Polyglossia: Modern multilingual typesetting with XeLaTeX and LuaLaTeX

François Charette Arthur Reutenauer Udi Fogiel
Bastien Roucariès Jürgen Spitzmüller

2025/06/30 v2.6*

(PDF file generated on 30 June 2025)

Contents

1 Introduction **4**

2 Setting up multilingual documents **4**
2.1 Activating languages 4
2.2 Supported languages 5
2.3 Relation to and use of Babel language names 6
2.4 Using IETF language tags 6
2.5 Global options 12
2.6 Shorthands 13

3 Language-switching commands **13**
3.1 Recommended commands 13
3.2 Babel commands 14
3.3 Other commands 15
3.4 Setting up alias commands 15

4 Font setup **16**

5 Adapting hyphenation **17**
5.1 Hyphenation exceptions 17
5.2 Hyphenation thresholds 17
5.3 Hyphenation disabling 18

*Please report bugs to http://github.com/reutenauer/polyglossia/issues

6 Language-specific options and commands 18
6.1 afrikaans . . . 18
6.2 arabic . . . 19
6.3 armenian . . . 20
6.4 belarusian . . . 20
6.5 bengali . . . 21
6.6 catalan . . . 22
6.7 chinese . . . 22
6.8 croatian . . . 22
6.9 czech . . . 23
6.10 dutch . . . 24
6.11 english . . . 25
6.12 esperanto . . . 25
6.13 finnish . . . 25
6.14 french . . . 26
6.15 gaelic . . . 28
6.16 georgian . . . 28
6.17 german . . . 29
6.18 greek . . . 31
6.19 hebrew . . . 31
6.20 hindi . . . 32
6.21 hungarian . . . 32
6.22 italian . . . 33
6.23 korean . . . 33
6.24 kurdish . . . 34
6.25 lao . . . 34
6.26 latin . . . 34
6.27 malay . . . 37
6.28 marathi . . . 37
6.29 mongolian . . . 37
6.30 norwegian . . . 39
6.31 odia . . . 39
6.32 persian . . . 39
6.33 polish . . . 39
6.34 portuguese . . . 40
6.35 punjabi . . . 41
6.36 russian . . . 41
6.37 sami . . . 43
6.38 sanskrit . . . 43
6.39 serbian . . . 43
6.40 slovak . . . 44

6.41 slovenian . . . 45
6.42 sorbian . . . 46
6.43 spanish . . . 46
6.44 syriac . . . 46
6.45 thai . . . 47
6.46 tibetan . . . 47
6.47 ukrainian . . . 47
6.48 urdu . . . 49
6.49 uyghur . . . 49
6.50 welsh . . . 50

7 Modifying or extending captions, date formats and language settings 50

8 Script-specific numbering 51
8.1 General localization of numbering . . . 52
8.2 Non-Western decimal digits . . . 52
8.3 Non-Latin alphabetic numbering . . . 53

9 Footnotes in right-to-left context 55
9.1 Horizontal footnote position . . . 55
9.2 Footnote rule length and position . . . 55

10 Calendars 56
10.1 Hebrew calendar (hebrewcal.sty) . . . 56
10.2 Islamic calendar (hijrical.sty) . . . 56
10.3 Farsi (jalālī) calendar (farsical.sty) . . . 57

11 Auxiliary commands 57

12 Accessing language information 57

13 Revision history 59

14 Acknowledgements (by François Charette) 78

15 More acknowledgements (by the current development team) 79

1 Introduction

Polyglossia is a package for facilitating multilingual typesetting with XeLaTeX and 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 XeTeX world (like font encodings, shorthands, etc.) are not supported. Generally speaking, polyglossia aims to remain as compatible as possible with the fundamental features of babel while being cleaner, light-weight, and modern. The package antomega has been very beneficial in our attempt to reach this objective.

Requirements The current version of polyglossia makes use of some convenient macros defined in the etoolbox package by Philipp Lehmann and Joseph Wright. Being designed for XeLaTeX and LuaLaTeX, it obviously also relies on fontspec by Will Robertson. For languages written from right to left, it needs the package bidi (for XeTeX) 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).

2 Setting up multilingual documents

2.1 Activating languages

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

\setdefaultlanguage

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

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

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

\setotherlanguage `\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 `\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.

Table 1. Languages currently supported in polyglossia

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

Nota bene Support for Chinese and Japanese is experimental. This particularly concerns the line breaking mechanism which is a proof of concept. Please use with care and report any issues and unexpected output you encounter.

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

v1.0.1 v1.1.1 v1.2.0 v1.43 v1.45 v1.46 v1.52 v1.59 v1.66

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. Version 1.52 ← introduces support for Uyghur. Version 1.59 ← adds support for (simplified and traditional) Chinese and for Punjabi. Version 1.66 ← introduces support for Odia.

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.

v1.46 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.

2.4 Using IETF language tags

v1.47 Polyglossia ← also supports the use of language tags that conform to the

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

Table 2. Babel-polyglossia language name matching

Babel name	Polyglossia name	Polyglossia options
acadian	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
classicallatin	latin	variant=classic
farsi	persian	
ecclesiasticallatin	latin	variant=ecclesiastic
friulan	friulian	
german[a]	german	spelling=old
irish	gaelic	variant=irish [*default*]
kurmanji	kurdish	variant=kurmanji
lowersorbian	sorbian	variant=lower
magyar	hungarian	
medievallatin	latin	variant=medieval
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 [*default*]

[a] Due to the name conflict only available in polyglossia as *germanb* (which is a babel synonym).

IETF BCP-47 *Best Current Practice*.[4] Thus, you can use tags such as `en-GB` (for British English) or `de-AT-1901` (for Austrian German, old spelling) in `\setdefaultlanguage` and `\setotherlanguage` as well as in the language switching command `\textlang{⟨tag⟩}`, the environment `\begin{lang}{⟨tag⟩}` … `\end{lang}` and the babel language switching commands/environments documented in section 3.2. Table 3 lists the currently supported tags.

Table 3. BCP47-polyglossia language name matching

BCP-47 tag	Polyglossia name	Polyglossia options
aeb	arabic	locale=tunisia
af	afrikaans	
afb	arabic	locale=default
am	amharic	
apd	arabic	locale=default
ar	arabic	
ar-IQ	arabic	locale=mashriq
ar-JO	arabic	locale=mashriq
ar-LB	arabic	locale=mashriq
ar-MR	arabic	locale=mauritania
ar-PS	arabic	locale=mashriq
ar-SY	arabic	locale=mashriq
ar-YE	arabic	locale=default
arq	arabic	locale=algeria
ary	arabic	locale=morocco
arz	arabic	locale=default
ast	asturian	
ayl	arabic	locale=libya
be	belarusian	
be-tarask	belarusian	spelling=classic
bg	bulgarian	
bn	bengali	
bo	tibetan	
br	breton	
bs	bosnian	
ca	catalan	
ckb	kurdish	variant=sorani [*default*]
ckb-Arab	kurdish	variant=sorani, script=Arabic [*default*]
ckb-Latn	kurdish	variant=sorani, script=Latin
cop	coptic	
cy	welsh	
cz	czech	
da	danish	

[4] Please refer to https://tools.ietf.org/html/bcp47 and https://en.wikipedia.org/wiki/IETF_language_tag for details.

Table 3. BCP47-polyglossia language name matching (*continued*)

BCP-47 tag	Polyglossia name	Polyglossia options
de	german	
de-AT	german	variant=austrian, spelling=new
de-AT-1901	german	variant=austrian, spelling=old
de-AT-1996	german	variant=austrian, spelling=new
de-CH	german	variant=swiss, spelling=new
de-CH-1901	german	variant=swiss, spelling=old
de-CH-1996	german	variant=swiss, spelling=new
de-DE	german	variant=german, spelling=new
de-DE-1901	german	variant=german, spelling=old
de-DE-1996	german	variant=german, spelling=new [*default*]
de-Latf	german	script=blackletter
de-Latf-AT	german	variant=austrian, spelling=new, script=blackletter
de-Latf-AT-1901	german	variant=austrian, spelling=old, script=blackletter
de-Latf-AT-1996	german	variant=austrian, spelling=new, script=blackletter
de-Latf-CH	german	variant=swiss, spelling=new, script=blackletter
de-Latf-CH-1901	german	variant=swiss, spelling=old, script=blackletter
de-Latf-CH-1996	german	variant=swiss, spelling=new, script=blackletter
de-Latf-DE	german	variant=german, spelling=new, script=blackletter
de-Latf-DE-1901	german	variant=german, spelling=old, script=blackletter
de-Latf-DE-1996	german	variant=german, spelling=new, script=blackletter
dsb	sorbian	variant=lower
dv	divehi	
el	greek	
el-monoton	greek	variant=monotonic [*default*]
el-polyton	greek	variant=polytonic
en	english	
en-AU	english	variant=australian
en-CA	english	variant=canadian
en-GB	english	variant=british
en-NZ	english	variant=newzealand
en-US	english	variant=us [*default*]
eo	esperanto	
es	spanish	
es-ES	spanish	variant=spanish [*default*]
es-MX	spanish	variant=mexican

Table 3. BCP47-polyglossia language name matching (*continued*)

BCP-47 tag	Polyglossia name	Polyglossia options
et	estonian	
eu	basque	
fa	persian	
fi	finnish	
fr	french	
fr-CA	french	variant=canadian
fr-CA-u-sd-canb	french	variant=acadian
fr-CH	french	variant=swiss
fr-FR	french	variant=french [*default*]
fur	friulian	
ga	gaelic	variant=irish [*default*]
gd	gaelic	variant=scottish
gl	galician	
grc	greek	variant=ancient
he	hebrew	
hi	hindi	
hr	croatian	
hsb	sorbian	variant=upper [*default*]
hu	hungarian	
hy	armenian	
ia	interlingua	
id	malay	variant=indonesian
is	icelandic	
it	italian	
ja	japanese	
ka	georgian	
km	khmer	
kmr	kurdish	variant=kurmanji
kmr-Arab	kurdish	variant=kurmanji, script=Arabic
kmr-Latn	kurdish	variant=kurmanji, script=Latin
kn	kannada	
ko	korean	
ku	kurdish	
ku-Arab	kurdish	script=Arabic
ku-Latn	kurdish	script=Latin
la	latin	
la-x-classic	latin	variant=classic
la-x-ecclesia	latin	variant=ecclesiastic
la-x-medieval	latin	variant=medieval
lo	lao	
lt	lithuanian	
lv	latvian	
mk	macedonian	
ml	malayalam	

Table 3. BCP47-polyglossia language name matching (*continued*)

BCP-47 tag	Polyglossia name	Polyglossia options
mn	mongolian	
mr	marathi	
nb	norwegian	variant=bokmal
nko	nko	
nl	dutch	
nn	norwegian	variant=nynorsk [*default*]
oc	occitan	
or	odia	changecounternumbering=false, numerals=Devanagari [*default*]
pa	punjabi	numerals=gurmukhi
pl	polish	
pms	piedmontese	
pt	portuguese	
pt-BR	portuguese	variant=brazilian
pt-PT	portuguese	variant=portuguese [*default*]
rm	romansh	
ro	romanian	
ru	russian	
ru-luna1918	russian	spelling=modern [*default*]
ru-petr1708	russian	spelling=old
sa	sanskrit	
sa-Beng	sanskrit	script=Bengali
sa-Deva	sanskrit	script=Devanagari [*default*]
sa-Gujr	sanskrit	script=Gujarati
sa-Knda	sanskrit	script=Kannada
sa-Mlym	sanskrit	script=Malayalam
sa-Telu	sanskrit	script=Telugu
se	sami	
sk	slovak	
sl	slovenian	
sq	albanian	
sr	serbian	
sr-Cyrl	serbian	script=Cyrillic
sr-Latn	serbian	script=Latin [*default*]
sv	swedish	
syr	syriac	
ta	tamil	
te	telugu	
th	thai	
tk	turkmen	
tr	turkish	
ug	uyghur	
uk	ukrainian	
ur	urdu	

Table 3. BCP47-polyglossia language name matching (*continued*)

BCP-47 tag	Polyglossia name	Polyglossia options
vi	vietnamese	
zh	chinese	
zh-CN	chinese	variant=simplified [*default*]
zh-TW	chinese	variant=traditional
zsm	malay	variant=malaysian [*default*]

2.5 Global options

Polyglossia can be loaded with the following global package options:

v1.1.1
- **babelshorthands** ← = *true or *false*
  Globally activates babel shorthands whenever available. Please refer to sec. 2.6 for details.
- **localmarks** = *true or *false*
  redefines the internal LaTeX macros \markboth and \markright to the effect that the header text is explicitly set in the currently active language (*i.e.*, wrapped into \foreignlanguage{⟨lang⟩}{⟨…⟩}).

v1.2.0 In earlier versions of polyglossia, ← this option was enabled by default, but we now realize that it causes more problems than it helps (since it breaks if a package or class redefines \markboth or \markright), so it is now disabled by default. For backwards compatibility, the option **nolocalmarks** which used to switch off the previous default, and now equals the default, is still available.

v1.50
- **luatexrenderer** ← = ⟨renderer⟩ (default value: Harfbuzz)
  determines which font renderer is used with LuaTeX output. The correct font renderer is essential particularly for non-Latin scripts. By default, polyglossia uses the Harfbuzz renderer that has been introduced to LuaTeX in 2019 (TeXLive 2020), as this gives the best results generally. If you want to use a different renderer, you can specify this here (or individually for specific fonts via the optional argument of the font selection commands). Please refer to the fontspec manual for supported values and for details on how to change the renderer for individual fonts.
  luatexrenderer=none disables polyglossia's automatic renderer setting.
- **verbose** = *true or false
  determines whether info messages and (some of the) warnings issued by LaTeX, fontspec and polyglossia are output.

2.6 Shorthands

Babel has introduced the nifty feature of "shorthands" for some of the languages it supports. Shorthands make use of active characters to provide quick access to glyphs and formatting peculiarities that are common in a respective language (*e.g.,* `"a` for German umlaut ⟨ä⟩ in case this is not available on your keyboard). Polyglossia has adopted this feature, we call it *babelshorthands*.

Shorthands need to be activated for the respective language, or can be activated globally, via the boolean option **`babelshorthands`** (either passed globally when loading polyglossia or to a specific language when loading it).

Shorthands are implemented currently for Afrikaans, Belarusian, Catalan, Croatian, Czech, Dutch, Finnish, Georgian, German, Italian, Latin, Mongolian, Occitan, Piedmontese, Polish, Portuguese, Russian, Slovak, and Ukrainian, and their number and function differs between these languages. Please refer to the respective language descriptions (sec. 6) for detailed elaborations.

