this may have adverse effects!</i>  |
| <code>\normalfontlatin</code>          | ▶ <code>\normalfontlatin</code> : in an environment where <code>\normalfont</code> has been re-defined to a non-latin script, this will reset to the font defined with <code>\setmainfont</code> etc. In a similar vein, it is possible to use <code>\rmfamilylatin</code> , |
| <code>\rmfamilylatin</code>            | <code>\sffamilylatin</code> , and <code>\ttfamilylatin</code> .  |
| <code>\sffamilylatin</code>            |  |
| <code>\ttfamilylatin</code>            | ▶ <code>\latinalph</code> : Representation of counter as a lower-case letter: 1 = a, 2 = b, etc.   |
| <code>\latinalph</code>                |  |
| <code>\latinAlph</code>                | ▶ <code>\latinAlph</code> : Representation of counter as a upper-case letter: 1 = A, 2 = B, etc.   |

## 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 partic-

v1.2.0

ular script or language, it can be defined by means of `\(\script)fontsf` or `\(\language)fontsf` and `\(\script)fonttt` or `\(\language)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 Adapting hyphenation

### 5.1 Hyphenation exceptions

$\TeX$  provides the command `\hyphenation{exceptions}` to globally define hyphenation exceptions which override the hyphenation patterns for specified words. The command takes as argument a space-separated list of words where hyphenation points are marked by a dash (if no dash is used, the respective word is not hyphenated at all):

```
\hyphenation{%
  po-ly-glos-sia
  LaTeX
}
```

These exceptions, however, apply to all languages. In addition to this, `polyglossia` provides the command `\pgghyphenation`

v.1.45

```
\pgghyphenation[options][lang]{exceptions}
```

which can be used to define exceptions that only apply to a specific language or language variant, respectively.

### 5.2 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`

`\enablehyphenation`

`\disablehyphenation` and `\enablehyphenation`. Note that if you select a new language while hyphenation is disabled, it will remain disabled. If you re-enable it, the hyphenation patterns of the currently selected language will be activated.

## 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*,<sup>6</sup> *mashriq*,<sup>7</sup> *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).
- ▶ **numerals** = *mashriq* or *maghrib* (the latter is the default when locale = *algeria*, *tunisia* or *morocco*)
- ▶ **abjadjimnotail** = *false* or *true*. ← Set this to true if you want the *abjad* form of the number three to be ٣ – as in the manuscript tradition – instead of the modern usage ٣.

v1.0.3

#### Commands:

- |                             |  |
|-----------------------------|--|
| <code>\abjad</code>         | ▶ <code>\abjad</code> and <code>\abjadmaghribi</code> (see section 8.3)                                      |
| <code>\abjadmaghribi</code> | ▶ <code>\aemph</code> to emphasize text with <code>\overline</code> . ← <code>\textarabic{\aemph{ب }}</code> |
| <code>\aemph</code>         | yields $\overline{ب}$ . This command is also available for Farsi, Urdu, etc.                                 |
- v1.2.0

### 6.2 armenian

#### Options:

- ▶ **variant** ← = *eastern* or *western*
  - ▶ **numerals** ← = *armenian* or *arabic*
- v1.45  
v1.45

### 6.3 bengali ←

v1.2.0

#### Options:

- ▶ **numerals** = *Western*, *Bengali* or *Devanagari*
- ▶ **changecounternumbering** = *true* or *false* (use specified numerals for headings and page numbers)

---

<sup>6</sup>For Egypt, Sudan, Yemen and the Gulf states.

<sup>7</sup>For Iraq, Syria, Jordan, Lebanon and Palestine.

## 6.4 catalan

### Options:

v1.1.1

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

### Commands:

- `\l.l` ▶ `\l.l` and `\L.L` ← can be used to type a geminated l, as in *collaborar*, similar to *babel* (the glyph U+00B7 MIDDLE DOT is used as a geminating sign).

v1.1.1

## 6.5 czech

### Options:

v1.45

- ▶ **babelshorthands** = *false* or *true*. ← if this is turned on, the following shorthands for Czech are activated:

"= for an explicit hyphen sign which is repeated at the beginning of the next line when hyphenated, as common in Czech typesetting (also see option `splithyphens`).

"‘ for Czech left double quotes (looks like „).

"’ for Czech right double quotes (looks like “).

"> for Czech left double guillemets (looks like »).

"< for Czech right double guillemets (looks like «).

v1.45

- ▶ **splithyphens** = *false* or *true*. ← According to Czech typesetting conventions, if a word with a hard hyphen (such as *je-li*) is hyphenated at this hyphen, a second hyphenation character is to be inserted at the beginning of the line that follows the hyphenation (*je-/li*).

