LuaTeX-CN Documentation

— High-Quality Classical Chinese Typesetting —

Auto-generated from GitHub Wiki

Table of Contents

English | 中文版

LuaTeX-CN Wiki

Welcome to the LuaTeX-CN Wiki. This site provides comprehensive documentation for users and developers to help you create high-quality classic Chinese typesettings using LuaTeX-CN.

> **Current Version**: v0.2.5 | **CTAN**: v0.1.1

Quick Access

- 🚀 Quick Start - Get started with your first project in 5 minutes
- 📦 Installation - Install LuaTeX-CN on your system
- 📚 Examples - Explore "Records of the Grand Historian" (Shiji) and "Dream of the Red Chamber"

Documentation

1. Fundamentals

- Templates & Format - Understand the "Banxin" (Page Center) concept
- Font Setup - Config Chinese fonts and fallback chains
- Layout Control - Chapters, TOC entries, and fine-grained layout control

2. Specialized Commands

- Interlinear & Side Notes - Detailed usage of jiazhu and side notes
- Annotations & Marginal Notes - Floating annotations and marginal notes (MeiPi)
- Punctuation - Traditional Chinese punctuation (Judou)
- Decorations - Underlines, book titles, and correction marks

3. Advanced Features

- Text Boxes - Absolute positioning and adaptive containers

- Seals - Adding and positioning electronic seals
- Debugging - Grid visualization and layout debugging

Developers

For information on contributing or understanding the internal architecture:

- Development Guide
- Release Process

Contact

- **Contact**: Sheldon Li
- **Email**: sheldonli.dev@gmail.com

English | 中文版

Installation Guide

This page describes how to install the `luatex-cn` package.

Method 1: Via CTAN/Package Manager (Recommended)

Install using your TeX distribution's package manager:

```
# TeX Live (Windows/Linux)
tlmgr install luatex-cn

# MiKTeX
# Open MiKTeX Console → Packages → Search for luatex-cn
```

> ⚠ **Note**: The CTAN version may lag behind GitHub. We recommend using GitHub Release.

Method 2: Manual Installation from GitHub Release

1. **Download**: Go to GitHub Releases and download the latest `luatex-cn-tex-v*.zip`
2. **Locate texmf directory**:
    - **Windows**: `C:\Users\<username>\texmf`
    - **macOS/Linux**: `~/texmf`
3. **Place files**: Put the contents of `tex/` into:

    ```
    texmf/tex/latex/luatex-cn/
    ```

4. **Refresh database**:

    ```
    texhash
    ```

Method 3: Direct Usage

Place the contents of `tex/` in the same directory as your `.tex` file and compile directly.

Verify Installation

Create a test file `test.tex`:

```
\documentclass[四库全书]{ltc-guji}
\begin{document}
\begin{正文}
天地玄黄，宇宙洪荒。
\end{正文}
\end{document}
```

Run:

```
lualatex test.tex
```

If compilation succeeds, installation is complete.

System Requirements

- LuaTeX (TeX Live 2024+ recommended)
- `luaotfload` and `fontspec`

Font Requirements

We recommend installing these fonts for best results:

- Noto Serif CJK SC (Source Han Serif)
- FangSong or KaiTi (for traditional book effects)

English | 中文版

Quick Start

This page helps you quickly get started with `luatex-cn` for traditional Chinese book typesetting.

Basic Template

```
\documentclass[四库全书]{ltc-guji}
% Optional: specify font
% \setmainfont{Noto Serif SC}

\title{钦定四库全书}
\chapter{史记\\卷一}

\begin{document}
\begin{正文}

This is sample vertical text for ancient book typesetting.

\end{正文}
\end{document}
```

Common Commands

Interlinear Notes (Double-line Small Text)

```
Main text\\夹注{This is interlinear note content, auto-balances}continued text.
```

Side Notes

```
Main text\\侧批[yoffset=10pt]{This is side note content}continued text.
```

Paragraph Indentation

```
\\begin{段落}[indent=2]
   天地玄黄\\\\
   宇宙洪荒
\\end{段落}
```

Spaces

```
Need\\空格[2]two spaces here.
```

Annotation Box

```
\\批注[x=5cm, y=2cm, height=6, color={1 0 0}]{Annotation content}
```

Seal

```
\\印章[page=1, xshift=2cm, yshift=5cm]{seal.png}
```

Force Column Break

```
End of this column\\\\
Start of new column
```

Metadata Setup (v0.2.0+)

```
\title{Book Title}
\publisher{Publisher Name}
\ChapterMarker{Volume 1}
```

Or using the setup command:

```
\metadataSetup{
  title = Book Title,
  publisher = Publisher Name
}
```

Debug Mode (v0.2.2+)

```
\LtcDebugOn   % Enable debug ruler, shows grid coordinates
\LtcDebugOff  % Disable debug mode
```

Preset Styles

Choose preset styles via document class options:

Option	Description
`四库全书`	Qing Dynasty Siku Quanshu style, single fishtail
`红楼梦甲戌本`	Manuscript style, no fishtail

```
\\documentclass[四库全书]{ltc-guji}
% or
\\documentclass[红楼梦甲戌本]{ltc-guji}
```

Next Steps

- See Examples for more complex usage
- Read Changelog for latest features

Templates and Customization

`luatex-cn` provides a flexible template system with built-in and custom templates.

