The babelbidi Package

Amer Amikhteh

Version 0.5.0
November 1, 2026

Abstract

babelbidi extends Babel's bidirectional support by fixing direction-sensitive behaviour and providing interfaces for explicit text direction, language selection, and font configuration, including glyph fallback with LuaTeX. It supports languages available through Babel and works with LuaTeX (recommended), XeTeX, or pdfTeX, with either an LTR or RTL main language.

1 Introduction

Babel provides the infrastructure for typesetting documents containing both left-to-right (LTR) and right-to-left (RTL) languages. Some layout components, however, do not always behave correctly in RTL contexts, and Babel does not provide all the basic direction interfaces that may be useful in a document.

babelbidi addresses these gaps in two ways:

1. it patches direction-sensitive layout behaviour
2. and provides a small set of bidi and language-selection APIs.

Most layout patches concern LuaLaTeX. For documents whose main language does not use the Latin script, code-related direction handling is provided with LuaLaTeX, XeLaTeX, and pdfLaTeX.

The basic direction commands follow interfaces found in bidi implementations such as luabidi. The commands `\lr`, `\rl`, and the `latin` environment are inspired in part by interfaces familiar to users of xepersian.

2 Obtaining and Installing

The official version of the `babelbidi` package is available from CTAN:

https://ctan.org/pkg/babelbidi

Once the TeX Live package database has been updated, the package can be installed with the following command:

```
tlmgr install babelbidi
```

3 Loading

The package is loaded as follows:

```
\usepackage[<options>]{babelbidi}
```

The package accepts options for both Babel and babelbidi. Options not defined by babelbidi are passed to Babel. The `bidi` and `layout` options are Babel options for which babelbidi provides engine-dependent defaults.

If `bidi` is not specified, babelbidi selects its default value according to the engine:

Engine	Default value of `bidi`
LuaTeX	`basic`
XeTeX	`bidi-r`
PdfTeX	`default`

With LuaTeX, babelbidi selects the following default value for `layout`:

```
counters graphics footnotes lists
```

Specifying `layout` replaces this default; it does not add to it. For example:

```
layout={counters footnotes}
```

This selects only `counters` and `footnotes` for the `layout` option. To retain `graphics` and `lists`, include them explicitly in the value. When the value contains multiple selectors, the value must be enclosed in **braces**.

For best results, consider the following recommendations:

1. For slides with an RTL main language, beamer-rl is recommended over beamer, though it is not required.

2. LuaLaTeX is recommended for its broader capabilities and Lua[1] extensibility. Separately, Babel's `onchar` option can automatically select a language based on the characters in the text. For Arabic:

   ```
   \babelprovide[import, mapdigits,
    ↪  onchar=ids fonts letters]{arabic}
   ```

   Punctuation such as parentheses may still require explicit language selection.

With LuaTeX, the `fontcache` option controls caching of font mappings used by Babel's `onchar=fonts` option. It is a boolean option and is enabled by default to improve compilation speed. In special cases, such as font changes made through the `rmfamily` hook, cached mappings may prevent these changes from taking effect. The cache can be disabled when necessary:

[1] Lua is a lightweight programming language designed to be embedded in applications.

```
\usepackage[fontcache=false]{babelbidi}
```

4 Font fallback

With LuaLaTeX, babelbidi can use fonts from other Babel languages when the current font lacks a required glyph. Declare an ordered fallback chain in the preamble:

```
\fontfallback{english}{persian,greek}
```

The source and candidate languages must be known to Babel. They can be specified as package options or loaded explicitly with `\babelprovide`. Their fonts must be declared with `\babelfont`. Candidates are tried in order, while the text language, direction, font family, series, shape, and size are preserved. Fallback applies to text fonts, including custom families, but not to math fonts. It works with or without `onchar=fonts` and also uses the `fontcache` setting. An explicit `\latinttfamily` selection uses the selected Latin language's fallback chain without changing the text language or direction.

A later declaration replaces the previous chain; an empty candidate list disables fallback:

```
\fontfallback{english}{}
```

Unknown languages, repeated candidates, and self-references are ignored with a warning where appropriate. Fallback is not recursive and is applied per glyph before shaping. It is available only with LuaLaTeX; XeLaTeX and pdfLaTeX warn and ignore explicit fallback declarations.