If you want to re-use the shorthands defined for a specific language in a different language, you can do so by using in preamble: ←

v2.0 `\inheritbabelshorthands`

```
\inheritbabelshorthands{⟨source language⟩}{⟨target language⟩}
```

in order to input and activate the shorthands defined for the `⟨source language⟩` in the `⟨target language⟩`. Both languages will be loaded as "other" languages if they have not been loaded yet. If one of the languages does not exist, or if the `⟨source language⟩` does not provide any shorthands, you will get an error message.

Note, also, that you can only use `\inheritbabelshorthands` once per language. Subsequent calls will overwrite previous ones (and also the shorthand that might be defined for a `⟨target language⟩` in the first place).

If you want to enable the shorthands of a specific language inline, also when the language itself is not active (but loaded in the preamble), you can do so by means of ←

v2.0 `\usebabelshorthands`

```
\usebabelshorthands{⟨language⟩}
```

The effect of this is local to groups. If you want to disable any active shorthands, you can use `\usebabelshorthands{none}`.

3 Language-switching commands

3.1 Recommended commands

`\text⟨lang⟩` `\textlang` v1.46

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 30 июня 2025 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 currently active 'outer' language).

`⟨lang⟩` (*env.*) The environment `⟨lang⟩`, which is also available for any activated language
`lang` (as well as the equivalent `\begin{lang}[⟨options⟩]{⟨lang⟩}` … `\end{lang}` ←), 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*:

v1.47

```
\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 Ἀτρεΐδης τε ἄναξ ἀνδρῶν καὶ δῖος Ἀχιλλεύς.

`Arabic` (*env.*) 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 XeLaTeX and LuaLaTeX format) are redefined, but keep a similar behaviour.

`\selectlanguage` ▸ `\selectlanguage[⟨options⟩]{⟨lang⟩}`

`\foreignlanguage` ▸ `\foreignlanguage[⟨options⟩]{⟨lang⟩}{⟨…⟩}`

`otherlanguage` (*env.*) ▸ `\begin{otherlanguage}[⟨options⟩]{⟨lang⟩}` … `\end{otherlanguage}`

`otherlanguage*` (*env.*) ▸ `\begin{otherlanguage*}[⟨options⟩]{⟨lang⟩}` … `\end{otherlanguage*}`

hyphenrules (*env.*)
v1.50

- `\begin{hyphenrules}[⟨options⟩]{⟨lang⟩}` … `\end{hyphenrules}` ← `\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 `\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`.

  The `⟨hyphenrules⟩` environment only switches the hyphenation patterns to the one associated with the language `⟨lang⟩` (or the language variety as specified via `⟨options⟩`). It does no further language-specific change.

  Since the XeLaTeX and LuaLaTeX format incorporate babel's `hyphen.cfg`, the low-level commands for hyphenation and language switching defined there are in principal also accessible. Note, however, that the availability of such low-level commands is not guaranteed, as `hyphen.cfg`, which is out of polyglossia's control, is (or at least has been) subject to regular change.

3.3 Other commands

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

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

\resetdefaultlanguage
- `\resetdefaultlanguage[⟨options⟩]{⟨lang⟩}` (experimental): Switches the default language to another one in the middle of a document: *this may have adverse effects*!

\normalfontlatin
\rmfamilylatin
\sffamilylatin
\ttfamilylatin
- `\normalfontlatin`: in an environment where `\normalfont` has been redefined to a non-latin script, this will reset to the font defined with `\setmainfont` etc. In a similar vein, it is possible to use `\rmfamilylatin`, `\sffamilylatin`, and `\ttfamilylatin`.

\latinalph
- `\latinalph{⟨counter⟩}`: Representation of counter as a lower-case letter: 1 = a, 2 = b, etc.

\latinAlph
- `\latinAlph{⟨counter⟩}`: Representation of counter as a upper-case letter: 1 = A, 2 = B, etc.

3.4 Setting up alias commands

v1.46
\setlanguagealias
By means of the macro ←

```
\setlanguagealias[⟨options⟩]{⟨language⟩}{⟨alias⟩}
```

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

```
\setlanguagealias[variant=austrian]{german}{AT}
```

will define a command `\textAT` as well as an environment `{AT}` 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 (the same holds true for any non-ASCII character and for digits).

For the latter case, and for the case where an alias would clash with an existing command (*e.g.,* \fi) or a \text⟨…⟩ command (*e.g.,* \textit), a starred version
\setlanguagealias* \setlanguagealias* is provided which does neither define a \text⟨alias⟩ command nor an ⟨alias⟩ environment, but which will set up the alias for everything else, including \textlang{⟨alias⟩} and \begin{lang}{⟨alias⟩}.

Polyglossia comes with some aliases predefined, namely aliases for babel language names (see sec. 2.3) and for IETF BCP-47 language tags (the latter via \setlanguagealias*; see sec. 2.4).

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. Note that ⟨language⟩ points to the polyglossia language name, ⟨script⟩ might point either to an OpenType script name as defined by fontspec (*e.g.,* Latin) ←or a BCP-47 script tag (*e.g.,* Latn), though the script is fully lower-cased in the \⟨script⟩font command.
V2.2

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⟩}
```

If you want to setup a blackletter font specifically for the German blackletter script variant, you might do so by:

```
\newfontfamily\latffont[⟨…⟩]{⟨font⟩}
```

Setting up a font specifically for traditional Chinese is possible via:

```
\newfontfamily\hantfont[Script=CJK,⟨…⟩]{⟨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.
V1.2.0

Whenever a new language is activated, polyglossia will first check whether a font has been defined for that language. If this is not the case, it will check whether a font associated with the BCP-47 script tag of the current language is defined. If this also isn't the case, polyglossia will finally check (but this only

for languages in non-Latin scripts) whether there is a font associated with the OpenType script tag. If none of these fonts is defined, polyglossia will use the currently active font as defined by `\setmainfont`, `\setsansfont` or `\setmonofont` 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 only to the currently active language.

v1.45 `\pghyphenation`

In addition to this, polyglossia provides the command ←

```
\pghyphenation[⟨options⟩]{⟨lang⟩}{⟨exceptions⟩}
```

which can be used to define exceptions that only apply to a specific language or language variant, respectively.[5]

5.2 Hyphenation thresholds

Polyglossia sets reasonable defaults for the hyphenation thresholds of each language, *i.e.,* the number of characters that must at least be there at the beginning or end of a word before it is hyphenated (`\lefthyphenmin` and `\righthyphenmin` in TEX). For instance, with English, this threshold is 2 at the beginning ('left') and 3 at the end ('right'), so a word will not be hyphenated within the first two characters at the beginning and the last three characters at the end.

v1.50 `\setlanghyphenmins`

To change this value, polyglossia provides the command ←

