The babelbidi Package

Amer Amikhteh

Version 0.6.0
November 8, 2026

Abstract

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

Contents

1 Introduction **3**

2 Obtaining and Installing **3**

3 Loading **4**

4 Quick Start **5**

5 Font Fallback **5**

6 Text Direction **6**
6.1 Inline Text 6
6.2 Paragraphs and Blocks 6

7 Boxes and Positioning **7**
7.1 Directional Boxes 7
7.2 Overlapping Text 7

8 Footnotes and Title Notes **7**
8.1 Footnote Direction 7
8.2 Footnote Rules 8
8.3 Extra Footnote Layouts 9

9 Lists, Cases, and Counters **10**
9.1 Lists and Bibliographies 10
9.2 Cases 11
9.3 Counter Separators 11

10 Margins and Columns **12**
10.1 Margin Notes 12
10.2 Two-Column Output 12

11 Language and Font APIs **12**
11.1 Font Selection 12
11.2 Language Commands 13
11.3 Language Environments 13
11.4 Language Selection 13

12 Patches **14**
12.1 Common Patches 14
12.2 XeLaTeX Patches 15
12.3 LuaLaTeX Patches 15

13 Minted Package **15**
13.1 Language and Direction 15
13.2 Character Directions 16
13.3 Layout 16

14 Limitations and Compatibility **16**

15 License **17**

16 Author and Maintainer **17**

1 Introduction

Babel provides a flexible framework for multilingual typesetting in LaTeX, including support for documents that combine left-to-right (LTR) and right-to-left (RTL) languages. However, supporting multiple writing directions involves more than selecting the appropriate language. Text direction also affects paragraph layout, footnotes, lists, tables, margins, and other document elements. Although Babel provides bidirectional support, some of these components may still require adjustments to behave consistently in RTL contexts.

The babelbidi package addresses these gaps while keeping Babel as the foundation for multilingual document processing. Rather than introducing a separate language-management system, it complements Babel's bidirectional support with direction interfaces and layout corrections. It works with either an LTR or RTL main language, allowing Babel's language facilities to be used in both types of documents.

The package also brings together facilities useful for multilingual documents, including explicit control over text direction, language-aware text switching, font configuration, and glyph fallback with LuaLaTeX. It aims to provide consistent user-level interfaces across the supported engines, while allowing engine-specific implementations where necessary.

The package provides user-level interfaces compatible with bidi, while allowing engine-specific implementations. Consequently, identical output across engines is not guaranteed. For bidirectional processing, XeTeX uses the bidi backend, while LuaTeX and pdfTeX use implementations supplied by babelbidi.

The main engine-dependent capabilities are summarized below. Here, `Native` means that the feature is implemented by babelbidi, while `Via bidi` indicates that it is provided by bidi.

Feature	LuaTeX	XeTeX	pdfTeX
Direction and layout interfaces	Native	Via bidi	Native
Unicode font commands	Native	Native	–
Glyph fallback	Native	–	–
Extra footnote layouts	Native	Via bidi	Native
minted `textlang` and `textdir`	Native	Native	Native
minted `chardir`	Native	–	–

The table describes interface availability, not identical behaviour or equal language coverage. In particular, pdfLaTeX still requires suitable input encodings, fonts, and Babel language support.

The following sections explain how to load and configure babelbidi, control text direction and layout, select languages and fonts, and use its additional interfaces and package patches. They also describe relevant engine-specific behaviour and limitations.

2 Obtaining and Installing

The official version of babelbidi is available from CTAN:

https://ctan.org/pkg/babelbidi

Once the TeX Live package database has been updated, install it with:

```
tlmgr install babelbidi
```

3 Loading

Load the package with:

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

Normally, babelbidi loads Babel and selects a bidirectional backend. With LuaLaTeX and beamer-rl, Babel setup is delegated to the class. 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, its default value is selected according to the engine:

Engine	Default value of bidi
LuaTeX	basic
XeTeX	bidi-r
pdfTeX	default

With LuaTeX, the default value of layout is:

```
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. To retain graphics and lists, include them explicitly. Values containing multiple selectors must be enclosed in braces. The package applies this layout setting only with LuaTeX; it has no effect in the XeTeX and pdfTeX loading branches.

LuaLaTeX is recommended for its broader capabilities and extensibility. Babel's onchar option can select a language from the characters in the text. For Arabic, for example:

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

Punctuation shared by several scripts, such as parentheses, may still require explicit language selection. For slides with an RTL main language, beamer-rl is recommended over beamer, although it is not required. With LuaLaTeX and beamer-rl, Babel setup is delegated to the class.

With LuaTeX, the boolean fontcache option controls caching of Babel font mappings used by automatic font selection and glyph fallback. It is enabled by

default to improve compilation speed. In special cases, such as font changes made through the `rmfamily` hook, cached mappings may prevent changes from taking effect. Disable the cache when needed:

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

To enable paragraph and multi-column footnote layouts, use the `extrafootnotefeatures` option:

```
\usepackage[extrafootnotefeatures]{babelbidi}
```

The option also accepts `=on` and `=off`; its default is off. With XeTEX, it is passed to bidi; with LuaTEX and pdfTEX, babelbidi provides its own implementation. See Section 8.3 for the available commands.

4 Quick Start

The following example works with LuaLATEX and XeLATEX and requires the fonts Latin Modern Roman and Amiri:

```
\documentclass{article}
\usepackage{babelbidi}

\babelprovide[import,main]{english}
\babelprovide[import]{arabic}

\babelfont{rm}{Latin Modern Roman}
\babelfont[arabic]{rm}{Amiri}

\begin{document}
    English text with \rl{نص عربي}.

    \begin{otherlanguage}{arabic}
        نص عربي مع \lr{English text}.
    \end{otherlanguage}
\end{document}
```

The lowercase commands `\lr` and `\rl` select a language context as well as its direction. By contrast, the uppercase commands `\LR` and `\RL` change direction only; they do not select a language or font. The `onchar` option can automate language selection for characters, as shown in the previous section.

5 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, and their fonts must be declared with `\babelfont`. These declarations should be made in the preamble. The `\fontfallback` declaration may precede them; fallback mappings are resolved when the document begins.

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 source or candidate languages produce warnings. Repeated candidates and self-references are ignored. Fallback is not recursive and is applied per glyph before shaping. If no candidate contains a glyph, it remains in the original font. Combining-mark placement, joining, and visual consistency across fonts are not guaranteed; choose compatible fonts for the scripts involved. XeLaTeX and pdfLaTeX warn and ignore explicit fallback declarations.

6 Text Direction

These interfaces control direction without changing the active language. Use inline commands for a passage within a paragraph, and switches or environments for paragraph-level direction.

6.1 Inline Text

Command	Description
`\beginL`, `\endL`	Begin and end an explicit LTR text group
`\beginR`, `\endR`	Begin and end an explicit RTL text group
`\LRE{<text>}`, `\LR{<text>}`	Inline LTR text
`\RLE{<text>}`, `\RL{<text>}`	Inline RTL text

Each `\beginL` must be matched by `\endL`, and each `\beginR` by `\endR`. Prefer the argument-taking commands for short passages.

6.2 Paragraphs and Blocks

Interface	Description
`\setLTR`, `\setLR`, `\unsetRTL`	Set paragraph direction to LTR
`\setRTL`, `\setRL`, `\unsetLTR`	Set paragraph direction to RTL
`LTR`, `RTL` environments	Typeset an enclosed block in LTR or RTL direction

The switches take no arguments; place them between paragraphs and use a group to limit their scope. End the final paragraph before closing the group. The environments handle paragraph boundaries and may contain several paragraphs.

7 Boxes and Positioning

7.1 Directional Boxes

These are directional variants of `\hbox`, `\vbox`, and `\vtop`; the suffix selects LTR or RTL direction.

Command	Description
`\hboxL`, `\hboxR`	Horizontal box without line breaking
`\vboxL`, `\vboxR`	Vertical box with a bottom reference point
`\vtopL`, `\vtopR`	Vertical box with a top reference point

Text in vertical boxes is broken into lines according to `\hsize`. For boxes containing lines of text, `\vbox` uses the last line's baseline and `\vtop` the first.

Before the opening brace, `to <dimension>` sets the final width of a horizontal box or height of a vertical box; `spread <dimension>` adds to its natural size. Stretchable or shrinkable space adjusts to the requested size. Use `\setbox` to store a box and `\box` to insert it.

7.2 Overlapping Text

These direction-sensitive variants of `\llap` and `\rlap` place text in a zero-width box.

Command	Description
`\bidillap{<text>}`	Overlap left in LTR text, right in RTL text
`\bidirlap{<text>}`	Overlap right in LTR text, left in RTL text

8 Footnotes and Title Notes

Footnote direction, rule appearance, and layout are controlled separately. Explicit commands insert an individual note; direction switches set the policy for subsequent ordinary footnotes.

8.1 Footnote Direction

The `LTR` and `RTL` forms select the direction of the footnote text, not its language. Use Babel's language commands when a footnote also requires a different language or font.

Command	Arguments and effect
`\LTRfootnote`, `\RTLfootnote`	`[`⟨num⟩`]{`⟨text⟩`}`: insert a footnote in the specified direction
`\LTRfootnotetext`, `\RTLfootnotetext`	`[`⟨num⟩`]{`⟨text⟩`}`: insert directed footnote text without a mark in the main text
`\LTRthanks`, `\RTLthanks`	`{`⟨text⟩`}`: insert a directed title note
`\setfootnoteLR`, `\unsetfootnoteRL`	Set ordinary footnotes to LTR
`\setfootnoteRL`, `\unsetfootnoteLR`	Set ordinary footnotes to RTL

The optional ⟨num⟩ has the same meaning as in `\footnote` or `\footnotetext`. The `\LTRfootnotetext` and `\RTLfootnotetext` commands do not insert a mark in the main text; use them with `\footnotemark` when a mark is needed.

The title-note commands are counterparts of `\thanks` and can be used in the `\author` or `\title` argument.

The switches take no arguments and insert no note. They set the direction policy for subsequent ordinary footnotes. Their assignments are local and remain in effect within their scope until changed. The `unset` forms select the opposite direction; they do not restore the initial Babel behaviour.

Explicitly directed footnotes take precedence over the policy: for example, `\RTLfootnote` still produces an RTL footnote after `\setfootnoteLR`. Before a direction policy is selected, the existing Babel behaviour is retained.

8.2 Footnote Rules

These controls affect the rule, not the footnote text direction.

Interface	Description
`\leftfootnoterule`, `\LRfootnoterule`	Left-aligned rule
`\rightfootnoterule`, `\RLfootnoterule`	Right-aligned rule
`\textwidthfootnoterule`	Full-width rule
`\autofootnoterule`	Rule alignment based on footnote direction
`\footnoterulewidth`	Length controlling the width of short footnote rules
`\SplitFootnoteRule`	Full-width rule above a continued footnote
`\debugfootnotedirection`	Direction messages when the automatic rule is evaluated

For example:

```
\setlength{\footnoterulewidth}{.3\columnwidth}
\rightfootnoterule
```

With LuaTEX and pdfTEX, rule selectors make global assignments. The initial short-rule width is `.4\columnwidth`, evaluated when the length is set. Automatic alignment is engine-dependent:

LuaTEX: Inspect the direction of the first line in the ordinary footnote insertion box; use a left rule if no line is found.

pdfTeX: Use the first explicitly directed footnote recorded since the preceding shipout; use a left rule if none was recorded. This is a fallback, not an inspection of the final footnote box.

XeTeX: Use the implementation supplied by bidi.

To diagnose automatic rule selection, use:

```
\autofootnoterule
\debugfootnotedirection
```

The LuaTeX and pdfTeX implementations provide no corresponding public command to disable these diagnostic messages. Remove the diagnostic command for normal runs.

Select the ordinary rule before `\SplitFootnoteRule`. It retains that rule on ordinary pages and uses a full-width rule above a continuation from the preceding page. In RTL-main documents, the package selects a right-aligned default rule only if no replacement rule has already been selected. Extra footnote layouts change rule handling as described below.

8.3 Extra Footnote Layouts

Enable these layouts with the `extrafootnotefeatures` package option. With LuaTeX and pdfTeX, they apply to page and minipage footnotes and preserve language, direction, labels, and hyperlinks.

Command	Layout for subsequent footnotes
`\normalfootnotes`	Ordinary one-column layout
`\paragraphfootnotes`	Paragraph layout
`\twocolumnfootnotes`	Multi-column layout with two to ten columns, as specified by the command name
`\threecolumnfootnotes`	
`\fourcolumnfootnotes`	
`\fivecolumnfootnotes`	
`\sixcolumnfootnotes`	
`\sevencolumnfootnotes`	
`\eightcolumnfootnotes`	
`\ninecolumnfootnotes`	
`\tencolumnfootnotes`	

Each command accepts an optional star. Without it, the footnote mark occupies a fixed-width box; with it, the box uses the mark's natural width:

```
\twocolumnfootnotes*
```

With LuaTeX and pdfTeX, paragraph and multi-column layouts collect RTL and LTR notes separately. Ordinary one-column footnotes are output first, followed by the RTL group and then the LTR group. Within each multi-column group, columns run right to left for RTL notes and left to right for LTR notes. Consequently, the visual order of notes may differ from the order of references in the text. With XeTeX, the layouts and their detailed behaviour are supplied by bidi.

Layout assignments are local, but collected notes are assembled later. Select a layout in the preamble or at a page boundary and keep it in force until the affected notes have been output. Do not rely on independent paragraph and multi-column layouts within the same pending stream. For a document-wide layout:

```
\usepackage[extrafootnotefeatures]{babelbidi}
\twocolumnfootnotes
```

To change the layout in the body, flush the preceding page first:

```
\clearpage
\paragraphfootnotes
```

With LuaTEX and pdfTEX, the extra-layout implementation installs a rule spanning `\columnwidth` and disables `\autofootnoterule`, `\leftfootnoterule`, `\LRfootnoterule`, `\rightfootnoterule`, `\RLfootnoterule`, and `\textwidthfootnoterule`. Calling `\normalfootnotes` does not restore these selectors. The rule spans the footnote area of the current document column, not an individual footnote column.

9 Lists, Cases, and Counters

9.1 Lists and Bibliographies

These environments set the direction of selected items within an existing list or bibliography; they do not create a new list.

Environment	Description
`LTRitems`, `RTLitems`	Set LTR or RTL direction for a group of `\item` entries inside a list such as `itemize` or `enumerate`
`LTRbibitems`, `RTLbibitems`	Set LTR or RTL direction for a group of `\bibitem` entries inside `thebibliography`

For example, a list can contain an RTL group between ordinary items:

```
\begin{itemize}
    \item The first item
    \begin{RTLitems}
        \item مورد دوم
        \item مورد سوم
    \end{RTLitems}
    \item The fourth item
\end{itemize}
```

For bibliography entries, use `\bibitem` rather than `\item`. The following example sets one entry in LTR direction without changing the bibliography heading:

```
\begin{thebibliography}{9}
    \begin{LTRbibitems}
        \bibitem{knuth}
        Donald E. Knuth. \textit{The TeXbook}. 1984.
    \end{LTRbibitems}
\end{thebibliography}
```

Here, `9` reserves space for a single-digit label, and `knuth` is the key used by `\cite{<knuth>}`.

To set the direction of all bibliography entries, use one of the following switches before `thebibliography` begins. The setting is local; the heading and running marks retain the surrounding direction.

Command	Description
`\setLTRbibitems`, `\setRTLbibitems`	Set all bibliography entries to LTR or RTL direction
`\setdefaultbibitems`	Restore the default bibliography-entry direction

9.2 Cases

The `RTLcases` environment typesets two columns with a brace on the right. Cells use text mode: separate columns with `&`, rows with `\\`, and enclose mathematics in `$...$`.