Built-in Templates

Template Name	Document Class	Description
`四库全书`	`ltc-guji`	Classic official book style (B&W)
`四库全书彩色`	`ltc-guji`	Classic official book style (Color)
`红楼梦甲戌本`	`ltc-guji`	Manuscript style with side/top notes
`default`	`ltc-book`	Default modern vertical style

Using Templates

Specify the template name in document class options:

```
% Use colored Siku Quanshu template
\documentclass[四库全书彩色]{ltc-guji}

% Use Hong Lou Meng Jiaxu edition template
\documentclass[红楼梦甲戌本]{ltc-guji}
```

Custom Templates

Method 1: Create Configuration File

Create a `luatex-cn-guji-<template-name>.cfg` file in the `configs/` directory:

```
% luatex-cn-guji-MyTemplate.cfg

% Can inherit from existing template
\gujiSetup{ template = default }

% Page setup
\pageSetup{
    paper-width = 1077.2pt,
    paper-height = 1077.2pt,
    margin-top = 226.8pt,
    margin-bottom = 113.4pt,
    margin-left = 25.5pt,
    margin-right = 25.5pt,
}

% Content setup
\contentSetup{
    n-column = 12,
    font-size = 30pt,
    line-spacing = 45pt,
    n-char-per-col = 18,
    border = true,
}

% Banxin setup
\banxinSetup{
    banxin-upper-ratio = 0.18,
    banxin-middle-ratio = 0.38,
    upper-yuwei = true,
    lower-yuwei = true,
}

\endinput
```

Method 2: Direct Configuration in Document

```
\documentclass{ltc-guji}

\gujiSetup{
    book-name = My Book Title,
    chapter-title = Chapter One,
}

\contentSetup{
    n-column = 10,
    font-size = 24pt,
    border = true,
}

\begin{document}
% ...
\end{document}
```

Method 3: Define Template with defineGujiTemplate

```
\defineGujiTemplate{MyTemplate}{
    book-name = Default Title,
    n-column = 10,
    border = true,
}

% Use custom template
\gujiSetup{ template = MyTemplate }
```

Configuration Reference Table

Command	Parameter	Description	Default
`\gujiSetup`	`book-name`	Book name	-
	`chapter-title`	Chapter title	-
	`template`	Template name	`default`
`\contentSetup`	`n-column`	Columns per half page	12
	`font-size`	Font size	30pt
	`line-spacing`	Line spacing	45pt
	`n-char-per-col`	Characters per column	18
	`border`	Show border	true
	`font-color`	Font color (RGB)	{0,0,0}
	`border-color`	Border color (RGB)	{0,0,0}
`\pageSetup`	`paper-width`	Paper width	-
	`paper-height`	Paper height	-
	`margin-*`	Margins	-
`\banxinSetup`	`banxin-upper-ratio`	Upper section ratio	0.18
	`upper-yuwei`	Upper fish tail	true
	`lower-yuwei`	Lower fish tail	true

👉 Next: See Debug Mode or Features

English | 中文版

Examples

This directory contains multiple usage examples for the `luatex-cn` package, demonstrating different features from traditional ancient book typesetting to modern vertical layouts.

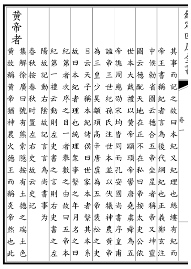
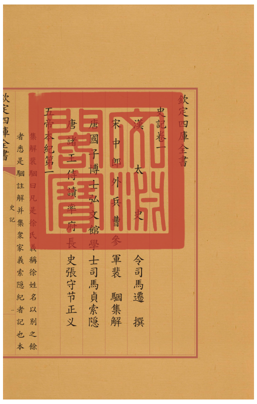
Example List

1. Records of the Grand Historian - Basic Annals of the Five Emperors

Features: Demonstrates highly complex ancient book typesetting.

Capabilities:

- Absolutely positioned red seals (overlaying text)
- Customized page center text with single fishtail
- Complex interlinear note typesetting
- Traditional black silk frame and indentation

Black & White	Color Preview
	

📂 View Source

2. Records of the Grand Historian - Table of Contents

Features: Following Qing Dynasty "Siku Quanshu" Northern Pavilion style.

Capabilities:

- Standard ancient book table of contents layout
- "Eight lines, twenty-one characters" format restriction
- Typical white mouth, double borders on all sides, single fishtail format

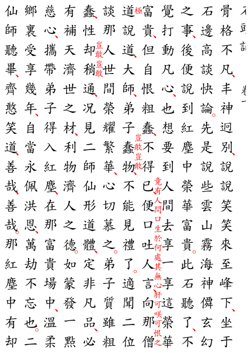
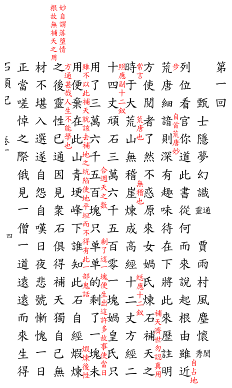
🗀 View Source

3. Dream of the Red Chamber - Jiaxu Manuscript

Features: Simulates manuscript/annotated copy style.

Capabilities:

- **Side and marginal notes**: Edge annotations using the sidenote system
- **Double-column small text**: Double-line small annotations in main text
- No fishtail page center, bottom page numbers

