Polyglossia: Modern multilingual typesetting with \texttt{XeLaTeX} and \texttt{LuaLaTeX}

François Charette        Arthur Reutenauer$^\ast$
Bastien Roucairiès       Jürgen Spitzmüller

2021/04/12  V1.53
(pdf file generated on 12 April 2021)

Contents

1 Introduction 4

2 Setting up multilingual documents 5
   2.1 Activating languages ................................. 5
   2.2 Supported languages ............................... 5
   2.3 Relation to and use of Babel language names 6
   2.4 Using IETF language tags .......................... 8
   2.5 Global options .................................. 12

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

$^\ast$Current maintainer
1 Introduction

Polyglossia is a package for facilitating multilingual typesetting with \LaTeX{} 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).\footnote{This is done by bundled sub-packages such as arabicnumbers.}
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 \LaTeX{} world (like font encodings, shorthands, etc.) are not supported. Generally speaking, polyglossia aims to remain as compatible as possible with the fundamental features of 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 \LaTeX{} and \Lualatex{}, it obviously also relies on fontspec by Will Robertson. For languages written from right to left, it needs the package bidi (for \LaTeX{}) or luabidi (for \Lualatex{}) by Vafa Khalighi (وفا خليقي) and the bidi-tex GitHub Organisation. Polyglossia also bundles three packages for calendaric computations (hebrewcal, hijrical, and farsical).
2 Setting up multilingual documents

2.1 Activating languages

The default language of a document is specified by means of the command
\setdefaultlanguage{(options)}{(lang)}
(or, equivalently, \setmainlanguage). Secondary languages are specified with
\setotherlanguage{(options)}{(lang)}.

All these commands allow you to set language-specific options.\[^2\] It is also possible to load a series of secondary languages at once (but without options) using
\setotherlanguages{(lang1),(lang2),(lang3),(...)}.

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

\textbf{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.

\textbf{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 (bahasa/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, Belarusan, Bosnian and Georgian. Version 1.52 ← introduces support for Uyghur.

\[^2\]Section 6 documents these options for the respective languages.
\[^3\]See acknowledgements at the end for due credit to the various contributors.
Table 1. Languages currently supported in polyglossia

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

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., bahasa for Bahasa Malayu) as well as endonyms, i.e., language names coming from the designated languages (such as bahasa, canadien or magyar). As opposed to this, polyglossia always uses spelled-out (lower-cased) English language names. Please refer to table 2 for the differing language names in both packages.

For convenience reasons, polyglossia also supports the use of babel names ← (for the few justified exceptions, please refer to the notes in table 2). The babel names listed in table 2 can be used instead of the corresponding polyglossia name/options in \setdefaultlanguage and \setotherlanguage as well as in the polyglossia and babel language switching commands/environments documented in section 3.1 and 3.2 (e.g., \textaustrian is synonymous to \textgerman[variant=austrian,spelling=old]). However, unless you have special reasons, we strongly encourage you to use the polyglossia names.
<table>
<thead>
<tr>
<th>Babel name</th>
<th>Polyglossia name</th>
<th>Polyglossia options</th>
</tr>
</thead>
<tbody>
<tr>
<td>acadien</td>
<td>french</td>
<td>variant=acadian</td>
</tr>
<tr>
<td>american</td>
<td>english</td>
<td>variant=american</td>
</tr>
<tr>
<td>australian</td>
<td>english</td>
<td>variant=australian</td>
</tr>
<tr>
<td>austrian</td>
<td>german</td>
<td>variant=austrian, spelling=old</td>
</tr>
<tr>
<td>bahasa</td>
<td>malay</td>
<td>variant=indonesian</td>
</tr>
<tr>
<td>bahasam</td>
<td>malay</td>
<td>variant=malaysian</td>
</tr>
<tr>
<td>brazil</td>
<td>portuguese</td>
<td>variant=brazilian</td>
</tr>
<tr>
<td>british</td>
<td>english</td>
<td>variant=british</td>
</tr>
<tr>
<td>canadian</td>
<td>english</td>
<td>variant=canadian</td>
</tr>
<tr>
<td>canadien</td>
<td>french</td>
<td>variant=canadian</td>
</tr>
<tr>
<td>classiclatina</td>
<td>latin</td>
<td>variant=classic</td>
</tr>
<tr>
<td>farsi</td>
<td>persian</td>
<td></td>
</tr>
<tr>
<td>ecclesiasticlatinb</td>
<td>latin</td>
<td>variant=ecclesiastic</td>
</tr>
<tr>
<td>friulan</td>
<td>friulan</td>
<td></td>
</tr>
<tr>
<td>germanc</td>
<td>german</td>
<td>spelling=old</td>
</tr>
<tr>
<td>irish</td>
<td>gaelic</td>
<td>variant=irish</td>
</tr>
<tr>
<td>kurmanji</td>
<td>kurdish</td>
<td>variant=kurmanji</td>
</tr>
<tr>
<td>lowersorbian</td>
<td>sorbian</td>
<td>variant=lower</td>
</tr>
<tr>
<td>magyar</td>
<td>hungarian</td>
<td></td>
</tr>
<tr>
<td>medievallatinb</td>
<td>latin</td>
<td>variant=medieval</td>
</tr>
<tr>
<td>naustrian</td>
<td>german</td>
<td>variant=austrian</td>
</tr>
<tr>
<td>newzealand</td>
<td>english</td>
<td>variant=newzealand</td>
</tr>
<tr>
<td>ngerman</td>
<td>german</td>
<td>variant=german</td>
</tr>
<tr>
<td>norsk</td>
<td>norwegian</td>
<td>variant=bokmal</td>
</tr>
<tr>
<td>nswissgerman</td>
<td>german</td>
<td>variant=swiss</td>
</tr>
<tr>
<td>nynorsk</td>
<td>norwegian</td>
<td>variant=nynorsk</td>
</tr>
<tr>
<td>polutonikogreek</td>
<td>greek</td>
<td>variant=polytonic</td>
</tr>
<tr>
<td>portuges</td>
<td>portuguese</td>
<td>variant=portuguese</td>
</tr>
<tr>
<td>samin</td>
<td>sami</td>
<td></td>
</tr>
<tr>
<td>scottish</td>
<td>gaelic</td>
<td>variant=scottish</td>
</tr>
<tr>
<td>serbian</td>
<td>serbian</td>
<td>script=Cyrillic</td>
</tr>
<tr>
<td>slovene</td>
<td>slovenian</td>
<td></td>
</tr>
<tr>
<td>spanishmx</td>
<td>spanish</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>swisssgerman</td>
<td>german</td>
<td>variant=swiss, spelling=old</td>
</tr>
<tr>
<td>uppersorbian</td>
<td>sorbian</td>
<td>variant=upper</td>
</tr>
</tbody>
</table>

*a In babel currently only selectable via dot modifier (*latin.classic*).

*b In babel currently only selectable via dot modifier (*latin.ecclesiastic*).

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

*d In babel currently only selectable via dot modifier (*latin.medieval*).
2.4 Using IETF language tags

Polyglossia ← also supports the use of language tags that conform to the IETF BCP-47 Best Current Practice. 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⟩} \ldots \end{lang} and the babel language switching commands/environments documented in section 3.2. Table 3 lists the currently supported tags.

