ΠΟΛΥΓΛΩΣΣΙΑ

Polyglossia: Modern multilingual typesetting with XƎLATEX and LuaLATEX

FRANÇOIS CHARETTE   ARTHUR REUTENAUER[*]
BASTIEN ROUCARIÈS   JÜRGEN SPITZMÜLLER

Contents

---

[*]Current maintainer

1

1   Introduction

Polyglossia is a package for facilitating multilingual typesetting with X∃LATEX and (with some exceptions) LuaLATEX. 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).[1]
8. Ensuring proper directionality if the document contains languages that are written from right to left (via the package bidi, available separately).

Several features of babel that do not make sense in the X∃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 X∃LATEX and LuaLATEX, it obviously also relies on fontspec by WILL ROBERTSON. For languages written from right to left, it needs the package bidi (for X∃TEX) or luabidi (for LuaTEX) by VAFA KHALIGHI (وفا خليقي) and the bidi-tex GitHub Organisation. Polyglossia also bundles three packages for calendaric computations (hebrewcal, hijrical, and farsical).

---

[1]This is done by bundled sub-packages such as arabicnumbers.

2 Setting up multilingual documents

2.1 Activating languages

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

`\setdefaultlanguage`

$$\text{\texttt{\textbackslash setdefaultlanguage[⟨options⟩]\{⟨lang⟩\}}}$$

`\setmainlanguage`
`\setotherlanguage`

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

$$\text{\texttt{\textbackslash setotherlanguage[⟨options⟩]\{⟨lang⟩\}.}}$$

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

`\setotherlanguages`

$$\text{\texttt{\textbackslash setotherlanguages\{⟨lang1⟩,⟨lang2⟩,⟨lang3⟩,…\}.}}$$

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.

2.2 Supported languages

Table 1 lists all languages currently supported. Those in red have specific options and/or commands that are explained in section 6 below.

v1.0.1
v1.1.1
v1.2.0

v1.43
v1.45

v1.46

Version Notes   The support for Amharic ← should be considered an experimental attempt to port the package ethiop; feedback is welcome. Version 1.1.1 ← added support for Asturian, Lithuanian, and Urdu. Version 1.2 ← introduced Armenian, Occitan, Bengali, Lao, Malayalam, Marathi, Tamil, Telugu, and Turkmen.[3] Version 1.43 ← brought basic support for Japanese (this is considered experimental, 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. Version 1.46 ← introduces support for Afrikaans, Belarusian, Bosnian and Georgian.

---

[2]Section 6 documents these options for the respective languages.

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

Table 1. Languages currently supported in polyglossia

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

2.3 Relation to and use of Babel language names

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 (*e.g., english*, `variant=american`).

Furthermore, babel sometimes uses abbreviated 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. Please refer to table 2 for the differing language names in both packages.

For convenience reasons, polyglossia also supports the use of babel names ← (for the few justified exceptions, please refer to the notes in table 2). The babel names listed in table 2 can be used instead of the corresponding polyglossia name/options in `\setdefaultlanguage` and `\setotherlanguage` as well as in the polyglossia and babel language switching commands/environments documented in section 3.1 and 3.2 (*e.g.,* `\textaustrian` is synonymous to `\textgerman[variant=austrian,spelling=old]`). However, unless you have special reasons, we strongly encourage you to use the polyglossia names.

Table 2. Babel-polyglossia language name matching

Babel name	Polyglossia name	Polyglossia options
acadien	french	variant=acadian
american	english	variant=american [*default*]
australian	english	variant=australian
austrian	german	variant=austrian, spelling=old
bahasa	malay	variant=indonesian [*default*]
bahasam	malay	variant=malaysian
brazil	portuguese	variant=brazilian
british	english	variant=british
canadian	english	variant=canadian
canadien	french	variant=canadian
farsi	persian	
friulan	friulian	
german[a]	german	spelling=old
irish	gaelic	variant=irish [*default*]
kurmanji	kurdish	variant=kurmanji
latin.classic[b]	latin	variant=classic
latin.ecclesiastic[c]	latin	variant=ecclesiastic
latin.medieval[d]	latin	variant=medieval
lowersorbian	sorbian	variant=lower
magyar	hungarian	
naustrian	german	variant=austrian
newzealand	english	variant=newzealand
ngerman	german	variant=german [*default*]
norsk	norwegian	variant=bokmal
nswissgerman	german	variant=swiss
nynorsk	norwegian	variant=nynorsk [*default*]
polutonikogreek	greek	variant=polytonic
portuges	portuguese	variant=portuguese [*default*]
samin	sami	
scottish	gaelic	variant=scottish
serbianc	serbian	script=Cyrillic
slovene	slovenian	
spanishmx	spanish	variant=mexican
swissgerman	german	variant=swiss, spelling=old
uppersorbian	sorbian	variant=upper