Page 2 (Punctuation)	Page 1 (Marginal Notes)
 石頭記 卷一 骨格不凡、丰神迥別、說說笑笑來至峰下、坐于 石邊高談快論。先是說些雲山霧海神僊玄幻 之事、後便說到紅塵中榮華富貴。此石聽了、不 覺打動凡心、也想要到人間去享一享這榮華 富貴、但自恨粗蠢、不得已、竟有人問口生於何處其無心肝可咲可恨之便口吐人言、向那僧 道極說道、大師、弟子蠢豈敢物、豈敢不能見禮了。適聞二位 談那人世間榮耀繁華、心切慕之。弟子質雖粗 蠢、性却豈敢稍豈敢通、况見二師仙形道體、定非凡品、必 有補天濟世之材、利物濟人之德。如蒙發一點 慈心、攜帶弟子得入紅塵、在那富貴場中、溫柔 鄉裏受享幾年、自當永佩洪恩、萬劫不忘也。二 仙師聽畢、齊憨笑道、善哉、善哉。那紅塵中有却	 第一回 甄士隱夢幻識通靈 賈雨村風塵懷閨秀 列位看官你道此書從何而來說起根由雖近自占地 荒步唐細諳自首荒唐妙則深有趣味待在下將此來歷註明 方使閱者了然不惑原來女媧氏煉石補天濟世勿認真用補天之 時常言于大荒荒唐也山無稽無稽也崖煉成高經十總應十二釵二丈方經二 十照應副十二釵四丈頑石三萬六千五百零一塊媧皇氏只 用了三萬六千五合週天之数百塊只单剩了這一塊便生出這許多故事使當日单的剩了一塊未 用雖不以此補天就該去補地之坑陷使地平坦而不得有此一部鬼話便棄在此山青埂峰下誰知此石自經煅煅煉後性煉 妙自謂落墮情 根故無補天之用 之方通甚哉人生不能學也後靈性已通因見眾石俱得補天獨自己無 材不堪入選遂自怨自嘆日夜悲號慚愧一日 正當嗟悼之際俄見一僧一道遠遠而來生得 石頭記 卷一 四

🗀 View Source

4. Modern Vertical Book

Features: Clean modern vertical layout style.

Capabilities:

- Demonstrates pure vertical text support without traditional book elements
- Suitable for modern literature or reports requiring vertical typesetting

> 📌 Modern vertical books are not currently the project focus; more features will be added later.

📂 View Source

Note: All ancient book examples are digital reconstructions based on historical document scans, intended to demonstrate `luatex-cn`'s adaptability to various complex vertical typesetting scenarios.

English | 中文版

Features Overview

`luatex-cn` provides a rich set of features for ancient Chinese book typesetting, ranging from core vertical engines to various decorative elements.

1. Core Typesetting Engine

A grid-based layout system built on LuaTeX for high-precision vertical text flows.

- **RTT Direction**: Native support for top-to-bottom, right-to-left direction.
- **Grid Positioning**: Characters are precisely aligned to a defined grid.
- **Pagination & Column Flow**: Handles complex page splitting and column transitions automatically.

2. Traditional Book Elements

Full support for visual elements required by traditional binding:

- **Banxin & Yuwei**: Support for single/double fishtails, black silk frames, etc.
- **Split Page**: Support for electronic reading and traditional "tube page" binding.
- **Border System**: Double borders, inner borders, and customizable thickness.

3. Annotation & Revision System

Multiple ways to add comments to the text:

- **Interlinear Notes**: Double-line small text within main body with auto-balancing.
- **Side Notes**: Vertical annotations between text columns.
- **Annotations**: Floating annotation boxes placed anywhere on the page.
- **TOC Entries**: Added `\条目` (v0.2.5+) with hierarchical indentation for table of contents.

4. Layout Control & Commands

- **Force Column**: `\列` (or `\Column`) forces content to start in a new column.
- **Last Column**: `\末列` (or `\LastColumn`) places content in the last column of the current page.
- **Page Breaks**: Supports standard `\newpage` and `\clearpage`.

5. Decorations & Utilities

- **Seals**: Add collection or personal seals anywhere with opacity support.
- **Judou**: Traditional punctuation marks (circles and dots). Enhanced support for more Chinese punctuation since v0.2.1.
- **Fonts**: Auto-detection, font families, and fallback chains for rare characters. Supports simplified/traditional Chinese font names since v0.2.2.
- **Textbox**: Grid-based layout with auto-compression and filling.
- **Correction & Decoration**: Correction marks (\fix) and character decoration (\decorate).

5. Architecture & Debug Features (v0.2.0+)

- **Style Registry**: Phase 2 multi-attribute style storage (color, font_size, grid_height), supports cross-page style preservation.
- **Metadata System**: Centralized book metadata management (`\title`, `\publisher`, `\ChapterMarker`).
- **Debug Mode**: `\LtcDebugOn` displays page grid coordinates, `\LtcShowGrid` shows coordinate grid for layout debugging.
- **Plugin Architecture**: Standardized plugin API with unified initialize/layout/render three-phase interfaces.

6. Template System

Preset classical styles available out-of-the-box. See Templates for details.

- `四库全书` (Siku Quanshu): Qing Dynasty imperial style.
- `四库全书彩色`: Colored version of Siku Quanshu style.
- `红楼梦甲戌本`: Manuscript style with extensive annotations.

👉 Next Steps: Check Templates | Debug Mode | Examples

English | 中文版

Interlinear & Side Notes

`luatex-cn` offers a comprehensive annotation system, including in-text interlinear notes and marginal side notes.

Interlinear Notes (Jiazhu / 夹注)

Commonly seen in ancient books as double-line small characters between main body text.

Usage

```
Main text\\夹注{This is interlinear note content. If it is long, it will automatically split into two col
```

Key Features

- **Auto-Balancing Algorithm**: The system calculates the remaining space in the current column and splits the note content evenly into right and left sub-columns.
- **Cross-Column/Page Flow**: When a note doesn't fit in one column, it flows to the next column or page automatically.
- **Alignment Control**: Supports `left`, `right`, `center`, `inward`, and `outward` (default) alignment modes.

Side Notes (SideNode / 侧批)

Small annotations placed between text columns or at page margins.

Usage

```
Main text\\侧批[yoffset=10pt, color=red]{This is the side note content}continued text.
```

Parameters

- `yoffset`: Vertical offset for fine-tuning positions.

- `color`: Annotation color (default: red).
- `font-size`: Font size for the note.
- `grid-height`: Custom grid height for the note block.
- `border-padding-top`: Top border padding.
- `border-padding-bottom`: Bottom border padding.