<table>
<thead>
<tr>
<th>BCP-47 tag</th>
<th>Polyglossia name</th>
<th>Polyglossia options</th>
</tr>
</thead>
<tbody>
<tr>
<td>aeb</td>
<td>arabic</td>
<td>locale=tunisia</td>
</tr>
<tr>
<td>af</td>
<td>afrikaans</td>
<td></td>
</tr>
<tr>
<td>afb</td>
<td>arabic</td>
<td>locale=default</td>
</tr>
<tr>
<td>am</td>
<td>amharic</td>
<td></td>
</tr>
<tr>
<td>apd</td>
<td>arabic</td>
<td>locale=default</td>
</tr>
<tr>
<td>ar</td>
<td>arabic</td>
<td></td>
</tr>
<tr>
<td>ar-IQ</td>
<td>arabic</td>
<td>locale=mashriq</td>
</tr>
<tr>
<td>ar-JO</td>
<td>arabic</td>
<td>locale=mashriq</td>
</tr>
<tr>
<td>ar-LB</td>
<td>arabic</td>
<td>locale=mashriq</td>
</tr>
<tr>
<td>ar-MR</td>
<td>arabic</td>
<td>locale=mauritania</td>
</tr>
<tr>
<td>ar-PS</td>
<td>arabic</td>
<td>locale=mashriq</td>
</tr>
<tr>
<td>ar-SY</td>
<td>arabic</td>
<td>locale=mashriq</td>
</tr>
<tr>
<td>ar-YE</td>
<td>arabic</td>
<td>locale=default</td>
</tr>
<tr>
<td>arq</td>
<td>arabic</td>
<td>locale=algeria</td>
</tr>
<tr>
<td>ary</td>
<td>arabic</td>
<td>locale=libya</td>
</tr>
<tr>
<td>arz</td>
<td>arabic</td>
<td>locale=default</td>
</tr>
<tr>
<td>ast</td>
<td>asturian</td>
<td></td>
</tr>
<tr>
<td>ayl</td>
<td>arabic</td>
<td>locale=libya</td>
</tr>
<tr>
<td>be</td>
<td>belarusian</td>
<td></td>
</tr>
<tr>
<td>be-tarask</td>
<td>belarusian</td>
<td>spelling=classic</td>
</tr>
<tr>
<td>bg</td>
<td>bulgarian</td>
<td></td>
</tr>
<tr>
<td>bn</td>
<td>bengali</td>
<td></td>
</tr>
<tr>
<td>bo</td>
<td>tibetan</td>
<td></td>
</tr>
<tr>
<td>br</td>
<td>breton</td>
<td></td>
</tr>
<tr>
<td>bs</td>
<td>bosnian</td>
<td></td>
</tr>
<tr>
<td>ca</td>
<td>catalan</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>BCP-47 tag</th>
<th>Polyglossia name</th>
<th>Polyglossia options</th>
</tr>
</thead>
<tbody>
<tr>
<td>ckb</td>
<td>kurdish variant=sorani</td>
<td>default</td>
</tr>
<tr>
<td>ckb-Arab</td>
<td>kurdish variant=sorani, script=Arabic</td>
<td>default</td>
</tr>
<tr>
<td>ckb-Latn</td>
<td>kurdish variant=sorani, script=Latin</td>
<td></td>
</tr>
<tr>
<td>cop</td>
<td>coptic</td>
<td></td>
</tr>
<tr>
<td>cy</td>
<td>welsh</td>
<td></td>
</tr>
<tr>
<td>cz</td>
<td>czech</td>
<td></td>
</tr>
<tr>
<td>da</td>
<td>danish</td>
<td></td>
</tr>
<tr>
<td>de</td>
<td>german</td>
<td></td>
</tr>
<tr>
<td>de-AT</td>
<td>german variant=austrian, spelling=new</td>
<td></td>
</tr>
<tr>
<td>de-AT-1901</td>
<td>german variant=austrian, spelling=old</td>
<td></td>
</tr>
<tr>
<td>de-AT-1996</td>
<td>german variant=austrian, spelling=new</td>
<td></td>
</tr>
<tr>
<td>de-CH</td>
<td>german variant=swiss, spelling=new</td>
<td></td>
</tr>
<tr>
<td>de-CH-1901</td>
<td>german variant=swiss, spelling=old</td>
<td></td>
</tr>
<tr>
<td>de-CH-1996</td>
<td>german variant=swiss, spelling=new</td>
<td></td>
</tr>
<tr>
<td>de-DE</td>
<td>german variant=german, spelling=new</td>
<td></td>
</tr>
<tr>
<td>de-DE-1901</td>
<td>german variant=german, spelling=old</td>
<td></td>
</tr>
<tr>
<td>de-DE-1996</td>
<td>german variant=german, spelling=new [default]</td>
<td></td>
</tr>
<tr>
<td>de-Latf</td>
<td>german script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-AT</td>
<td>german variant=austrian, spelling=new, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-AT-1901</td>
<td>german variant=austrian, spelling=old, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-AT-1996</td>
<td>german variant=austrian, spelling=new, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-CH</td>
<td>german variant=swiss, spelling=new, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-CH-1901</td>
<td>german variant=swiss, spelling=old, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-CH-1996</td>
<td>german variant=swiss, spelling=new, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-DE</td>
<td>german variant=german, spelling=new, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-DE-1901</td>
<td>german variant=german, spelling=old, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>de-Latf-DE-1996</td>
<td>german variant=german, spelling=new, script=blackletter</td>
<td></td>
</tr>
<tr>
<td>dsb</td>
<td>sorbian variant=lower</td>
<td></td>
</tr>
<tr>
<td>dv</td>
<td>divehi</td>
<td></td>
</tr>
<tr>
<td>el</td>
<td>greek variant=monotonic</td>
<td>default</td>
</tr>
<tr>
<td>el-polyton</td>
<td>greek variant=polytonic</td>
<td></td>
</tr>
<tr>
<td>en</td>
<td>english variant=austalian</td>
<td></td>
</tr>
<tr>
<td>en-AU</td>
<td>english variant=canadian</td>
<td></td>
</tr>
<tr>
<td>en-GB</td>
<td>english variant=british</td>
<td></td>
</tr>
<tr>
<td>en-NZ</td>
<td>english variant=newzealand</td>
<td></td>
</tr>
<tr>
<td>en-US</td>
<td>english variant=us [default]</td>
<td></td>
</tr>
<tr>
<td>eo</td>
<td>esperanto</td>
<td></td>
</tr>
</tbody>
</table>
Table 3. BCP47-polyglossia language name matching (continued)

<table>
<thead>
<tr>
<th>BCP-47 tag</th>
<th>Polyglossia name</th>
<th>Polyglossia options</th>
</tr>
</thead>
<tbody>
<tr>
<td>es</td>
<td>spanish</td>
<td>variant=spanish [default]</td>
</tr>
<tr>
<td>es-ES</td>
<td>spanish</td>
<td>variant=spanish [default]</td>
</tr>
<tr>
<td>es-MX</td>
<td>spanish</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>et</td>
<td>estonian</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>eu</td>
<td>basque</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>fa</td>
<td>persian</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>fi</td>
<td>finnish</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>fr</td>
<td>french</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>fr-CA</td>
<td>french</td>
<td>variant=canadian</td>
</tr>
<tr>
<td>fr-CH</td>
<td>french</td>
<td>variant=swiss</td>
</tr>
<tr>
<td>fr-FR</td>
<td>french</td>
<td>variant=french [default]</td>
</tr>
<tr>
<td>fur</td>
<td>friulian</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ga</td>
<td>gaelic</td>
<td>variant=irish [default]</td>
</tr>
<tr>
<td>gd</td>
<td>gaelic</td>
<td>variant=irish</td>
</tr>
<tr>
<td>gl</td>
<td>galician</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>grc</td>
<td>greek</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>he</td>
<td>hebrew</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>hi</td>
<td>hindi</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>hr</td>
<td>croatian</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>hsb</td>
<td>sorbian</td>
<td>variant=upper [default]</td>
</tr>
<tr>
<td>hu</td>
<td>hungarian</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>hy</td>
<td>armenian</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ia</td>
<td>interlingua</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>id</td>
<td>malay</td>
<td>variant=indonesian</td>
</tr>
<tr>
<td>is</td>
<td>icelandic</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>it</td>
<td>italian</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ja</td>
<td>japanese</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ka</td>
<td>georgian</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>kn</td>
<td>kannada</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ko</td>
<td>korean</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ku</td>
<td>kurdish</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>kmr</td>
<td>kurdish</td>
<td>variant=kurmanji</td>
</tr>
<tr>
<td>kmr-Arab</td>
<td>kurdish</td>
<td>variant=kurmanji, script=Arabic</td>
</tr>
<tr>
<td>kmr-Latn</td>
<td>kurdish</td>
<td>variant=kurmanji, script=Latin</td>
</tr>
<tr>
<td>kn</td>
<td>kannada</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ko</td>
<td>korean</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ku</td>
<td>kurdish</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>ku-Arab</td>
<td>kurdish</td>
<td>script=Arabic</td>
</tr>
<tr>
<td>ku-Latn</td>
<td>kurdish</td>
<td>script=Latin</td>
</tr>
<tr>
<td>la</td>
<td>latin</td>
<td>variant=mexican</td>
</tr>
<tr>
<td>la-x-classic</td>
<td>latin</td>
<td>variant=classic</td>
</tr>
</tbody>
</table>
Table 3. BCP47-polyglossia language name matching (continued)