[a]Due to the name conflict only available in polyglossia as *germanb* (which is a babel synonym).
[b]Available in polyglossia as *latinclassic* (as dots are invalid in command names).
[c]Available in polyglossia as *latinecclesiastic* (as dots are invalid in command names).
[d]Available in polyglossia as *latinmedieval* (as dots are invalid in command names).

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 Afrikaans, Belarusian, Catalan, Czech, Dutch, Finnish, Georgian, German, Italian, Latin, Mongolian, Russian, and Slovak. 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⟩
\textlang

1.46

For each activated language the command \text⟨lang⟩[⟨options⟩]{…} (as well as the synonymous \textlang[⟨options⟩]{⟨lang⟩}{…} ←) becomes available for short insertions of text in that language. For example \textrussian{\today} and \textlang{russian}{\today} yield 15 ноября 2019 г. The commands switch to the correct hyphenation patterns, they activate some extra features for the selected language (such as extra spacing before punctuation in French), and they translate the date when using \today. They do 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⟩ and \textlang commands: It does everything the latter do, 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 commands, 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{quote}
 \begin{greek}[variant=ancient]
   μῆνιν ἄειδε θεὰ Πηληϊάδεω Ἀχιλῆος οὐλομένην, ἣ μυρί' Ἀχαιοῖς
   ἄλγε' ἔθηκε, πολλὰς δ' ἰφθίμους ψυχὰς Ἄϊδι προΐαψεν ἡρώων,
   αὐτοὺς δὲ ἑλώρια τεῦχε κύνεσσιν οἰωνοῖσί τε πᾶσι, Διὸς δ'
   ἐτελείετο βουλή, ἐξ οὗ δὴ τὰ πρῶτα διαστήτην ἐρίσαντε Ἀτρείδης
   τε ἄναξ ἀνδρῶν καὶ δῖος Ἀχιλλεύς.
 \end{greek}
\end{quote}
```

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

<div style="margin-left: auto">Arabic</div> 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 X҅LaTeX and LuaLaTeX 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 the ⟨lang⟩ environment, except that `\selectlanguage{⟨lang⟩}` does not need to be explicitly closed. The command `\foreinlanguage{⟨lang⟩}{…}` and the `otherlanguage*` environment are identical with the use of the `\text⟨lang⟩` or `\textlang` command, with the one notable exception that they do not translate the date with `\today`.

Since the X҅LaTeX and LuaLaTeX 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.[4]

\selectbackgroundlanguage

‣ \selectbackgroundlanguage{⟨lang⟩}: this selects the global font setup and the numbering definitions for the default language.

\resetdefaultlanguage

‣ \resetdefaultlanguage[⟨options⟩]{⟨lang⟩} (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 redefined to a non-latin script, this will reset to the font defined with \setmainfont etc. In a similar vein, it is possible to use \rmfamilylatin, \sffamilylatin, and \ttfamilylatin.

\rmfamilylatin
\sffamilylatin
\ttfamilylatin
\latinalph
\latinAlph

‣ \latinalph: Representation of counter as a lower-case letter: 1 = a, 2 = b, etc.

‣ \latinAlph: Representation of counter as a upper-case letter: 1 = A, 2 = B, etc.

3.4 Setting up alias commands

By means of the macro

\setlanguagealias
v1.46

$$\setlanguagealias[⟨options⟩]{⟨language⟩}{⟨alias⟩} \qquad \leftarrow$$

you can define alias commands for specific language (variants). *E.g.,*

\setlanguagealias[variant=austrian]{german}{deAT}

will define a command \textdeAT as well as an environment {deAT} which will link towards the command \textgerman[variant=austrian] and the environment {german}[variant=austrian], respectively. The aliases can also be used in the language switching commands described in section 3.1 and 3.2. Note, though, that the usual restrictions for command names apply, so something such as de-AT or de_AT will not work since - and _ are not allowed in command names.

For the latter case, and for the case where an alias would clash with an existing \text⟨...⟩ command (*e.g.,* \textit), a starred version \setlanguagealias*

\setlanguagealias*

is provided which does not define a \text⟨alias⟩ command (but which will set up the alias for everything else, including \textlang{⟨alias⟩}).

---

[4]Note that these commands require polyglossia language names; they do not support language/ babel aliases.

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 defined as is, so assure to do all necessary settings (please refer to the fontspec documentation for details). For instance, if `\arabicfont` is explicitly defined, then the option `Script=Arabic` should be included in that definition.

If a specific sans serif or monospace ('teletype') font is needed for a particular script or language, it can be defined by means of ← `\⟨script⟩fontsf` or `\⟨language⟩fontsf` and `\⟨script⟩fonttt` or `\⟨language⟩fonttt`, respectively.