5 Direction APIs

babelbidi provides commands and environments for controlling text direction without switching the language context, matching the traditional bidi and luabidi approach.

5.1 Commands

Direction commands without arguments

Command	Description
`\beginL`, `\endL`	Begin and end an explicit LTR text group
`\beginR`, `\endR`	Begin and end an explicit RTL text group
`\setRTL`, `\setRL`, `\unsetLTR`	Set paragraph direction to RTL
`\setLTR`, `\setLR`, `\unsetRTL`	Set paragraph direction to LTR

Footnote-rule controls

Command	Description
\leftfootnoterule	Left-aligned footnote rule
\rightfootnoterule	Right-aligned footnote rule
\textwidthfootnoterule	Full-width footnote rule
\footnoterulewidth	Length controlling the width of short footnote rules
\autofootnoterule	Automatically selects the rule alignment based on footnote direction

Commands taking text argument: {text}

Command	Description
\LRE, \LR	Inline LTR embedding
\RLE, \RL	Inline RTL embedding

Footnote commands: [num]{text}

Command	Description
\LTRfootnote	Typeset footnote in LTR direction
\RTLfootnote	Typeset footnote in RTL direction

Directional boxes

Command	Description
\hboxL, \hboxR	Horizontal box with LTR or RTL direction
\vboxL, \vboxR	Vertical box with LTR or RTL direction
\vtopL, \vtopR	Top-aligned vertical box with LTR or RTL direction

The commands \hbox, \vbox, and \vtop construct boxes containing the material enclosed in braces. An \hbox arranges its contents horizontally, without line breaking. A \vbox arranges its contents vertically; text inside it is broken into lines according to \hsize. A \vtop works like \vbox, but aligns with surrounding material at its top rather than its bottom: for boxes containing lines of text, the reference point is on the first line's baseline for \vtop and the last line's baseline for \vbox.

The directional variants listed above set the box direction to LTR or RTL without changing the language. Before the opening brace, an optional to <dimension> specifies the final width of a horizontal box or height of a vertical box; spread <dimension> adds the given amount to its natural width or height. These sizes are achieved by adjusting stretchable or shrinkable space within the box. A box can also be stored with \setbox and later inserted with \box.

5.2 Environments

Environment	Description
RTL	Typesets its contents from right to left
LTR	Typesets its contents from left to right

6 Language and Font APIs

babelbidi provides commands, environments, and package options for selecting languages and setting fonts.

6.1 Font Selection

When running under XeLaTeX or LuaLaTeX, babelbidi provides the following commands:

- \settextfont[⟨options⟩]{⟨font⟩}: Sets the rm, sf, and tt fonts for the main language.
- \setlatintextfont[⟨options⟩]{⟨font⟩}: Sets the rm, sf, and tt fonts for the selected Latin language.
- \deflatinfont{⟨command⟩}[⟨options⟩]{⟨font⟩}: Defines a Latin font family command.
- \setlatinsansfont[⟨options⟩]{⟨font⟩}: Sets the sans-serif font for the selected Latin language.
- \setlatinmonofont[⟨options⟩]{⟨font⟩}: Sets the monospaced font for the selected Latin language and configures it for use by \latinttfamily.
- \setmonofont[⟨options⟩]{⟨font⟩}: In the preamble, also sets Babel's monospaced font for the main language, so it is available to font fallback.

6.2 Commands

These commands provide language selection, Latin-script dates, and Latin monospaced or verbatim text. The commands \lr and \rl switch the active language context in addition to setting the script direction, and are not part of the standard bidi package.

Command	Arguments	Description
\lr	{text}	Typeset text using LTR language
\rl	{text}	Typeset text using RTL language
\latintoday	–	Print current date in Latin script
\latinttfamily	–	Select Latin monospaced font family
\latinverb	verb	Typeset verbatim code in Latin
\latinverb *	verb	Typeset verbatim with visible spaces

6.3 Environments

Environment	Description
`latin`	Typesets its contents in the selected Latin language
`latinverbatim`	Typesets a verbatim block in the `code` language
`latinverbatim*`	Same, displaying spaces

6.4 Language Selection

The following package keys select the languages used for Latin text, directional text, and code-related material:

Key	Used for
`latin`	The `latin` environment and Latin font commands
`ltr`	The `\lr` command
`rtl`	The `\rl` command
`code`	Code-related material

For example:

```
\usepackage[code=dutch,rtl=arabic]{babelbidi}
```

Unless specified explicitly, the values of these keys are selected automatically and updated after each `\babelprovide` call. Explicit values remain unchanged.