<table>
<thead>
<tr>
<th>BCP-47 tag</th>
<th>Polyglossia name</th>
<th>Polyglossia options</th>
</tr>
</thead>
<tbody>
<tr>
<td>la-x-ecclesia</td>
<td>latin</td>
<td>variant=ecclesiastic</td>
</tr>
<tr>
<td>la-x-mediieval</td>
<td>latin</td>
<td>variant=medieval</td>
</tr>
<tr>
<td>lo</td>
<td>lao</td>
<td></td>
</tr>
<tr>
<td>lt</td>
<td>lithuanian</td>
<td></td>
</tr>
<tr>
<td>lv</td>
<td>latvian</td>
<td></td>
</tr>
<tr>
<td>mk</td>
<td>macedonian</td>
<td></td>
</tr>
<tr>
<td>ml</td>
<td>malayalam</td>
<td></td>
</tr>
<tr>
<td>mn</td>
<td>mongolian</td>
<td></td>
</tr>
<tr>
<td>mr</td>
<td>marathi</td>
<td></td>
</tr>
<tr>
<td>nb</td>
<td>norwegian</td>
<td>variant=bokmal</td>
</tr>
<tr>
<td>nko</td>
<td>nko</td>
<td></td>
</tr>
<tr>
<td>nl</td>
<td>dutch</td>
<td></td>
</tr>
<tr>
<td>nn</td>
<td>norwegian</td>
<td>variant=nynorsk [default]</td>
</tr>
<tr>
<td>oc</td>
<td>occitan</td>
<td></td>
</tr>
<tr>
<td>pl</td>
<td>polish</td>
<td></td>
</tr>
<tr>
<td>pms</td>
<td>piedmontese</td>
<td></td>
</tr>
<tr>
<td>pt</td>
<td>portuguese</td>
<td></td>
</tr>
<tr>
<td>pt-BR</td>
<td>portuguese</td>
<td>variant=brazilian</td>
</tr>
<tr>
<td>pt-PT</td>
<td>portuguese</td>
<td>variant=portuguese [default]</td>
</tr>
<tr>
<td>rm</td>
<td>romansh</td>
<td></td>
</tr>
<tr>
<td>ro</td>
<td>romanian</td>
<td></td>
</tr>
<tr>
<td>ru</td>
<td>russian</td>
<td></td>
</tr>
<tr>
<td>ru-luna1918</td>
<td>russian</td>
<td>spelling=modern [default]</td>
</tr>
<tr>
<td>ru-petr1708</td>
<td>russian</td>
<td>spelling=old</td>
</tr>
<tr>
<td>sa</td>
<td>sanskrit</td>
<td></td>
</tr>
<tr>
<td>sa-Beng</td>
<td>sanskrit</td>
<td>script=Bengali</td>
</tr>
<tr>
<td>sa-Deva</td>
<td>sanskrit</td>
<td>script=Devanagari [default]</td>
</tr>
<tr>
<td>sa-Gujr</td>
<td>sanskrit</td>
<td>script=Gujarati</td>
</tr>
<tr>
<td>sa-Knda</td>
<td>sanskrit</td>
<td>script=Kannada</td>
</tr>
<tr>
<td>sa-Mlym</td>
<td>sanskrit</td>
<td>script=Malayalam</td>
</tr>
<tr>
<td>sa-Telu</td>
<td>sanskrit</td>
<td>script=Telugu</td>
</tr>
<tr>
<td>se</td>
<td>sami</td>
<td></td>
</tr>
<tr>
<td>sk</td>
<td>slovak</td>
<td></td>
</tr>
<tr>
<td>sl</td>
<td>slovenian</td>
<td></td>
</tr>
<tr>
<td>sq</td>
<td>albanian</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>serbian</td>
<td></td>
</tr>
<tr>
<td>sr-Cyrl</td>
<td>serbian</td>
<td>script=Cyrillic</td>
</tr>
<tr>
<td>sr-Latn</td>
<td>serbian</td>
<td>script=Latin [default]</td>
</tr>
<tr>
<td>sv</td>
<td>swedish</td>
<td></td>
</tr>
</tbody>
</table>
Table 3. BCP47-polyglossia language name matching (continued)

<table>
<thead>
<tr>
<th>BCP-47 tag</th>
<th>Polyglossia name</th>
<th>Polyglossia options</th>
</tr>
</thead>
<tbody>
<tr>
<td>syr</td>
<td>syriac</td>
<td></td>
</tr>
<tr>
<td>ta</td>
<td>tamil</td>
<td></td>
</tr>
<tr>
<td>te</td>
<td>telugu</td>
<td></td>
</tr>
<tr>
<td>th</td>
<td>thai</td>
<td></td>
</tr>
<tr>
<td>tk</td>
<td>turkmen</td>
<td></td>
</tr>
<tr>
<td>tr</td>
<td>turkish</td>
<td></td>
</tr>
<tr>
<td>ug</td>
<td>uyghur</td>
<td></td>
</tr>
<tr>
<td>uk</td>
<td>ukrainian</td>
<td></td>
</tr>
<tr>
<td>ur</td>
<td>urdu</td>
<td></td>
</tr>
<tr>
<td>vi</td>
<td>vietnamese</td>
<td></td>
</tr>
<tr>
<td>zsm</td>
<td>malay variant=malaysian [default]</td>
<td></td>
</tr>
</tbody>
</table>

2.5 Global options

Polyglossia can be loaded with the following global package options:

- **babelshorthands** ← = *true or false
  
  Globally activates babel shorthands whenever available. Currently shorthands are implemented for Afrikaans, Belarusian, Catalan, Croatian, Czech, Dutch, Finnish, Georgian, German, Italian, Latin, Mongolian, Russian, Slovak, and Ukrainian. Please refer to the respective language descriptions (sec. 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⟩}⟨…⟩). 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.

- **luatexrenderer** ← = (renderer) (default value: Harfbuzz)
  
  determines which font renderer is used with Lua\TeX{} output. The correct font renderer is essential particularly for non-Latin scripts. By default, polyglossia uses the Harfbuzz renderer that has been introduced to Lua\TeX{}
in 2019 (\TeX\Live 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.

\texttt{\luatexrenderer=none} disables \polyglossia’s automatic renderer setting.

- \texttt{\texttt{\verb|\verbose|= \texttt{*true} or \texttt{false}}
  
  determines whether info messages and (some of the) warnings issued by \TeX, \fontspec and \polyglossia are output.

3 Language-switching commands

3.1 Recommended commands

\begin{verbatim}
\texttt{\text\langle lang\rangle} \langle options\rangle\{\langle…\rangle\} \langle text\rangle\langle lang\rangle
\texttt{\textlang\{\langle lang\rangle\}\langle options\rangle\{\langle…\rangle\} \langle text\rangle\langle lang\rangle}
\end{verbatim}

For each activated language the command \texttt{\text\langle lang\rangle\{\langle options\rangle\}{\langle…\rangle}} (as well as the synonymous \texttt{\textlang\{\langle options\rangle\}{\langle lang\rangle}{\langle…\rangle} \langle text\rangle\langle lang\rangle}) becomes available for short insertions of text in that language.

For example \texttt{\textrussian{\today}} and \texttt{\textlang{russian}{\today}} yield 12 апреля 2021 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 \texttt{\today}. They do not, however, translate so-called \textsl{caption strings}, i.e., “chapter”, “figure” etc., to the local language (these remain in the currently active ‘outer’ language).

The environment \texttt{\langle lang\rangle}, which is also available for any activated language \texttt{\langle lang\rangle} (as well as the equivalent \texttt{\begin{lang}\{\langle options\rangle\}{\langle lang\rangle}\{\langle…\rangle\} \langle text\rangle\langle lang\rangle} \langle end\rangle\langle lang\rangle), is meant for longer passages of text. It behaves slightly different than the \texttt{\text\{\langle lang\rangle\}} and \texttt{\textlang\{\langle lang\rangle\}} 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 \textsl{Iliad}:

\begin{quote}
\begin{greek}[variant=ancient]
\begin{verbatim}
μὴνιν ἀείδε θεά Πηληνάδεω Αχιλήδος οὐλομένην, ἢ μυρί’ Ἀχιλαῖς
ἀλγε’ ἔθηκε, πολλὰς δ’ ἱσθίμους ψυχὰς Ἀϊδί προίαψεν ἤρων,
αὐτοὺς δὲ ἐλώρια τεῦχε κύνεσσιν οἰνοσίσι τε πάσι, Διὸς δ’
\end{verbatim}
\end{greek}
\end{quote}
ἐτελείετο βουλή, ἐξ οὗ δὴ τὰ πρῶτα διαστήτην ἐρίσαντε Ἀτρεί̈δης τε ἄναξ ἀνδρῶν καὶ δίος Ἀχιλλεύς.