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 ←

\pghyphenation   \pghyphenation[⟨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   \disablehyphenation and \enablehyphenation. Note that if you select a new
\enablehyphenation   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 com-
mands are defined. The default value of each option is given in italic.

6.1   afrikaans

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
  Afrikaans 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
       TEX).

  "~   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.

6.2 arabic

Options:

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

<div style="float:left">v1.0.3</div>

Commands:

<div style="float:left">\abjad
\abjadmaghribi
\aemph
v1.2.0</div>

- ▸ \abjad and \abjadmaghribi (see section 8.3)
- ▸ \aemph to emphasize text with \overline. ← \textarabic{\aemph{اب}} yields اب̄. This command is also available for Farsi, Urdu, etc.

6.3 armenian

Options:

<div style="float:left">v1.45
v1.45</div>

- ▸ **variant** ← = eastern or *western*
- ▸ **numerals** ← = armenian or *arabic*

6.4 belarusian ←

<div style="float:left">v1.46</div>

Options:

- ▸ **babelshorthands** = *false* or true. If this is turned on, the following shorthands 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).

---

[5]For Egypt, Sudan, Yemen and the Gulf states.
[6]For Iraq, Syria, Jordan, Lebanon and Palestine.

<dl>

"--- 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 »).

</dl>

‣ **numerals** = *arabic* or cyrillic. Uses either Arabic numerals or Cyrillic alphanumerical numbering.

‣ **spelling** = *modern* or classic (= tarask). With spelling=classic, captions and dates adhere to the Taraškievica (or Belarusian classical) orthography rather than the standard orthography.

Commands:

\Asbuk   ‣ \Asbuk: produces uppercased Cyrillic alphanumerals, for environments such as enumerate. The command takes a counter as argument, *e.g.*, \textbelarusian{\Asbuk{page}} produces ІД.

\asbuk   ‣ \asbuk: same in lowercase

6.5 bengali ←

_{v1.2.0}

Options:

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

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

6.6 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 *col·laborar*, similar

\L.L   to babel (the glyph U+00B7 MIDDLE DOT is used as a geminating sign).

_{v1.1.1}

6.7 czech

Options:

‣ **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 (only needed with `splithyphens=false`).

    `"'`   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 «).

‣ **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*). By default, this is done automatically ← (if you are using LuaTEX, the luavlna package is loaded to achieve this). Set this option to `false` to disable the feature.

‣ **vlna** = false or *true*. ← According to Czech typesetting conventions, single-letter words (non-syllable prepositions) must not occur at line ends. Polyglossia takes care of this automatically by default ← (if you are using LuaTEX, the luavlna package is loaded to achieve this). Set this option to `false` to disable the feature.

6.8 dutch

Options:

‣ **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 TEX).

    `"~`   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 `"-`).

6.9 english

Options:

- ‣ **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)

6.10 esperanto

Commands:

`\hodiau`
`\hodiaun`

- ‣ `\hodiau` and `\hodiaun` 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.

6.11 finnish

Options:

- ‣ **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.

6.12 french

Options:

> ‣ **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. 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.

> ‣ **thincolonspace** ← = true or *false*. Normally, a full (non-breaking) interword space is inserted before a colon. Use this option if you prefer a thinner space as used before ;, !, and ?.

> ‣ **autospaceguillemets**[7] = *true* or false. 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.

> ‣ **autospacetypewriter**[8] ← = 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, depending on the *autospacing* 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.

> ‣ **frenchitemlabels** ← = true or *false* (default value = true). If *true*, itemize item labels use em-dashes throughout, as common in French typography. Note that this option is only functional (by design) if French is the main language. Also, it might interfere with list packages such as enumitem.

> ‣ **itemlabels** ← = ⟨cmd⟩ (default value = \textemdash). If *frenchitemlabels* is true, you can customize here the used item label of all levels.