If this option is true, this is done automatically (so the shorthand "=" is not needed). Note, however, that the option is only available for X<sub>Y</sub>TeX. If you are using LuaTeX, load the `luavlna` package to the same effect.

v1.45

- ▶ **vlna** = *false* or *true*. ← According to Czech typesetting conventions, single-letter words (non-syllable prepositions) must not occur at line ends.

If this option is true, this is taken care of. Note, however, that the option is only available for X<sub>Y</sub>TeX. If you are using LuaTeX, load the `luavlna` package to the same effect.

## 6.6 dutch

### Options:

v1.1.1

- ▶ **babelshorthands** = *false* or *true*. ← if this is turned on, the following shorthands defined for fine-tuning hyphenation and micro-typography of Dutch words are activated:
  - " - adds a hyphenation point that does still allow for hyphenation at the points preset in the hyphenation patterns (as opposed to \- in default T<sub>E</sub>X).
  - "~ for a hyphen sign without a breakpoint. Useful for cases where the hyphen should stick at the following syllable.
  - "| disables a ligature at this position.
  - " " allows for a line break at this position (without hyphenation sign).
  - "/ a slash that allows for a subsequent line break. As opposed to \slash, hyphenation at the breakpoints preset in the hyphenation patterns is still allowed.

\- In addition, the macro \- is redefined to allow hyphens in the rest of the word (equivalent to "-).

## 6.7 english

### Options:

v1.45

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

## 6.8 esperanto

### Commands:

- \hodiaun and \hodiau are special forms of \today. The former produces the date in Esperanto preceded by the article (*la*), which is the most common date format. The latter produces the same date format in accusative case.

## 6.9 finnish

### Options:

v1.45

- ▶ **babelshorthands** = *false* or *true*. ← if this is turned on, the following shorthands for fine-tuning hyphenation and micro-typography of Finnish words are activated:
  - " - adds a hyphenation point that does still allow for hyphenation at the points preset in the hyphenation patterns (as opposed to \-).
  - "~ for a hyphen sign without a breakpoint. Useful for cases where the hyphen should stick at the following syllable.
  - "| disables a ligature at this position.
  - " " allows for a line break at this position (without hyphenation sign).
  - "/ a slash that allows for a subsequent line break. As opposed to \slash, hyphenation at the breakpoints preset in the hyphenation patterns is still allowed.

## 6.10 french

### Options:

v1.45

- ▶ **variant** = *french* or *canadian* (= *acadian*). ← Currently, the three variants do not differ; they are supported for compatibility with **babel** (where they do not differ either).
- ▶ **autospacing** = *true* or *false* (default value = *true*). One of the most distinct features of French typography is the addition of extra spacing around punctuation and quotation marks (guillemets). By default, polyglossia adds these spaces automatically, so you don't need to enter them. This options allows you to switch this feature off globally.
- ▶ **autospaceguillemets**<sup>8</sup> = *true* or *false* (default value = *true*). If you only want to disable the automatic addition of spacing after opening and before closing guillemets (and not at punctuation), set this to *false*. Note that the more general option *autospacing* overrides this.
- ▶ **autospacetyewriter**<sup>9</sup> ← = *true* or *false* (default value = *true*). By default, automatic spacing is disabled in typewriter font. If this is enabled, spacing in typewriter context is the same as with roman and sans serif font,

1.45

---

<sup>8</sup>Up to version 1.44, the option was called *automaticspacesaroundguillemets*. For backwards compatibility reasons, the more verbose old option is still supported.

<sup>9</sup>Babel's syntax *OriginalTypewriter* is also supported.

depending on the *autospadding* and *autospaceguillemets* settings (note that this was the default up to v. 1.44).

- ▶ **frenchfootnote** = *true* or *false* (default value = *true*). If *true*, footnotes start with a non-superscripted number followed by a dot, as common in French typography. Note that this might interfere with the specific footnote handling of classes or packages. Also note that this option is only functional (by design) if French is the main language.

#### Commands:

`\NoAutoSpacing`  
v1.45

- ▶ `\NoAutoSpacing` ← disables automatic spacing around punctuation and quotation marks in all following text. The command can also be used locally if braces are used for grouping: `{\NoAutoSpacing foo:bar}`

`\AutoSpacing`  
v1.45

- ▶ `\AutoSpacing` ← enables automatic spacing around punctuation and quotation marks in all following text. The command can also be used locally if braces are used for grouping: `{\AutoSpacing regarde!}`

v1.45

#### 6.11 gaelic ←

##### Options:

- ▶ **variant** = *irish* or *scottish*

#### 6.12 german

##### Options:

- v1.33.4
- ▶ **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  $\TeX$ Live 2008 and later.
- v1.0.3
- ▶ **babelshorthands** = *false* or *true*: ← if this is turned on, all shorthands defined in `babel` for fine-tuning hyphenation and micro-typography of German words are activated.  
"ck for ck to be hyphenated as k-k (1901 spelling).