Global Configuration

```
\sidenoteSetup{
  color = red,
  font-size = 19pt,
  grid-height = 20pt,
  yoffset = 0em,
  border-padding-top = 0.5em,
  border-padding-bottom = 0.5em
}
```

Cross-Page Color Preservation (v0.2.1+)

v0.2.1 introduced the style registry mechanism. Side note color settings now persist correctly across pages. Even when side note content spans multiple pages, colors remain consistent.

Technical Implementation (For Developers)

- **Interlinear Notes**: Uses LuaTeX attributes to mark nodes. During layout, the system detects these attributes, pauses normal typesetting, and calls `core_textflow.lua` for column splitting and re-mapping to the grid. Migrated to style stack architecture since v0.2.1.
- **Side Notes**: Managed by `core_sidenote.lua`. It captures the content as an independent box and calculates the offset based on the current reference character's coordinates. Supports Phase 2 style registry for color preservation since v0.2.1.
- **Style Registry**: `luatex-cn-style-registry.lua` provides multi-attribute style storage, supporting deduplication and cross-page preservation for color, font_size, grid_height, and other attributes.

☞ Next Steps: Check Annotations or Judou.

English | 中文版

Annotations (PiZhu / 批注)

The `\批注` command allows placing floating annotation boxes anywhere on the page, simulating post-publication notes or commentaries in ancient manuscripts.

Usage

```
\批注[x=5cm, y=2cm, height=6, color={1 0 0}]{Annotation content}
```

Parameters

- `x` : Horizontal position of the top-left corner (relative to page center).
- `y` : Vertical position of the top-left corner (relative to page center).
- `height` : Number of grid rows (height) allocated for the box.
- `color` : Color in RGB format (e.g., `{1 0 0}` for red).
- `font-size` : Font size inside the annotation.
- `grid-width` : Custom grid cell width for the box.
- `grid-height` : Custom grid cell height for the box.

Features

- **Absolute Positioning**: Does not occupy main body space; overlaid directly at specified coordinates.
- **Auto-Columnizing**: If content exceeds the set `height` , it automatically starts a new column within the box.
- **Clip Prevention**: Position calculations account for Banxin (page center) to prevent content being cut off by borders.
- **Multi-column Support**: Internal columns can be precisely controlled via `n-cols` .
- **Independent Indentation** (v0.2.2+): Annotation content no longer inherits paragraph indentation from main text (fix #37).

Technical Implementation

Under the hood, annotations are based on `core_textbox.lua`. Content is rendered into an independent vertical grid system. This produced box is then overlaid during the page rendering stage using `shipout` logic in `render_page.lua`.

Marginal Notes (MeiPi / 眉批) v0.2.3+

The `\眉批` command places annotations at the top of the page (above the main text), a common format in traditional Chinese book annotations. Unlike `\批注`, MeiPi automatically calculates positions, arranging from right to left to avoid overlap.

Usage

```
% Auto-positioning (recommended)
\眉批[height=5]{Marginal note content}

% Manual Y coordinate
\眉批[height=5, y=50pt]{Marginal note content}

% Fully manual positioning
\眉批[height=5, x=100pt, y=50pt]{Marginal note content}
```

Parameters

- `height` : Number of grid rows (height) for the note, **required**.
- `x` : Horizontal position. Auto-calculated if not specified, arranging right-to-left.
- `y` : Vertical position. Auto-calculated if not specified, aligning bottom edge to main text top.
- `color` : Color, default red `{1 0 0}` .
- `font-size` : Font size, default 18pt.
- `grid-width` : Grid width, default 20pt.
- `grid-height` : Grid height, default 19pt.

Features

- **Auto-arrangement** (v0.2.3+): Multiple MeiPi notes automatically arrange right-to-left. Spacing configurable via `\meipiSetup{spacing=...}` .
- **Smart Positioning**: Y coordinate auto-calculated so bottom edge aligns with main text top.

- **Cross-page Support** (v0.2.3+): MeiPi correctly appears on the page where its anchor is located, not the previous page.
- **Independent Indentation**: MeiPi content does not inherit paragraph indentation from main text.

Configuration

```
\meipiSetup{
  spacing = 10pt,  % Spacing between MeiPi notes
  gap = 0pt        % Gap between MeiPi bottom and main text top
}
```

Technical Implementation

MeiPi is based on `guji/luatex-cn-guji-meipi.lua`. It inserts anchor nodes into the node list to record positions, then during layout phase determines which page the MeiPi should appear on based on the **first node with layout information after** the anchor.

☞ Next Steps: Check Seals or Judou.

English | 中文版

Seals (YinZhang / 印章)

The `\印章` command adds images like collection marks or personal seals to the page, supporting absolute positioning and opacity.

Usage

```
\印章[page=1, xshift=2cm, yshift=5cm, opacity=0.8, width=3cm]{seal.png}
```

Parameters

- `page`: Page number where the seal appears (default: 1).
- `xshift` / `yshift`: Offsets from the page center or margin.
- `opacity`: Opacity (0.0 to 1.0). For ancient books, 0.7-0.9 is recommended to simulate ink absorption.
- `width`: Display width of the seal image.
- `color`: (Optional) Color for vector-based seals.

Features

- **Layer Management**: Seals are rendered during the `shipout/background` hook by default, appearing below the text. This can be customized if foreground placement is needed.
- **Multi-point Layout**: Multiple seal commands can be used on the same page; they are positioned independently based on their coordinates.
- **Absolute Positioning**: Uses TikZ's coordinate system for pinpoint accuracy on the page.

Technical Implementation

Seals leverage LaTeX `shipout` hooks. Each call to `\印章` registers a `tikzpicture` block inside the appropriate hook, wrapped in an `\ifnum\value{page}=N` conditional to ensure images are output on the correct page and at the correct layer.