> ‣ **itemlabeli** ← = ⟨cmd⟩ (default value = \textemdash). If *frenchitemlabels* is

---

[7]Up to version 1.44, the option was called *automaticspacesaroundguillemets*. For backards compatibility reasons, the more verbose old option is still supported.

[8]Babel's syntax *OriginalTypewriter* is also supported.

17

true, you can customize here the used item label of the first level.

▸ **itemlabelii** ← = ⟨cmd⟩ (default value = \textemdash). If *frenchitemlabels* is true, you can customize here the used item label of the second level.

▸ **itemlabeliii** ← = ⟨cmd⟩ (default value = \textemdash). If *frenchitemlabels* is true, you can customize here the used item label of the third level.

▸ **itemlabeliv** ← = ⟨cmd⟩ (default value = \textemdash). If *frenchitemlabels* is true, you can customize here the used item label of the fourth level.

Commands:

`\NoAutoSpacing`
▸ \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`
▸ \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!}

6.13   gaelic ←

Options:
▸ **variant** = *irish* or scottish

6.14   georgian ←

Options:
▸ **babelshorthands** = *false* or true. If this is turned on, the following shorthands 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 georgian. Uses either Arabic numerals or Georgian alphanumerical numbering.
- **oldmonthnames** = *true* or false (default: false). Uses traditional Georgian month names.

6.15 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 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 blackletter ← (= fraktur ←). Setting script=blackletter
  adapts the captions for typesetting German in blackletter type (using the
  long s (ſ) where appropriate).

6.16   greek

Options:

▸ **variant** = *monotonic* (= mono), polytonic (= poly), or ancient

▸ **numerals** = *greek* or arabic

▸ **attic** = *false*/true

Commands:

\Greeknumber   ▸ \Greeknumber and \greeknumber (see section 8.3).

\greeknumber   ▸ The command \atticnumeral (= \atticnum) (activated with the option

\atticnumeral   attic=true), displays numbers using the acrophonic numbering system

\atticnum   (defined in the Unicode range U+10140–U+10174).[9]

6.17   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 מרחשון (Marcheshvan) rather
  than חשון (Heshvan), which is the default.

---

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

Commands:

v1.2.0

6.18   hindi ←

Options:

‣ **numerals** = Western or *Devanagari*

6.19   hungarian

Options:

v1.46

‣ **swapstrings** = *all*, captions, headings, headers, hheaders or none ←
In Hungarian, some caption strings need to be in a different order than in other languages (*e.g., 1. fejezet* instead of *Chapter 1*). By default, polyglossia tries hard to provide the correct order for different classes and packages (standard classes, KOMA-script, memoir, and titlesec package should work, as well as fancyhdr and caption). However, since the definition of these strings is not standardized, the redefinitions might not work and even interfere badly if you use specific classes or packages that redefine the respective strings themselves. In this case, you can disable some or all changes. The possibilities are:

     ‣ `all`: Redefine figure and table captions, part and chapter headings, and running headers (= default setting)

     ‣ `captions`: Redefine figure and table captions only

     ‣ `headings`: Redefine part and chapter headings only

     ‣ `headers`: Redefine running headers only

     ‣ `hheaders`: Redefine part and chapter headings as well as running headers

     ‣ `none`: Do not redefine anything

Commands:

\ontoday   ‣ \ontoday (= \ondatehungarian): special form of \today which produces a

\ondatehungarian   slightly different date format as used in prepositional phrases (such as 'on February 10th') in Hungarian.

6.20 italian

Options:

> ‣ **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 \-).

6.21 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

6.22 kurdish ←

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.
> ‣ **abjadjimnotail** = *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.

Table 3. Spelling differences between the Latin language variants.
The capitalization of month names and the use of *i/j* may be affected by
the `capitalizemonth` and the `usej` option.

classic	medieval	modern	ecclesiastic
Ianuarii	Ianuarii	Ianuarii	ianuarii
Nouembris	Nouembris	Novembris	novembris
Praefatio	Præfatio	Praefatio	Præfatio

`\MakeUppercase{Iulius}` yields:

IVLIVS	IVLIVS	IULIUS	IULIUS

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 6.2)

6.23 lao ←

Options:
- ▸ **numerals** = lao or *arabic*