- "ff for ff to be hyphenated as ff-f (1901 spelling); this is also available for the letters l, m, n, p, r and t.
- "| disables a ligature at this position (e.g., Auf" |lage).
- "= for an explicit hyphen with a breakpoint, allowing for hyphenation at the other points preset in the hyphenation patterns (as opposed to plain -).
- "~ for a hyphen sign without a breakpoint. Useful for cases where the hyphen should stick at the following syllable, e.g., bergauf und "~ab.
- "- adds a hyphenation point that does still allow for hyphenation at the points preset in the hyphenation patterns (as opposed to \-).
- " " allows for a line break at this position (without hyphenation sign); e.g., (pseudo"~)" "wissenschaftlich.
- "/ a slash that allows for a subsequent line break. As opposed to \slash, hyphenation at the breakpoints preset in the hyphenation patterns is still allowed.

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

## 6.13 greek

### Options:

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

### Commands:

- |               |  |
|---------------|--|
| \Greeknuber   | ▶ \Greeknuber and \greeknumber (see section 8.3).  |
| \greeknumber  | ▶ 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). <sup>10</sup> |
| \atticnumeral |  |
| \atticnum     |  |

<sup>10</sup>See the documentation of the [xgreek](#) package for more details.



## 6.14 hebrew

### Options:

- ▶ **numerals** = hebrew or *arabic*
- ▶ **calendar** = hebrew or *gregorian*
- ▶ **marcheshvan** = true or *false* (default value = true). If true, the second month of the civil year will be output as מַרְחֶשְׁוָן (Marcheshvan) rather than חֶשְׁוָן (Heshvan), which is the default.

### Commands:

- `\hebrewnumeral` ▶ `\hebrewnumeral` (= `\hebrewalph`) (see section 8.3).
- `\hebrewalph` ▶ `\aemph` (see section 6.1).
- `\aemph`

v1.2.0

## 6.15 hindi ←

### Options:

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

## 6.16 hungarian

### Commands:

- `\ontoday` ▶ `\ontoday` (= `\ondatehungarian`): special form of `\today` which produces a slightly different date format as used in prepositional phrases (such as ‘on February 10th’) in Hungarian.
- `\ondatehungarian`

## 6.17 italian

### Options:

v1.2.0cc

- ▶ **babelshorthands** = *false* or true. ← Activates the " character as a switch to perform etymological hyphenation when followed by a letter. Furthermore, the following shorthands are activated:
  - " " double raised open quotes (the Italian keyboard misses the backtick).
  - "< open guillemet (looks like «).
  - "> closing guillemet (looks like »).
  - "/ a slash that allows for a subsequent line break. As opposed to `\slash`, hyphenation at the breakpoints preset in the hyphenation patterns is still allowed.

"- adds a hyphenation point that does still allow for hyphenation at the points preset in the hyphenation patterns (as opposed to \-).

v1.40.0

## 6.18 korean ←

### Options:

- ▶ **variant** = *plain*, *classic* or *modern*: for spacing before/after CJK punctuations. ‘classic’ is suitable for text with no interword spaces, ‘modern’ for text with interword spaces.
- ▶ **captions** = *hangul* or *hanja*

v1.45

## 6.19 kurdish ←

Kurdish support includes Sorani Kurdish and Kurmanji Kurdish, both in either Arabic or Latin script.

### Options:

- ▶ **variant** = *kurmanji* or *sorani*
- ▶ **script** = Arabic or Latin. Defaults are Arabic for Sorani and Latin for Kurmanji.
- ▶ **numerals** = *western* or *eastern*. Defaults are *western* for Latin and *eastern* for Arabic script, depending on the selection above.
- ▶ **abjadjimmnotail** = *false* or *true*. Set this to *true* if you want the *abjad* form of the number three to be  $\text{چ}$  – as in the manuscript tradition – instead of the modern usage  $\text{ج}$ .
- ▶ **locale** (not yet implemented)
- ▶ **calendar** (not yet implemented)

### Commands:

- `\ontoday` ▶ `\ontoday`: special form of `\today` which produces a slightly different date format as used in prepositional phrases (as in ‘on February 10th’). Only available for Latin script.
- `\abjad` ▶ `\abjad` (see section 8.3)
- `\aemph` ▶ `\aemph` (see section 6.1)

v1.2.0

## 6.20 lao ←

### Options:

- ▶ **numerals** = *lao* or *arabic*

## 6.21 latin

### Options:

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

## 6.22 malay

### Options:

- v1.45 ▸ **variant** ← = *indonesian* (= bahasai in **babel**) or *malaysian* (= bahasam in **babel**)

## 6.23 mongolian ←

v1.45

Currently, only the Khalkha variety in Cyrillic script is supported.