```
\setlanghyphenmins[⟨options⟩]{⟨lang⟩}{⟨l⟩}{⟨r⟩}
```

where `⟨lang⟩` is to be replaced with the respective language name or alias, `⟨options⟩` can be used to delimit the modification to a particular variety (*e.g.,*

[5] More precisely, it applies to all languages or varieties that share the same hyphenation patterns. Hence, for instance, all variants of **`german`** with **`spelling=new`** will share the same exceptions, as opposed to **`spelling=old`** which uses different patterns.

via `variant` or `spelling`), ⟨l⟩ with the left threshold value (*e.g.,* 3), and ⟨r⟩ with the right one (*e.g.,* `\setlanghyphenmins[spelling=old]{german}{4}{4}`). This setting can be changed repeatedly in the preamble and the document body. It applies to all subsequent text in the respective language (variety), but only after the next language switch.

5.3 Hyphenation disabling

In some very specific contexts (such as music score creation), TEX hyphenation is something to avoid completely as it may cause troubles. Polyglossia provides
`\disablehyphenation` two functions: `\disablehyphenation` and `\enablehyphenation`. Note that if you
`\enablehyphenation` 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. Note the following conventions:

- The preset value of each option (*i.e.,* the setting that applies by default, if the option is not explicitly set) is given in *`italics`*.
- If an option key may be used without a value, the value that applies for value-less keys is marked by a preceding `*asterisk`.

For instance, `babelshorthands = *true` or *`false`* means that `babelshorthands` is `false` by default in the respective language, and that passing `babelshorthands` (without value) is equivalent to passing `babelshorthands=true`.

6.1 afrikaans

Options:

V1.1.1

- **`babelshorthands`** ⟵ = `*true` or *`false`*
  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`)
- **`hijricorrection`** = ⟨`integer`⟩ (default value: `0`)
  If `calendar=hijri` or `calendar=islamic` is selected, `\today` will output the date according to the lunar Islamic (Hijra) calendar. This option allows to shift the day of the output with respect to the current date. Both positive and negative integer values are allowed (negative decrease the day value). This corresponds to the optional argument of the `\Hijritoday` command which can be used to output Hijri dates irrespective of the `calendar` option (see sec. 10.2).
- **`locale`** = *`default`*, `mashriq`, `libya`, `algeria`, `tunisia`, `morocco`, `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). Recommended settings are `default` for Egypt, Sudan, Yemen and the Gulf states, and `mashriq` for Iraq, Syria, Jordan, Lebanon and Palestine.
- v1.63 v1.63 **`numerals`** = *`mashriq`* (= `eastern` ←) or `maghrib` (= `western` ←)
  Use Eastern Arabic (a.k.a. Indic-Arabic) numerals (`mashriq` or `eastern`) or Western numerals (`maghrib` or `western`). The latter is the default when `locale=algeria`, `tunisia`, or `morocco`, the former is the default with all other Arabic locales.
- v1.63 **`sectionsep`** ← = ⟨`code`⟩ (default value: `.`)
  This option allows to customize the separator between chapters, sections, and subsections (a dot by default), *e.g.*, **`sectionsep=-`**. This might be useful particularly with **`numerals=mashriq`** or **`eastern`**, as the dot looks too similar to the zero in many fonts.
- v1.50 **`abjadalph`** ← = `*true` or *`false`*
  Set this to true if you want the alphabetic counters to be output using `\abjadalph` rather than `\abjad`. Note that this limits the counter scope to 28 (see `\abjadalph` below).
- v1.0.3 **`abjadjimnotail`** ← = `*true` or *`false`*
  Set this to true if you want the *abjad* form of the number three to be ڜ – as in the manuscript tradition – instead of the modern usage ج.

Commands:

`\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.

`\abjadalph` v1.50 ▸ `\abjadalph` ⟵ steps through the Arabic alphabet, thus it can only be used up to 28. Example: `\textarabic{\abjadalph{18}}` yields ص.

`\aemph` v1.2.0 ▸ `\aemph` to emphasize text with `\overline`. ⟵ `\textarabic{\aemph{ا}}` yields $\overline{\text{ا}}$ب. This command is also available for Farsi, Urdu, etc.

6.3 armenian

Options:

- **`variant`** ⟵ = `eastern` or *`western`*
- **`numerals`** ⟵ = `armenian` or *`arabic`* (v1.45)
- **`capitalyiwn`** ⟵ = `*true` or *`false`* (v1.45) (v1.63)
  If this is set to true, `\MakeUppercase` will upcase the ligature of letters ech and yiwn to capital ech and yiwn. This conforms to the output outside Armenian language selection. By default (and with this option set to `false`), `\MakeUppercase` upcases the ech and yiwn ligature to capital ech and *vew*, adhering to Armenian reformed spelling regulations. Note that this feature requires LaTeX kernel 2023/06 at least.

6.4 belarusian ⟵ (v1.46)

Options:

- **`babelshorthands`** = `*true` or *`false`*
  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 `\-`).
  - `"=` adds 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.
  - `"|` disables a ligature at this position.
  - `""` allows for a line break at this position (without hyphenation sign).
  - `",` 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 »).

There are also three shorthands for the Cyrillic dash (тире), which is shorter than the emdash but longer than the endash (namely 0.8 em). Note that, since it is not covered by unicode, this character is faked by telescoping two endashes:

`"---` Cyrillic dash for the use in normal text. This requires preceding space in input (trailing space is optional) and prints with a non-breakable thin space before and after the dash.

`"--~` Cyrillic dash for the use in compound names (surnames). As opposed to `"---` this removes any space before and after the dash.

`"--*` Cyrillic dash for denoting direct speech. This adds a larger space after the dash. Space before the dash is output as is.

- **`numerals`** = *`arabic`*, `cyrillic-alph` or `cyrillic-trad`
  Uses either Arabic numerals or Cyrillic alphanumerical numbering. The two Cyrillic variants differ as follows:
  - `cyrillic-alph` steps through the Cyrillic alphabet. Thus it can only be used up to 30.
  - `cyrillic-trad` (= `cyrillic`) uses a traditional Cyrillic alphanumeric system.[6] It supports numbers up to 999 999.
- **`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`. It steps through the Cyrillic alphabet and thus it can only be used up to 30. The command takes a counter as argument, *e.g.*, `\textbelarusian{\Asbuk{page}}` produces C.

`\asbuk`
- `\asbuk`: same as `\Asbuk` but in lowercase.

`\AsbukTrad`
- `\AsbukTrad`: same as `\Asbuk` but using the traditional Cyrillic alphanumeric numbering which supports numbers up to 999 999.
  E.g., `\textbelarusian{\AsbukTrad{page}}` produces КА.

`\asbukTrad`
- `\asbukTrad`: same as `\AsbukTrad` but 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] See https://en.wikipedia.org/wiki/Cyrillic_numerals.

6.6 catalan

Options:

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

Commands:

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

v1.1.1

6.7 chinese ←

v1.59

Options:

- **variant** = *simplified* (= cn), or traditional (= tw)
- **numerals** = *arabic* or chinese
  Uses either Arabic numerals or Chinese ideographic numbering.

Commands:

\chinesenumeral
- \chinesenumeral (see section 8.3).

6.8 croatian

Options:

v1.47
- **babelshorthands** ← = *true or *false*
  If this is turned on, the following shorthands for fine-tuning hyphenation and micro-typography of Croatian words are activated.

  "| disables a ligature at this position.

  "= 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.

  "- 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.

  Furthermore, the following shorthands generate easy access to Croatian digraphs:

`"dz` Generates the digraph dž if the font provides it. If not, the two consecutive glyphs are output mimicking the digraph. Also available for `"Dz` (Dž) and `"DZ` (DŽ).

`"lj` Generates the digraph lj if the font provides it. If not, the two consecutive glyphs are output mimicking the digraph. Also available for `"Lj` (Lj) and `"LJ` (LJ).

`"nj` Generates the digraph nj if the font provides it. If not, the two consecutive glyphs are output mimicking the digraph. Also available for `"Nj` (Nj) and `"NJ` (NJ).

Finally, there are also four shorthands for quotation marks:

`"`` for Croatian left double quotes („).

`"'` for Croatian right double quotes (”).

`">` for Croatian left guillemets (»).

`"<` for Croatian right guillemets («).

v1.47 ▸ **disabledigraphs**[7] ⟵ = `*true` or *`false`*
If this is `true`, all Croatian digraphs (such as dž) will be replaced by the two consecutive letters, which is the most common way of typesetting them in Croatian. This can be useful if the Unicode digraphs in your font are broken (if the font does not have them, they are automatically mimicked by the two consecutive glyphs).

v1.53 ▸ **localalph** ⟵ = *`true` or `false`
If `true`, alphanumeric counters will use a locally established version which excludes the characters ⟨q⟩, ⟨w⟩, ⟨x⟩ and ⟨y⟩ from alphabetic counting. Obviously this limits the counting range to 22.

v1.51 ▸ **splithyphens** ⟵ = *`true` or `false`
According to Croatian typesetting conventions, if a word with a hard hyphen (such as *hrvatsko-engleski*) is hyphenated at this hyphen, a second hyphenation character is to be inserted at the beginning of the line that follows the hyphenation (*hrvatsko-/-engleski*). 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.

6.9 czech

Options:

v1.45 ▸ **babelshorthands** ⟵ = `*true` or *`false`*
If this is turned on, the following shorthands for Czech are activated:

[7] Up to version 1.53, the option was called `disableligatures`. The old option is kept for backwards compatibility, but the use is discouraged.

`"=` 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 („).

`"’` for Czech right double quotes (“).

`">` for Czech left double guillemets (»).

`"<` for Czech right double guillemets («).

v1.45 ▸ **`splithyphens`** ⟵ `=` *`true` or `false`

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.

v1.46

v1.45 ▸ **`vlna`** ⟵ `=` *`true` or `false`

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.

v1.46

6.10 dutch

Options:

v1.63 ▸ **`tremahyphenation`** ⟵ `=` *`true` or `false`

Following Dutch orthography, words with tremata (such as *geëxecuteerd*) lose the trema on hyphenation before the trematized letter (*ge-executeerd*). This is achieved by polyglossia by making those letters active. If you set this option to false, this mechanism is deactivated (and the hyphenation wrong, but you can still use the babelshorthands mentioned below).

v1.1.1 ▸ **`babelshorthands`** ⟵ `=` *`true` or `false`*

If this is turned on, the following shorthands defined for fine-tuning hyphenation and micro-typography of Dutch words are activated:

v1.63 `"a` ⟵ an ⟨ä⟩ (a with trema) which is hyphenated as ⟨`-a`⟩ as required by the Dutch standards. Also implemented for the other vowels (e, i, o, u), both lower and uppercase.

v1.63 `"y` ⟵ the ij ligature.

v1.63 `"Y` ⟵ the IJ ligature.

`"-` 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 `"-`).

6.11 english

Options:

- **`variant`** = *american* (= us), usmax (same as american but with additional hyphenation patterns), british (= uk), australian, canadian ⟵, or newzealand

v1.45

- **`ordinalmonthday`** = *true or *false*
  The default value is true for variant=british.

6.12 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.

6.13 finnish

Options:

v1.45

- **`babelshorthands`** ⟵ = *true or *false*
  If this is turned on, the following shorthands for fine-tuning hyphenation and micro-typography of Finnish words are activated:

  `"-` adds an explicit hyphen without a breakpoint, allowing for hyphenation at the other points preset in the hyphenation patterns. Useful for cases where the hyphen should stick at the following syllable (*e.g.*, pakastekaapit ja "-arkut).

`"=` for a hyphen sign without a breakpoint that does also remove all other breakpoints in the word (as opposed to `"-`).

`"~` functionally equivalent to `"-` (provided for backwards compatibility reasons).

`"|` 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.

v2.0 ▸ **hyphens** ⟵ = *default*, babel, school
Hyphenation in Finnish is peculiar since there are many rather long words. In order to select a suitable hyphenation pattern, three different options are provided:

- The `default` patterns as used by LaTeX are rather strict and try to prevent hyphenation at points which are valid but considered sub-optimal from an ortho-typographic point of view (*e.g.*, hyphenation between two vowels), with the drawback that simple words are sometimes not being hyphenated at points where this would be valid, and lines might become unevenly spaced.
- The babel support for Finnish by default tweaks some hyphenation-related values (such as `\hyphenpenalty`) for Finnish. If you want to replicate this with polyglossia, select `babel` (this still uses the default patterns).
- The `school` setting, finally, employs alternative (experimental) hyphenation patterns rather than the default Finnish patterns. These (so-called 'School') patterns allow hyphenation at all valid points, including hyphenation points that are considered sub-optimal from an ortho-typographic point of view.[8] See `https://github.com/hyphenation/basic-finnish` for details.

6.14 french

Options:

v1.45 v1.47 ▸ **variant** = *french*, canadian, acadian ⟵ or swiss ⟵
Currently, the only difference between the four variants is that `swiss` uses `thincolonspace=true` by default since this conforms to the Swiss conventions.

▸ **autospacing** = **true* or false

[8] This equals the previous boolean option **schoolhyphens**, which is still supported for backwards compatibility.

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.

v1.46 ▸ **thincolonspace** ⟵ = *true or *false*
With variant=swiss, the default value is true. If false, a full (non-breaking) interword space is inserted before a colon. If true, a thinner space – as before ;, !, and ? – is used. Note that this option must be set after the variant option.

▸ **autospaceguillemets**[9] = **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.

v1.45 ▸ **autospacetypewriter**[10] ⟵ = *true or *false*
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*
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.

v1.46 ▸ **frenchitemlabels** ⟵ = *true or *false*
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.

v1.51 ▸ **frenchpart** ⟵ = **true* or false
By default, polyglossia modifies part headings to match French conventions (*Première partie* rather than *Partie I*). Next to the standard classes, specifics of KOMA-script, memoir and the titlesec package are taken into account. With other classes or packages, redefinition might fail if these have particular part settings. In such case, or if you don't want the redefinition, you can switch off the feature by passing *false* to this option.

v1.46 ▸ **itemlabels** ⟵ = ⟨code⟩ (default value: \textemdash)
If *frenchitemlabels* is true, you can customize here the used item label of all levels.

v1.46 ▸ **itemlabeli** ⟵ = ⟨code⟩ (default value: \textemdash)
If *frenchitemlabels* is true, you can customize here the used item label of

[9] Up to version 1.44, the option was called automaticspacesaroundguillemets. For backwards

the first level.

v1.46 ▸ **itemlabelii** ← = ⟨code⟩ (default value: \textemdash)
If *frenchitemlabels* is true, you can customize here the used item label of the second level.

v1.46 ▸ **itemlabeliii** ← = ⟨code⟩ (default value: \textemdash)
If *frenchitemlabels* is true, you can customize here the used item label of the third level.

v1.46 ▸ **itemlabeliv** ← = ⟨code⟩ (default value: \textemdash)
If *frenchitemlabels* is true, you can customize here the used item label of the fourth level.

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 6.15 gaelic ←

Options:

▸ **variant** = *irish* or scottish

v1.46 6.16 georgian ←

Options:

▸ **babelshorthands** = *true or *false
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 \-).

"= adds 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.

"| disables a ligature at this position.

"" allows for a line break at this position (without hyphenation sign).

compatibility reasons, the more verbose old option is still supported.

[10] Babel's syntax OriginalTypewriter is also supported.

`",` thinspace for initials with a breakpoint in following surname.

`"‘` for German-style left double quotes (looks like „).

`"’` for German-style right double quotes (looks like “).

`"<` for French-style left double quotes (looks like «).

`">` for French-style right double quotes (looks like »).

There are also three shorthands for the Cyrillic dash (тире), which is shorter than the emdash but longer than the endash (namely 0.8 em). Note that, since it is not covered by unicode, this character is faked by telescoping two endashes:

`"---` Cyrillic dash for the use in normal text. This requires preceding space in input (trailing space is optional) and prints with a non-breakable thin space before and after the dash.

`"--~` Cyrillic dash for the use in compound names (surnames). As opposed to `"---` this removes any space before and after the dash.

`"--*` Cyrillic dash for denoting direct speech. This adds a larger space after the dash. Space before the dash is output as is.

- **`numerals`** = *`arabic`* or `georgian`
  Uses either Arabic numerals or Georgian alphanumerical numbering.
- **`oldmonthnames`** = `*true` or *`false`*
  Uses traditional Georgian month names.

6.17 german

Options:

- **`variant`** = *`german`*, austrian, or swiss ← (v1.33.4)
  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.
- **`capitaleszett`** ← = `*true` or *`false`* (v1.63)
  If this is set to true, `\MakeUppercase` will upcase the eszett character ⟨ß⟩ to the capital counterpart ⟨ẞ⟩ that has been introduced to Unicode in 2008 and standardized in German regulations in 2017, rather than ⟨SS⟩. Note that this feature requires LaTeX kernel 2023/06 at least and a font that contains the respective glyph.
- **`script`** ← = *`latin`* or `blackletter` ← (= `fraktur`) (v1.2.0 v1.46)
  Setting `script=blackletter` adapts the captions for typesetting German in blackletter type, using the ‘long s’ ⟨ſ⟩ where appropriate.

v1.0.3 ▸ **babelshorthands** ⟵ = *true or *false*

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.

v2.0 - `"*` ⟵ An asterisk which assures the word can still be hyphenated at its defined breakpoints. Useful if you want to employ gender-sensitive writing (‚gender star'). Similar shorthands are available for the alternative gender-sensitive spellings, `":` and `"_`.

v2.0 - `"x` ⟵ Inserts a gender mark which assures the word can still be hyphenated at its defined breakpoints. This is predefined to * but can be globally redefined by redefining the macro `\mkgender` (see below).

There are also four shorthands for quotation signs:

- `"`` for German-style left double quotes („)
- `"'` for German-style right double quotes (“)
- `"<` for French-style left double quotes («)
- `">` for French-style right double quotes (»).

Commands:

`\mkgender` v2.0 ▸ `\mkgender` ⟵ Defines which gender mark is output by the babelshorthand `"x`. Predefined to `\def\mkgender{*}`.

6.18 greek

Options:

- **variant** = *monotonic* (= mono), polytonic (= poly), or ancient
- **numerals** = *greek* or arabic
- **capitaliota** ← = *true or *false*
  v1.63
  If this is set to true, \MakeUppercase will upcase the *ypogegrammeni* (subscript muted iota) to capital iota. By default (and with this option set to false), \MakeUppercase retains the subscript versions. Note that this feature requires LaTeX kernel 2023/06 at least.
- **koppa** ← = *modern* or archaic
  v2.6
  While modern Greek uses the letter koppa in its modern form (ϟ) to represent the number 90, classicists often prefer the archaic koppa variant (ϙ). Polyglossia uses by default the modern form in all variants (with **numerals=greek**). If you need the archaic form, use **koppa=archaic**.

Commands:

- \Greeknumeral (= \Greeknumber) and \greeknumeral (= \greeknumber) (see section 8.3).
  \Greeknumber
  \greeknumeral
  \greeknumber
- The command \atticnumeral (= \atticnum) displays numbers using the acrophonic numbering system (defined in the Unicode range U+10140–U+10174). Please refer to the documentation of the xgreek package for more details on Attic numbering.
  \atticnumeral
  \atticnum
- ← The macros \greeknumeralsix, \greeknumeralSix, \greeknumeralninety and \greeknumeralNinety determine how the numbers 6 and 90 are represented in lower and uppercase output. By default, this is ϛ (lowercase 6), Ϛ (uppercase 6), ϟ (lowercase ninety), and Ϟ (uppercase ninety) – or, with **koppa=arachaic**, ϙ (lowercase ninety), and Ϙ (uppercase ninety). You can redefine the macros if you need other representations.
  v2.6
  \greeknumeralsix
  \greeknumeralSix
  \greeknumeralninety
  \greeknumeralNinety

6.19 hebrew

Options:

- **numerals** = hebrew or *arabic*
- **calendar** = hebrew or *gregorian*
- **marcheshvan** = *true or *false*
  If true, the second month of the civil year will be output as מרחשון (Marcheshvan) rather than חשון (Heshvan), which is the default.
- **fullyear** ← = *true or *false*
  v1.57
  causes years from the current millennium to be printed with the thousands digit (he-tav-shin-samekh-gimel). Without this option, thousands are not printed for the current millennium.

v1.57
- **transliteration** ← = academy or *alt*
  With value academy, transliteration follows the recommendations of the Hebrew Language Academy. The default (alt) uses the received settings of babel (hebcal) and polyglossia (hebrewcal).

Commands:

\hebrewnumeral
- \hebrewnumeral (= \hebrewalph) (see section 8.3).

\hebrewalph
- \aemph (see section 6.2).

\aemph

6.20 hindi ←

v1.2.0

Options:

- **numerals** = western or *devanagari*

6.21 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

v1.58
- **forceheadingpunctuation** ← = **true* or false
  Section numbers always have a trailing punctuation in Hungarian (as in *1.1.* as opposed to *1.1*). For compatibility reasons, the default option is false, thus polyglossia does not touch heading punctuation, so this will be whatever the class or a package determines. Set this option to true, and polyglossia appends a period after the section counters, and adjusts the header punctuation (as in *1. fejezet.* as opposed to *1. fejezet*).

Commands:

\ontoday
\ondatehungarian

- `\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.

6.22 italian

Options:

v1.2.0cc

- **babelshorthands** ← = *`true` or *`false`*
  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 `\-`).

6.23 korean ←

v1.40.0

Options:

- **variant** = *`plain`*, `classic`, or `modern`
  These variants control spacing before/after CJK punctuations.
  - `plain`: Do nothing
  - `classic`: Suitable for text with no interword spaces. This option forces CJK punctuations to half-width, and inserts half-width glue around them.
  - `modern`: Suitable for text with interword spaces. This option forces CJK punctuations to half-width, and inserts small (half of interword space) glue around them.
- **captions** = *`hangul`* or `hanja`
- **swapstrings** ← = *`all`*, `headers`, `headings`, or `none`

  v1.47