👉 Next Steps: Check Judou or Features Overview.

[English Version] | 中文版

Judou (Punctuation)

Judou (jù dòu) refers to the traditional Chinese system of punctuation and pausing marks. `luatex-cn` can automatically convert modern punctuation into traditional styles suitable for ancient book typesetting.

What is Judou Mode?

Traditional ancient books typically do not use modern full-width punctuation marks. Instead, they use small circles ("Ju") or dots ("Dou") attached to the characters.

- **Ju (句)**: Corresponds to periods (.), question marks (?), and exclamation marks (!). Rendered as a small circle (○).
- **Dou (读)**: Corresponds to commas (,), semicolons (;), and enumeration commas (、). Rendered as a small dot or tick (●).

Supported Punctuation (v0.2.1+)

Since v0.2.1, the range of supported punctuation has been extended to include:

- Basic punctuation: periods, commas, enumeration commas, semicolons, colons
- Paired punctuation: quotation marks (「」『』""""), book title marks (《》〈〉), parentheses
- Special punctuation: dashes, ellipses, interpuncts, etc.

Usage

1. Global Activation

Enable it in the document class options:

```
\documentclass[judou-on=true]{ltc-guji}
```

2. Commands

You can switch modes anywhere in the document:

- `\JudouOn` : Enable the Judou processing mechanism.
- `\JudouOff` : Disable Judou processing.

Switching **Punctuation Styles**:

- `\JudouPunctuationMode` (or `\句读模式`): Convert modern punctuation to Ju/Dou marks.
- `\NonePunctuationMode` (or `\无标点模式`): Hide all punctuation (Plain text / "Baiwen").
- `\NormalPunctuationMode` (or `\正常标点模式`): Show standard modern full-width punctuation.

Example

```
\JudouOn
\JudouPunctuationMode
人之初，性本善。
% "," becomes a dot, "." becomes a circle at the bottom-right.

\NonePunctuationMode
子曰学而时习之不亦说乎
% All punctuation is hidden.
```

Parameters

Fine-tune Judou appearance via parameters:

Key	Description	Default
`judou-color`	Color of the marks	`red`
`judou-size`	Size of the marks	`1em`
`judou-pos`	Position relative to character	`right-bottom`

Technical Details

Implemented via `luatex-cn-guji-judou.lua`:

1. **Identification**: Scans node stream for punctuation Unicode code points.
2. **Conversion**: In `judou` mode, punctuation nodes are removed and replaced by a `WHATSIT` node attached to the preceding CJK character.
3. **Rendering**: The renderer draws the circle/dot at the specified anchor point (usually bottom-right) of the character.
   - v0.1.5 optimized the positioning algorithm to ensure marks adhere closely to the character bounding box without extra gaps.
   - v0.2.1 fixed punctuation mode application for sidenote content (fix #39).
   - v0.2.2 extended the range of supported Chinese punctuation symbols (fix #40).

👉 Next: Side-Note or Features.

[English Version] | 中文版

Fonts

`luatex-cn` provides intelligent cross-platform font detection and a powerful multi-font fallback mechanism to gracefully handle missing characters (rare glyphs).

1. Automatic Font Detection

By default, the `ltc-guji` document class automatically selects the most suitable installed fonts based on your operating system:

- **Windows**: Prioritizes SimSun, SimHei, etc.
- **macOS**: Prioritizes Songti SC, PingFang SC, etc.
- **Linux**: Prioritizes Fandol series or Noto CJK series.

To manually trigger and apply the auto-detected font:

```
\ApplyAutoFont
```

Chinese Font Name Support (v0.2.2+)

Since v0.2.2, font detection supports simplified and traditional Chinese font names:

```
\设置字体{宋体}          % Simplified Chinese name
\设置字体{標楷體}        % Traditional Chinese name
\设置字体{思源宋体}      % Open source font Chinese name
```

The system automatically maps Chinese font names to corresponding system fonts.

2. Font Family & Fallback

v0.1.5 introduces the concept of **Font Family**. By setting a font family, the system establishes a "fallback chain". When the main font lacks a specific character (e.g., a rare CJK ideograph), it automatically looks for it in the next font in the chain.

`\setFontFamily`

This command accepts a comma-separated list of fonts. The first font in the list becomes the main font, and the rest serve as fallbacks.

```
% Sets "Source Han Serif SC" as the main font.
% If a character is missing, it tries "SimSun", then "SimHei".
\setFontFamily{Source Han Serif SC, SimSun, SimHei}
```

Chinese Alias:

```
\设置字体族{Source Han Serif SC, SimSun, SimHei}
```

`\DeclareFontFallback`

To define a named fallback chain for custom usage:

```
% Define a chain named "mybackup"
\DeclareFontFallback{mybackup}{SimSun, SimHei}

% Apply this chain to a specific font
\setmainfont{MyCustomFont}[RawFeature={fallback=mybackup}]
```

3. Standard Font Setting

If you only need to set a single font without fallback logic:

```
\setmainfont{KaiTi}
% or
\设置字体{KaiTi}
```

> [!NOTE] For vertical layout, `luatex-cn` generally handles vertical features (`RawFeature={+vert}`) automatically.

☞ Next: Correction or Features.

[English Version] | 中文版

Textbox

`luatex-cn` provides powerful vertical text box capabilities with grid alignment.

Basic Textbox

The `\TextBox` (or `\文本框`) command creates an independent vertical text area.