### Options:

- **babelshorthands** = *false* or true. If this is turned on, the following short-hands are activated:
  - " - adds a hyphenation point that does still allow for hyphenation at the points preset in the hyphenation patterns (as opposed to \-).
  - "~ for a hyphen sign without a breakpoint. Useful for cases where the hyphen should stick at the following syllable.
  - "| disables a ligature at this position.
  - " " allows for a line break at this position (without hyphenation sign).
  - " - - - Cyrillic emdash in plain text.
  - " - - Cyrillic emdash in compound names (surnames).
  - " - -\* Cyrillic emdash for denoting direct speech.
  - " , thinspace for initials with a breakpoint in following surname.
  - " ‘ for German left double quotes (looks like „).
  - " ’ for German right double quotes (looks like “).
  - " < for French left double quotes (looks like «).
  - " > for French right double quotes (looks like »).
- **numerals** = *arabic* or *cyrillic*. Uses either Arabic numerals or Cyrillic alphanumerical numbering.

### Commands:

- `\Asbuk` ▶ `\Asbuk`: produces uppercased Cyrillic alphanumerals, for environments such as `enumerate`. The command takes a counter as argument, e.g., `\textrussian{\Asbuk{page}}` produces  $\text{I}\Theta$ .
- `\asbuk` ▶ `\asbuk`: same in lowercase

## 6.24 norwegian

### Options:

v1.45

- ▶ **variant** ← = `bokmal` (= ‘norsk’ in `babel`) or `nynorsk`

## 6.25 persian

### Options:

v1.0.3

- ▶ **numerals** = western or *eastern*
- ▶ **abjadjimmotail** = *false* or true. ← Set this to true if you want the *abjad* form of the number three to be  $\text{ژ}$  – as in the manuscript tradition – instead of the modern usage  $\text{ج}$ .
- ▶ **locale** (not yet implemented)
- ▶ **calendar** (not yet implemented)

### Commands:

- `\abjad` ▶ `\abjad` (see section 8.3)
- `\aemph` ▶ `\aemph` (see section 6.1).

## 6.26 portuguese

### Options:

v1.45

- ▶ **variant** ← = `brazilian` or `portuguese`

## 6.27 russian

### Options:

- ▶ **babelshorthands** = *false* or true. If this is turned on, the following short-hands are activated:
  - "- adds a hyphenation point that does still allow for hyphenation at the points preset in the hyphenation patterns (as opposed to `\-`).
  - "~ for a hyphen sign without a breakpoint. Useful for cases where the hyphen should stick at the following syllable.

- "| disables a ligature at this position.
- "" allows for a line break at this position (without hyphenation sign).
- " - - Cyrillic emdash in plain text.
- " - - Cyrillic emdash in compound names (surnames).
- " - - \* Cyrillic emdash for denoting direct speech.
- ▶ **spelling** = *modern* or *old* (for captions and date only, not for hyphenation)
- ▶ **numerals** = *arabic* or *cyrillic*. Uses either Arabic numerals or Cyrillic alphanumerical numbering.

#### Commands:

- \Asbuk ▶ \Asbuk: produces uppercased Cyrillic alphanumerals, for environments such as `enumerate`. The command takes a counter as argument, e.g., `\textrussian{\Asbuk{page}}` produces KA.
- \asbuk ▶ \asbuk: same in lowercase

v1.45

### 6.28 sami ←

Currently support for Sami is limited to Northern Sami.

### 6.29 sanskrit

#### Options:

v1.0.2

- ▶ **Script** = *Devanagari* ←, Gujarati, Malayalam, Bengali, Kannada, Telugu or Latin. The value is passed to `fontspec` in cases where the respective `\<script>font` is not defined. This can be useful if you typeset Sanskrit texts in scripts other than Devanagari.

v1.45

- ▶ **Numerals** = *Devanagari* ← or Western

### 6.30 serbian

#### Options:

- ▶ **Script** = *Cyrillic* or Latin. Will likely go to a variant.
- ▶ **numerals** = *arabic* or *cyrillic*. Uses either Arabic numerals or Cyrillic alphanumerical numbering.

### Commands:

- `\Asbuk` ▶ `\Asbuk`: produces uppercased Cyrillic alphanumerals, for environments such as `enumerate`. The command takes a counter as argument, Example: `\textserbian[numerals=cyrillic]{\Asbuk{page}}` produces KA.
- `\asbuk` ▶ `\asbuk`: same in lowercase

## 6.31 slovenian

### Options:

- ▶ `localalph` = true or *false*

## 6.32 sorbian

### Options:

- ▶ `variant` ← = lower or *upper*
- ▶ `olddate` ← = true or *false* (default value = true). If true, `\today` will use traditional Sorbian month names (*i.e.*, it will be synonymous to `\oldtoday` below)

v1.45

v1.45

### Commands:

- `\oldtoday` ▶ `\oldtoday`: outputs the current date using traditional Sorbian month names, even if `olddate` is false.