μῆνιν ἄειδε θεὰ Πηληϊάδεω Ἀχιλῆος οὐλομένην, ἣ μυρί’ Ἀχαῖοι ἀλγε’ ἠθηκε, πολλὰς δ’ ἱφήμους ψυχὰς Ἀϊδί προίασεν ἤρωας, αὐτοὺς δὲ ἐλώρια τεύχε κύνεσιν οἰνονοσί τε πάσι, Διὸς δ’ ἐτελείετο βουλή, ἐξ οὗ δὴ τὰ πρῶτα διαστήτην ἐρίσαντε Ἀτρεί̈δης τε ἄναξ ἀνδρῶν καὶ δίος Ἀχιλλεύς.

Note that for Arabic one cannot use the environment arabic, as \arabic is defined internally by Ξῆ PX. 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 Ξῆ PX and LuaΞῆ PX format) are redefined, but keep a similar behaviour.

- `\selectlanguage{⟨lang⟩}`
- `\foreignlanguage{⟨lang⟩}{⟨…⟩}`
- `\begin{otherlanguage}{⟨lang⟩}…\end{otherlanguage}`
- `\begin{otherlanguage*}{⟨lang⟩}…\end{otherlanguage*}`
- `\begin{hyphenrules}{⟨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 \foreignlanguage{⟨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 Ξῆ PX and LuaΞῆ PX 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{(lang)}: this selects the global font setup and the numbering definitions for the default language.
- \resetdefaultlanguage{(options)}{(lang)} (experimental): completely switches the default language to another one in the middle of a document: this may have adverse effects!
- \normalfontlatin: in an environment where \normalfont has been re-defined 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: Representation of counter as a lower-case letter: 1 = a, 2 = b, etc.
- \latinAlph: Representation of counter as a upper-case letter: 1 = A, 2 = B, etc.

3.4 Setting up alias commands

By means of the macro

\setlanguagealias{(options)}{(language)}{(alias)} ← \setlanguagealias*[variant=austrian]{{german}}{AT}

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* 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 \langle script \rangle font or \langle language \rangle font. For instance, if the default font defined by \setmainfont does not support Greek, then one can define the font used to display Greek with:

\newfontfamily\greekfont[Script=Greek,\ldots]{\fontfamily}\.

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 \langle script \rangle fonts or \langle language \rangle fonts and \langle script \rangle fonttt or \langle language \rangle fonttt, respectively.

Whenever a new language is activated, polyglossia will first check whether a font has been defined for that language or – for languages in non-Latin scripts – for the script it uses. If it is not defined, it will use the currently active font and – in the case of OpenType fonts – will attempt to turn on the appropriate OpenType tags for the script and language used, in case these are available in the font, by means of fontspec’s \addfontfeature. If the current font does not appear to support the script of that language, an error message is displayed.

5 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):
These exceptions, however, apply to all languages. 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.

5.2 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.

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., via variant or spelling), (l) with the left threshold value (e.g., 3), and (r) with the right threshold value (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 two functions: \disablehyphenation and \enablehyphenation. Note that if you select a new language while hyphenation is disabled, it will remain disabled. If you re-enable it, the hyphenation patterns of the currently selected language will be activated.
6 Language-specific options and commands

This section gives a list of all languages for which options and end-user commands are defined. 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:

- 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)
- locale = default\(^5\), mashriq\(^6\), libya, algeria, tunisia, morocco, mauritania

\(^5\)For Egypt, Sudan, Yemen and the Gulf states.
\(^6\)For Iraq, Syria, Jordan, Lebanon and Palestine.
This setting influences the spelling of the month names for the Gregorian calendar, as well as the form of the numerals (unless overridden by the following option).

- **numerals** = mashriq or maghrib
  The latter is the default when locale=algeria, tunisia, or morocco.

- **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).

- **abjadimnotail** ← *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 ٣.

**Commands:**

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

- **\abjadmaghribi** outputs Arabic *abjad* numbers according to the Maghrib varieties. Example: \abjadmaghribi{1863} yields شظصج.

- **\abjadalph** ← steps through the Arabic alphabet, thus it can only be used up to 28. Example: \textarabic{\abjadalph{18}} yields ص. This command is also available for Farsi, Urdu, etc.

- **\aemph** to emphasize text with \overline.

---

**6.3 armenian**

**Options:**

- **variant** ← eastern or western

- **numerals** ← armenian or arabic

---

**6.4 belarusian** ←

**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. 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 alphanumeral, 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 Ф.
\asbuk
- \asbuk: same as \Asbuk but in lowercase.
\AsbukTrad
- \AsbukTrad: same as \Asbuk but using the traditional Cyrillic alphanumerical numbering which supports numbers up to 999 999. E.g., \textbelarusian{\AsbukTrad{page}} produces KA.
\asbukTrad
- \asbukTrad: same as \AsbukTrad but in lowercase.

6.5 bengali ←

Options:
- numerals = Western, Bengali, or Devanagari
- changecounternumbering = *true or false
  Use specified numerals for headings and page numbers.

6.6 catalan

Options:
- babelshorthands ← *true or false
  Activates the shorthands "l and "L to type geminated l or L.

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

6.7 croatian

Options:
- 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.

\footnote{See \url{https://en.wikipedia.org/wiki/Cyrillic_numerals}.}
"=" 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 (ligatures):

"dz" Generates the ligature dž if the font provides it. If not, the two characters are output separately. Also available for "Dz (Dž) and "DZ (DŽ).

"lj" Generates the ligature lj if the font provides it. If not, the two characters are output separately. Also available for "Lj (Lj) and "LJ (LJ).

"nj" Generates the ligature nj if the font provides it. If not, the two characters are output separately. 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 («).

**disableligatures** ← = *true or false

If this is true, all Croatian ligatures (for digraphs such as dž) will be replaced by single characters. This can be useful if the ligatures on your font are broken (if the font does not have them, they are automatically replaced).

**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.

v1.47

v1.53
• **splithyphens** ← = *true* or *false*

According to Croatian 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.

### 6.8 Czech

**Options:**

• **babelshorthands** ← = *true* or *false*

If this is turned on, the following shorthands for Czech are activated:

```
“ = for an explicit hyphen sign which is repeated at the beginning of the next line when hyphenated, as common in Czech typesetting (only needed with splithyphens=false).
‘ ‘ for Czech left double quotes („).
‘ ‘ for Czech right double quotes (”).
“> for Czech left double guillemets (»).
“< for Czech right double guillemets («).
```

• **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.

• **vlna** ← = *true* or *false*

According to Czech typesetting conventions, single-letter words (nonsyllable 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.
6.9  Dutch

Options:

- **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:

  - "- 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 ".-.").

6.10  English

Options:

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

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

6.11  Esperanto

Commands:

- \hodiau and \odiaun 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.
6.12 finnish

Options:

- **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 a hyphenation point that does still allow for hyphenation at the points preset in the hyphenation patterns (as opposed to \-).
  
  - " for a hyphen sign without a breakpoint. Useful for cases where the hyphen should stick at the following syllable.
  
  - "| disables a ligature at this position.
  
  - " " allows for a line break at this position (without hyphenation sign).
  
  - \/ a slash that allows for a subsequent line break. As opposed to \\slash, hyphenation at the breakpoints preset in the hyphenation patterns is still allowed.

6.13 french

Options:

- **variant** ← french or canadian (= acadian) ← 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

  One of the most distinct features of French typography is the addition of extra spacing around punctuation and quotation marks (guillemets). By default, polyglossia adds these spaces automatically, so you don’t need to enter them. This options allows you to switch this feature off globally.

- **thincolonspace** ← *true or false

  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.

- **autospacenguillemets** ← *true or false

  Up to version 1.44, the option was called automaticspacesaroundguillemets. For backwards compatibility reasons, the more verbose old option is still supported.
If you only want to disable the automatic addition of spacing after opening and before closing guillemets (and not at punctuation), set this to `false`. Note that the more general option `autospacing` overrides this.

- **`autospacetypewriter`** ← *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.

- **`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`.

- **`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.

- **`itemlabels`** ← ⟨code⟩ (default value: \textemdash)
  
  If `frenchitemlabels` is true, you can customize here the used item label of all levels.

- **`itemlabeli`** ← ⟨code⟩ (default value: \textemdash)
  
  If `frenchitemlabels` is true, you can customize here the used item label of the first level.

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

- **`itemlabeliii`** ← ⟨code⟩ (default value: \textemdash)
  
  If `frenchitemlabels` is true, you can customize here the used item label of

---

*Babel’s syntax `OriginalTypewriter` is also supported.*
the third level.

- **itemlabeliv** ← \{code\} (default value: \textemdash)
  
  If `frenchitemlabels` is true, you can customize here the used item label of the fourth level.