6.24 latin

Options:

- ▸ **variant** = classic, medieval, *modern*, or ecclesiastic ←
  These variants refer to different spelling conventions. The `classic` and the `medieval` variant do not use the letters *U* and *v*, but only *V* and *u*. This concerns predefined terms like month names as well as the behaviour of the `\MakeUppercase` and the `\MakeLowercase` command. The `medieval` and the `ecclesiastic` variant use the ligatures æ and œ. See table 3 for examples.
  Furthermore, the `ecclesiastic` variant takes care for a punctuation spacing similar to French, but with smaller spaces, as provided for PDFTEX by the ecclesiastic package.

- ▸ **hyphenation** ← = classic, modern, or liturgical

Table 4. Latin default hyphenation styles

Language variant	Default hyphenation style
classic	classic
medieval	modern
modern	modern
ecclesiastic	modern

There are three different sets of hyphenation patterns for Latin. Separate documention for them is available on the Internet.[10] Each of the four variants mentioned above has its default set of hyphenation patterns as indicated by table 4. Use the hyphenation option if the default style does not fit your needs. Note that the liturgical hyphenation patterns are the default of none of the language variants. To use them, you have to say hyphenation=liturgical in any case.

v.1.46

‣ **ecclesiasticfootnotes** ← = true or *false*

Use footnotes as provided by the ecclesiastic package, which typesets footnotes with ordinary instead of superior numbers and without indentation. As many ecclesiastic documents and liturgical books use footnotes that are very similar to the ordinary LaTeX ones, we do not use this footnote style as default even for the ecclesiastic variant.

Note that this option is only possible if Latin is the main language of your document.

v.1.46

‣ **usej** ← = true or *false*

Use *J/j* in predefined terms. The letter *j* is not of ancient origin. In early modern times, it was used to distinguish the consonantic *i* from the vocalic *i*. Nowadays, the use of *j* has disappeared from most Latin publications. So false is the default value for all four language variants. Use this option if you prefer *Januarii* and *Maji* to *Ianuarii* and *Maii*.

v.1.46

‣ **capitalizemonth** ← = true or false

Capitalize the month name when printing dates (using the \today command). Traditionally, month names are capitalized. However, in recent liturgical books they are lowercase. So true is the default value for the variants classic, medieval, and modern, whereas false is the default value for the ecclesiastic variant.

---

[10]https://github.com/gregorio-project/hyphen-la/blob/master/doc/README.md#hyphenation-styles

‣ **babelshorthands** = true or *false*

Enable the following shorthands inherited from babel-latin and the ecclesiastic package.

"< for « (left guillemet)

"> for » (right guillemet)

"   If no other shorthand applies, " before any letter character defines an optional break point allowing further break points within the same word (as opposed to the \- command).

"| the same as ", but also possible before non-letter characters

'a for á (a with acute), also available for é, í, ó, ú, ý, ǽ, and ǿ

'A for Á (A with acute), also available for É, Í, Ó, Ú, V́, Ý, Ǽ, and Œ

The following shorthands are only available for the `medieval` and the `ecclesiastic` variant.

"ae for æ (ae ligature), also available for œ

"Ae for Æ (AE ligature), also available for Œ

"AE for Æ (AE ligature), also available for Œ

'ae for ǽ (ae ligature with acute), also available for ǿ

'Ae for Ǽ (AE ligature with acute), also available for Œ́

'AE for Ǽ (AE ligature with acute), also available for Œ́

‣ **prosodicshorthands** ← = true or *false*

Enable shorthands for prosodic marks (macrons and breves) very similiar to those provided by babel-latin using the `withprosodicmarks` modifier. Note that the active = character used for macrons will cause problems with commands using `key=value` interfaces, *e.g.,* `\includegraphics[scale=2]{...}`. Use `\shorthandoff{=}` before such commands (and `\shorthandon{=}` thereafter) within every environment with prosodic shorthands enabled. The following shorthands are available.

=a   for ā (a with macron), also available for ē, ī, ō, ū, and ȳ

=A   for Ā (A with macron), also available for Ē, Ī, Ō, Ū, V̄, and Ȳ. Note that a macron above the letter V is only displayed if your font supports the Unicode character 0304 (*combining macron*).

=ae for a͞e (ae diphthong with macron), also available for a͞u, e͞u, and o͞e. Note that macrons above diphthongs are only displayed if your font supports the Unicode character 035E (*combining double macron*).