## 6.33 syriac

### Options:

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

v1.0.1

### Commands:

- `\abjadsyriac` ▶ `\abjadsyriac` (see section 8.3)
- `\aemph` ▶ `\aemph` (see section 6.1).

### 6.34 thai

#### Options:

- ▶ **numerals** = thai or *arabic*

To insert 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.

### 6.35 tibetan

#### Options:

- ▶ **numerals** = tibetan or *arabic*

### 6.36 ukrainian

#### Commands:

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

### 6.37 welsh

#### Options:

- ▶ **date** = long or *short*

## 7 Modifying or extending captions, date formats and language settings

[Polyglossia](#) uses the following macros to define language-specific captions (*i.e.*, strings such as “chapter”), date formats and additional language settings (`\lang` is to be replaced with the respective language name):

- `\captions<lang>` ▶ `\captions<lang>` stores definitions of caption strings (such as, in the case of English, `\def\chaptername{Chapter}`)
- `\date<lang>` ▶ `\date<lang>` stores definitions of date formats (usually redefinitions of `\today`, in some cases also definitions of additional date commands)
- `\blockextras<lang>` ▶ `\blockextras<lang>` stores macros that are to be executed when the language `<lang>` is activated via `\selectlanguage` command or the `<lang>` environment
- `\inlineextras<lang>` ▶ `\inlineextras<lang>` stores macros that are to be executed when the language `<lang>` is activated locally via `\text<lang>` command

`\noextras{lang}`      ▶ `\noextras{lang}` stores macros that are to be executed when the language `{lang}` is closed

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

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

Note that this needs to be done after the respective language has been loaded with `\setmainlanguage` or `\setotherlanguage`.

Specifically for package authors, analogous commands are provided which are only executed if a specific language *variety* is used. As opposed to the macros above, these refer to babel names. Other than that, the function is identical:

```
\captions@bbl@{babelname}      ▶ \captions@bbl@{babelname}
   \date@bbl@{babelname}        ▶ \date@bbl@{babelname}
\blockextras@bbl@{babelname}   ▶ \blockextras@bbl@{babelname}
\inlineextras@bbl@{babelname} ▶ \inlineextras@bbl@{babelname}
\noextras@bbl@{babelname}     ▶ \noextras@bbl@{babelname}
```

By default, these macros are undefined. If they are defined (*e.g.*, by an external package), they will be executed after their respective `{lang}` counterpart and thus can be used to overwrite definitions of the former. Again, use `\gappto` to define/modify these macros. For instance, to add a new caption `\footnotename` to the Swiss variety of German (babel name `nswissgerman`), you can do this:

```
\gappto\captions@bbl@nswissgerman{\def\footnotename{Fussnote}}
```

If you do this in a document preamble rather than in a package, you need to embrace the redefinition by `\makeatletter` and `\makeatother` due to the `@` in the macro names.

## 8 Script-specific numbering

Languages and scripts have specific numbering conventions. Some use decimal digits (*e.g.*, Arabic numerals), some use alphabetic or alphanumerical notation (*e.g.*, Roman numbering). In some cases, different conventions are available (*e.g.*, Mashriq or Maghrib numbering in Arabic script, Arabic or Hebrew [= alphanumerical] numbering in Hebrew).



If the latter is the case, `polyglossia` provides language options which allow you to select or switch to the suitable convention. 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.

For manual input of numbers, macros are provided. These convert Arabic numeric input to the respective local decimal digit (see sec. 8.2), alphanumeric representation (see sec. 8.3) or whatever is appropriate (see sec. 8.1). The possibilities are described in turn.

## 8.1 General localization of numbering

1.45 As of 1.45,  $\leftarrow$  `polyglossia` provides a generic macro `\localnumeral` which converts numbers to the current local form (which might be script-specific decimal digit, an alphabetic numbering or something else). For instance in an Arabic environment `\localnumeral{42}` yields  $\text{٤٢}$ , whereas in an Hebrew environment, it results in  $\text{כב}$  with `numerals=hebrew`, and  $42$  with `numerals=arabic`. Note that, as opposed to the various `digits` macros (described in sec. 8.2), the argument of `\localnumeral` must consist of numbers only.

1.45 For  $\leftarrow$  the conversion of counters, the starred version `\localnumeral*` is provided. This takes a counter as argument. For instance in an Arabic environment `\localnumeral*{page}` yields  $\text{٢٥}$ .

`\Localnumeral` For scripts with alphanumeric numbering, the variants `\Localnumeral` and `\Localnumeral*` provide the uppercased versions.

