The kanbun package

Yuanhao Chen (陳元錦)

13 FEBRUARY 2022, V1.2

Contents

1 Introduction 1

3 Usage 2

3.1 Package options . . . . 2

3.2 Basic usage without LuaTEX 4

3.3 Usage with LuaTEX . . . . 5

2 Basic example with LuaTEX

1 Introduction

The kanbun package, like other kanbun-kundoku (漢文訓読) \TeX\ 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 Lua\TeX, which allows typesetting kanbun-kundoku paragraphs efficiently¹.

2 Basic example with Lua\TeX

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 \TeX\ macros through Lua, rather than having users type in macros themselves.

¹The idea comes from 漢文 HTML by UntPhesoca, which is a JavaScript and css implementation.
月落チ烏啼キテ霜満ツレニ，
江楓漁火対スニ愁眠ニ一。
姑（こ）蘇（そ）城外ノ寒山寺，
夜半ノ鐘声到ルニ客船ニ一。

夜 姑 江 月
半 め 塔 落
鐘 城 漁 鳥
声 外 火 喋
到 寒 對 霜
客 山 愁 滿
船 寺 眠 天

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.

okuriinclusion=⟨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 Lua\TeX{} (not recommended)

When not using the advanced \textit{kanbun}-annotation functionality, it is possible to typeset \textit{kanbun} with any engine with \textit{cjk} support, such as using Xe\LaTeX{} with the \texttt{xeCJK} package, or using up\LaTeX{} with \texttt{utarticle} or other appropriate class.

\begin{verbatim}
\kanjiunit{ \furiokuri{(right furigana)}{{(right okurigana)}} }\linebreak
{(left (top) punctuation (e.g. ’‘)}\linebreak
{(kanji)}\linebreak
{(other punctuation)}\linebreak
{(kaeriten)}\linebreak
\furiokuri{(left furigana)}{(left okurigana)} }
\end{verbatim}

\texttt{\kanbunfont}

Sets the font size of \textit{kanji}. Use when the unit option is set different to the document’s default font size.

Use \texttt{\multifuriokuri} instead of \texttt{\furiokuri} if you are putting \textit{furigana} to multiple \textit{kanji}.

\begin{verbatim}
\multifuriokuri{(length by which ruby is raised)}{(furigana)}{(okurigana)}
\end{verbatim}

For example, the code

\begin{verbatim}
\% example text from https://phesoca.com/kanbun-html/
\kanbunfont
\kanjiunit{}{}{子}{}{}}\linebreak
\kanjiunit{}{}{ク}{曰}{}{}}\linebreak
\kanjiunit{}{}{ゾ}{曰}{}{}}\linebreak
\kanjiunit{}{}{ハ}{言}{}{}}\linebreak
\kanjiunit{}{}{ノ}{爾}{}{}}\linebreak
\kanjiunit{}{}{ヲ}{志}{}{}}\linebreak
\par
\end{verbatim}

outputs

\begin{verbatim}
言, 子
爾, 曰
志, 曰
各
\end{verbatim}

with tateaki set to 1.
3.3 Usage with Lua\TeX

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 (堅点) can be input with either — (U+2015), ― (U+2014) or ㈠ (U+3190).

\begin{verbatim}
\Kanbun
\end{verbatim}

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

\begin{verbatim}
\printkanbun
\end{verbatim}

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.

\begin{verbatim}
\printkanbuncode
\end{verbatim}

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

\begin{verbatim}
\printkanbunnopar
\end{verbatim}

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

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}
\begin{document}
\Kanbun
此レ乃チ信（しん）之‘所－[三]以’（ゆゑん）為ル[二]陛下ノ禽（とりこ）ト[一]也。
\EndKanbun
\let\信\printkanbun

\Kanbun
\EndKanbun
\let\孔明\printkanbun
\begin{document}
\Par\bfseries孔明
\end{document}