The kanbun package

以‍₂羅蝶‍₁作‍₂漢文訓讀‍₁用 包

Yuanhao Chen (陳元鎬)

13 FEBRUARY 2022, V1.2

Contents

1 Introduction

The kanbun package, like other *kanbun-kundoku* (漢文訓讀) LaTeX packages (such as gckanbun), allows users to manually input macros for elements in a *kanbun-kundoku* paragraph.

More importantly, it accepts input with light markup in the "*kanbun* annotation" form when used with LuaLaTeX, which allows typesetting *kanbun-kundoku* paragraphs efficiently[1].

2 Basic example with LuaLaTeX

As seen in the following example, typesetting a *kanbun-kundoku* paragraph with the kanbun package requires only light annotations — it automatically transforms the annotated plain text into LaTeX macros through Lua, rather than having users type in macros themselves.

---

[1] The idea comes from 漢文 HTML by UntPhesoca, which is a JavaScript and CSS implementation.

```
\documentclass{ltjtarticle}
\usepackage[kumi=aki, tateaki=1]{kanbun}
\begin{document}
\Kanbun
月落チ烏啼キテ霜満ツ[レ]天ニ，
江楓漁火対ス[二]愁眠ニ[一]。
姑(こ)蘇(そ)城外ノ寒山寺，
夜半ノ鐘声到ル[二]客船ニ[一]。
\EndKanbun
\printkanbun
\end{document}
```

月落チ烏啼キテ霜満ツレ天ニ，
江楓漁火對ス二愁眠ニ一。
姑ニ蘇ソ城外ノ寒山寺，
夜半ノ鐘聲到ル二客船ニ一。

  Note that if you want to use this functionality, you have to run this document with LuaLaTeX.

3 Usage

3.1 Package options

  Load the package with

`\usepackage[`⟨*kanbun options*⟩`]{kanbun}`

  This package provides a variety of customisable features in *kanbun-kundoku*.

`scale=`⟨*ratio*⟩

Sets the ratio of the size of *kanji* to that of ruby texts. Default: 2.

fontcmd=⟨*font command*⟩

Sets the font command to use for *kanji*. If `luatexja-fontspec` is loaded, it is set default to `\addjfontfeatures{RawFeature={+trad}}` to obtain traditional *kanji* if applicable.

rubyfontcmd=⟨*font command*⟩

Sets the font command to use for ruby texts. If `luatexja-fontspec` is loaded, it is set default to `\addjfontfeatures{RawFeature={+ruby}}` to obtain ruby glyphs when applicable.

unit=⟨*length*⟩

Sets the dimensions of *kanji* (assuming it is a square). Default: `\kanbun_init_zw`, which is a length equal to `1em`.

yokoaki=⟨*ratio*⟩

Sets the horizontal space between *kanji* with respect to the size of ruby texts. Default: `2`.

tateaki=⟨*ratio*⟩

Sets the vertical space between *kanji* with respect to the size of ruby texts. Default: `2`.

okuriintrusion=⟨*ratio*⟩

Sets the intrusion of *okurigana* (how much *okurigana* should be vertically tucked into the space of *kanji*) with respect to the size of ruby texts, if that does not cause an overlap with *furigana*. Default: `1`.

kumi=aki, beta

Sets whether to use *aki-gumi* (typeset with uniform inter-character spacing) or *beta-gumi* (typeset with no inter-character space between adjacent character frames). Or simply call `aki` or `beta` without `kumi=`. Default: `aki`.

After initialising the options, you can still change the option values through the command `\setkanbun`. For example, to switch to *beta-gumi*, you could use

```
\setkanbun{beta}
```

3.2 Basic usage without LuaLaTeX (not recommended)

When not using the advanced *kanbun*-annotation functionality, it is possible to typeset *kanbun* with any engine with CJK support, such as using XƎLaTeX with the xeCJK package, or using upLaTeX with utarticle or other appropriate class.