All these macros provide the following options:

- [lang=]
    - **lang** = *local*, *main*, or  $\langle$ language $\rangle$ .
- Output number in the local form of the currently active language for *local*, the main language of the document for *main*, and any (loaded) language for  $\langle$ language $\rangle$  (e.g., `\localnumeral[lang=arabic]{42}`).

## 8.2 Non-Western decimal digits

v1.1.1 In addition  $\leftarrow$  to the generic macros described above, `polyglossia` provides language-specific conversion macros which can be used if the generic ones do not suit the need.<sup>11</sup> The macros have the form `\<script>digits`. They convert

<sup>11</sup>A third method are so-called TECKit fontmappings. Those can be activated with the `fontspec` Mapping option, using `arabicdigits`, `farsidigits` or `thaidigits`. For instance if `\arabicfont` is defined with the option `Mapping=arabicdigits`, typing `\textarabic{2010}` results in  $\text{٢٠١٠}$ . Note

Arabic numerical input and leave every other input untouched. In an Arabic context, for instance, `\arabicdigits{9182/738543-X}` yields ٩١٨٢/٧٣٨٥٤٣-X.

Currently, the following macros are provided:

|                                |                                  |
|--------------------------------|----------------------------------|
| <code>\arabicdigits</code>     | ▶ <code>\arabicdigits</code>     |
| <code>\bengalidigits</code>    | ▶ <code>\bengalidigits</code>    |
| <code>\devanagaridigits</code> | ▶ <code>\devanagaridigits</code> |
| <code>\farsidigits</code>      | ▶ <code>\farsidigits</code>      |
| <code>\kannadadigits</code>    | ▶ <code>\kannadadigits</code>    |
| <code>\khmerdigits</code>      | ▶ <code>\khmerdigits</code>      |
| <code>\laodigits</code>        | ▶ <code>\laodigits</code>        |
| <code>\nkodigits</code>        | ▶ <code>\nkodigits</code>        |
| <code>\thaidigits</code>       | ▶ <code>\thaidigits</code>       |
| <code>\tibetandigits</code>    | ▶ <code>\tibetandigits</code>    |

### 8.3 Non-Latin alphabetic numbering

For languages which use special (non-Latin) alphanumerical notation<sup>12</sup>, dedicated macros are provided.

They work in a similar way than the `\<script>digits` macros described above: They take Arabic numerical input and output the respective value in the local alphabetic numbering scheme (most of these macros are equivalent to `\localnumeral` and `\Localnumeral` in the respective context).

The following macros are provided:

|                               |   |
|-------------------------------|---|
| <code>\abjad</code>           | ▶ <code>\abjad</code> outputs Arabic <i>abjad</i> numbers according to the Mashriq varieties. Example: <code>\abjad{1863}</code> yields غضسج.                 |
| <code>\abjadmaghribi</code>   | ▶ <code>\abjadmaghribi</code> outputs Arabic <i>abjad</i> numbers according to the Maghrib varieties. Example: <code>\abjadmaghribi{1863}</code> yields شظصج. |
| <code>\abjadsyriac</code>     | ▶ <code>\abjadsyriac</code> outputs Syriac abjad numerals. <sup>13</sup> Example: <code>\abjadsyriac{463}</code> yields ٤٦٣.                                  |
| <code>\armeniannumeral</code> | ▶ <code>\armeniannumeral</code> produces Armenian alphabetic numbering. Example: <code>\armeniannumeral{1863}</code> yields ՌՊԿԳ.                             |
| <code>\greeknumeral</code>    | ▶ <code>\greeknumeral</code> produces Greek alphabetic numbering, <code>\Greeknnumeral</code> out-  |
| <code>\Greeknnumeral</code>   |   |

that this method has some drawbacks, though, for instance when the value of a counter has to be written and read from auxiliary files. So please use this with care.

<sup>12</sup>For instance, see [http://en.wikipedia.org/wiki/Greek\\_numerals](http://en.wikipedia.org/wiki/Greek_numerals), [http://en.wikipedia.org/wiki/Abjad\\_numerals](http://en.wikipedia.org/wiki/Abjad_numerals), [http://en.wikipedia.org/wiki/Hebrew\\_numerals](http://en.wikipedia.org/wiki/Hebrew_numerals), and [http://en.wikipedia.org/wiki/Syriac\\_alphabet](http://en.wikipedia.org/wiki/Syriac_alphabet).

