

The babelbidi Package

Amer Amikhteh

Version 0.3.0
October 14, 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 T_EX 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 <code>bidi</code>
LuaTeX	<code>basic</code>
XeTeX	<code>bidi-r</code>
PdfTeX	<code>default</code>

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 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.

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:

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

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

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.

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 and Language APIs

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

5.1 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`.

5.2 Commands

Command	Description
<code>\LRE</code>	Typesets its argument from left to right
<code>\RLE</code>	Typesets its argument from right to left
<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
<code>\latinttfamily</code>	Selects the configured Latin monospaced font family

5.3 Environments

Environment	Description
<code>RTL</code>	Typesets its contents from right to left
<code>LTR</code>	Typesets its contents from left to right
<code>latin</code>	Typesets its contents in the selected Latin language

5.4 Language Selection

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

Key	Used for
<code>latin</code>	The <code>latin</code> environment and Latin font commands
<code>ltr</code>	The <code>\lr</code> command
<code>rtl</code>	The <code>\rl</code> command
<code>code</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.

6 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.
- (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) With XeLaTeX and LuaLaTeX, typeset the footnote rule in a stable LTR direction when the main language is RTL.

7 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.

8 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.

9 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>