  With this option, Korean-style part and chapter headings, and running headers are available. It is similar to Hungarian (see 6.21) except that figure and table captions are not touched.
  - `all`: Redefine part and chapter headings, and running headers (= default setting)
  - `headings`: Redefine part and chapter headings only

- `headers`: Redefine running headers only
- `none`: Do not redefine anything

v1.45

6.24 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.

v1.63

- **`sectionsep`** ← = ⟨`code`⟩ (default value: `.`)
  This option allows to customize the separator between chapters, sections, and subsections (a dot by default), *e.g.*, **`sectionsep=-`**. This might be useful particularly with **`numerals=eastern`**, as the dot looks too similar to the zero in many fonts.
- **`abjadjimnotail`** = `*true` or *`false`*
  Set this to true if you want the *abjad* form of the number three to be ﺝ – as in the manuscript tradition – instead of the modern usage ﺟ.

Commands:

`\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)

v1.2.0

6.25 lao ←

Options:

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

6.26 latin

Options:

v1.46

- **`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 4 for examples.

Table 4. 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

Table 5. Latin default hyphenation styles

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

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.

v1.46 ▸ **`hyphenation`** ⟵ = `classic`, `modern`, or `liturgical`
There are three different sets of hyphenation patterns for Latin. Separate documentation for them is available on the Internet.[11] Each of the four variants mentioned above has its default set of hyphenation patterns as indicated by table 5. 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.

v1.46 ▸ **`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.

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

v1.46 ▸ **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*.

v1.46 ▸ **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.

▸ **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, æ, and œ
'A	for Á (A with acute), also available for É, Í, Ó, Ú, Ý, Æ, 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 œ́
'Ae	for Ǽ (AE ligature with acute), also available for Œ́
'AE	for Ǽ (AE ligature with acute), also available for Œ́

v1.46 ▸ **prosodicshorthands** ⟵ = *true or *false*
Enable shorthands for prosodic marks (macrons and breves) very similar 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, such as the command \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, for the `classic` and the `modern` variant) or ǣ (ae ligature with macron, for the `medieval` and the `ecclesiastic` variant), respectively; also available for a͞u, e͞u, and o͞e/œ̄. Note that macrons above diphthongs are only displayed if your font supports the Unicode character `035E` (*combining double macron*).

`=Ae` for A͞e (Ae diphthong with macron, for the `classic` and the `modern` variant) or Ǣ (AE ligature with macron, for the `medieval` and the `ecclesiastic` variant), respectively; also available for A͞u, E͞u, and O͞e/Œ.

`=AE` for A͞E (AE diphthong with macron, for the `classic` and the `modern` variant) or Ǣ (AE ligature with macron, for the `medieval` and the `ecclesiastic` variant), respectively; also available for A͞U, E͞U, and O͞E/Œ.

`^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 supports the Unicode character `0306` (*combining breve*).

6.27 malay

Options:

- **`variant`** ← = `indonesian` or *`malaysian`*

v1.45

6.28 marathi

Options:

- **`numerals`** = *`devanagari`* or `western`

6.29 mongolian ←

v1.45

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

Options:

- **babelshorthands** = *true or *false*
  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 \-).
  - "= adds 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.
  - "| disables a ligature at this position.
  - "" allows for a line break at this position (without hyphenation sign).
  - ", thinspace for initials with a breakpoint in following surname.
  - "‘ for German-style left double quotes (looks like „).
  - "’ for German-style right double quotes (looks like “).
  - "< for French-style left double quotes (looks like «).
  - "> for French-style right double quotes (looks like »).

  There are also three shorthands for the Cyrillic dash (тире), which is shorter than the emdash but longer than the endash (namely 0.8 em). Note that, since it is not covered by unicode, this character is faked by telescoping two endashes:
  - "--- Cyrillic dash for the use in normal text. This requires preceding space in input (trailing space is optional) and prints with a non-breakable thin space before and after the dash.
  - "--~ Cyrillic dash for the use in compound names (surnames). As opposed to "--- this removes any space before and after the dash.
  - "--* Cyrillic dash for denoting direct speech. This adds a larger space after the dash. Space before the dash is output as is.
- **numerals** = *arabic*, cyrillic-alph or cyrillic-trad
  Uses either Arabic numerals or Cyrillic alphanumerical numbering. The two Cyrillic variants differ as follows:
  - cyrillic-alph steps through the Cyrillic alphabet. Thus it can only be used up to 30.
  - cyrillic-trad (= cyrillic) uses a traditional Cyrillic alphanumeric system.[12] It supports numbers up to 999 999.

Commands:

\Asbuk
- \Asbuk: produces uppercased Cyrillic alphanumerals, for environments

[12] See https://en.wikipedia.org/wiki/Cyrillic_numerals.

such as `enumerate`. It steps through the Cyrillic alphabet and thus it can only be used up to 30. The command takes a counter as argument, *e.g.*, `\textmongolian{\Asbuk{section}}` produces Ҭ.

`\asbuk` ▸ `\asbuk`: same as `\Asbuk` but in lowercase.

`\AsbukTrad` ▸ `\AsbukTrad`: same as `\Asbuk` but using the traditional Cyrillic alphanumeric numbering which supports numbers up to 999 999.
E.g., `\textmongolian{\AsbukTrad{section}}` produces S.

`\asbukTrad` ▸ `\asbukTrad`: same as `\AsbukTrad` but in lowercase.

6.30 norwegian

Options:

v1.45 ▸ **`variant`** ← = `bokmal` or *`nynorsk`*

6.31 odia ←

v1.66

Options:

▸ **`numerals`** = `western`, *`devanagari`*, or `odia`

▸ **`changecounternumbering`** = `*true` or *`false`*
Use specified numerals for headings and page numbers.

6.32 persian

Options:

▸ **`numerals`** = `western` or *`eastern`*

▸ **`sectionsep`** ← = ⟨`code`⟩ (default value: `.`)
v1.63 This option allows to customize the separator between chapters, sections, and subsections (a dot by default), *e.g.*, **`sectionsep=-`**. This might be useful particularly with **`numerals=eastern`**, as the dot looks too similar to the zero in many fonts.

▸ **`abjadjimnotail`** ← = `*true` or *`false`*
v1.0.3 Set this to `true` if you want the *abjad* form of the number three to be ﺝ – as in the manuscript tradition – instead of the modern usage ﺟ.

Commands:

`\abjad` ▸ `\abjad` (see section 8.3)

`\aemph` ▸ `\aemph` (see section 6.2).

6.33 polish

Options:

▸ **`babelshorthands`** ← = `*true` or *`false`*
v1.55 If this is turned on, the following shorthands for Polish are activated:

`"=` for an explicit hyphen sign which is repeated at the beginning of the next line when hyphenated, as common in Polish typesetting (also if `splithyphens=false`). Unlike plain `-`, this also allows for hyphenation at the other points preset in the hyphenation patterns.

`"|` 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 Polish left double quotes (looks like „).

`"'` for Polish right double quotes (looks like ”).

`"<` for French left double guillemets (looks like « – used in Polish as second level quotes).

`">` for French right double guillemets (looks like »).

v1.55 ▸ **splithyphens** ← = **true* or `false`
According to Polish typesetting conventions, if a word with a hard hyphen (such as *czerwono-złote*) is hyphenated at this hyphen, a second hyphenation character is to be inserted at the beginning of the line that follows the hyphenation (*czerwono-/-złote*). 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.

v1.55 ▸ **vlna** ← = **true* or `false`
According to Polish 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.

6.34 portuguese

Options:

v1.45 ▸ **variant** ← = `brazilian` or *portuguese*

v1.54 ▸ **babelshorthands** ← = `*true` or *false*
If this is turned on, the following shorthands for fine-tuning hyphenation and micro-typography of Portuguese words are activated.

`"|` disables a ligature at this position.

`"=` for an explicit hyphen sign which is repeated at the beginning of the next line when hyphenated, as common in Portuguese typesetting (also if `splithyphens=false`). Unlike plain `-`, this also allows for hyphenation at the other points preset in the hyphenation patterns.

`"~` 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 French left guillemets («).

`">` for French right guillemets (»).

v1.54 ▸ **splithyphens** ← = **true* or `false`
According to Portuguese typesetting conventions, if a word with a hard hyphen (such as *para-brisas*) is hyphenated at this hyphen, a second hyphenation character is to be inserted at the beginning of the line that follows the hyphenation (*para-/-brisas*). 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.

6.35 punjabi ←

v1.59

Options:

▸ **numerals** = `western` or *gurmukhi*

6.36 russian

Options:

▸ **babelshorthands** = *`true` or *false*
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 `\-`).

`"=` adds an explicit hyphen with a breakpoint, allowing for hyphenation at the other points preset in the hyphenation patterns (as opposed to plain `-`).

`"~` adds an explicit hyphen 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).

There are also three shorthands for the Cyrillic dash (тире), which is shorter than the emdash but longer than the endash (namely 0.8 em). Note that, since it is not covered by unicode, this character is faked by telescoping two endashes:

`"---` Cyrillic dash for the use in normal text. This requires preceding space in input (trailing space is optional) and prints with a non-breakable thin space before and after the dash.

`"--~` Cyrillic dash for the use in compound names (surnames). As opposed to `"---` this removes any space before and after the dash.

`"--*` Cyrillic dash for denoting direct speech. This adds a larger space after the dash. Space before the dash is output as is.

v1.50 ▸ **`forceheadingpunctuation`** ⟵ = **true* or `false`
By default, chapter and section numbers always have a trailing punctuation in Russian (as in *1.1.* as opposed to *1.1*). If this option is set to `false`, polyglossia will not touch heading punctuation, so this will be whatever the class or a package determines.

v1.46 ▸ **`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.

v1.52 ▸ **`mathfunctions`** ⟵ = **true* or `false`
By default, some specific math macros are defined for Russian (see below). In order to prevent command clashes (*e.g.*, with the chemformula package), you can switch these definitions off by passing `false` to this option.

▸ **`spelling`** = *modern* or `old`
This option is for captions and date only, not for hyphenation.

▸ **`numerals`** = *arabic*, `cyrillic-alph` or `cyrillic-trad`
Uses either Arabic numerals or Cyrillic alphanumerical numbering. The two Cyrillic variants differ as follows:

- `cyrillic-alph` steps through the Cyrillic alphabet. Thus it can only be used up to 30.
- `cyrillic-trad` (= `cyrillic`) uses a traditional Cyrillic alphanumeric system.[13] It supports numbers up to 999 999.

Commands:

`\Asbuk` ▸ `\Asbuk`: produces uppercased Cyrillic alphanumerals, for environments such as `enumerate`. It steps through the Cyrillic alphabet and thus it can

[13] See https://en.wikipedia.org/wiki/Cyrillic_numerals.

only be used up to 30. The command takes a counter as argument, *e.g.*, `\textrussian{\Asbuk{section}}` produces Ѣ.

`\asbuk` ▸ `\asbuk`: same as `\Asbuk` but in lowercase.

`\AsbukTrad` ▸ `\AsbukTrad`: same as `\Asbuk` but using the traditional Cyrillic alphanumeric numbering which supports numbers up to 999 999.
E.g., `\textrussian{\AsbukTrad{page}}` produces МГ.

`\asbukTrad` ▸ `\asbukTrad`: same as `\AsbukTrad` but in lowercase.

If the **mathfunctions** option is `true`, loading Russian defines a few macros than can be used independently of the current language. These are nine macros to be used in math mode to type the names of trigonometric functions common for Russian documents: `\sh`, `\ch`, `\tg`, `\ctg`, `\arctg`, `\arcctg`, `\th`, `\cth`, and `\cosec`. Cyrillic letters in math mode can be typed with the aid of text commands such as `\textbf`, `\textsf`, `\textit`, `\texttt`, etc. The macros `\Prob`, `\Variance`, `\NOD`, `\nod`, `\NOK`, `\nok`, `\Proj` print some rare Russian mathematical symbols.

6.37 sami ←

v1.45

Currently support for Sami is limited to Northern Sami.

6.38 sanskrit

Options:

v1.0.2 ▸ **script** ← = *devanagari*, `gujarati`, `malayalam`, `bengali`, `kannada`, `telugu`, `latin`
The value is passed to fontspec should the respective `\(script)font` not be defined. This can be useful if you typeset Sanskrit texts in scripts other than Devanagari.

v1.45 ▸ **numerals** ← = *devanagari* or `western`

6.39 serbian

Options:

v1.63 ▸ **variant** ← = *ekavian* or `ijekavian`
With `ijekavian`, the different spelling is reflected in some caption strings.

▸ **script** = *cyrillic* or `latin`

v1.63 ▸ **datei** ← = `*true` or *false*
If this is `true`, the sixth and seventh month of the year are written in the date with the trailing `⟨i⟩` or `⟨и⟩` (*i.e., juni*, *juli* or *juни*, *juли* rather than *jun*, *jul* or *juн*, *juл*).

v1.54 ▸ **splithyphens** ← = `*`*true* or `false`
According to Serbian typesetting conventions, if a word with a hard hyphen (such as *калцијум-карбонат*) is hyphenated at this hyphen, a second hyphenation character is to be inserted at the beginning of the line that

follows the hyphenation (*калцијум-/-карбонат*). 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.

- **`numerals`** = *`arabic`*, `cyrillic-alph` or `cyrillic-trad`
  Uses either Arabic numerals or Cyrillic alphanumerical numbering. The two Cyrillic variants differ as follows:
  - `cyrillic-alph` steps through the Cyrillic alphabet. Thus it can only be used up to 30.
  - `cyrillic-trad` (= `cyrillic`) uses a traditional Cyrillic alphanumeric system.[14] It supports numbers up to 999 999.

Commands:

`\Asbuk`
- `\Asbuk`: produces uppercased Cyrillic alphanumerals, for environments such as `enumerate`. It steps through the Cyrillic alphabet and thus it can only be used up to 30. The command takes a counter as argument, *e.g.*, `\textserbian{\Asbuk{section}}` produces Ђ.

`\asbuk`
- `\asbuk`: same as `\Asbuk` but in lowercase.

`\AsbukTrad`
- `\AsbukTrad`: same as `\Asbuk` but using the traditional Cyrillic alphanumeric numbering which supports numbers up to 999 999.
  E.g., `\textserbian{\AsbukTrad{page}}` produces МД.

`\asbukTrad`
- `\asbukTrad`: same as `\AsbukTrad` but in lowercase.

`\today*`
- `\today*`: as opposed to the unstarred counterpart, this does not end the date with a punctuation.

`\todayGen`
- `\todayGen`: outputs the date with months in genitive inflection. This form is suggested when the date is integrated into a sentence (as opposed to a

`\todayGen*`
  standalone date statement). The starred form `\todayGen*` omits the trailing punctuation.

`\todayArabic`
- `\todayArabic`: outputs the date with months in Arabic number. The

`\todayArabic*`
  starred form `\todayArabic*` omits the trailing punctuation.

`\todayRoman`
- `\todayRoman`: outputs the date with months in Roman number. The starred

`\todayRoman*`
  form `\todayRoman*` omits the trailing punctuation.

6.40 slovak

Options:

v1.46
- **`babelshorthands`** ← = `*true` or *`false`*
  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 (also

[14] See https://en.wikipedia.org/wiki/Cyrillic_numerals.

if `splithyphens=false`). Unlike plain `-`, this also allows for hyphenation at the other points preset in the hyphenation patterns.

`"|` 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 «).

v1.46 ▸ **splithyphens** ⟵ = **true* or `false`
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.

v1.46 ▸ **vlna** ⟵ = **true* or `false`
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.

6.41 slovenian

Options:

▸ **localalph** = *`true` or *`false`*
If `true`, alphanumeric counters will use a localized version including characters with caron (*a, b, c, č, d, …*) and excluding the characters `⟨q⟩`, `⟨w⟩`, `⟨x⟩` and `⟨y⟩` from alphabetic counting.

6.42 sorbian

Options:

- **variant** ← = `lower` or *upper* (v1.45)
- **olddate** ← = `*true` or *false* (v1.45)
  If `true`, `\today` will use traditional Sorbian month names (*i.e.,* it will be synonymous to `\oldtoday` below).

Commands:

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

6.43 spanish

Options:

- **variant** ← = *spanish* or `mexican` (v1.46)
- **spanishoperators** ← = `*all`, `accented`, `spaced`, `none`, or *false* (v1.46)
  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: ← (v1.46)

- `\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}`.

6.44 syriac

Options:

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

v1.63

- **sectionsep** ⟵ = ⟨code⟩ (default value: .)
  This option allows to customize the separator between chapters, sections, and subsections (a dot by default), *e.g.*, **sectionsep=-**. This might be useful particularly with **numerals=eastern**, as the dot looks too similar to the zero in many fonts.

Commands:

\abjadsyriac
- \abjadsyriac (see section 8.3)

\aemph
- \aemph (see section 6.2).

6.45 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.

6.46 tibetan

Options:

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

6.47 ukrainian

Options:

- **babelshorthands** = *true or *false*
  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 \-).

  "= adds 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.

  "| disables a ligature at this position.

  "" allows for a line break at this position (without hyphenation sign).

  There are also three shorthands for the Cyrillic dash (тире), which is shorter than the emdash but longer than the endash (namely 0.8 em). Note that, since it is not covered by unicode, this character is faked by telescoping two endashes:

`"---` Cyrillic dash for the use in normal text. This requires preceding space in input (trailing space is optional) and prints with a non-breakable thin space before and after the dash.

`"--~` Cyrillic dash for the use in compound names (surnames). As opposed to `"---` this removes any space before and after the dash.

`"--*` Cyrillic dash for denoting direct speech. This adds a larger space after the dash. Space before the dash is output as is.

v1.52

- **mathfunctions** ⟵ = **true* or false
  By default, some specific math macros are defined for Ukrainian (see below). In order to prevent command clashes (*e.g.,* with the chemformula package), you can switch these definitions off by passing `false` to this option.
- **numerals** = *arabic*, cyrillic-alph or cyrillic-trad
  Uses either Arabic numerals or Cyrillic alphanumerical numbering. The two Cyrillic variants differ as follows:
  - `cyrillic-alph` steps through the Cyrillic alphabet. Thus it can only be used up to 30.
  - `cyrillic-trad` (= `cyrillic`) uses a traditional Cyrillic alphanumeric system.[15] It supports numbers up to 999 999.

Commands:

\Asbuk
- `\Asbuk`: produces uppercased Cyrillic alphanumerals, for environments such as `enumerate`. It steps through the Cyrillic alphabet and thus it can only be used up to 30. The command takes a counter as argument, *e.g.,* `\textukrainian{\Asbuk{section}}` produces Ъ.

\asbuk
- `\asbuk`: same as `\Asbuk` but in lowercase.

\AsbukTrad
- `\AsbukTrad`: same as `\Asbuk` but using the traditional Cyrillic alphanumeric numbering which supports numbers up to 999 999.
  E.g., `\textukrainian{\AsbukTrad{page}}` produces МИ.

\asbukTrad
- `\asbukTrad`: same as `\AsbukTrad` but in lowercase.

If the **mathfunctions** option is `true`, loading Ukrainian defines a few macros than can be used independently of the current language. These are nine macros to be used in math mode to type the names of trigonometric functions common for Ukrainian documents: `\sh`, `\ch`, `\tg`, `\ctg`, `\arctg`, `\arcctg`, `\th`, `\cth`, and `\cosec`. Cyrillic letters in math mode can be typed with the aid of text commands such as `\textbf`, `\textsf`, `\textit`, `\texttt`, etc. The macros `\Prob`, `\Variance`, `\NOD`, `\nod`, `\NOK`, `\nok`, `\NSD`, `\nsd`, `\NSK`, `\nsk`, `\Proj` print some rare Ukrainian mathematical symbols.

[15] See https://en.wikipedia.org/wiki/Cyrillic_numerals.

6.48 urdu

Options:

- **calendar** = *gregorian* or hijri
- **hijricorrection** = ⟨integer⟩ (default value: 0)
  If calendar=hijri is selected, \today will output the date according to the lunar Islamic (Hijra) calendar. This option allows to shift the day of the output with respect to the current date. Both positive and negative integer values are allowed (negative decrease the day value). This corresponds to the optional argument of the \Hijritoday command which can be used to output Hijri dates irrespective of the calendar option (see sec. 10.2).
- **numerals** = western or *eastern*
- v1.63 **sectionsep** ← = ⟨code⟩ (default value: .)
  This option allows to customize the separator between chapters, sections, and subsections (a dot by default), *e.g.*, **sectionsep=-**. This might be useful particularly with **numerals=eastern**, as the dot looks too similar to the zero in many fonts.
- **abjadjimnotail** = *true or *false*
  Set this to true if you want the *abjad* form of the number three to be ـجـ – as in the manuscript tradition – instead of the modern usage ج.

6.49 uyghur

Options:

- v1.53 **chapterformat** ← = **ordinal*, roman, arabic
  In Uyghur, chapter and part headings usually use ordinals rather than numbers. Alternatively Roman numbers are common as well. By default, polyglossia uses ordinals. This option allows to change this. Choices are:
  - ordinal: Use ordinals for chapter and part headings (= default setting)
  - roman: Use Roman numbering for chapter and part heading.
  - arabic: Use Arabic numbering for chapter and part heading (= LaTeX default).

  Note that this option only works if **swapstrings** is set to all, headings or hheaders. The caveats for that options (see below) apply.
- v1.53 **swapstrings** ← = **all*, captions, headings, headers, hheaders, or none
  In Uyghur, table and figure caption strings need to be in a different order and format than in other languages. By default, polyglossia provides the correct 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 spe-

cific 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:

`\uyghurordinal`
- `\uyghurordinal`: outputs Uyghur ordinals. Supported input: integer numbers in the range of 1–100.

`\uyghurord`
- `\uyghurord`: same as `\uyghurordinal` but with counter as input (*e.g.*, `\textuyghur{\uyghurord{page}}`).

6.50 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

`\nestedextras@⟨lang⟩` v2.3
- `\nestedextras@⟨lang⟩` ← stores macros that are to be executed when a

nested language is closed and ⟨lang⟩ is the active language again. This is useful for settings that are only possible on global scale.

In order to redefine internal macros, we recommend to use the command \addto. For instance, to change the \chaptername for language lingua, you can do this:

```
\addto\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.

Finally, as soon as the language has been switched (either inline or as a block), polyglossia executes the (by default empty) hook

\polyglossia@language@switched
- \polyglossia@language@switched

to which you can append arbitrary code (via \gappto) that should be executed if (a particular) language is being activated. This is done before any of the above macros are issued (so you can still alter them), but at a point where \languagename, \babelname and \languageid are already set, so you can condition on specific languages in your code. This hook is particularly provided for package authors.

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

v1.45 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`

v1.45 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 ٥٢.

`\localnumeral*`

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

`\Localnumeral*`

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

v1.1.1 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.[16] The macros have the form `\⟨script⟩digits`. They convert

[16] 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.

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`

`\gurmukhidigits` ▸ `\gurmukhidigits`

`\kannadadigits` ▸ `\kannadadigits`

`\khmerdigits` ▸ `\khmerdigits`

`\laodigits` ▸ `\laodigits`

`\nkodigits` ▸ `\nkodigits`

`\odiadigits` ▸ `\odiadigits`

`\thaidigits` ▸ `\thaidigits`

`\tibetandigits` ▸ `\tibetandigits`

8.3 Non-Latin alphabetic numbering

For languages which use special (non-Latin) alphanumerical notation[17], 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.[18] Example: `\abjadsyriac{463}` yields ܣܟܓ.

`\armeniannumeral` ▸ `\armeniannumeral` produces Armenian alphabetic numbering. Example: `\armeniannumeral{1863}` yields ՌՊԿԳ.

`\belarusiannumeral` `\Belarusiannumeral` ▸ `\belarusiannumeral` produces Belarusian numbering, with uppercased variant (for alphanumerical variant) via `\Belarusiannumeral`. Depending on the `numerals` option in the Belarusian language selection, this is either Arabic digit or Cyrillic alphanumercial output.

[17] 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.

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

Example: With `numerals=latin` `\belarusiannumeral{19}` yields 19, with `numerals=cyrillic-trad` `\belarusiannumeral{19}` results in іѳ, with `numerals=cyrillic-alph` `\belarusiannumeral{19}` results in y.

`\chinesenumeral` ▸ `\chinesenumeral` produces Chinese numbering which, depending on the `numerals` option in the Chinese language selection, produces is either Arabic digit or Chinese ideographic output.
Example: With `numerals=arabic` `\chinesenumeral{753}` yields 753, with `numerals=chinese` `\chinesenumeral{753}` results in 七百五十三.

`\georgiannumeral` ▸ `\georgiannumeral` produces Georgian alphabetic numbering.
Example: `\georgiannumeral{1863}` yields ჩყჟგ.

`\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` produce variants of Hebrew alphanumeric numerals. The commands behave identical as in babel: `\hebrewnumeral` outputs the numbers without any decoration at all, `\Hebrewnumeral` adds *gereshayim* before the last letter, and `\Hebrewnumeralfinal` uses in addition the final forms of Hebrew letters. Examples: `\hebrewnumeral{1750}` yields אתשנ, `\Hebrewnumeral{1750}` yields אתש״נ, and `\Hebrewnumeralfinal{1750}` yields אתש״ן.

`\mongoliannumeral` `\Mongoliannumeral` ▸ `\mongoliannumeral` produces Mongolian numbering, with uppercased variant (for alphanumerical variant) via `\Mongoliannumeral`. Depending on the `numerals` option in the Mongolian language selection, this is either Arabic digit or Cyrillic alphanumercial output.
Example: With `numerals=latin` `\mongoliannumeral{19}` yields 19, with `numerals=cyrillic-trad` `\mongoliannumeral{19}` results in іѳ, with `numerals=cyrillic-alph` `\mongoliannumeral{19}` results in y.

`\punjabinumeral` ▸ `\punjabinumeral` produces Punjabi numbering, depending on the setting of the `numerals` option in the Punjabi language selection, this is either Gurmukhi or Arabic (Western).

`\russiannumeral` `\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.
Example: With `numerals=latin` `\russiannumeral{19}` yields 19, with `numerals=cyrillic-trad` `\russiannumeral{19}` results in іѳ, with `numerals=cyrillic-alph` `\russiannumeral{19}` results in y.

`\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-trad` `\serbiannumeral{19}` results in іѳ, with `numerals=cyrillic-alph` `\serbiannumeral{19}` results in ıı.

`\ukrainiannumeral` `\Ukrainiannumeral`

- `\ukrainiannumeral` produces Ukrainian numbering, with uppercased variant (for alphanumerical variant) via `\Ukrainiannumeral`. Depending on the `numerals` option in the Ukrainian language selection, this is either Arabic digit or Cyrillic alphanumercial output.
  Example: With `numerals=latin` `\ukrainiannumeral{19}` yields 19, with `numerals=cyrillic-trad` `\ukrainiannumeral{19}` results in іѳ, with `numerals=cyrillic-alph` `\ukrainiannumeral{19}` results in у.

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 XeTeX and LuaTeX output (this is due to differences in the bidi and luabidi packages). With XeTeX, 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

\Hebrewtoday 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.[19] It provides two macros for the end-user. The command

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

\Hijritoday sets the counters `Hijriday`, `Hijrimonth` and `Hijriyear`. `\Hijritoday` formats the Hijri date for the current day. This command is now locale-aware ←: its output will differ depending on the currently active language. Presently polyglossia's language definition files for Arabic, Farsi, Urdu, Turkish 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 al-

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

gorithm.[20] 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.[21] Here we have renamed the command `\ftoday` to `\Jalalitoday`. Example: today is 9 Tīr 1404.

`\Jalalitoday`

11 Auxiliary commands

v1.47 `\charifavailable`

The macro ←

```
\charifavailable{⟨char code⟩}{⟨substitution⟩}
```

checks whether the character with the specified `⟨char code⟩` (*i.e.,* unicode utf-16 code without preceding `0x`) exists in the current font. If so, the character is printed, if not, the `⟨substitution⟩` is printed.

Example: `\charifavailable{1E9E}{SS}` prints the capital version of the German letter ⟨ß⟩ if available (*i.e.,* ẞ), else it prints the substitution digraph SS.

v1.66 `\IfCharIsAvailableTF`

With the test ←

```
\IfCharIsAvailableTF{⟨char code⟩}{⟨true condition⟩}{⟨false condition⟩}
```

you can test for the availability of a character and execute different code depending whether or not this is the case (this replaces the undocument internal command `\xpg@if@char@available`).

12 Accessing language information

The following is specifically relevant to package authors who need information about the languages in use and the details of the respective locale. In order to get such information, the recommended way is to use the `\BCPdata{⟨type⟩}` ← command to get (parts of) the BCP-47 language tag. This command is used by the LaTeX kernel as of 6/2023, and the syntax is standardized at least between babel and polyglossia.

`\BCPdata{⟨type⟩}` v1.61

`\BCPdata` supports the following `{⟨type⟩}` arguments:

- **`language`** The language subtag (*e.g.,* `de`)
- **`region`** The region subtag (*e.g.,* `AT`)

[20] 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.

[21] One day we may rewrite farsical from scratch using the algorithm in Reingold & Gershowitz (ref. n. 19).

- **`script`** The script subtag (*e.g.,* `Latn`)
- **`variant`** The variant subtag (*e.g.,* `1901` for German old spelling)
- **`extension.t`** The subtag indicating transformation between languages or scripts (*e.g.,* `ja` as in `en-t-ja`: English transformed to Japanese)
- **`extension.u`** The subtag indicating additional locale information (*e.g.,* `nu-latn` as in `ar-u-nu-latn`: Arabic with Latin Numbering)
- **`extension.x`** The private usespace subtag (*e.g.,* `classic` as in `la-x-classic` for classic Latin)
- **`casing`** is a special semantic value which returns whatever is considered suitable for casing commands such as `\MakeUppercase`. In many cases, this is identical to `language` but it could also be something more specific such as `el-x-iota` or `ckb-Latn` if this is needed for proper casing. Note that this information is retrieved by the LaTeX kernel for casing as of 06/2023.
- **`tag`** returns the registered full tag (identical to `\languageid` below). This however omits certain subtags if these are considered obvious (*e.g.,* for German, only `de-DE` is returned, although script [`Latn`] and variant [`1996`] are defined as well).