<sup>13</sup>A fine guide to numerals in Syriac can be found at <http://www.garzo.co.uk/documents/syriac-numerals.pdf>.

|  |  |
|--|--|
|  | puts uppercased variants. Example: <code>\greeknumeral{1863}</code> yields $\alpha\omega\xi\gamma'$ , <code>\Greeknumeral{1863}</code> results in $A\Omega\xi\Gamma'$ .  |
| <code>\hebrewnumeral</code><br><code>\Hebrewnumeral</code><br><code>\Hebrewnumeralfinal</code> | <ul style="list-style-type: none"> <li>▶ <code>\hebrewnumeral</code>, <code>\Hebrewnumeral</code> and <code>\Hebrewnumeralfinal</code> generate variants of Hebrew alphanumeric numerals. The commands behave exactly as they do in <code>babel</code>: <code>\hebrewnumeral</code> outputs the numbers without any decoration, <code>\Hebrewnumeral</code> adds <i>gereshayim</i> before the last letter, <code>\Hebrewnumeralfinal</code> uses in addition the final forms of Hebrew letters. Examples: <code>\hebrewnumeral{1750}</code> yields <math>\text{אָשׂתֵּיָא}</math>, <code>\Hebrewnumeral{1750}</code> yields <math>\text{אָשׂתֵּיָא}</math>, and <code>\Hebrewnumeralfinal{1750}</code> yields <math>\text{אָשׂתֵּיָא}</math>.</li> </ul> |
| <code>\russiannumeral</code><br><code>\Russiannumeral</code>                                   | <ul style="list-style-type: none"> <li>▶ <code>\russiannumeral</code> produces Russian numbering, with uppercased variant (for alphanumeric variant) via <code>\Russiannumeral</code>. Depending on the <code>numerals</code> option in the Russian language selection, this is either Arabic digit or Cyrillic alphanumeric output. Example: With <code>numerals=latin</code> <code>\russiannumeral{19}</code> yields 19, with <code>numerals=cyrillic</code> <code>\russiannumeral{19}</code> results in <math>\text{іѳ}</math>.</li> </ul>  |
| <code>\serbiannumeral</code><br><code>\Serbiannumeral</code>                                   | <ul style="list-style-type: none"> <li>▶ <code>\serbiannumeral</code> produces Serbian numbering, with uppercased variant (for alphanumeric variant) via <code>\Serbiannumeral</code>. Depending on the <code>numerals</code> option in the Serbian language selection, this is either Arabic digit or Cyrillic alphanumeric output. Example: With <code>numerals=latin</code> <code>\serbiannumeral{19}</code> yields 19, with <code>numerals=cyrillic</code> <code>\serbiannumeral{19}</code> results in <math>\text{іѳ}</math>.</li> </ul>  |

## 9 Footnotes in right-to-left context

With languages that use right-to-left scripts, footnote apparatuses are usually placed at the right side of the page bottom. Consequently, the footnote rule also is to be placed right. Things get more tricky, though, if right-to-left and left-to-right scripts are mixed. Then you might want to put the footnotes on some pages left, on some right, or even mix positions on a page. Thus, footnote handling in right-to-left context sometimes needs manual intervention. This is described in what follows.

### 9.1 Horizontal footnote position

When right-to-left languages are used, the `\footnote` command becomes sensitive to the text directionality. The footnote is always placed on the side that is currently the origin of direction: on the left side of the page in LTR paragraphs and on the right in RTL paragraphs.

For cases where this is not desired, two additional footnote commands are provided: `\RTLfootnote` and `\LTRfootnote`. `\LTRfootnote` always places the footnote on the left side, notwithstanding the current directionality. Likewise, `\RTLfootnote` always places it on the right side. Like `\footnote`, `\RTLfootnote` and `\LTRfootnote` provide an optional argument to customize the number.

## 9.2 Footnote rule length and position

The default placement of the footnote rule differs in  $X_{\text{Y}}\text{T}_{\text{E}}\text{X}$  and  $\text{LuaT}_{\text{E}}\text{X}$  output (this is due to differences in the `bidi` and `luabidi` packages). With  $X_{\text{Y}}\text{T}_{\text{E}}\text{X}$ , footnote rules are always placed left, which is often wrong in RTL context. With  $\text{LuaT}_{\text{E}}\text{X}$ , by contrast, the rule is placed always right if the main language is a right-to-left language, and always left if the main language is a left-to-right language, which is the right thing in many cases.

In both cases, you can change the default behavior as follows:

- `\leftfootnoterule` ▶ Put `\leftfootnoterule` in the preamble to have all rules left-aligned.
- `\rightfootnoterule` ▶ Put `\rightfootnoterule` in the preamble to have all rules right-aligned.
- `\autofootnoterule` ▶ Put `\autofootnoterule` in the preamble to have automatic placement depending on the context (see below for elaboration).
- `\textwidthfootnoterule` ▶ Put `\textwidthfootnoterule` in the preamble to have a rule that spans the whole text width.

With `\autofootnoterule`, the first footnote on the current page determines the placement. Note that this automatic can fail with footnotes at page boundaries that differ in directionality from the first footnote on the page. You can work around such cases by switching to `\rightfootnoterule` or `\leftfootnoterule` on these pages.

Note also that the rule switches might interfere in bad ways with packages or classes that redefine footnotes themselves. This is also the reason why `\autofootnoterule` is not used by default.