**Commands:**

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

6.14 gaelic ←

**Options:**

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

6.15 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).

  ", 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.

### 6.16 german

Options:

- variant = german, austrian, or swiss ←
  Setting variant=austrian or variant=swiss uses some lexical variants. With spelling=old, variant=swiss furthermore loads specific hyphenation patterns.

- spelling = new (=1996) or old (=1901)
  Indicates whether hyphenation patterns for traditional (1901) or reformed (1996) orthography should be used. The latter is the default.

- 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).
The command \atticnumeral (= \atticnum) (activated with the option \attic=true), displays numbers using the acrophonic numbering system (defined in the Unicode range U+10140–U+10174).\footnote{See the documentation of the \texttt{xgreek} package for more details.}
6.18 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 מְרָחָשְׁוָן (Marcheshvan) rather than חָשְׁוָן (Heshvan), which is the default.

Commands:

- \hebrewnumeral
- \hebrewalph
- \aemph

6.19 hindi

Options:

- **numerals** = Western or Devanagari

6.20 hungarian

Options:

- **swapstrings** = all, captions, headings, headers, hheaders, or none

In Hungarian, some caption strings need to be in a different order than in other languages (e.g., 1. fejezet instead of Chapter 1). By default, polyglossia tries hard to provide the correct order for different classes and packages (standard classes, KOMA-script, memoir, and titlesec package should work, as well as fancyhdr and caption). However, since the definition of these strings is not standardized, the redefinitions might not work and even interfere badly if you use specific classes or packages that redefine the respective strings themselves. In this case, you can disable some or all changes. The possibilities are:

- **all**: Redefine figure and table captions, part and chapter headings, and running headers (= default setting)
- **captions**: Redefine figure and table captions only
- **headings**: Redefine part and chapter headings only
- **headers**: Redefine running headers only
- **hheaders**: Redefine part and chapter headings as well as running headers
- **none**: Do not redefine anything
Commands:
\today (\ontoday): special form of \today which produces a
slightly different date format as used in prepositional phrases (such as 'on
February 10th') in Hungarian.

6.21 italian

Options:
• 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 \-).

6.22 korean ←

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
  With this option, Korean-style part and chapter headings, and running
  headers are available. It is similar to Hungarian (see 6.20) 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

6.23 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.
• abjad jimnotail = *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 گ.

Commands:
\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 (see section 8.3)
\aemph (see section 6.2)

6.24 lao ←

Options:
• numerals = lao or arabic

6.25 latin

Options:
• variant = classic, medieval, modern, or ecclesiastic ←
  These variants refer to different spelling conventions. The classic and the medieval variant do not use the letters U and V, but only V and u. This concerns predefined terms like month names as well as the behaviour of the \MakeUppercase and the \MakeLowercase command. The medieval
Table 4. Spelling differences between the Latin language variants.
The capitalization of month names and the use of \textit{i/j} may be affected by
the \texttt{capitalizemonth} and the \texttt{usej} option.

\begin{tabular}{lcccc}
\textbf{Month} & \textbf{classic} & \textbf{medieval} & \textbf{modern} & \textbf{ecclesiastic} \\
\hline
Ianuarii & Ianuarii & Ianuarii & ianuarii \\
Novembris & Novembris & Novembris & novembris \\
Praefatio & Praefatio & Praefatio & Praefatio \\
\end{tabular}

\begin{verbatim}
\MakeUppercase{Iulius} yields:
IVLIVS IVLIVS IULIUS IULIUS
\end{verbatim}

Table 5. Latin default hyphenation styles

\begin{tabular}{l|c}
\textbf{Language variant} & \textbf{Default hyphenation style} \\
\hline
classic & classic \\
medieval & modern \\
modern & modern \\
ecclesiastic & modern \\
\end{tabular}

and the ecclesiastic variant use the ligatures \textit{æ} and \textit{œ}. See table 4 for examples.
Furthermore, the ecclesiastic variant takes care for a punctuation spacing similar to French, but with smaller spaces, as provided for PDF\TeX \ by the ecclesiastic package.

\begin{itemize}
\item \texttt{hyphenation} ← classic, modern, or liturgical
\item \texttt{ecclesiasticfootnotes} ← true or false
\end{itemize}

There are three different sets of hyphenation patterns for Latin. Separate documentation for them is available on the Internet.\textsuperscript{11} Each of the four variants mentioned above has its default set of hyphenation patterns as indicated by table 5. Use the \texttt{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 \texttt{hyphenation=liturgical} in any case.

\textsuperscript{11} \url{https://github.com/gregorio-project/hyphen-la/blob/master/doc/README.md#hyphenation-styles}

11
notes 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.

- **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.*

- **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, âé, and òé
  - `'A` for Á (A with acute), also available for Ê, Î, Ô, Ù, Ý, ÀÉ, and ÓÉ
  
  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 ÒÉ
  - "AE` for Æ (AE ligature), also available for ÒÉ
  - `‘ae` for áé (ae ligature with acute), also available for œé
 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 ų.
- `=A` for Ā (A with macron), also available for Ė, 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 aë (ae diphthong with macron), also available for ēu, ēu, and ẽ. Note that macrons above diphthongs are only displayed if your font supports the Unicode character 035E (`combining double macron`).
- `=Ae` for Aē (Ae diphthong with macron), also available for Ėu, Ėu, and Oē.
- `=AE` for AĒ (AE diphthong with macron), also available for AĒ, EU, and OĒ.
- `^a` for ă (a with breve), also available for ē, ī, ō, ū, and ų. Note that a breve above the letter y is only displayed if your font supports the Unicode character 0306 (`combining breve`).
- `^A` for A (A with breve), also available for Ė, Ī, Ō, Ū, V̄, and Ȳ. Note that breves above the letters V and Y are only displayed if your font supports the Unicode character 0306 (`combining breve`).

### 6.26 malay

**Options:**

- `variant ←= indonesian` or `malaysian`
6.27 marathi

Options:

‣ **numerals** = Devanagari or Western

6.28 mongolian ←

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 telescop-
  ing 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 op-
  posed to "--- this removes any space before and after the dash.

36
—• 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 alphanumerical system.\(^{12}\) It supports numbers up to 999,999.

**Commands:**

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

\texttt{\textasciitilde asbuk}: same as \texttt{\textasciitilde Asbuk} but in lowercase.

\texttt{\textasciitilde AsbukTrad}: same as \texttt{\textasciitilde Asbuk} but using the traditional Cyrillic alphanumerical numbering which supports numbers up to 999,999. \textit{E.g.}, \texttt{\textasciitilde AsbukTrad\{section\}} produces \texttt{S}.

\texttt{\textasciitilde asbukTrad}: same as \texttt{\textasciitilde AsbukTrad} but in lowercase.

6.29 **norwegian**

Options:

• **variant = bokmal or nynorsk**

6.30 **persian**

Options:

• **numerals = western or eastern**

• **abjadjimmotail = *true or false**
  
  Set this to \texttt{true} if you want the \textit{abjad} form of the number three to be \texttt{ج} – as in the manuscript tradition – instead of the modern usage \texttt{چ}.

**Commands:**

\texttt{\textasciitilde abjad} (see section 8.3)

\texttt{\textasciitilde aemph} (see section 6.2).

6.31  portuguese

Options:

- **variant** ← *brazilian or portuguese*

6.32  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 telescop-
ing 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 op-
    posed 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.

- **forceheadingpunctuation** ← *true or false*
  
  By default, chapter and section numbers always have a trailing punctu-
  ation 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.
• **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{}.

• **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.\(^\text{13}\) It supports numbers up to 999 999.

**Commands:**

\textbf{\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.*, \texttrussian{\Asbuk{section}} produces Е.

\textbf{\asbuk}

• \asbuk: same as \Asbuk but in lowercase.

\textbf{\AsbukTrad}

• \AsbukTrad: same as \Asbuk but using the traditional Cyrillic alphanumerical numbering which supports numbers up to 999 999.

  *E.g.*, \texttrussian{\AsbukTrad{page}} produces ЛѲ.

• \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{tf}`, `\textsf{sf}`, `\textit{it}`, `\texttt{tt}`, etc. The macros `\Prob`, `\Variance`, `\NOD`, `\nod`, `\NOK`, `\nok`, `\Proj` print some rare Russian mathematical symbols.

6.33 sami
Currently support for Sami is limited to Northern Sami.