If a subtag is not defined for a given language, an empty string is returned.

The values above return information for the language currently in use. If you need to access information for the main language of the document, prepend **`main.`** to the respective argument (*e.g.,* **`main.language`**). This works for all supported arguments.

Next to `\BCPdata`, polyglossia provides the following legacy macros which are partly unique to polyglossia, partly re-implementations of legacy babel commands:

`\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.

`\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.

`\languageid{(type)}` v1.47
- `\languageid{(type)}` ← stores the identifier tag of the current language.

Currently supported ⟨types⟩:
- bcp-47 (alias bcp47): IETF BCP-47 language identifier

\mainlanguageid{⟨type⟩} v1.47
- \mainlanguageid{⟨type⟩} ← stores identifier tag of the main language. Currently supported ⟨types⟩: see \languageid.

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

\xpg@bcp@loaded v1.47
- \xpg@bcp@loaded ← stores a comma-separated list of the BCP-47 IDs of all language variants

Whether a language is loaded can be tested by

\iflanguageloaded
```
\iflanguageloaded{⟨lang⟩}{⟨true⟩}{⟨false⟩}
```
where ⟨lang⟩ is a polyglossia language name, by

\ifbabellanguageloaded
```
\ifbabellanguageloaded{⟨lang⟩}{⟨true⟩}{⟨false⟩}
```
where ⟨lang⟩ is a babel language name (see table 2 on p. 7), or by ← v1.47

\iflanguageidloaded
```
\iflanguageidloaded{⟨type⟩}{⟨id⟩}{⟨true⟩}{⟨false⟩}
```
where ⟨type⟩ is a supported language id type (such as bcp-47) and ⟨id⟩ is a language id (such as en-US; see table 3 on p. 8).

Finally, if you want to know whether a specific language option has been set, you can use ← v1.47

\iflanguageoption
```
\iflanguageoption{⟨lang⟩}{⟨key⟩}{⟨value⟩}{⟨true⟩}{⟨false⟩}
```

13 Revision history

2.6 (2025/06/30)

Bug fixes

- Reset splithyphens and vlna state of embedding language after embedded language has been closed (#682).
- Fix space after \greeknumeral with numbers > 100 (#684).
- Adapt **localmarks** to work with LaTeX 6/2025.

New Features

- Add **koppa** option to Greek (#683).
- Add possibility to customize the numbers 6 and 90 in Greek (#683).

2.5 (2025/05/12)

Interface and defaults changes

- The babelnames `classiclatin` and `ecclesiasticlatin` are changed to `classicallatin` and `ecclesiasticallatin`. This reflects a change in the babel-latin package.

Bug fixes

- Fix a bug with linebreaking in CJK languages (#675).
- Fix global setting of **`babelshorthands`** (#680).
- Fix mapping between upper and lower case for classic and medieval Latin: *u/V, ú/Ú, ū/Ū, ŭ/Ŭ*.

2.4 (2025/01/31)

Bug fixes

- Fix **`changecounternumbering`** option of Bengali (#672).
- Fix testing of languages in groups (#669).
- Fix error with undefined internal command (`\xpg_pop_language:`) that should be defined (#671).
- The command `\familytype` is now predefined again independent of font settings, and its name was changed to `\l_xpg_familytype_tl`.
- A warning is now issued if babel is loaded in parallel with polyglossia.

2.3 (2024/09/23)

Bug fixes

- Line-breaking for Japanese and Chinese has been implemented. This is still considered experimental, please test and report any issues (#635).
- Prevent the Latin gloss from writing undefined commands to the aux file (#643).
- Fix use of `\pghyphenation` in preamble (#654).
- Fix handling of current language options in lists of figures/tables (#657).
- Fix definition of `\cyrdash` so that no linebreak is added when it is used at paragraph begin.
- Fix test for KOMA class in Hungarian and Russian glosses (#447).
- Fix display of font name in package error.
- Remove spurious spaces from Korean, Sanskrit, and Serbian glosses (#659, #660)

2.2 (2024/07/15)

Bug fixes

- Fix `lang` environment (#633).
- Fix first number of Maghrib abjad numbering for Algeria, Morocco and Tunisia (#640).
- Fix some captions strings in Bulgarian, Chinese, Esperanto, Estonian, Hindi, Latvian, and Slovak (#651)
- Improve performance with multilingual documents that have a TOC (#641).

New Features

- Add possibility to associate a font to a language/variant with reference to a BCP-47 script tag (#636).

Internal work

- The major rewrite of the package code has been finished. More code has been cleaned up, and polyglossia employs modern concepts (latex3 and modern LaTeX 2_ε). All key-val handling uses l3keys now rather than xkeyval.

2.1 (2024/03/07)

Bug fixes

- Make `\foreignlanguage` `\long` again (#622).
- Fix `otherlanguage*` environment (#622).

2.0 (2024/02/17)

Bug fixes

- Fix error with undefined property list that unhides with l3kernel version 2024-02-13 (#626).
- Fix **`frenchfootnote`** option (#620).
- The error message on improper font setting is now correct if the main script is non-Latin, for non-Roman fonts (#571), and specifically for the CJK case (#621).
- Fix expansion of localnumeral (#622).

New Features

- Add `\inheritbabelshorthands` (#604).
- Add `\usebabelshorthands`.
- Add shorthands for German gender-sensitive writing.
- Implement babel-style hyphenation for Finnish.

Documentation improvements

- Fix documentation of Finnish shorthands (#623).

Internal work

- The package code has been subjected to a major rewrite. Code has been cleaned up, and in general, polyglossia more consistently employs latex3 now. Note that this is an ongoing and yet unfinished project that will continue over the next releases.

1.66 (2023/12/11)

Bug fixes

- Fix OpenType language and script tags in Chinese (#606).
- Fix resetting of language specifications at the end of a local font set with babel legacy means (`\foreignlanguage` or `otherlanguage*` environment) (#607).
- Fix lowercasing of `\textlang`'s first mandatory argument. Now the casing does not change (language tag was lower-cased always) (#608).
- Add missing BCP-47 alias for Khmer (#611).
- Stop `\charifavailable` from looking ahead for more numbers or gobbling a space (#613).

New Features

- Add support for Odia, provided by Avisek Jena and Ashok Das.
- New auxiliary package `odiadigits`
- Add auxiliary command `\IfCharIsAvailableTF` to complement `\charifavailable`.

1.65 (2023/10/01)

Bug fixes

- Fix spurious space in Hebrew captions (#601).
- Fix resetting of language specifications at end of inline switch (#603).

Documentation improvements

- Refer to `\greeknumeral` and `\Greeknumeral` rather than to the synonyms `\greeknumber` and `\Greeknumber` (#602).

1.64 (2023/07/21)

Bug fixes

- Fix conflict of soul with **`splithyphens`** on XeLaTeX (#600).

1.63 (2023/06/10)

Bug fixes

- Language options are now lowercased before they are handled. This fixes errors when using the options in `\MakeUppercase` context (*e.g.,* in headings).
- Fix casing value for medieval Latin (this concerns ⟨u⟩ and ⟨V⟩ casing).
- Remove hacks in some gloss files where `\MakeUppercase` has been redefined to a no-op since the respective scripts do not have uppercasing. As `\MakeUppercase` is now locale-aware, these hacks are no longer needed. This concerns Arabic, Divehi, Hebrew, Hindi, Kurdish, Persian, Punjabi, Syriac, and Urdu.
- Fix Eastern (Mashriq) Arabic numbering direction with LuaLaTeX for Arabic, Kurdish, Persian, Syriac, and Urdu (amends previous fix to #213).
- French **`variant=acadian`** refers to the Canadian French subvariety as spoken in the Acadian region (as opposed to Quebec French). To differentiate it we now use a dedicated subtag which can also be used to access the language (`fr-CA-u-sd-canb`). Note that Quebec French is currently connected with `fr-CA`, not the more specific `fr-CA-u-sd-caqc`, although the u extension is returned if requested via `\BCPdata`. In practice, the Canadian subvarieties of French are identical currently, anyway.
- Do not blindly de-activate the quote character in languages that provide shorthands if the shorthands are not used. This allows to activate the character with other packages such as csquotes (#592).
- Do not flood the LaTeX hooks with redundant French itemlabels and footnote settings (#593).
- Fix Korean patching of part format with hyperref (#595).
- Make Hebrew numerals expandable (#596).
- Avoid superfluous writing of language switches to aux file (#593).

New features

- New option **capitalyiwn** for Armenian.
- New option **capitaleszett** for German.
- New option **capitaliota** for Greek.
- New option **datei** for Serbian.
- New option **sectionsep** for Arabic, Kurdish, Persian, Syriac, and Urdu (#589).
- Add aliases `eastern` (= `mashriq`) and `western` (= `maghrib`) to Arabic **numerals**.
- New date commands `\today*`, `\todayGen`, `\todayArabic`, and `\todayRoman` (all with starred counterparts) for Serbian (#524).
- Implement basic support for **variant=ijekavian** in Serbian.
- Dutch now hyphenates properly on letters with tremata. The new option, **tremahyphenation**, is provided to opt-out this feature.
- Add shorthands `"a`, `"e`, `"i`, `"o`, `"u`, `"A`, `"E`, `"I`, `"O`, `"U`, `"y`, and `"Y` to Dutch.

Interface changes

- All language options are now lowercase by default (though the mechanism is case-insensitive anyway). Changes cover Kurdish, Sanskrit, Serbian.

1.62 (2023/04/22)

Bug fixes

- `\BCPdata` was not fully expandable as required. Now it is.

1.61 (2023/04/16)

Bug fixes

- Properly reset fonts on change to Latin-script other language (#580)
- Turn ^ shorthand off in aux file for Latin (#582)
- Fix stacking of nested languages and resulting bug in TOC language assignment (#585)
- Fix `\greeknumeral{6}` (#587)
- Fix (re)setting of scripts in inline language commands.

New features

- Introduce `\BCPdata`, a standardized way to gather localization information (see sec. 12).

1.60 (2023/02/11)

Bug fixes

- Remove stray `U+FEFF` glyphs in gloss files (#574)
- Fix vertical whitespace in captions (#575)
- Fix whitespace in dutch with shorthands.

1.59 (2022/11/29)

Bug fixes

- Do not error, but warn, after language change due to non-updated aux file (#565).
- Fix catcode issue for Latin (#566).
- Improve error message with missing fonts (#571)

New features

- Add support for (simplified and traditional) Chinese (#65, provided by Takuji Tanaka).
- Add support for Punjabi (#572), provided by Arvinder Singh.

1.58 (2022/10/26)

Bug fixes

- The option **`forceheadingpunctuation`** is introduced for correct running headers in Hungarian documents (#557).
- Fix language setting in list of figure and table if captions float to different language areas (#542).
- Make Croatian digraphs robust (#552).
- Fix definition of some Finnish shorthands (#554).
- Re-fix Hungarian patching of parts with hyperref (#555).
- Correct running headers in Hungarian documents (#557).
- Fix option expansion issue (#559).

1.57 (2022/07/18)

Bug fixes

- Fix passing of macros to `\setmainlanguage` and `\setotherlanguage` (#543).
- Fix `otherlanguage*` environment with bidi text (#544).
- Re-fix patching of French part headings with hyperref (#546).
- Fix Latin prosodic shorthand issues (#547, Babel/#126, Babel/#129).

New features

- New option **`transliteration`** to Hebrew and hebrewcal (#540).
- New option **`fullyear`** to Hebrew.

1.56 (2022/04/20)

Bug fixes

- Fix English setup via aliases (#539).

1.55a (2022/04/10)

Bug fixes

- Fix `\foreignlanguage` command (#538).

1.55 (2022/04/09)

New features

- New option **splithyphens** and **vlna** for Polish (#535).
- Add **babelshorthands** to Polish.

Bug fixes

- Add missing `gloss-latex.lde` file to release (#537).

1.54 (2022/03/27)

New features

- New option **splithyphens** for Serbian (#496) and Portuguese (#534).
- Add **babelshorthands** to Portuguese.
- Add **schoolhyphens** option to Finnish (#525).

Interface and defaults changes

- Rename **disableligatures** to **disabledigraphs** for Croatian (#497, #500).
- Fix output with **numerals=cyrillic-alph** (part of #503).
- Standardize February and November in Indonesian according to the Great Dictionary of the Indonesian Language of the Language Center (*Kamus Besar Bahasa Indonesia*) (#526).

Bug fixes

- Fix robustification of font family switches (#428).
- Preserve font family switches across languages (#519).
- Fix TEX dash ligatures with **splithyphens** (#502).
- Prevent *missing hyphenmins value* LATEX error with unknown languages (#513).
- Fix global **babelshorthands**, **localmarks** and **verbose** options (#515).
- Fix Latin shorthands (#516).
- Remove spurious space in Bosnian date (#528).
- Fix `\languagevariant` and `\mainlanguagevariant` macros (#530, #531).
- Make `\iflanguageloaded` and friends work in preamble (#532).
- Fix deactivation of shorthands (#320).
- Fix deactivation of numerals.
- Update deprecated LATEX hooks (#523).
- Fix **luatexrenderer** option which was not considered appropriately (#533).
- Turn warning about `totalhyphenmin` to info, which is more appropriate (#111).

Documentation improvements

- Document in this manual where to report bugs (#512).

1.53 (2021/04/12)

New features

- New option **localalph** for Croatian (#486).
- Use ordinals in part and chapter headings. A new option **chapterformat** has been added to disable this or change format to Roman numbering (#479).

- New command `\uyghurordinal` and `\uyghurord` to produce Uyghur ordinals from 1–100.

Bug fixes

- Fix typo in Uyghur language definition file (#479).
- Fix Uyghur date format (#479).
- Fix Uyghur caption format and add **`swapstrings`** option (#479).
- Add overlooked digraphs in Croatian (#484).
- Fix typo in Serbian `\thepart` (#485).
- Fix part modifications in Hungarian with hyperref (#493).

1.52 (2021/03/16)

New features

- Adaptations to LaTeX 2021/05/01 pre-release 2 for Korean (#477).
- Add support for Uyghur (#475).
- New option **`mathfunctions`** for Russian and Ukrainian allows to disable the definitions of math macros that might clash with other packages (#465).
- Support LaTeX's new NFSS hooks (#471).

Bug fixes

- Fix French part modifications with hyperref (#469).
- Fix markup of French `\see` and `\alsoname` (#468).