## 10 Calendars

### 10.1 Hebrew calendar (`hebrewcal.sty`)

The package `hebrewcal.sty` is almost a verbatim copy of `hebcals.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.<sup>14</sup> It provides two macros for the end-user. The command

`\HijriFromGregorian`                    `\HijriFromGregorian{<year>}{<month>}{<day>}`

`\Hijritoday` sets the counters `Hijriday`, `Hjrimonth` and `Hjriyear`. `\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 and Malay 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.<sup>15</sup> For instance if `\Hijritoday` yields the date “7 Rajab 1429” (which is the date that was displayed on the front page of [aljazeera.net](http://aljazeera.net) on 11th July 2008), `\Hijritoday[1]` would rather print “8 Rajab 1429” (the date indicated the same day on the site [gulffnews.com](http://gulffnews.com)).

## 10.3 Farsi (jalālī) calendar (farsical.sty)

This package is an almost verbatim copy of `ArabiToday.sty` (in the `Arabi` package), itself a slight modification of `ftoday.sty` in Farsi $\TeX$ .<sup>16</sup> Here we have renamed the command `\ftoday` to `\Jalalitoday`. Example: today is 5 Ābān 1398.

`\Jalalitoday`

## 11 Accessing language information

The following is specifically relevant to package authors who need information about the languages in use.

In order to get such information, `polyglossia` provides the following macros:

---

<sup>14</sup>It makes use of the arithmetical algorithm in chapter 6 of Reingold & Gershowitz, *Calendrical calculation: the Millenium edition* (Cambridge University Press, 2001).

<sup>15</sup>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.

<sup>16</sup>One day I may rewrite `farsical` from scratch using the algorithm in Reingold & Gershowitz (ref. n. 14).

|                                   |   |
|-----------------------------------|---|
| <code>\language</code>            | ▶ <code>\language</code> stores the currently active (polyglossia) language name.   |
| <code>\mainlanguage</code>        | ▶ <code>\mainlanguage</code> stores the (polyglossia) language name of the main document language.  |
| <code>\languagevariant</code>     | ▶ <code>\languagevariant</code> stores the language variant if set. The macro is empty if no variant has been set.  |
| <code>\mainlanguagevariant</code> | ▶ <code>\mainlanguagevariant</code> stores the language variant of the main document language if set. The macro is empty if no variant has been set.  |
| <code>\babelname</code>           | ▶ <code>\babelname</code> stores the corresponding name of the currently active language (variant) in <code>babel</code> . This might not only be useful if you want to support both <code>babel</code> and <code>polyglossia</code> , but also since this name is unique for a given language variety (e.g., <code>ngerman</code> , <code>german</code> , <code>swissgerman</code> etc.). Note that this macro is also defined for languages that are not supported in <code>babel</code> . In that case, they are equal to the polyglossia language name. |
| <code>\mainbabelname</code>       | ▶ <code>\mainbabelname</code> analogously stores the name of document's main language (variant) in <code>babel</code> .   |

If you want to have a full list of loaded languages/variants, use the following macros:

|                           |  |
|---------------------------|--|
| <code>\xpg@loaded</code>  | ▶ <code>\xpg@loaded</code> stores a comma-separated list of all loaded languages (polyglossia name)            |
| <code>\xpg@vloaded</code> | ▶ <code>\xpg@vloaded</code> stores a comma-separated list of all loaded variants                               |
| <code>\xpg@bloaded</code> | ▶ <code>\xpg@bloaded</code> stores a comma-separated list of <code>babel</code> names of all language variants |

Finally, you can test whether a language is loaded by

|                                      |   |
|--------------------------------------|---|
| <code>\iflanguageloaded</code>       | <code>\iflanguageloaded{&lt;lang&gt;}{&lt;true&gt;}{&lt;false&gt;}</code>       |
|                                      | where <code>&lt;lang&gt;</code> is a <code>polyglossia</code> language name, or |
| <code>\ifbabbellanguageloaded</code> | <code>\ifbabbellanguageloaded{&lt;lang&gt;}{&lt;true&gt;}{&lt;false&gt;}</code> |
|                                      | where <code>&lt;lang&gt;</code> is a <code>babel</code> language name.          |

## 12 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 splendid 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` (`xpersian` and `bidi`). The work of `MOJCA MIKLAVEC` and `ARTHUR REU-`

TENAUER 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), ZDENĚK WAGNER (Hindi), MIKHAL OREN (Hebrew), SERGEY ASTANIN (Russian), KHALED HOSNY (Arabic), SERTAÇ Ö. YILDIZ (Turkish), KAMAL ABDALI (Urdu), and several other members of the X<sub>Y</sub>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 X<sub>Y</sub>TeX!

### 13 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 contributor statistics 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!