The automatic selections are determined as follows:

`latin`: The main language if it uses the Latin script; otherwise, the first loaded Latin-script language; otherwise, `english`.

`ltr`: The main language if it is left to right; otherwise, the first loaded left-to-right language; otherwise, `english`.

`rtl`: The main language if it is right to left; otherwise, the first loaded right-to-left language; otherwise, `arabic`.

`code`: The main Latin-script language, if available; otherwise, `english`, if loaded; otherwise, the first loaded Latin-script language; otherwise, the main or first loaded left-to-right language; finally, the main language.

Languages should be declared before their fonts are configured.

7 Patches

The package applies the following patches automatically when the corresponding engine or package is in use.

1. **Common patches**

(a) **Code support**

i. **LaTeX verbatim interfaces:** With a non-Latin main language, adjusts the font and direction handling of `\verb`, `\verb*`, `verbatim`, and `verbatim*`. With Unicode engines, the font patch selects `\latinttfamily` without changing the text language.

ii. listings: With a non-Latin main language, selects the `code` language for display and inline listings, with direction handling for inline listings.

iii. minted: With a non-Latin main language, applies the `code` language and direction handling to the `minted` environment and its processing commands. LuaLaTeX also receives the required layout adjustments.

(b) `\label`: prevents errors caused by Babel's direction commands in label keys derived from enumeration counters.

2. **XeLaTeX patches**

(a) natbib: corrects citation links in RTL text with recent hyperref versions.

(b) fontspec: disables its math-mode handling when Babel bidi mode is active.

3. **LuaLaTeX patches**

(a) `\makebox` and tables: adjust the positions of `l` and `r` material and swap table-column alignment in RTL text.

(b) Footnotes: adjust footnote paragraph direction in RTL contexts; footnotes issued in math mode in RTL-main documents are treated as RTL.

(c) TikZ: use LTR paragraph direction inside `tikzpicture`, except with beamer-rl.

(d) fancyhdr: increase the head height and typeset headers and footers in LTR direction.

(e) beamer: correct the layout of rounded boxes, except when beamer-rl is loaded.

4. **RTL-main footnote-rule patch**

(a) Typesets the default rule on the right in RTL-main documents, with stable LTR geometry, preserving explicit rule selections.

8 Minted package

BabelBidi supports the minted package:

```
\usepackage{minted}
```

The textlang option selects a Babel language already declared with \babelprovide or loaded as a Babel package option:

```
\begin{minted}[textlang=persian]{latex}
    متن فارسی
\end{minted}
```

It can also be set with \setminted:

```
\setminted{textlang=persian}
```

The textdir option sets the code direction: rtl (right-to-left) or ltr (left-to-right).

With LuaLaTeX, chardir uses Babel's \babelcharproperty to temporarily set character directions. It takes comma-separated {character}{direction} pairs. Direction values include l, al, r, en, an, on, et, and cs. Previous values are restored after the environment ends.

This example sets right-to-left code with the backslash and closing brace assigned direction l:

```
\begin{minted}[textdir=rtl,
    chardir={{\\}{l},{\}}{l}}]{latex}
    این روش در
    \lr{Chapter 2 (Examples)}
    .توضیح داده شده است
\end{minted}
```

With LuaLaTeX, frames retain left-to-right geometry. Line numbers use Latin digits.

9 License

The BabelBidi package is released under the LaTeX Project Public License (LPPL), version 1.3c or later. The full text of the license is available in the LICENSE file.

10 Author and Maintainer

The babelbidi package is developed and maintained by **Amer Amikhteh**.

The project's source code, development versions, and issue tracker are available in the project repository:

https://github.com/gambi-shah/babelbidi