1.51 (2020/12/08)

New features

- New option **`frenchpart`** for French (#458).
- New option **`splithyphens`** for Croatian (#454).

Bug fixes

- Use new LaTeX core hooks rather than filehook package. This fixes a recent breakage of filehook with other external packages (#453).
- Remove very old code that pretends polyglossia is babel (#455).
- Fix spelling of Albanian contentsname (#456).
- Fix part heading modification in French (#458).
- Fix extra space in Hebrew (#459).
- Register main polyglossia language earlier (#461).
- Allow for hyphenations in words following opening guillemet in French with XeTeX (#462).

1.50a (2020/10/15)

Bug fixes

- Assure `\autodot` is defined with `\KOMAScript` in Russian.

1.50 (2020/10/09)

New features

- Polyglossia now uses the Harfbuzz renderer by default with LuaTeX output. This brings LuaTeX on par with XeTeX for all scripts (#337). The renderer can be changed via the new global **`luatexrenderer`** option.
- The (previously inadvertently working) `hyphenrules` environment that ceased to work after a recent babel update is back and now officially supported. The environment now also supports language options and aliases (#427).
- New command `\setlanghyphenmins` to adapt hyphenation thresholds of languages and varieties.
- New command `\abjadalph` for Arabic with corresponding option (#431).
- Replace consecutive glues around punctuation by the correct amount of space with lualatex for French, ecclesiastic Latin, and Sanskrit (#437).

Bug fixes

- Remove warning about missing Brazil patterns (#404).
- Fix incompatibility with recent babel release (#408).
- Fixed some spellings in Marathi (#409).
- Fix spacing of geminating dot in Catalan (#410).
- Fix incompatibility of Marathi with beamer.
- Correct `\partname` in Hindi (#416).
- Updates and improvements to Kurdish (#418).
- Only activate shorthand character if **`babelshorthands`** is `true` (#421).
- Fix whitespace issue in Czech and Slovak with `vlna=true` (#423).
- Fix whitespace issue in Danish (#424).
- Fix catcode conflicts that might occur in language definition files f. ex. when loaded from a LaTeX3 class (#67, #425).
- Robustify font family switches (#428).
- Fix whitespace issue in Russian **`indentfirst`** option (#433).
- In Russian, **`indentfirst`** is now again default (#434).
- Fix LaTeX error with arabic numbering in Ukrainian (#440).
- Fix directionality after Hebrew decimal numbers (#441).
- Fix `babelname` of Latin Serbian (#442).
- Fix recording of secondary languages in `\xpg@bloaded` and `\xpg@bcp@loaded` lists (#443).
- Simplify and robustify section heading modification in Russian and introduce option **`forceheadingpunctuation`** (#444).
- Fix Cyrillic dash (via babelshorthand `"---`) when TeX ligatures are disabled (#445).
- Fix problem with large character indices in Lua module for punctuation spacing

Interface and defaults changes

- Polyglossia now uses the Harfbuzz renderer by default with LuaTeX output. See new features section.

Build fixes

- Fix an embarrassing bug in the dtx build script which was the reason for an utterly incomplete `polyglossia.dtx` file (#420).

Documentation improvements

- Document how to change `\lefthyphenmin` and `\righthyphenmin` for a language (#435).

1.49 (2020/04/08)

New features

- Add hook `\polyglossia@language@switched` to the external package interface (#398).
- Real fix for #400, that wasn't properly taken care of in 1.48.

Bug fixes

- Fix compilation error with some **`swapstring`** options in Hungarian (#373).
- Fix whitespace problem in Greek language.

Interface and defaults changes

- Changed Finnish caption for "Table of Contents" to "Sisällys" (#403).

1.48 (2020/03/25)

- No new features

Bug fixes

- Fix use of Hebrew with LuaLaTeX (#389).
- Do not overwrite footnote redefinitions of other packages with Latin and French (#391).
- Fix Serbian cyrillic numerals code (#392).
- Fix **`[no]localmarks`** option, whose logic was swapped (part of #395).
- Protect **`localmarks`** function against uppercased language names (part of #395).
- Fix buggy redefinition of `\@markright` with option localmarks (#396).
- Fix incompatibility between Latin and unicode-math (#394).
- Make (undocumented) `\defineshorthand` command (imported from babel) work.
- Fix usage of **`localmarks`** option without value.
- Emergency fixes for bugs caused by updates in babel's `switch.def` (#399 and #400).

Interface and defaults changes

- Use private macros in keyval choice keys (#390).

1.47 (2020/01/29)

New features

- IETF BCP-47 compliant language tags can now be used for loading and switching languages alternatively to language names (#226).
- New commands `\languageid{<type>}` and `\mainlanguageid{<type>}`.
- New test `\iflanguageidloaded`.
- New list `\xpg@bcp@loaded`.
- New environment `{lang}{<lang>}` (this is equivalent to `{<lang>}`, but also available with `\setlanguagealias*` which does not define dedicated alias environments).
- New gloss option `totalhyphenmin` (corresponds to LuaTeX's `\hyphenationmin`) (#111).

- New test `\iflanguageoption{<lang>}{<key>}{<val>}` (#364).
- Restore simple alphabetic numbering for `\asbuk` and `\Asbuk` in Belarusian, Mongolian, Russian, Serbian, and Ukrainian (#377).
- New command `\AsbukTrad` and `\asbukTrad` for Belarusian, Mongolian, Russian, Serbian, and Ukrainian which uses traditional alphanumerical numbering.
- New numerals option `cyrillic-trad` and `cyrillic-alph` to differentiate simple alphabetic and traditional alphanumerical Cyrillic numbering.
- `\selectbackgroundlanguage` and `\resetdefaultlanguage` now also support language aliases.
- New macro `\charifavailable{<char code>}{<substitution>}`.
- Add French language variant `swiss`.
- Implement **`babelshorthands`** for Croatian.
- Implement `\localnumeral` for Japanese.

Bug fixes

- Fix font family issue in headers (#355).
- Fix whitespace issues in `\text<lang>` (#356).
- Fix option-less `\babelname` in multi-variant languages (#357).
- Fix some spacing inconsistencies with French, Latin, and Sanskrit (#358).
- Fix issues with **`babelshorthands`** and graphics package (#368).
- Fix some captions and improve numbering in Marathi (#370).
- Fix Hungarian **`swapstrings`** feature (#373).
- Fix lua punctuation code problem (#374).
- Fix Bengali changecounternumbering option (#381).
- Fix whitespace issue in Japanese (#387).
- Fix `\text<lang>` command with multiple paragraphs.
- Actually implement documented german spelling variant `1996` (= `new`).
- Fix Slovenian **`localalph`** option.
- Fix Czech and Slovak **`splithyphens`** with typewriter fonts.
- `farsical.sty`: fix spacing issue with some month names.
- Fix directionalty of numbers in Hebrew with XeTeX.
- Improve interoperatability with biblatex (some language variants did not work yet).

Interface and defaults changes

- Some boolean options had `false` value by default, which meant if you passed them without value, the logic was reversed. This has been changed, leading to change of behavior should you have used one of these options without value (#363). Concerned are the following options:
  - **`babelshorthands`** in language Belarusian, Mongolian, Ukrainian, and Russian (now **`babelshorthands`** equals **`babelshorthand=true`**, no longer **`babelshorthands=false`**).
  - **`localalph`** in language Slovenian (**`localalph`** now equals **`localalph=true`**).
  - **`fullyear`** in package hebrewcal (**`fullyear`** now equals **`fullyear=true`**).
- The option **`latesthyphen`** in language German is deprecated. XeTeX and LuaTeX nowadays always use the latest German hyphens.
- The command `\setlanguagealias*` (introduced in v1.46) does no longer define dedicated alias environments.
- The babelnames for Latin variants are now changed to `classiclatin`, `ecclesiasticlatin` and `medievallatin`. This is how the hyphenation patterns and babel `\extras` are named, even though the variants can currently be selected in babel only via appended "dot modifier".

- In accordance with the respective l3kernel change, `\str_lower_case:n` has been renamed to `\str_lowercase:n` where used in `polyglossia.sty`. Thus polyglossia 1.47 requires l3kernel 2020-01-12 at least.

1.46 (2019/11/15)

New features

- Add option **`indentfirst`** to Russian (#78).
- Add options to set and customize French-style itemize item labels to French (#89).
- Polyglossia now decodes all supported babel language names in `\setdefaultlanguage`, `\setotherlanguage` and the language switching commands (#112, #132).
- Add optional localized math operators to Spanish (#123).
- Swap section headings in Hungarian (#344). New option **`swapstrings`** provides control over this.
- Introduce macro `\setlanguagealias` and `\setlanguagealias*`.
- Introduce language switching command `\textlang{lang}{...}` (these are equivalent to `\text<lang>`, but also available with `\setlanguagealias*` which does not define `\text<alias>`).
- Add support for Afrikaans.
- Add support for Belarusian.
- Add support for Bosnian.
- Add support for Georgian.
- Add Spanish variant `mexican`.
- Add babelshorthands as well as options **`splithyphens`** and **`vlna`** to Slovak.
- Add Latin language variant `ecclesiastic`.
- Add Latin language options `capitalizemonth`, `ecclesiasticfootnotes`, `hyphenation`, `prosodicshorthands`, and `usej`.
- Add Latin shorthands for «, », æ, Æ, œ, and Œ.
- Add French language option **`thincolonspace`**.

Bug fixes

- Fix problems with fragile font settings (#24).
- Fix clash of French punctuation spacing with the soul package (#52).
- Re-enable the possibility to pass a macro as main argument to `\setmainlanguage` and `\setotherlanguage` (#331).
- Fix detection of default `\languagevariant` (#332).
- Fix LaTeX error with undefined hyphenation pattern (#346).
- Fix some babel shorthand issues by updating the shorthand code from recent babel.
- Fix some problems with French and Latin auto-spacing (#345, #352).
- Fix an expl3 declaration (#348).

Interface and defaults changes

- The sub-package `cyrillicnumbers.sty` has been renamed to `xpg-cyrillicnumbers.sty` (per TeXLive request).
- In Russian, all paragraphs are now indented by default, as common in Russian typography. The behavior can be opted out by **`indentfirst=false`**.
- In Czech, **`splithyphens`** and **`lvna`** are enabled by default. Also, the option does now work as well with LuaTeX.

- Changed option name `fraktur` to `blackletter` in German (the former is still available as an alias).
- In French, high punctuation characters and guillemets are spaced by half an interword space now instead of a `\thinspace` (cf. #345).

1.45 (2019/10/27)

New features

- Introduce a framework for external packages to access language variants. This fixes, among other things, long-standing problems in the interaction of biblatex and polyglossia.
- Add new macros `\languagevariant` and `\mainlanguagevariant` as well as `\babelname` and `\mainbabelname` for package authors to access language information.
- Add new test `\iflanguageloaded{<language>}{<true>}{<false>}` where <language> can be a polyglossia or babel language name.
- Add new macros `\localnumeral, \localnumeral*, \Localnumeral` and `\Localnumeral*` that convert Arabic digitals to the local number scheme.
- Add new macro `\pghyphenation` to add language-specific hyphenation exceptions (#18).
- Add support form (Khalkha & Cyrillic) Mongolian in line with babel-mongolian (#23).
- Add option **`splithyphens`** and **`vlna`** to Czech (XeTeX only; for LuaTeX, use the package luavlna to get these features) (#32).
- Add support for Kurdish, both Kurmanji and Sorani (#277).
- Implement proper Cyrillic (alphanumeric) numbering (#285).
- Add new language `friulian`. This deprecates `friulan` (which is still supported for backwards compatibility).
- Add new language `malay` with variants `indonesian` and `malaysian`. This deprecates `bahasai` and `bahasam` (which are still supported for backwards compatibility).
- Add new language `gaelic` with variants `irish` and `scottish`. This deprecates `irish` and `scottish` as own polyglossia languages (which are still supported for backwards compatibility).
- Add new language `hungarian`. This deprecates `magyar` (which is still supported for backwards compatibility).
- Add new language `sorbian` with variants `lower` and `upper`. This deprecates `lsorbian` and `usorbian` (which are still supported for backwards compatibility).
- Add new language `portuguese` with variants `portuguese` and `brazilian`. This deprecates `brazil` and `portuges` (which are still supported for backwards compatibility).
- Add new language `norwegian` with variants `nynorsk` and `bokmal`. This deprecates `nynorsk` and `norsk` (which are still supported for backwards compatibility).
- Add new language `persian`. This deprecates `farsi` (which is still supported for backwards compatibility).
- Add new language `sami`. Currently only Northern Sami is supported. This deprecates `samin` (which is still supported for backwards compatibility).
- `gloss-serbian`: add **`numerals=cyrillic`** option. Add `\asbuk` and `\Asbuk` (#285).
- Implement basic support for (French) `canadien` and (English) `canadian` (#22).
- Improve support for Armenian (#79): Add captions, Eastern month names (accessible via variant=eastern) and Armenian alphabetic numbering (via **`numerals=armenian`** and `\armenicnumeral`).
- Add french option **`autospacing`** and commands `\AutoSpacing, \NoAutoSpacing` This allows to switch off autospacing globally or locally (#113).
- Fixup `\normalfont` (#203).
- Fix directionality issues in mixed RTL/LTR paragraphs (#204).
- Implement **`babelshorthands`** for Finnish (#212) and Czech.

- Implement access to current language via Lua (#243).
- Introduce french option option **autospacetypewriter** alias **OriginalTypewriter**.
- Support `\aemph` with lualatex
- Rename **automaticspacesaroundguillemets** to **autospaceguillemets** The old option is kept for backwards compatibility.

Bug fixes

- Fix equation number in Arabic and Farsi (#7).
- Simplify and document Hebrew **marcheshvan** option (#16).
- Fix hyphenation of Greek with LuaTeX (#55).
- Fix N'ko date format (#63).
- Disable the extras of a language when a nested language starts (#66, #169).
- Properly implement Bengali numbers (#69, #184).
- Fix conflicts with other packages caused by premature shorthand activation in preamble (#81, #200).
- Fix kerning in math with French (#92).
- Fix expansion issue in Hebrew (#93).
- Fix numbering expansion issue in Greek (#110).
- Postpone `\disablehyphenation` in preamble until after setting of document language (#125).
- Postpone the assignment of defaultfamily to `\AtBeginDocument`, thus do not overwrite `\familydefault` redefinitions in the preamble (#127).
- Reset number settings when switching language (#133).
- Hebrew: Properly store `\MakeUppercase` for later restoration (#152).
- Fix whitespace issue in `\datewelsh` (#158).
- When switching language, set the language/script specific font families (#164).
- Correct some Bengali captions (#165).
- Fix documentation of Serbian (#168).
- Reset ucl codes in Latin only if the respective variant is used (#172).
- Fix `\disablehyphenation` with LuaTeX (#187).
- Fix typos in Hindi captions (#202).
- Pass language options to the aux files (#205).
- Rewrite and fix English variant handling (#208).
- Define magyar caption formats in `\blockextras` and undef them in `\noextras` (#209).
- Ensure proper direction with arabic digits in Arabic and Farsi (#213).
- Fix `\linespread` with Korean (#218).
- Define Russian caption before key allocation (#219).
- Register current language in polyglossia lua module after selection (#234).
- Fix babel language switching commands (#239): `\foreignlanguage` and the starred `otherlanguage*` environment are not supposed to change dates.
- Fix French spacing leaking beyond French (#270).
- Redefine font families for French only if language is loaded (#270).
- `gloss-russian`:
  - Check whether command exist before redefining (#280).
  - Fix some whitespace issues.
- Fix and simplify `\frenchfootnote` definition (#294).
- Fix footnote numbering in Farsi.
- Fix Latin footnotes in Arabic documents.
- Set the correct main direction with luabidi.
- Fix **autospaceguillemets** option in French.

