

The babelbidi Package

Amer Amikhteh

Version 0.1.0
October 9, 2026

Abstract

babelbidi extends Babel's bidirectional support by correcting direction-sensitive behaviour and adding complementary interfaces for text direction and language selection. It works with LuaLaTeX, XeLaTeX, and pdfLaTeX, with either an LTR or an RTL main language and with languages supported by Babel.

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 Loading

The package is loaded as follows:

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

The package accepts Babel options. The `bidi` and `layout` options are handled by `babelbidi` and passed to Babel; the remaining options are passed directly to Babel.

If `bidi` is not specified, its default value depends on the engine:

Engine	Default value of <code>bidir</code>
LuaTeX	<code>basic</code>
XeTeX	<code>bidir</code>
Pdftex	<code>default</code>

With LuaTeX, the default value of `layout` is:

```
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, enclose it 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 Lua¹ 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.

3 Direction and Language APIs

`babelbidir` provides commands and environments for selecting text direction and for selecting the languages used by auxiliary text, including Latin text, left-to-right text, right-to-left text, and code.

3.1 Commands

Command	Description
<code>\LRE</code>	Typesets its argument with left-to-right text direction
<code>\RLE</code>	Typesets its argument with right-to-left text direction
<code>\LR</code>	Alias for <code>\LRE</code>
<code>\RL</code>	Alias for <code>\RLE</code>
<code>\lr</code>	Typesets its argument in the selected left-to-right language
<code>\rl</code>	Typesets its argument in the selected right-to-left language
<code>\latintoday</code>	Prints today's date in the selected Latin language

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

3.2 Environments

Environment	Description
RTL	Typesets its contents with right-to-left text direction
LTR	Typesets its contents with left-to-right text direction
latin	Typesets its contents in the selected Latin language

3.3 Language Selection

The languages used by the `latin` environment, the commands `\lr` and `\rl`, and code-related material can be specified with `\babelbidilangs`:

Key	Language selected for
<code>latin</code>	The <code>latin</code> environment
<code>ltr</code>	The <code>\lr</code> command
<code>rtl</code>	The <code>\rl</code> command
<code>code</code>	Code-related material

For example:

```
\babelbidilangs{latin=english,rtl=arabic}
```

If a key is not specified, `babelbidi` determines its value from the languages loaded by `Babel`. The values of the `latin`, `ltr`, `rtl`, and `code` keys are selected according to the main language of the document:

latin: The main language if it uses the Latin script; otherwise, the first loaded Latin-script language, or `english` if no such language is available.

ltr: The main language if it is left to right; otherwise, the first loaded left-to-right language, or `english` if no such language is available.

rtl: The main language if it is right to left; otherwise, the first loaded right-to-left language, or `arabic` if no such language is available.

code: The main language if it uses the Latin script; otherwise, `english` if it has been loaded; otherwise, the first loaded Latin-script language; otherwise, the main language if it is left to right; otherwise, the first loaded left-to-right language; and, finally, the main language.

The automatically selected languages can be overridden individually by passing the corresponding keys to `\babelbidilangs`.

4 Patches

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

1. Common patches

- (a) Code language selection: when the main language does not use the Latin script, the code-related patches use the language selected by the code key. These patches cover the `verbatim` environment, the `listings` package, and the `minted` environment and inline command. Documents with a Latin-script main language do not need these patches.

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) `multicol`: selects RTL or LTR column order according to the current text direction.
- (c) Footnotes: adjust footnote paragraph direction in RTL contexts; footnotes issued in math mode in RTL-main documents are treated as RTL.
- (d) `TikZ`: use LTR paragraph direction inside `tikzpicture`, except with `beamer-rl`.
- (e) `fancyhdr`: increase the head height and typeset headers and footers in LTR direction.
- (f) `beamer`: correct the layout of rounded boxes, except when `beamer-rl` is loaded.

4. RTL-main footnote-rule patch

- (a) With XeLaTeX and LuaLaTeX, typeset the footnote rule in a stable LTR direction when the main language is RTL.

5 Current Limitations and Possible Directions

1. **Incomplete API coverage.** Several APIs provided by `luabidi` are not yet defined by `babelbidi`. Extending this API coverage is a priority for the next release.
2. **Limited compatibility testing.** The package has not been tested with every combination of document class and third-party package. Broader testing may help identify interactions that require further work.

6 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.