25

=Ae for A͞e (Ae diphthong with macron), also available for A͞u, E͞u, and
O͞e.

=AE for A͞E (AE diphthong with macron), also available for A͞U, E͞U, and
O͞E.

^a for ă (a with breve), also available for ĕ, ĭ, ŏ, ŭ, and y̆. Note that a
breve above the letter y is only displayed if your font supports the
Unicode character `0306` (*combining breve*).

^A Ă (A with breve), also available for Ĕ, Ĭ, Ŏ, Ŭ, V̆, and Y̆. Note that
breves above the letters V and Y are only displayed if your font sup-
ports the Unicode character `0306` (*combining breve*).

6.25 malay

Options:

‣ **variant** ← = *indonesian* (= bahasai in babel) or malaysian (= bahasam in
babel)

6.26 mongolian ←

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.,* \textmongolian{\Asbuk{page}} produces КЗ.

\asbuk   ▸ \asbuk: same in lowercase

6.27 norwegian

Options:

- ▸ **variant** ← = bokmal (= 'norsk' in babel) or *nynorsk*

6.28 persian

Options:

- ▸ **numerals** = western or *eastern*

- ▸ **abjadjimnotail** = *false* or true. ← Set this to true if you want the *abjad* form of the number three to be ـج – as in the manuscript tradition – instead of the modern usage جـ.

Commands:

\abjad   ▸ \abjad (see section 8.3)

\aemph   ▸ \aemph (see section 6.2).

6.29 portuguese

Options:

- ▸ **variant** ← = brazilian or *portuguese*

6.30 russian

Options:

- ‣ **babelshorthands** = *false* or true. If this is turned on, the following shorthands 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.

- ‣ **indentfirst** ← = *true* or false. By default, all paragraphs are indented in Russian, also those after a chapter or section heading. If this option is false, the latter paragraphs are not indented, as normal in LaTeX.
- ‣ **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 КИ.

\asbuk   ‣ \asbuk: same in lowercase

6.31 sami ←

Currently support for Sami is limited to Northern Sami.

6.32 sanskrit

Options:

- ‣ **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.

‣ **numerals** = *Devanagari* ← or Western

6.33 serbian

Options:
- ‣ **script** = *Cyrillic* or Latin.
- ‣ **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 КѲ.

\asbuk

‣ \asbuk: same in lowercase

6.34 slovak

Options:

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

"=	for an explicit hyphen sign which is repeated at the beginning of the next line when hyphenated, as common in Slovak typesetting (only needed with splithyphens=false).
"|	disables a ligature at this position.
"~	for a hyphen sign without a breakpoint. Useful for cases where the hyphen should stick at the following syllable.
"-	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).
"/	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.
"'	for Slovak left double quotes (looks like „).
"'	for Slovak right double quotes (looks like ").
">	for Slovak left double guillemets (looks like »).

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

▸ **splithyphens** = *false* or true. ← According to Slovak 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*). By default, this is done automatically (if you are using LuaTEX, the luavlna package is loaded to achieve this). Set this option to false to disable the feature.

▸ **vlna** = *false* or true. ← According to Slovak typesetting conventions, single-letter words (non-syllable prepositions) must not occur at line ends. Polyglossia takes care of this automatically by default (if you are using LuaTEX, the luavlna package is loaded to achieve this). Set this option to false to disable the feature.

6.35 slovenian

Options:
▸ **localalph** = true or *false*

6.36 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)

Commands:
\oldtoday ▸ \oldtoday: outputs the current date using traditional Sorbian month names, even if **olddate** is false.

6.37 spanish

Options:
▸ **variant** = *spanish* or mexican. ←
▸ **spanishoperators** = all, accented, spaced or none or *false* (default value = all). ← Determines of and how math operators are localized to Spanish.
  ▸ accented causes some math operators to use accents where usual in Spanish (*lím, lím sup, lím inf, máx, mín, ínf, mód*).

- ‣ spaced causes some math operators to use spaces where usual in Spanish (*arc cos, arc sen, arc tg*).
- ‣ all activates accented and spaced and furthermore provides Spanish localizations of \sin (*sen*), \tan (*tg*), \sinh (*senh*), and \tanh (*tgh*).
- ‣ none does no localization at all (default setting).

Commands: ←

\arcsen
\arctg
\sen
\senh
\tg
\tgh
\spanishoperator

- ‣ \arcsen: alias to \arcsin (babel compatibility)
- ‣ \arctg: alias to \arctan (babel compatibility)
- ‣ \sen: alias to \sin (babel compatibility)