- Fix grouping in `gloss-danish.ldf`.
- Properly store `\MakeUppercase` and `\@arabic` for later restoration.

Documentation

- Add documentation about footnotes in RTL context
- Document Tibetan numerals option (#109).
- Improve `\frenchfootnote` documentation.
- Mention Japanese support in the docs.

1.44 (2019/04/04)

- Correction to Russian language file, by Maksim Zholudev (commit d2f383e).
- Added Macedonian language file, by Stefan Zlatinov (commit cd379e1).

1.43 (2019/03/05)

- Correction to Hindi language file, by Zdeněk Wagner.

1.42.5 (2017/04/13)

- Many changes to the French language file, by Maïeul Rouquette.

1.42.4 (February, March 2016)

- Remedial actions for the babel changes.
- Fixed side effect of pull request #122 (see commit d2a34ff).
- Added automatic Josa selection, variant, and captions options to Korean, by Dohyun Kim (pull request #128).
- Updated `gloss-occitan` from CTAN.

18-01-2016

- Fixed issue #124 (minor typo in `polyglossia-frpt.lua`)
- Merged pull request #117 for more French guillemet spacing
- Merged pull request #121 to add `\bbl@loaded`; fixes issue #120
- Merged pull request #122 that build on #121
- Merged pull request #116 for French (spacing around guillemets)
- Fixed issue #115 (spurious spaces in Arabic)

19-08-2015

- Fixed issue #107 for Marathi (labels and month names)

1.42.0 (2015/08/06)

- Add Bengali digits package, and option to reset all numbering functions.
- Add `long` option for Welsh date.
- Add local alphabet in Slovenian, for enumerations and such.
- Fix long-standing bug with Welsh: date should use ordinals.
- Fix for Latin with LuaTeX: all variants had same problems as Classic.
- Fixed error with British variant of English and LuaTeX (issue #86).

1.41.0 (2015/07/16)

- Added support for Khmer, by Say Ol (private email)

1.40.1 (2015/07/14)

- Bugfix for Korean, by Dohyun Kim (pull request #103)

1.40.0 (2015/07/07)

- `gloss-korean.ldf` contributed by Dohyun Kim (pull request #102)

1.33.7 (2015/07/04)

- Release to CTAN, no code change
- Fixed extraneous space in code for Swiss German (pull request #101)
- Fixed a typo in Ukrainian alphabet, for `\Asbuk` (pull request #99)
- Fix for Classic Latin: load patterns for LuaTeX
- Made `\rmfamily`, `\sffamily` and `\ttfamily` robust again
- Merged fix for Hebrew date format, by Guy Rutenberg (pull request #94)
- Merged fix for spurious space, by Caleb McKennan (pull request #91)
- Merged pull request #84 by Élie Roux for Tibetan
- Added support for Swiss German (pull request #75)
- Added commands `\Asbuk` and `\asbuk` for Ukrainian (pull request #76), similar to Russian
- Documented changes to Latin from last year.
- Be friendlier to right-to-left languages with LuaTeX
- Enhanced Latin support by Claudio Beccari

1.33.6 (2015/05/15)

- Introduce a `classical` and `medieval` variant of Latin
- Add `\asbuk` and `\Asbuk` for Ukrainian (after their Russian counterpart)
- Fix a number of bugs

1.33.5 (2014/05/21)

- Option to disable hyphenation entirely, by Élie Roux
- Fix spurious spaces in gloss-russian.ldf, by Oleg Domanov
- Support for the Austrian variant of German, by Jürgen Spitzmüller
- Changes to the Croatian translations, by Ivan Kokan
- Correction to the Lithuanian translations, by Ignas Anikevičius

1.33.4 (2013/06/27)

- Emergency release for a bug introduced in `babelsh.def`

1.33.3 (2013/05/28)

- Changed formatting of some error messages (emergency fixes for TeX Live 2013)

1.33.2 (2013/05/26)

- Added `\disablehyphenation` and `\enablehyphenation`, contributed by Élie Roux.
- Fixed bug related to package inclusion. Polyglossia would break if we loaded `breqn.sty`, and then called `\setmainlanguage{english}`, this is no longer the case.
- Removed spurious space introduced by `\captionswedish`.

1.33.1 (2013/05/23)

- Editorial changes to the documentation
- Hunted and documented bugs

1.33.0 (2013/05/20)

- Added support for N'Ko.
- Bugfixes for LuaTeX
- More work in progress on Bidi in LuaTeX.

1.32.0 (2013/05/15)

- Transitional version to support right-to-left languages with LuaTeX.

1.31 (2013/05/10) / 1.3 (2013/05/11)

- Several bugfixes.
- Sync with babel 3.9.
- Consolidated support for LuaTeX for all languages but the ones using South and South-East Asian scripts, and languages written from right to left. Many thanks to Élie Roux for his help.
- Added support for Tibetan, contributed by Élie Roux (end of lines are experimental).

1.30 (2012/08/06)

- Added support for LuaTeX. Many languages don't work yet. Please be patient.

1.2.0e (2012/04/28)

- Fixed a number of outstanding and not very interesting bugs.
- Added gloss files for Romansh and Friulan, contributed by Claudio Beccari.

1.2.0d (2012/01/12)

- Removed `\makeatletter` and `\makeother` from gloss files entirely.

1.2.0c (2011/10/12) [First update by Arthur Reutenauer]

- Update to `gloss-italian.ldf` by Claudio Beccari, incorporating changes by Enrico Gregorio.
- Conclude every gloss file with `\makeatother` to match the initial `\makeatletter`. (Not necessary from a technical point of vue, because of one of the changes by Enrico reported below, but I like it better that way :-)
- Conclude `polyglossia.sty` with `\ExplSyntaxOff` to match the initial `\ExplSyntaxOn`.
- Added gloss file for Kannada, contributed by Aravinda VK and others.

- Corrections to the gloss-dutch.ldf thanks to Wouter Bolsterlee.
- Several patches by Enrico Gregorio, fixing long-standing bugs. From the git log:
  - Deleted setup for right-to-left writing direction, see `http://tug.org/pipermail/xetex/2011-April/020319.html`
  - Changed three appearances of `\newcommand` to `\newrobustcmd`, as the commands needs to be protected. Bug reported by kamensky.
  - Corrected `\datepolish` as suggested by Piotr Kempa
  - Changed `\lccode"` into `\lccode\string"`, because it might come into action at wrong times when `"` is active
  - Changed definition of key `\xpg@setup`, as `\@tmpfirst` and `\@tmpsecond` were not expanded, causing dependence of `\lefthyphenmin` and `\righthyphenmin` to the last loaded language. Raised by Vadim Rodionov on the XeTeX mailing list.
  - Deleted `\bgroup` and `\egroup` tokens from the definition of `otherlanguage*`; they serve no purpose (we are already inside an environment) and conflict with csquotes. Noticed by P. Lehman.
  - Changed the calls of `\input` to `\xpg@input`, which inputs the required file and resets the catcode of `@` to the same value as it had before the input. Since each `.ldf` file starts with `\makeatletter`, the old behaviour would leave a category 11 @, which is wrong.
  - Added `\csuse{date#2}` to the definition of `otherlanguage*`.

1.2.0b (2011/10/03) [Update by Philipp Stephani]

- Load xkeyval package explicitly since newer versions of fontspec don't load it any more.

1.2.0a (2010/07/27) [Last update by François Charette]

- Initialize `\fontfamily` acc to value of `\familydefault` (Fixes a bug when using polyglossia with beamer)
- Remove spurious space in def of `\dateenglish`
- Add missing English variant `american`
- Serbian: fix date format and captions (Latin+Cyrillic)
- Fix `\atticnumeral` in `gloss-greek`
- Small improvements and fixes in documentation

1.2.0 (2010/07/15)

- Adapted for fontspec 2.0 (will not work with earlier versions!)
- Implementation of a `\PolyglossiaSetup` interface that simplifies writing `gloss-*.ldf` files
- Many internal enhancements and refactoring (including a patch by Dirk Ulrich)
- Improved automatic font setup when `\<lang>font` is not defined
- New environment `otherlanguage*` (equivalent to `\foreignlanguage` (Enrico Gregorio)
- Bugfix to prevent bogus expansion of `\{rm,sf,tt}family` even in aux files (Enrico Gregorio)
- New gloss files for Armenian, Bengali, Occitan, Bengali, Lao, Malayalam, Marathi, Tamil, Telugu, and Turkmen.
- New auxiliary packages `devanagaridigits` and `bengalidigits`
- hijrical no longer loads bidi and checks for incompatible l3calc
- Add babel shorthands for Russian (based on a patch by Vladimir Lomov)
- Fix `\fnum@{table,figure}` for Lithuanian
- Various improvements in `gloss-russian` (provided by Vladimir Lomov and others)
- Corrected captions for Bahasai, Lithuanian, Russian, Croatian

- Add option **indentfirst=true** for Spanish, Croatian and other languages (**indentfirst** was previously named **frenchindent**)
- New option **script** for German: Setting **script=fraktur** modifies the captions for typesetting in Fraktur.
- New command `\aemph` for Arabic, Farsi, Urdu, etc. to mark emphasis through overlining.
- Package option **nolocalmarks** is now true by default: to activate it the option **localmarks** must be passed instead.
- Loading languages à la babel as package options is no longer possible (this feature was deprecated since v1.1.0).

1.1.1 (2010/03/23)

- Bugfix for French: explicit spaces before/after double punctuation signs (Par exemple : les grands « espaces » du Canada !) are now replaced by the appropriate non-breaking spaces, as in babel.
- Bugfix for font switching mechanism within Latin script (pending a complete re-implementation in v1.2): the font shape and series are no longer reset when switching language.
- New macros for non-Western decimal digits (instead of fontmappings)
- New gloss files for Asturian, Lithuanian and Urdu
- `hijrical.sty` is now locale-aware: `\hijritoday` is formatted differently in Arabic, Farsi, Urdu, Turkish and Bahasa Indonesia.
- Enable **babelshorthands** for Dutch
- Add missing macro `\allowhyphens`
- Add global option **babelshorthands**
- Support Catalan geminated l
- Bugfix when declaring more than one pkg option
- Protect `\reset@font`
- Add missing requirement makecmds
- Bugfix for smallcaps in captions
- Typo for `ccname` in Hebrew
- Add option **numerals** to `gloss-russian`
- Provide `\newXeTeXintercharclass` when undefined
- Bugfix for Russian `\alph`
- Remove superfluous level of `{}` in definition of `\markright`
- Bugfix for `\datecatalan`
- Change `hyphenmins` for Sanskrit

1.1.0b (2009/11/22)

- Modify `\hyphenmins` for Sanskrit (Yves Codet)
- Bugfixes for Serbian and Bulgarian (Enrico Gregorio)

1.1.0a (2009/11/22)

- Bugfix for interchar tokens

1.1.0 (2009/11/20)

- Use `\newXeTeXintercharclass` (thanks to Enrico Gregorio)
- Fixed implementation of shorthands for German (babel code in file `babelsh.def`)
- Arabic (Khaled Hosny):
  - Fix abjad form for 3 and 5 and add option `\abjadjimnotail`

    - bugfix for `\arabicnumber`
    - make Gregorian calendar the default
    - fixed typos in the sample text
- Turkish (S. Ö. Yildiz):
    - fix white-space before : and !
    - also check if the font specified TRK for language
    - added missing Turkish translation of "Glossary"
- Suppress `nopattern` warning for non-hyphenated scripts
- Changed U+0163 to U+021B for Romanian (Elie Roux)
- Stylistic fixes and use macro `\xpg@option` for package options (E. Gregorio)
- Fix month names in Dutch (A. Ledda)
- Add Brazilian translation for "glossary"
- Remove spurious space generated by `gloss-spanish`
- Fix `ldf` file for brazilian
- Various improvements in the code communicated by E. Gregorio:
    - remove superfluous `\protect\language`
    - change default language from `0` to `\l@nohyphenation=255`
    - localize `lccode` handling of apostrophe in French; add it to Italian
- Fix `frenchspacing` for Vietnamese
- Other minor bugfixes

1.0.2 (2009/01/27)

- Captions corrected in Hebrew, Russian and Spanish
- Removed all `\text<lang>` wrappers within caption definitions
- Improved compatibility with babel
- New option **`babelshorthands`** for German
- New option **`Script`** for Sanskrit

1.0.1 (2008/07/31)

- Improved documentation (added sections on font setup and numeration mappings)
- Improvements and bug fixes for English and German
- Bugfix in `gloss-syriac.ldf` (spurious space after `\textsyriac{...}`)
- Extended the scope of `\syriacabjad`
- Added `gloss-amharic.ldf` (ported from `ethiop.ldf` in the package ethiop)

1.0 (2008/07/13)

- Initial release on CTAN.

14 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 (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 XeTeX 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 XeTeX!

15 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, Matthew Bertucci, Wouter Bolsterlee, Christian Buhtz, Ashok Das, Oleg Domanov, user fakhriaunur, Philipp Gesang, Kevin Godby, Enrico Gregorio, Khaled Hosny, Stephen Huan, Najib Idrissi, user julroy67, Dohyun Kim, Phil Kime, Mike Kroutikov, Ivan Kokan, Caleb Maclennan, Gyula Magyarkuti, José Mancera, Miquel Ortega, Yevgen Pogribnyi, Will Robertson, Maïeul Rouquette, Elie Roux, Hugo Roy, Guy Rutenberg, Marcin Serwin, Arvinder Singh, Avisek Jena, Philipp Stephani, Niranjan Tambe, Takuji Tanaka, Osman Tursun, Keno Wehr, Dominik Wujastyk, Sertaç Ö. Yildiz, Andrey Zgarbul, 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!