`\kanjiunit{` `\furiokuri{`⟨*right furigana*⟩`}{`⟨*right okurigana*⟩`}` `}`
`{`⟨*left (top) punctuation (e.g. 「)*⟩`}`
`{`⟨*kanji*⟩`}`
`{`⟨*other punctuation*⟩`}`
`{`⟨*kaeriten*⟩`}`
`{` `\furiokuri{`⟨*left furigana*⟩`}{`⟨*left okurigana*⟩`}` `}`

`\kanbunfont`

Sets the font size of *kanji*. Use when the unit option is set different to the document's default font size.

Use `\multifuriokuri` instead of `\furiokuri` if you are putting *furigana* to multiple kanji.

`\multifuriokuri[`⟨*length by which ruby is raised*⟩`]{`⟨*furigana*⟩`}{`⟨*okurigana*⟩`}`

For example, the code

```
% example text from https://phesoca.com/kanbun-html/
\kanbunfont
\kanjiunit{}{}{子}{}{}{}
\kanjiunit{\furiokuri{}{ク}}{}{曰}{，}{}{}
\kanjiunit{\furiokuri{}{ゾ}}{}{盍}{}{三}{\furiokuri{}{ル}}
\kanjiunit{}{}{各}{ゝ}{}{}
\kanjiunit{\furiokuri{}{ハ}}{}{言}{}{二}{}
\kanjiunit{\furiokuri{}{ノ}}{}{爾}{}{}{}
\kanjiunit{\furiokuri{}{ヲ}}{}{志}{。}{一}{}
\par
```

outputs

子
曰ク，
盍ゾ三
各ゝ
言ハ二
爾ノ
志ヲ一。

with tateaki set to 1.

4

3.3 Usage with LuaLaTeX

Kanbun annotation uses the following brackets to mark different elements in *kanbun-kundoku* (as described in 漢文 HTML by UntPhesoca).

- ( ): *furigana* (振仮名)

- { }: *okurigana* (送仮名) (these brackets can be omitted)

- ‹ ›: *furigana* (振仮名) of *saidoku-moji* (再読文字)

- « »: *okurigana* (送仮名) of *saidoku-moji* (再読文字)

- [ ]: *kaeriten* (返点)

- ' ': multiple *kanji*, potentially with *tateten* inserted, as a ruby base; group ruby (グループルビ)

- no annotation: *kanji* (漢字) and punctuation

Tateten (竪点) can be input with either — (U+2015), — (U+2014) or 〡 (U+3190).

\Kanbun
\EndKanbun

Write the annotated *kanbun* between the commands \Kanbun and \EndKanbun, and it will be processed and saved, ready to be used later.

\printkanbun

Where you would like to use the most recently saved *kanbun-kundoku* paragraph, use \printkanbun. It automatically calls \kanbunfont and \par to set the font size and line spacing and make paragraphs.

\printkanbuncode

If you wish to make modifications on the result or to use the result with a non-LuaTEX engine, it is possible to obtain the macros using \printkanbuncode (prints in the terminal), and continue to work from there.

\printkanbunnopar

The no-paragraph counterpart of \printkanbun, which does not invoke \kanbunfont, and uses
instead of \par.

The no-paragraph counterpart of \printkanbuncode.

You can always save \printkanbun to a macro for use later and start a new annotated *kanbun* block, as in the following example.

```
% example text from https://phesoca.com/kanbun-html/
\documentclass{ltjtarticle}
\usepackage[kumi=beta]{kanbun}

\Kanbun
此レ乃チ信(しん)之'所一[三]以'(ゆゑん)為ル[二]陛下ノ禽(とりこ)ト[一]也。
\EndKanbun
\let\信\printkanbun

\Kanbun
孤之有ルハ[二]孔明[一]，猶ホ‹ごと›«キ»[二]魚之有ルガ[一レ]水也。
\EndKanbun
\let\孔明\printkanbun

\begin{document}
\孔明\par\bfseries\信
\end{document}
```

孤之有ルハ孔明，猶ホ魚之有ルガ水也。
此乃信之所以爲陛下禽也。