```
\begin{RTLcases}
    $x$  & $x \geq 0$ \\
    $-x$ & $x < 0$
\end{RTLcases}
```

9.3 Counter Separators

With XeTEX, `\SepMark` is provided by `bidi`; its counter-formatting behaviour follows that implementation. `\SepMark{⟨separator⟩}` globally sets the separator in supported compound counters; the default is a period. With LuaTEX and pdfTEX, the patch replaces a period immediately following the parent counter in standard sectioning formats, chapter-based equation, figure, and table numbers, and formats created by unstarred `\numberwithin`. It also sets the theorem-counter separator.

This is not a general replacement of punctuation in counter definitions. Custom formats without the expected parent-counter pattern are left unchanged. Other formatting, including Babel's localized digits, is retained.

10 Margins and Columns

10.1 Margin Notes

Explicit commands insert a directed note; switches configure ordinary `\marginpar` commands without inserting a note.

Command	Arguments and effect
`\LTRmarginpar`, `\RTLmarginpar`	`[`⟨left-text⟩`]{`⟨right-text⟩`}`: insert a note with LTR or RTL text
`\setLTRmarginpar`, `\setRTLmarginpar`	Set ordinary notes to LTR or RTL direction and placement
`\setdefaultmarginpar`	Restore behaviour based on surrounding text direction

The optional ⟨left-text⟩ argument has the same meaning as in `\marginpar`: alternative text for the left margin. For ordinary `\marginpar` commands, the direction switches apply when no optional left-hand text is supplied. With the optional form, the surrounding text direction is retained. Explicit commands change the note's text direction, not its placement policy. The switches are local; page parity, reversed margins, and document columns still participate in placement. By default, ordinary margin notes follow the surrounding text direction.

With XeTeX, these commands use the native bidi implementation, which may apply the explicit direction only to the first paragraph of a multi-paragraph margin note. With LuaTeX and pdfTeX, babelbidi keeps the selected direction throughout the note.

10.2 Two-Column Output

These commands change the order of completed document columns, not the direction of their contents. They do not enable two-column mode.

Command	Description
`\LTRdblcol`	Place the first column on the left
`\RTLdblcol`	Place the first column on the right

The default order follows the main document direction. For multi-column footnotes, see Section 8.3.

11 Language and Font APIs

11.1 Font Selection

With XeLaTeX or LuaLaTeX, the following commands are available:

- `\settextfont[`⟨options⟩`]{`⟨font⟩`}`: Set the `rm`, `sf`, and `tt` fonts for the main language.
- `\setlatintextfont[`⟨options⟩`]{`⟨font⟩`}`: Set all three font families for the selected Latin language.

- `\deflatinfont{⟨command⟩}[⟨options⟩]{⟨font⟩}`: Define a font-family command with `Script=Latin`; this does not select a language.
- `\setlatinsansfont[⟨options⟩]{⟨font⟩}`: Set the sans-serif font for the selected Latin language.
- `\setlatinmonofont[⟨options⟩]{⟨font⟩}`: Set the monospaced font for the selected Latin language and configure it for use by `\latinttfamily`.
- `\setmonofont[⟨options⟩]{⟨font⟩}`: In the preamble, also register the main language's monospaced font with Babel, making it available to font fallback.

The first two commands set all three font families (`rm`, `sf`, and `tt`), not just the roman family. For individual language–family assignments, use `\babelfont` directly.

11.2 Language Commands

Unlike the direction-only commands, `\lr` and `\rl` select a Babel language context as well as the corresponding direction. The uppercase commands `\LR` and `\RL` change direction only; they do not select a language. The language-switching commands are not part of the standard `bidi` interface.

Command	Description
`\lr{<text>}`	Typeset text using the selected LTR language
`\rl{<text>}`	Typeset text using the selected RTL language
`\latintoday`	Print the current date using the selected Latin language
`\latinttfamily`	Select the Latin monospaced font family; Unicode engines only
`\latinverb|<text>|`	Inline verbatim in the selected `code` language
`\latinverb*|<text>|`	The same, with visible spaces

Despite its name, `\latinverb` selects the `code` language, not the `latin` alias.

11.3 Language Environments

Environment	Description
`latin`	Typeset contents in the selected Latin language
`latinverbatim`	Typeset a verbatim block in the `code` language
`latinverbatim*`	The same, with visible spaces

The Latin verbatim interfaces are available regardless of the main script.

11.4 Language Selection

The following package keys select the languages used by these interfaces:

Key	Used for
`latin`	The `latin` environment, `\latintoday`, and Latin font commands
`ltr`	`\lr`
`rtl`	`\rl`
`code`	Latin verbatim interfaces and code-package integration

These keys select language names; they do not load the languages. For example:

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

Unless specified explicitly, the values 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 LTR; otherwise, the first loaded LTR language; otherwise, `english`.

`rtl`: The main language if it is RTL; otherwise, the first loaded RTL language; otherwise, `arabic`.

`code`: The main language if it uses the Latin script; otherwise, loaded `english`; otherwise, the first loaded Latin-script language; otherwise, the main or first loaded LTR language; finally, the main language. If the main language is Babel's `nil` placeholder, the final fallback is `english`.

Selections use Babel's language metadata and loading order. Ensure that a selected language is declared before using it, including a default such as `english` or `arabic`. Declare languages before configuring their fonts.

12 Patches

Patches are applied automatically when their engine and package conditions are met. Load related packages in the preamble. The code integrations can be loaded before or after babelbidi.

12.1 Common Patches

- **LaTeX verbatim interfaces:** With a non-Latin main script, adjust direction and font handling for `\verb`, `\verb*`, `verbatim`, and `verbatim*`. With Unicode engines, the font patch selects `\latinttfamily` without changing the text language.
- listings: With a non-Latin main script, select the `code` language for display and inline listings, with direction handling for inline listings.
- minted: Register the options described in Section 13 for both Latin and non-Latin main scripts. Additional display wrappers are installed for non-Latin main scripts.
- **Labels and link targets:** Remove Babel counter-direction wrappers while constructing label keys and hyperlink target names, without removing them from displayed counter values.

12.2 XeLaTeX Patches

- natbib: Adjust citation-link handling in RTL text with recent hyperref versions.
- fontspec: Disable its automatic math-font setup when Babel bidi mode is active. This is not a general mathematics-direction patch.

12.3 LuaLaTeX Patches

- **Boxes and tables:** Adjust `\makebox` positioning and interchange the handling of `l` and `r` alignment in RTL contexts. The table patch changes cell alignment, not the declared column sequence.
- **Footnotes:** Adjust paragraph direction in RTL contexts and handle footnotes issued in mathematics in RTL-main documents.
- beamer: Adjust rounded-box layout, except with beamer-rl.

For RTL-main documents, additional LuaLaTeX patches apply:

- **TikZ:** Use LTR paragraph direction inside `tikzpicture`, except with beamer-rl.
- fancyhdr: Set `\headheight` to `1.2em` and wrap headers and footers in LTR runs. This value is not an automatic measurement of the header contents.

13 Minted Package

Load minted as usual and meet its external-tool requirements:

```
\usepackage{minted}
```

The options below are added by babelbidi, not by minted. They are registered for both Latin and non-Latin main scripts.

13.1 Language and Direction

The `textlang` option selects an already declared Babel language. If it is empty, the selected `code` language is used. The `textdir` option accepts `ltr` or `rtl`. It is applied after language selection, so an explicit direction overrides the direction established by `textlang`.

```
\begin{minted}[textlang=persian,textdir=rtl]{text}
    متن فارسی
\end{minted}
```

The language `persian` must already be declared and have a suitable font. The argument `text` above is the minted lexer, not the Babel language. Defaults can also be set with `\setminted`:

```
\setminted{textlang=english,textdir=ltr}
```

13.2 Character Directions

With LuaLaTeX, `chardir` temporarily changes Babel character- direction properties. Its value is a comma-separated list of `{character}{direction}` pairs. Direction values include `l`, `r`, `al`, `en`, `an`, `on`, `et`, and `cs`.

For example, assign LTR direction to a backslash and closing brace:

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

Saved properties are restored after the `minted` environment. The restoration hook is environment-based; do not assume equivalent restoration for inline commands or file-input interfaces. A nonempty `chardir` value raises an error with XeLaTeX or pdfLaTeX; it is not silently ignored.

13.3 Layout

Inside `minted` environments, line numbers use Latin digits. With LuaLaTeX, frame lines and padding receive LTR geometry adjustments. For a non-Latin main script, the additional LuaLaTeXdisplay wrapper uses a minipage. Such wrapped displays cannot break across pages; keep this in mind for long listings.

14 Limitations and Compatibility

- Shared interfaces do not imply identical implementations or output across engines. With XeTeX, behaviour may also depend on the installed `bidi` version.
- Direction commands do not provide language data, input encodings, fonts, or glyphs. Some patches depend on Babel or third-party package internals, so successful compilation alone does not guarantee correct bidirectional layout.
- Extra footnote layouts modify insertion and output handling. Check their compatibility with classes or packages that modify these mechanisms; universal compatibility is not guaranteed.

15 License

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

16 Author and Maintainer

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

Source code, development versions, and the issue tracker are available in the project repository:

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

For bug reports, include a minimal example, the TeX engine and its version, and the complete log file. Add \listfiles to the preamble to record the loaded file versions in the log.