Usage

```
\TextBox[options]{Content}
```

Options

Key	Description	Default
`height`	Height in grid units. 0 for auto-extension.	`0`
`n-cols`	Number of columns.	`0`
`border`	Show border.	`false`
`background-color`	Background color.	None
`floating`	Enable floating mode (needs x, y).	`false`

Features

- **Auto-Extension**: If `height` is 0, the box extends naturally.
- **Compression (Flat Text)**: If `height` is fixed and content is too long, text is vertically compressed (v0.1.5).
- **Independent Indentation** (v0.2.2+): Textbox content no longer inherits paragraph indentation from main text, ensuring layout independence (fix #37).

Fill Textbox

`\FillTextBox` (or `\填充文本框`) is a variant of `\TextBox` with default fill alignment.

```
\FillTextBox[height=10]{Content}
```

☞ Next: Features.

[English Version] | 中文版

Correction & Decoration

`luatex-cn` provides tools for classical text correction and decoration, including correction marks, emphasis marks, and underlines (proper name marks / book title marks).

Correction (GaiZi)

The `\fix` (or `\改`) command marks corrections in classical text editing. It retains the original character with a deletion mark (丶) and places the replacement character at the bottom right.

Usage

```
\fix{replacement}
% or Chinese alias
\改{替换字}
```

Example

```
This has a \fix{correct}wrong text.
```

In this example, "wrong" is marked with a deletion symbol, and "correct" is placed at its bottom-right corner at 60% size.

Technical Details

- **Deletion Mark**: Automatically overlays a deletion point (丶) on the character.
- **Replacement Text**: Placed at bottom-right corner at 60% of original font size.
- **Non-destructive**: Does not affect the original grid layout.

Text Decoration

Preset commands for marking text according to classical or modern standards.

1. Underline Marks

Command	Alias	Description	Use Case
`\Underline`	`\下划线`	Straight line on the left	General highlighting
`\专名号`	-	Straight line on the left	Proper names (Person, Place)
`\WavyUnderline`	`\波浪线`	Wavy line on the left	General highlighting
`\书名号`	-	Wavy line on the left	Book/Article titles

Options (`[options]`):

- `color` : Line color (default: `red`).
- `offset` : Distance from text (default: `0.6em`).

Example:

```
\专名号[color=black]{Confucius}
\书名号{The Analects}
```

2. Emphasis Marks

Command	Alias	Description
`\EmphasisMark`	`\着重号`	Add small circles (◦) to the right of each char
`\圈点`	-	Same as above

Example:

```
\EmphasisMark{Very Important}
```

Generic Decoration

The `\decorate` (or `\装饰`) command allows overlaying arbitrary content on **each character**.

Usage

```
\decorate[options]{content}
```

Parameters

Key	**Description**	**Default**
`char`	Decoration character	`◦`
`color`	Color	`red`
`xoffset`	Horizontal offset (xshift)	`0pt`
`yoffset`	Vertical offset (yshift)	`0pt`
`scale`	Scale factor	`1.0`
`font-size`	Force specific font size	auto

Example

```
% Add a red circle on each character in "important"
\decorate[char=○, scale=0.8, color=red]{important}
```

☞ Next: Textbox or Features.

Debug Mode

`luatex-cn` provides comprehensive debugging features to help troubleshoot layout issues.

Enable/Disable Debug

```
\LtcDebugOn     % or \开启调试
\LtcDebugOff    % or \关闭调试
```

Module-level Debug

Enable debugging for specific modules:

```
\LtcDebugModuleOn{vertical}   % or \开启调试模块{vertical}
\LtcDebugModuleOff{vertical}  % or \关闭调试模块{vertical}
```

Display Helper Tools

Show Page Frame

```
\LtcShowFrame   % or \显示边框
```

Show Grid Coordinates

Overlay grid coordinates on the page for element positioning:

```
% Show grid (default unit: cm)
\LtcShowGrid              % or \显示网格

% Specify unit
\LtcShowGrid[measure=cm]  % or \显示网格[measure=cm]
\LtcShowGrid[measure=pt]
\LtcShowGrid[measure=mm]
```

```
% Hide grid
\LtcHideGrid              % or \隐藏网格
```

Supported units: `cm` (default), `pt`, `mm`

Show Coordinates

`\显示坐标` is an alias for `\显示网格`:

```
\显示坐标[measure=pt]
\隐藏坐标
```

Debug Color Settings

Set different debug colors for different modules:

```
\LtcDebugColor{vertical}{blue}
\LtcDebugColor{banxin}{red}
```

Usage Example

```
\documentclass[四库全书彩色]{ltc-guji}

% Enable debug mode
\LtcDebugOn
\LtcShowGrid[measure=cm]

\begin{document}
\begin{正文}
Debug content...
\end{正文}
\end{document}
```

Debug Output

When debug is enabled, the compilation log outputs detailed layout information including:

- Grid position calculations
- Character positioning
- Banxin rendering process
- Plugin invocation order

☞ Back to: Features | Home

English | 中文版

Changelog

All notable changes to this project will be documented here.

[0.2.5] - 2026-02-05

- ✨ **Catalog Entry Command**: Added `\条目` command with hierarchical indentation for table of contents
- ✨ **Column Control Commands**: Added `\列` (Column) and `\末列` (LastColumn) commands for better layout control (fix #30)
- 🐛 **Bug Fix**: Fixed decoration underlines (专名号/书名号) not drawn on last character at column end (fix #54)
- 🐛 **Bug Fix**: Fixed character centering offset for some characters (e.g., "漢"), main text now uses width-based centering (fix #55)
- 🐛 **Bug Fix**: Fixed sidenote color leakage to banxin when crossing page boundaries (fix #48)