6.34 sanskrit
Options:
  - **script** ← Devanagari, Gujarati, Malayalam, Bengali, Kannada, Telugu, or Latin
    The value is passed to `fontspec` in cases where the respective `(script)` font
    is not defined. This can be useful if you typeset Sanskrit texts in scripts
    other than Devanagari.
  - **numerals** ← Devanagari or Western

6.35 serbian
Options:
  - **script** = Cyrillic or Latin
  - **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:
\*`\textserbian{\Asbuk{section}}`* produces E.
\*\*`\textserbian{\AsbukTrad{page}}`* produces M.

6.36  slovak

Options:

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

  `|` disables a ligature at this position.

  `-` for a hyphen sign without a breakpoint. Useful for cases where the hyphen should stick at the following syllable.

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

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

  `/` a slash that allows for a subsequent line break. As opposed to `\slash`, hyphenation at the breakpoints preset in the hyphenation patterns is still allowed.

  `\` for Slovak left double quotes (looks like „).

  `\` for Slovak right double quotes (looks like “).

  `>\` for Slovak left double guillemets (looks like »).

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

- **splithyphens** ← *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 LuaLaTeX, the `luavlna` package is loaded to achieve this). Set this option to `false` to disable the feature.

- **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 LuaLaTeX, the `luavlna` package is loaded to achieve this). Set this option to `false` to disable the feature.
6.37 slovenian

Options:

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

6.38 sorbian

Options:

- `variant ← lower or upper`
- `olddate ← *true or false`

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.

6.39 spanish

Options:

- `variant ← spanish or mexican`
- `spanishoperators ← *all, accented, spaced, none, or false`

Determines of and how math operators are localized to Spanish.

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

Commands:

- `\arcsen`: 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 \textit{cotg} in math. The optional argument of the command lets you specify the spelling, if needed, \textit{e.g.}, \spanishoperator[arc\,ctg]{arcctg}.

6.40 syriac

Options:
- \textit{numerals} ← = western (\textit{i.e.}, 1234567890), eastern (for which the Oriental Arabic numerals are used: ٠٩٨٧٦٥٤٣٢١), or abjad

Commands:
- \abjadsyriac (see section 8.3)
- \aemph (see section 6.2).

6.41 thai

Options:
- \textit{numerals} = thai or arabic

To insert word breaks, you need to use an external processor. See the documentation to \texttt{thai-late x} and the file \texttt{testthai.tex} that comes with this package.

6.42 tibetan

Options:
- \textit{numerals} = tibetan or arabic

6.43 ukrainian

Options:
- \textit{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 \texttt{-}).
- "= adds an explicit hyphen with a breakpoint, allowing for hyphenation at the other points preset in the hyphenation patterns (as opposed to plain \texttt{-}).
"- 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.

\begin{itemize}
  \item \texttt{mathfunctions} \leftarrow \ast \texttt{true or false}
  \begin{itemize}
    \item By default, some specific math macros are defined for Ukrainian (see below). In order to prevent command clashes (e.g., with the \texttt{chemformula} package), you can switch these definitions off by passing \texttt{false} to this option.
  \end{itemize}
  \item \texttt{numerals} = \texttt{arabic}, \texttt{cyrillic-alph} or \texttt{cyrillic-trad}
  \begin{itemize}
    \item Uses either Arabic numerals or Cyrillic alphanumerical numbering. The two Cyrillic variants differ as follows:
      \begin{itemize}
        \item \texttt{cyrillic-alph} steps through the Cyrillic alphabet. Thus it can only be used up to 30.
        \item \texttt{cyrillic-trad} (= \texttt{cyrillic}) uses a traditional Cyrillic alphanumerical system.\footnote{See \url{https://en.wikipedia.org/wiki/Cyrillic_numerals}.} It supports numbers up to 999 999.
      \end{itemize}
  \end{itemize}
\end{itemize}
meric numbering which supports numbers up to 999 999.
E.g., \textukrainian{\AsbukTrad{page}} produces МД.

\asbukTrad: same as \AsbukTrad but in lowercase.

If the \texttt{mathfunctions} option is true, loading Ukrainian defines a few macros
that 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, \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.

6.44 uyghur

Options:

- \texttt{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 \texttt{swapstrings} is set to all, headings or
hheaders. The caveats for that options (see below) apply.

- \texttt{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 \texttt{titlesec} package should work, as well as \texttt{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 speci-
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}}).

6.45 welsh

Options:
• date = long or short

7 Modifying or extending captions, date formats and language settings

Polyglossia uses the following macros to define language-specific captions (i.e., strings such as “chapter”), date formats and additional language settings (\lang is to be replaces with the respective language name):
\captions{\lang}
• \captions{\lang} stores definitions of caption strings (such as, in the case of English, \def\chaptername{Chapter})
\date{\lang}
• \date{\lang} stores definitions of date formats (usually redefinitions of \today, in some cases also definitions of additional date commands)
\blockextras{\lang}
• \blockextras{\lang} stores macros that are to be executed when the language (\lang) is activated via \selectlanguage command or the (\lang) environment
\inlineextras{\lang}
• \inlineextras{\lang} stores macros that are to be executed when the language (\lang) is activated locally via \text{\lang} command
\noextras{\lang}
• \noextras{\lang} stores macros that are to be executed when the language (\lang) is closed

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

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

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

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

- \captions@bbl@⟨babelname⟩
- \date@bbl@⟨babelname⟩
- \blockextras@bbl@⟨babelname⟩
- \inlineextras@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` 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`, `\bbl@⟨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, \texttt{polyglossia} provides language options which allow you to select or switch to the suitable convention. With the appropriate language option set, \texttt{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.

\section{General localization of numbering}

As of 1.45, \texttt{polyglossia} provides a generic macro \texttt{\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 \texttt{\localnumeral{42}} yields \texttt{٢٤}, whereas in an Hebrew environment, it results in \texttt{מב} with \texttt{numerals=hebrew}, and \texttt{42} with \texttt{numerals=arabic}. Note that, as opposed to the various \texttt{digits} macros (described in sec. 8.2), the argument of \texttt{\localnumeral} must consist of numbers only.

For the conversion of counters, the starred version \texttt{\localnumeral*} is provided. This takes a counter as argument. For instance in an Arabic environment \texttt{\localnumeral*[page]} yields \texttt{ףי}.

\texttt{\Localnumeral} and \texttt{\Localnumeral*} provide the uppercased versions.

All these macros provide the following options:

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

\section{Non-Western decimal digits}

In addition to the generic macros described above, \texttt{polyglossia} provides language-specific conversion macros which can be used if the generic ones do
not suit the need. The macros have the form \(\text{\langle script\rangle digits}\). They convert Arabic numerical input and leave every other input untouched. In an Arabic context, for instance, \(\text{\arabicdigits}\{9182/738543-X\}\) yields ٩١٨٢/٧٣٨٥٤٣-X.

Currently, the following macros are provided:

- \texttt{\arabicdigits}
- \texttt{\bengalidigits}
- \texttt{\devanagaridigits}
- \texttt{\farsidigits}
- \texttt{\kannadadigits}
- \texttt{\khmerdigits}
- \texttt{\laodigits}
- \texttt{\kodigits}
- \texttt{\thaidigits}
- \texttt{\tibetandigits}

### 8.3 Non-Latin alphabetic numbering

For languages which use special (non-Latin) alphanumerical notation, dedicated macros are provided.

They work in a similar way than the \(\text{\langle script\rangle digits}\) 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 \texttt{\localnumeral} and \texttt{\Localnumeral} in the respective context).

The following macros are provided:

- \texttt{\abjad} outputs Arabic \textit{abjad} numbers according to the Mashriq varieties. Example: \texttt{\abjad}\{1863\} yields \textit{غضسج}.
- \texttt{\abjadmaghribi} outputs Arabic \textit{abjad} numbers according to the Maghrib varieties. Example: \texttt{\abjad\text{maghribi}}\{1863\} yields \textit{شظصج}.
- \texttt{\abjadsyriac} outputs Syriac abjad numerals. A fine guide to numerals in Syriac can be found at [http://www.garzo.co.uk/documents/syriac-numerals.pdf](http://www.garzo.co.uk/documents/syriac-numerals.pdf).

\footnote{A third method are so-called TECKit fontmappings. Those can be activated with the \texttt{fontspec} Mapping option, using \texttt{arabicdigits}, \texttt{farsidigits} or \texttt{thaidigits}. For instance if \texttt{\arabicfont} is defined with the option \texttt{Mapping=arabicdigits}, typing \texttt{\textarabic}\{2010\} results in \textit{٠١٠٢}. 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.}


\footnote{A fine guide to numerals in Syriac can be found at [http://www.garzo.co.uk/documents/syriac-numerals.pdf](http://www.garzo.co.uk/documents/syriac-numerals.pdf).}
\armeniannumeral \armeniannumeral produces Armenian alphabetic numbering. Example: \armeniannumeral{1863} yields ՌՊԿԳ.

