# Unicode Bidi Algorithm Implementation for OpTEX, LATEX, and Plain

Version: 0.3, 2026-01-13

*Udi Fogiel, 2025-2026*

The unibidi-lua LuaTeX package is an implementation of the Unicode Annex #9 for OpTEX, LATEX and Plain LuaTEX formats. It allows to typeset bidirectional documents without the need of special markup.

This package is still in early phase, and the interface might still change.

The requirements are the LuaTEX engine, and the `luaotfload` font loader.

## 1 Usage

Currently there is only one macro, `\unibidilua`, which accepts key-value pairs separated by a space.

The macro accepts the following keywords:

- **enable:** This key accepts a boolean value. When true, the unibidi-lua process function is active in the `pre_shaping_filter` callback. When false, the callback remains registered but processing is skipped. Default is true when the package is loaded.
- **fences:** This key accepts a boolean value. It allows to disable or enable step N0 of the UAX9 algorithm (directional formatting characters for paired brackets). Default is false when the package is loaded.
- **removecontrols:** This key accepts a boolean value. When true, bidirectional control characters are removed from the processed text. When false, they are preserved. Default is true when the package is loaded.
- **setdir:** This key accepts an integer (or a range of integers) representing Unicode code points, and a direction value. It sets the directional property for the specified character(s). Example: `setdir `\A l` or `setdir `\A-`\Z al`.
- **setmirror:** This key accepts an integer (or a range of integers) representing Unicode code points, and mirror data. It sets the mirroring property for the specified character(s), indicating what character they should mirror to in RTL contexts. Example: `setmirror `\( `\)`.
- **settextclass:** This key accepts an integer (or a range of integers) representing Unicode code points, and a text class value. It sets the text class property for the specified character(s) according to UAX9 classification.
- **baselevel:** This key accepts a Lua function that determines the base directionality level of the text. The function is called during processing to establish whether the paragraph should be treated as LTR or RTL.

To use the package, as with other packages, you can do `\load[unibidi-lua]`, `\usepackage{unibidi-lua}` or `\input unibidi-lua` if you are using OpTEX, LATEXor Plain respectively. The process function is added to the `pre_shaping_filter` when you load the package.

## 2 Lua API

The package exposes the following interface from Lua when you require it:

```
local unibidi = require('unibidi-lua')
```

- `unibidi.setbaselevel(func)`: Set a Lua function that determines the base directionality level
- `unibidi.directions`: Table containing character directional properties
- `unibidi.mirrors`: Table containing character mirroring mappings
- `unibidi.textclasses`: Table containing character text classifications