[0.2.4] - 2026-02-03

- ✨ **Underline Support**: Added `\Underline` / `\下划线` commands, implemented with the `|` character (fix #5)
- ✨ **Book Title & Proper Name Marks**: Added `\书名号` (wavy line) and `\专名号` (straight line) commands (fix #16)
- 🐛 **Bug Fix**: Fixed sidenote column alignment when switching columns (fix #47)
- 🐛 **Bug Fix**: Fixed sidenote going to second column when crossing pages (fix #49)
- 🐛 **Bug Fix**: Fixed jiazhu alignment issue (fix #51)
- 🐛 **Bug Fix**: Fixed `\newgeometry` not supporting paperwidth/paperheight (fix #52)
- 🐛 **Bug Fix**: Fixed sidenote color leakage causing banxin to turn red when crossing pages

[0.2.3] - 2026-02-01

- ✨ **MeiPi (Marginal Notes)**: Added `\眉批` command with auto-positioning and manual coordinate support (fix #46)
- ✨ **Special TextBox**: TextBox frame feature with border shape customization (fix #25)
- ✨ **Publisher Display**: Added publisher display below banxin with customizable font (fix #26)
- 🐛 **Bug Fix**: Fixed annotation and textbox incorrectly inheriting paragraph indentation (fix #37)
- 🐛 **Bug Fix**: Fixed MeiPi appearing on wrong page when crossing page boundaries
- 🐛 **Bug Fix**: Fixed font detection support for simplified/traditional Chinese names (fix #35)
- ♻️ **Refactoring**: Refactored border style system with style stack management for border, outer_border attributes

[0.2.2] - 2026-01-31

- 🐛 **Bug Fix**: Fixed preamble chapter display on first page (fix #45)
- 🐛 **Bug Fix**: Fixed jiazhu column jumping at column end (fix #42)
- ✨ **Judou Enhancement**: Support for more Chinese punctuation symbols (fix #40)
- 🐛 **Bug Fix**: Fixed TextBox and annotation inheriting paragraph indent (fix #37)
- ✨ **Debug Feature**: Added debug ruler showing page grid coordinates (fix #31)
- ✨ **Font Detection**: Support simplified/traditional Chinese font names (fix #35)
- ♻️ **Refactoring**: Consolidated banxin text rendering logic
- ♻️ **Refactoring**: Moved title/publisher commands to metadata.sty
- ☑ **Module Cleanup**: Cleaned up banxin, guji, yinzhang module dependencies

[0.2.1] - 2026-01-31

- ✨ **Color Preservation**: Sidenote color preservation across pages (fix #38, #44)
- ✨ **Style Stack Architecture**: Implemented Phase 3 style stack for multi-attribute support
- ✨ **Style Registry**: Phase 2 multi-attribute style storage (color, font_size, grid_height)

- 🐛 **Bug Fix**: Fixed sidenote rendering order, now correctly displays above silk columns (fix #38)
- 🐛 **Bug Fix**: Punctuation mode now correctly applies to sidenote content (fix #39)
- ♻️ **Jiazhu Migration**: Migrated jiazhu functionality to style stack architecture
- ♻️ **Refactoring**: Unified color registration mechanism

[0.2.0] - 2026-01-31

- ✨ **Plugin Architecture**: Refactored to standardized plugin API for consistent component behavior
- ✨ **Global State Management**: Unified configuration via global tables for all modules
- ✨ **Metadata System**: Centralized book metadata management (title, publisher, chapter markers)
- ♻️ **Module Namespace**: Clear separation of core/, guji/, banxin/, util/, decorate/
- ♻️ **Major Refactoring**: Plugin system refactoring with standardized module interfaces

[0.1.5] - 2026-01-28

- ✨ **Multi-font Support**: Added multi-font fallback chain and Chinese font setting commands (fix #23).
- ✨ **New Features**: Implemented `\decorate` and `\fix` functionalities, supporting vertical typesetting correction marks (fix #12).
- ✨ **Textbox Enhancements**: Textbox supports vertical compression (flattening) to fit grid positions (fix #24).
- 🐛 **Bug Fixes**: Fixed layout bugs including 0pt dimension parsing error, Judou positioning offsets, and long title squeezing (fix #19, #20).
- ✅ **Regression Tests**: Introduced automated regression testing program based on PDF rendering comparison.
- ♻️ **Refactoring**: Refactored core rendering logic and improved overall code stability.

[0.1.4] - 2026-01-27

- ✨ Added Judou (punctuation marks) functionality
- 🐛 Fixed paragraph indentation and side note column switching issues

- ♻️ Major code refactoring for better readability and maintainability
- ☑️ Added more test files

[0.1.3] - 2026-01-25

- 🐛 Fixed inaccurate page breaks and page number positioning
- ✨ Annotation coordinates now more accurate, skipping page center to prevent clipping
- 📚 Completely typeset Dream of the Red Chamber (Jiaxu manuscript) Chapter 1
- ♻️ Better compliance with Expl3 syntax
- ☑️ Added l3build test files for regression testing
- ☑️ Established comprehensive Lua unit testing framework with `l3build test` command support
- 🐛 Completely fixed Mock logic and API inconsistency issues in unit tests

[0.1.2] - 2026-01-22

- ♻️ Renamed src to tex
- 🐛 Completely fixed CRLF and BOM character issues on Windows
- 📦 Adopted l3build, see build.lua
- 🐛 Fixed package require paths to ensure release version works out of the box

[0.1.1] - 2026-01-19

Features

- ✨ Added annotation command, still using floating text box underneath

Fixes

- 🐛 Fixed CRLF and BOM character issues on Windows

Documentation

- 📖 Updated documentation with bilingual support

[0.1.0] - 2026-01-18

Major Features

- **Native Geometric Vertical Typesetting Engine**: Deep Lua customization with grid layout system designed for ancient book typesetting
- **Traditional Book Core Elements**: Complete support for Banxin (page center), Yuwei (fishtail), and traditional silk frames
- **Comprehensive Annotation System**: Flexible interlinear notes and side note functionality
- **Ready-to-use Template System**: Preset visual styles including Siku Quanshu and Dream of the Red Chamber manuscripts
- **Cross-platform Font Auto-detection**: Smart recognition of Windows, macOS, and Linux system fonts
- **Electronic Scroll/Traditional Page Support**: Traditional "tube page" logic and electronic reading adaptation
- **Absolute Positioning Seal System**: Support for placing collection seals anywhere on the page

Project Infrastructure

- Automated CTAN packaging and release workflow
- Comprehensive bilingual user manuals and multiple classic ancient book reproduction examples

English | 中文版

Release Process

`luatex-cn` follows an automated release workflow to ensure compliance with CTAN (Comprehensive TeX Archive Network) standards.