\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.
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 ў.

\georgiannumeral \georgiannumeral produces Georgian alphabetic numbering.
Example: \georgiannumeral{1863} yields ჩყჲგ.

\greeknumeral \greeknumeral produces Greek alphabetic numbering, \Greeknumeral outputs uppercased variants. Example: \greeknumeral{1863} yields ͵αωξγʹ, \Greeknumeral{1863} results in ᾱΩΞΓʹ.

\hebrewnumeral, \Hebrewnumeral and \Hebrewnumeralfinal generate variants of Hebrew alphanumeric numerals. The commands behave exactly as they do in babel: \hebrewnumeral outputs the numbers without any decoration, \Hebrewnumeral adds gereshayim before the last letter, \Hebrewnumeralfinal uses in addition the final forms of Hebrew letters. Examples: \hebrewnumeral{1750} yields י״תשנ, \Hebrewnumeral{1750} yields י״תש״נ, and \Hebrewnumeralfinal{1750} yields י״תש״ן.

\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 ў.

\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 ў.

\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 alphanumerical output.
Example: With \serbiannumeral{19} yields 19, with \serbiannumeral{19} results in й.
with \serbiannumeral{19} results in у.

\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 alphanumerical output.
Example: With \ukrainiannumeral{19} yields 19, with \ukrainiannumeral{19} results in й,
with \ukrainiannumeral{19} results in y.

9 Footnotes in right-to-left context

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

9.1 Horizontal footnote position

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

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

The default placement of the footnote rule differs in \TeX and \luatex output (this is due to differences in the bidi and luabidi packages). With \TeX, footnote rules are always placed left, which is often wrong in RTL context. With \luatex, by contrast, the rule is placed always right if the main language is a right-to-left language, and always left if the main language is a left-to-right language, which is the right thing in many cases.

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

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

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

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

10 Calendars

10.1 Hebrew calendar (hebrewcal.sty)

The package \hebrewcal.sty is almost a verbatim copy of \hebcal.sty that comes with babel. The command \Hebrewtoday formats the current date in the Hebrew 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.\footnote{It makes use of the arithmetical algorithm in chapter 6 of Reingold & Gershowitz, \textit{Calendrical calculation: the Millenium edition} (Cambridge University Press, 2001).} It provides two macros for the end-user. The command
\HijriFromGregorian \Hijritoday \v1.1.1

\DefineHijriDateFormat \v1.47

\charifavailable \v1.47

\HijriFromGregorian{{\textlangle year\rangle}}{{\textlangle month\rangle}}{{\textlangle day\rangle}}
\Hijritoday

\DefineHijriDateFormat{{\textlangle lang\rangle}}{{\textlangle code\rangle}}.

The command \Hijritoday also accepts an optional argument to add or subtract a correction (in days) to the date computed by the arithmetical algorithm.\footnote{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.} For instance if \Hijritoday yields the date “7 Rajab 1429” (which is the date that was displayed on the front page of \texttt{aljazeera.net} on 11th July 2008), \Hijritoday[1] would rather print “8 Rajab 1429” (the date indicated the same day on the site \texttt{gulfnews.com}).

10.3 Farsi (jalālī) calendar (farsical.sty)

This package is an almost verbatim copy of Arabiftoday.sty (in the Arabi package), itself a slight modification of ftoday.sty in Farsi\TeX.\footnote{One day we may rewrite farsical from scratch using the algorithm in Reingold & Gershowitz (ref. n. 19).} Here we have renamed the command \ftoday to \Jalalitoday. Example: today is 23 Farvardin 1400.

11 Auxiliary commands

The macro

\charifavailable \v1.47

\charifavailable{{\textlangle char code\rangle}}{{\textlangle substitution\rangle}}
\charifavailable checks whether the character with the specified \textlangle char code\rangle (i.e., unicode utf-16 code without preceding 0x) exists in the current font. If so, the character is printed, if not, the \textlangle substitution\rangle 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.
12 Accessing language information

The following is specifically relevant to package authors who need information about the languages in use. In order to get such information, polyglossia provides the following macros:

- \languagename stores the currently active (polyglossia) language name.
- \mainlanguagename stores the (polyglossia) language name of the main document language.
- \languagevariant stores the language variant if set. The macro is empty if no variant has been set.
- \mainlanguagevariant stores the language variant of the main document language if set. The macro is empty if no variant has been set.
- \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 analogously stores the name of document’s main language (variant) in babel.
- \languageid{⟨type⟩} \← stores the identifier tag of the current language. Currently supported ⟨types⟩:
  - bcp-47 (alias bcp47): IETF BCP-47 language identifier
- \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 stores a comma-separated list of all loaded languages (polyglossia name)
- \xpg@vloaded stores a comma-separated list of all loaded variants
- \xpg@bloaded stores a comma-separated list of babel names of all language variants
- \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{⟨lang⟩}{⟨true⟩}{⟨false⟩}

where (lang) is a polyglossia language name, by
\texttt{\ifbabellanguageloaded}\langle\text{lang}\rangle\langle\text{true}\rangle\langle\text{false}\rangle\texttt{\iflanguageidloaded}\langle\text{type}\rangle\langle\text{id}\rangle\langle\text{true}\rangle\langle\text{false}\rangle\texttt{\iflanguageoption}\langle\text{lang}\rangle\langle\text{opt. key}\rangle\langle\text{opt. value}\rangle\langle\text{true}\rangle\langle\text{false}\rangle\texttt{\fi}}\texttt{\fi}}$

where \langle\text{lang}\rangle\texttt{\fi} is a babel language name (see table 2 on p. 7), or by
\texttt{\iflanguageidloaded}\langle\text{type}\rangle\langle\text{id}\rangle\langle\text{true}\rangle\langle\text{false}\rangle\texttt{\fi}←\texttt{\iflanguageoption}\langle\text{lang}\rangle\langle\text{opt. key}\rangle\langle\text{opt. value}\rangle\langle\text{true}\rangle\langle\text{false}\rangle\texttt{\fi}}$

where \langle\text{type}\rangle\texttt{\fi} is a supported language id type (such as bcp-47) and \langle\text{id}\rangle\texttt{\fi} 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
\texttt{\iflanguageoption}\langle\text{lang}\rangle\langle\text{opt. key}\rangle\langle\text{opt. value}\rangle\langle\text{true}\rangle\langle\text{false}\rangle\texttt{\fi}}\texttt{\fi}}

13 Revision history

1.53 (12-04-2021)

New features

- New option \texttt{localalph} for Croatian (#486).
- Use ordinals in part and chapter headings. A new option \texttt{chapterformat} has been added to disable this or change format to Roman numbering (#479).
- New command \texttt{uyghurord} and \texttt{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 \texttt{swapstrings} option (#479).
- Add overlooked digraphs in Croatian (#484).
- Fix typo in Serbian \texttt{theprt} (#485).
- Fix part modifications in Hungarian with \texttt{hyperref} (#493).

1.52 (16-03-2021)

New features

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

Bug fixes

- Fix French part modifications with \texttt{hyperref} (#469).
- Fix markup of French \texttt{\see} and \texttt{\alsoname} (#468).
1.51 (08-12-2020)

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 (15-10-2020)

Bug fixes

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

1.50 (09-10-2020)

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 babel\name 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 a 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 (08-04-2020)**

**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 (25-03-2020)

- 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 \texttt{[no]localmarks} option, whose logic was swapped (part of #395).
- Protect \texttt{localmarks} function against uppercased language names (part of #395).
- Fix buggy redefinition of \texttt{@markright} with option \texttt{localmarks} (#396).
- Fix incompatibility between Latin and unicode-math (#394).
- Make (undocumented) \texttt{defineshorthand} command (imported from babel) work.
- Fix usage of \texttt{localmarks} option without value.
- Emergency fixes for bugs caused by updates in \texttt{babel’s switch.def} (#399 and #400).