- ‣ \senh: alias to \sinh (babel compatibility)
- ‣ \tg: alias to \tan (babel compatibility)
- ‣ \tgh: alias to \tanh (babel compatibility)
- ‣ \spanishoperator: allows you to define further localized operators. For instance, \spanishoperator{cotg} defines a command \cotg that outputs *cotg* in math. The optional argument of the command lets you specify the spelling, if needed, *e.g.,* \spanishoperator[arc\,ctg]{arcctg}.

6.38 syriac

Options:

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

Commands:

\abjadsyriac
\aemph

- ‣ \abjadsyriac (see section 8.3)
- ‣ \aemph (see section 6.2).

6.39 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.

6.40 tibetan

Options:

- ‣ **numerals** = tibetan or *arabic*

6.41 ukrainian

Commands:

\Asbuk   ▸ \Asbuk: produces the uppercase Ukrainian alphabet, for environments such as `enumerate`

\asbuk   ▸ \asbuk: same in lowercase

6.42 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 replaces 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 [= alphanumeric] 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

`\localnumeral`
As of 1.45, ← 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 ٤٢, whereas in an Hebrew environment, it results in מב 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.

`\localnumeral*`
For ← 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 ٣٤.

`\Localnumeral`
`\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 <language>.
   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 `<language>` (*e.g.,* `\localnumeral[lang=arabic]{42}}`).

8.2 Non-Western decimal digits

In addition ← 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.[11] The macros have the form `\<script>digits`. They convert 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:

`\arabicdigits`   ‣ `\arabicdigits`
`\bengalidigits`   ‣ `\bengalidigits`
`\devanagaridigits`   ‣ `\devanagaridigits`
`\farsidigits`   ‣ `\farsidigits`
`\kannadadigits`   ‣ `\kannadadigits`
`\khmerdigits`   ‣ `\khmerdigits`

---

[11] 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 ٢٠١٠. Note 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.

\laodigits	‣ \laodigits
\nkodigits	‣ \nkodigits
\thaidigits	‣ \thaidigits
\tibetandigits	‣ \tibetandigits

8.3 Non-Latin alphabetic numbering

For languages which use special (non-Latin) alphanumerical notation[12], 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:

\abjad ‣ \abjad outputs Arabic *abjad* numbers according to the Mashriq varieties. Example: \abjad{1863} yields غضسج.

\abjadmaghribi ‣ \abjadmaghribi outputs Arabic *abjad* numbers according to the Maghrib varieties. Example: \abjadmaghribi{1863} yields شظصج.
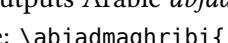
\abjadsyriac ‣ \abjadsyriac outputs Syriac abjad numerals.[13] Example: \abjadsyriac{463} yields ܠܐܣܝ.
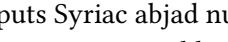
\armeniannumeral ‣ \armeniannumeral produces Armenian alphabetic numbering. Example: \armeniannumeral{1863} yields ՈՊԿԳ.
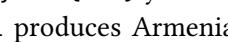
\georgiannumeral ‣ \georgiannumeral produces Georgian alphabetic numbering. Example: \georgiannumeral{1863} yields ჩყჲგ.
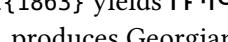
\greeknumeral
\Greeknumeral ‣ \greeknumeral produces Greek alphabetic numbering, \Greeknumeral outputs uppercased variants. Example: \greeknumeral{1863} yields ͵αωξγ´, \Greeknumeral{1863} results in ͵ΑΩΞΓ´.
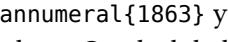
\hebrewnumeral
\Hebrewnumeral
\Hebrewnumeralfinal ‣ \hebrewnumeral, \Hebrewnumeral and \Hebrewnumeralfinal generate variants of Hebrew alphanumeric numerals. The commands behave exactly as they do in babel: \hebrewnumeral outputs the numbers without any decoration, \Hebrewnumeral adds *gereshayim* before the last letter, \Hebrewnumeralfinal uses in addition the final forms of Hebrew letters. Examples: \hebrewnumeral{1750} yields אתשנ, \Hebrewnumeral{1750}

---
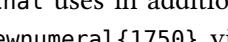
[12]For instance, see http://en.wikipedia.org/wiki/Greek_numerals, http://en.wikipedia.org/wiki/Abjad_numerals, http://en.wikipedia.org/wiki/Hebrew_numerals, and http://en.wikipedia.org/wiki/Syriac_alphabet.

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

yields אתש״נ, and \Hebrewnumeralfinal{1750} yields אתש״ן.

\russiannumeral	‣ \russiannumeral produces Russian numbering, with uppercased variant (for alphanumerical variant) via \Russiannumeral. Depending on the numerals option in the Russian language selection, this is either Arabic digit or Cyrillic alphanumercial output.
\Russiannumeral	

Example: With numerals=latin \russiannumeral{19} yields 19, with numerals=cyrillic \russiannumeral{19} results in ιθ.

\serbiannumeral