Development Workflow

Building and testing are powered by `l3build`:

```
# Run all Lua unit tests
l3build test

# Install to local TEXMFHOME
l3build install

# Create release bundle (.zip)
l3build bundle
```

Automated Publishing (CI/CD)

GitHub Actions handles the packaging:

1. **Source Sync**: The `main` branch is used for active development.
2. **Packaging**:
    - Extracts `src/`, `文档/` (docs), and `示例/` (examples).
    - Translates folders and filenames to English as required by CTAN, using `scripts/file_name_translation.json`.
    - Generates the standardized directory structure (doc, example, tex).
3. **Release Branch**: The packaged content is pushed to the `ctan` branch and published via GitHub Releases.

Versioning

- Version numbers are stored in the root `VERSION` file.
- Significant changes are recorded in the Changelog.
- We follow Semantic Versioning (SemVer) principles.

☞ Next Steps: Check Development Documentation.

English | 中文版

Development Documentation

This document summarizes the core design philosophy, modular architecture, and the complete development workflow for the `luatex-cn` typesetting engine.

1. Core Architecture: Three-Layer Design

`luatex-cn` utilizes a three-layer architecture to decouple layout logic from rendering:

1. **LaTeX Interface Layer (`.sty` , `.cls`)**: Defines user commands; sets attributes and parameters via `luatexbase`.
2. **Coordination Layer (`core_*.lua`)**: Manages logic flow. For instance, `core_main` orchestrates the main process, while `core_textflow` handles interlinear note splitting.
3. **Processing Layer (`layout_*.lua` , `render_*.lua`)**:
   - **Flattening**: Converts TeX node lists into linear Lua tables.
   - **Layout**: Calculates grid coordinates for each character and its associated elements.
   - **Rendering**: Reassembles content based on coordinates and outputs PDF instructions.

Module Directory Structure (v0.2.0+)

```
tex/
├── core/               # Core vertical engine (26 files)
├── guji/               # Ancient book specific features (6 files)
├── banxin/             # Page center system (5 files)
├── splitpage/          # Tube page splitting (2 files)
├── fonts/              # Font auto-detection
├── util/               # Utility modules (including style registry)
├── decorate/           # Text decoration
├── debug/              # Debug tools
└── configs/            # Preset configuration files
```

Plugin System (v0.2.0+)

v0.2.0 introduced a standardized plugin API where each module implements a three-phase interface:

```
plugin = {
  initialize = function(params, engine_ctx) end,  -- Initialization
  layout = function(list, layout_map, engine_ctx, context) end,  -- Layout
  render = function(head, layout_map, params, context, ...) end   -- Rendering
}
```

Style Registry (v0.2.1+)

`luatex-cn-style-registry.lua` provides multi-attribute style storage:

```
-- Register a style, returns unique ID
local style_id = style_registry.register({
  color = {1, 0, 0},
  font_size = 19,
  grid_height = 20
})

-- Retrieve style by ID
local style = style_registry.get(style_id)
```

Supported attributes: `color`, `font_size`, `grid_height`, `font`, etc.

2. Participating in Development

We welcome contributions and Pull Requests (PRs).

2.1 Fork & Workflow

1. **Fork the Repo**: Click the Fork button on GitHub.
2. **Clone**: Clone your forked repository locally.
3. **Coding**: Main source code is located in the `tex/` directory.

> [!TIP] VS Code with LaTeX Workshop is recommended for development.

2.2 Testing

Please ensure all tests pass before submitting a PR.

1. **Install Dependencies**: You need `l3build` installed.

2. **Unit Tests**: Run Lua unit tests:

```
   l3build test
   ```

   This invokes `test/run_all.lua`.

3. **Regression Tests**: We use automated regression testing based on PDF rendering comparison to prevent layout regressions.

   ```
   l3build check
   ```

   This compiles `.lvt` files in `testfiles/` and compares the output logs (`.tlg`) against baselines. If your changes affect layout, ensure these pass. If layout changes are intentional, update baselines:

   ```
   l3build save test-name
   ```

2.3 Release

For release procedures, please refer to Release.

3. Key Technical Details

Vertical Direction (RTT)

Leverages LuaTeX's `dir RTT` attribute. In RTT mode, text flows top-to-bottom, and lines stack right-to-left.

Attribute Management

Uses LuaTeX attributes to pass metadata across languages (e.g., "this glyph belongs to an interlinear note").

> [!WARNING] \selectfont clears all active attributes. Ensure attributes are reset AFTER setting the font size.

Node Ownership

Passing nodes to TeX via `tex.box[n] = node` transfers ownership. Use `node.copy_list()` if you need to reuse nodes.

4. Development Lessons (LEARNING)

- **Color Commands**: Must use normalized RGB (e.g., `0 0 0 rg`), not names.
- **Rendering Order**: PDF follows the "painter's model"—later content overlays earlier content. Insert backgrounds at the head of the list to ensure they are at the bottom.
- **Module Loading**: Standardize on `require()` and utilize `package.loaded` for caching.