Interface and defaults changes

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

1.47 (29-01-2020)

New features

- IETF BCP-47 compliant language tags can now be used for loading and switching languages alternatively to language names (#226).
- New commands \texttt{\languageid{<type>}} and \texttt{\mainlanguageid{<type>}}.
- New test \texttt{\iflanguageidloaded}.
- New list \texttt{\xpg@bcp@loaded}.
- New environment \texttt{\{lang\}{<lang>}} (this is equivalent to \texttt{\{lang\}, but also available with \texttt{\setlanguagealias*} which does not define dedicated alias environments).
- New gloss option totalhyphenmin (corresponds to LuaTeX’s \texttt{\hyphenationmin}) (#111).
- New test \texttt{\iflanguageoption{<lang>}{<key>}{<val>}} (#364).
- Restore simple alphabetic numbering for \texttt{\asbuk} and \texttt{\Asbuk} in Belarusian, Mongolian, Russian, Serbian, and Ukrainian (#377).
- New command \texttt{\AsbukTrad} and \texttt{\asbukTrad} for Belarusian, Mongolian, Russian, Serbian, and Ukrainian which uses traditional alphanumerical numbering.
- New numerals option \texttt{cyrillic-trad} and \texttt{cyrillic-alph} to differentiate simple alphabetic and traditional alphanumerical Cyrillic numbering.
- \texttt{\selectbackgroundlanguage} and \texttt{\resetdefaultlanguage} now also support language aliases.
- New macro \texttt{\charifavailable{<char code>}{<substitution>}}.
- Add French language variant \texttt{swiss}.
- Implement \texttt{babelshorthands} for Croatian.
- Implement \texttt{\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).
- 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).
  - latesthyphen in language German (latesthyphen now equals latesthyphen=true).
  - fullyear in package hebrewcal (fullyear now equals fullyear=true).
  - The command \setlanguagealias* (introduced in v1.46) does no longer define dedicated alias environments.

- The babelnames for Latin variants have been corrected 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 (15-11-2019)

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 (⑪112, ⑪132).
  • Add optional localized math operators to Spanish (⑪123).
  • Swap section headings in Hungarian (⑪344). New option `swapstrings` provides control over this.
  • Introduce macro `\setlanguagealias` and `\setlanguagealias*`.
  • Introduce language switching command `\textlang{lang}{...}` (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 (⑪24).
• Fix clash of French punctuation spacing with the **soul** package (⑪52).
• Re-enable the possibility to pass a macro as main argument to `\setmainlanguage` and `\setotherlanguage` (⑪331).
• Fix detection of default `\languagevariant` (⑪332).
• Fix **LaTeX** error with undefined hyphenation pattern (⑪346).
• Fix some babel shorthand issues by updating the shorthand code from recent **babel**.
• Fix some problems with French and Latin auto-spacing (⑪345, ⑪352).
• Fix an **expl3** declaration (⑪348).

**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 `vlna` 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. ⑪345).
1.45 (27-10-2019)

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, \mainlanguagevariant, \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 vina 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 bahasan (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 \ynorsk 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 autospaning and commands \AutoSpacing, \NoAutoSpacing This allows to switch off autospaning 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 \texttt{autospacetyperwriter} alias \texttt{OriginalTypewriter}.
• Support \texttt{\LaTeXe} with lualatex
• Rename \texttt{autospacesaroundguillemets} to \texttt{autospacenguillemets} The old option is kept for backwards compatibility.

Bug fixes

• Fix equation number in Arabic and Farsi (#7).
• Simplify and document Hebrew \texttt{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 \texttt{\disablehyphenation} in preamble until after setting of document language (#125).
• Postpone the assignment of defaultfamily to \texttt{\AtBeginDocument}, thus do not overwrite \texttt{\familydefault} redefinitions in the preamble (#127).
• Reset number settings when switching language (#133).
• Hebrew: Properly store \texttt{\MakeUppercase} for later restoration (#152).
• Fix whitespace issue in \texttt{\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 \texttt{\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 \texttt{\blockextrasp} and undef them in \texttt{\noextrasp} (#209).
• Ensure proper direction with arabic digits in Arabic and Farsi (#213).
• Fix \texttt{\linespread} with Korean (#218).
• Define Russian caption before key allocation (#219).
• Register current language in polyglossia lua module after selection (#234).
• Fix \texttt{babel} language switching commands (#239): \texttt{\foreignlanguage} and the starred \texttt{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).
• \texttt{gloss-russian}:
  • Check whether command exist before redefining (#280).
  • Fix some whitespace issues.
• Fix and simplify \texttt{\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 (04-04-2019)

• Correction to Russian language file, by Maksim Zhouludev (commit d2f5383e).
• Added Macedonian language file, by Stefan Zlatinov (commit cd379e1).

1.43 (05-03-2019)

• Correction to Hindi language file, by Zdenĕk Wagner.

1.42.5 (13-04-2017)

• Many changes to the French language file, by Maiul Rouquette.

1.42.4 (February, March 2016)

• Remedial actions for the babel changes.
• Fixed side effect of pull request #122 (see commit d2a34f4f).
• 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 (06-08-2015)**
- 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 (16-07-2015)**
- Added support for Khmer, by Say Ol (private email)

**1.40.1 (14-07-2015)**
- Bugfix for Korean, by Dohyun Kim (pull request #103)

**1.40.0 (07-07-2015)**
- gloss-korean.ldf contributed by Dohyun Kim (pull request #102)

**1.33.7 (04-07-2015)**
- 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 (15-05-2015)**
- 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 (21-05-2014)
- 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 (27-06-2013)
- Emergency release for a bug introduced in babelsh.def

1.33.3 (28-05-2013)
- Changed formatting of some error messages (emergency fixes for TeX Live 2013)

1.33.2 (26-05-2013)
- 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 (23-05-2013)
- Editorial changes to the documentation
- Hunted and documented bugs

1.33.0 (20-05-2013)
- Added support for N’Ko.
- Bugfixes for LuaTeX
- More work in progress on Bidi in LuaTeX.

1.32.0 (15-05-2013)
- Transitional version to support right-to-left languages with LuaTeX.

1.31 (10-05-2013) / 1.3 (11-05-2013)
- 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 (06-08-2012)
- Added support for LuaTeX. Many languages don’t work yet. Please be patient.

1.2.0e (28-04-2012)
- Fixed a number of outstanding and not very interesting bugs.
- Added gloss files for Romansh and Friulan, contributed by Claudio Beccari.

1.2.0d (12-01-2012)
- Removed \makeatletter and \makeother from gloss files entirely.

1.2.0c (12-10-2011) [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:
  - 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 (03-10-2011) [Update by Philipp Stephani]
- Load xkeyval package explicitly since newer versions of fontspec don’t load it any more.
1.2.0a (27-07-2010) [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 (15-07-2010)

- Adapted for fontspec 2.0 (will not work with earlier versions!)
- New 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 \foreignlanguage of \PolyglossiaSetup interface that simplifies writing gloss-*\.ldf files
- 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 Bahasaai, Lithuanian, Russian, Croatian
- Add option indentfirst=true for Spanish, Croatian and other languages (NB: indentfirst was previously named frenchindent)
- New option script for German: Setting script=fraktur modifies the captions for type-setting 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 (23-03-2010)

- 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 (22-11-2009)
• Modify \hyphenmins for Sanskrit (Yves Codet)
• Bugfixes for Serbian and Bulgarian (Enrico Gregorio)

1.1.0a (22-11-2009)
• Bugfix for interchar tokens

1.1.0 (20-11-2009)
• 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. O. 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 \xp@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 tdf file for brazilian
Various improvements in the code communicated by E. Gregorio:
- remove superfluous \protect\language
- change default language from 0 to \@nohyphenation=255
- localize \lccode handling of apostrophe in French; add it to Italian
- Fix frenchspacing for Vietnamese
- Other minor bugfixes

1.0.2 (27-01-2009)
- 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 (31-07-2008)
- 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 (13-07-2008)
- 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 Jafari (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), Mikhail Oren (Hebrew), Sergey Astanin (Russian), Khaled Hosny (Arabic), Sertaç Ö. Yıldız (Turkish), Kamal Abdali (Urdu), and several other members of the XeLaTeX user community, notably Enrico Gregorio, who has sent me many useful suggestions and corrections and contributed the \newXeTeXintercharclass mech-
anism 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, Wouter Bolsterlee, Christian Buhtz, Zagbul Andrey, Oleg Domanov, Philipp Gesang, Kevin Godby, Enrico Gregorio, Khaled Hosny, Najib Idrissi, user julroy67, Dohyun Kim, Phil Kime, Mike Kroutikov, Ivan Kokan, Caleb MacLennan, José Mancera, Miquel Ortega, Yevgen Pogribnyi, Will Robertson, Maëul Rouquette, Elie Roux, Hugo Roy, Guy Rutenberg, Philipp Stephani, Niranjan Tambe, Osman Tursun, Keno Wehr, Dominik Wujastyk, Sertaç Ö. Yildiz, Maksim Zhuleudev, 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!