\Serbiannumeral

‣ \serbiannumeral produces Serbian numbering, with uppercased variant (for alphanumerical variant) via \Serbiannumeral. Depending on the numerals option in the Serbian language selection, this is either Arabic digit or Cyrillic alphanumercial output.

Example: With numerals=latin \serbiannumeral{19} yields 19, with numerals=cyrillic \serbiannumeral{19} results in ιθ.

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.

\RTLfootnote
\LTRfootnote

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⫬TEX and LuaTEX output (this is due to differences in the bidi and luabidi packages). With X⫬TEX, footnote rules are always placed left, which is often wrong in RTL context. With LuaTEX, 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 `hebcal.sty` that comes with babel. The command `\Hebrewtoday` formats the current date in the Hebrew
`\Hebrewtoday` 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.[14] It provides two macros for the end-user. The command

---

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

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

\Hijritoday sets the counters Hijriday, Hijrimonth and Hijriyear. \Hijritoday formats the

v1.1.1 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

\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.[15] 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 FarsiTₑX.[16] Here we have re-

\Jalalitoday named the command \ftoday to \Jalalitoday. Example: today is 24 Ābān 1398.

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:

\languagename   ▸ \languagename stores the currently active (polyglossia) language name.

\mainlanguagename   ▸ \mainlanguagename stores the (polyglossia) language name of the main document language.

\languagevariant   ▸ \languagevariant stores the language variant if set. The macro is empty if no variant has been set.

---

[15]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.

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

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

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

`\xpg@loaded`	‣ `\xpg@loaded` stores a comma-separated list of all loaded languages (polyglossia name)
`\xpg@vloaded`	‣ `\xpg@vloaded` stores a comma-separated list of all loaded variants
`\xpg@bloaded`	‣ `\xpg@bloaded` stores a comma-separated list of babel names of all language variants

Finally, you can test whether a language is loaded by

`\iflanguageloaded`	`\iflanguageloaded{⟨lang⟩}{⟨true⟩}{⟨false⟩}`

where ⟨lang⟩ is a polyglossia language name, or

`\ifbabellanguageloaded`	`\ifbabellanguageloaded{⟨lang⟩}{⟨true⟩}{⟨false⟩}`

where ⟨lang⟩ is a babel 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 comparision: Johannes Braams and the numerous contributors to the babel package (in particular Boris Lavva and others for its Hebrew support), Alexej Kryukov (antomega), Will Robertson (fontspec), Apostolos Syropoulos (xgreek), Youssef Jabri (arabi), and Vafa Khalighi (xepersian and bidi). The work of Mojca Miklavec and Arthur Reutenauer on hyphenation patterns with their package hyph-utf8 is of course invaluable. I should also thank other individuals for their assistance in supporting specific languages: Yves Codet (Sanskrit), 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 XƎTEX

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∃TEX!

13 More acknowledgements (by the current development team)

Many thanks to all the people who have contributed bugfixes and new features to polyglossia since we took over. In alphabetical order: IGNAS ANIKEVICIUS, SINA AHMADI, WOUTER BOLSTERLEE, CHRISTIAN BUHTZ, ZGARBUL ANDREY, OLEG DOMANOV, PHILIPP GESANG, KEVIN GODBY, ENRICO GREGORIO, KHALED HOSNY, NAJIB IDRISSI, user JULROY67, DOHYUN KIM, PHIL KIME, MIKE KROUTIKOV, IVAN KOKAN, CALEB MACLENNAN, JOSÉ MANCERA, YEVGEN POGRIBNYI, MAÏEUL ROUQUETTE, ELIE ROUX, HUGO ROY, GUY RUTENBERG, PHILIPP STEPHANI, KENO WEHR, DOMINIK WUJASTYK, SERTAÇ Ö. YILDIZ, MAKSIM ZHOLUDEV, YAN ZHOU, and STEFAN ZLATINOV. Their respective contributions can be identified from the contributor statistics on GitHub.

Among the ones who sent contributions directly to us we would like to especially thank CLAUDIO BECCARI, the indefatigable champion of Romance languages, and beyond! Furthermore, kudos go to MORITZ WEMHEUER (of biblatex) who has helped a lot to improve polyglossia interaction with biblatex and csquotes.

Not at least, we are very grateful for all bug reports and feature enhancement requests we received from the numerous users we cannot list all here (but again, you can find all names on GitHub). Please go on with that, you are keeping polyglossia running!