

Guide to Japanese typesetting with \LaTeX

Japanese \TeX Development Community*

December 6, 2020

Typesetting requirements of Japanese are completely different from those of Western languages, so setting parameter values and additional hyphenation patterns are not enough.

The package `babel-japanese` provides “japanese” option for Babel package. However, it only defines translated replacement texts for keywords and dates (e.g. “Contents” → “目次”, `\today` → “令和2年12月6日”) and it does not care about any Japanese typesetting requirements.

This document describes how to get an “acceptable” Japanese typesetting result and the eminent differences between Western and Japanese typesetting requirements.

1 Short introduction: using \LuaTeX

To obtain a properly formatted Japanese document, the easiest way is to use \LuaTeX with \LuaTeX -ja package.

- Using the `jlreq` class:

```
1 \documentclass[book]{jlreq}
2 \begin{document}
3 \chapter{日本語の文書}% Japanese document
4 こんにちは, 日本。% Hello, Japan.
5 \end{document}
```

Save the above as `jp1.tex` and run the following command.

```
$ lualatex jp1
```

You will get `jp1.pdf` which is properly formatted and a Japanese font family “HaranoAji Mincho/Gothic” is embedded.

* <https://texjp.org>, e-mail: [issue\(at\)texjp.org](mailto:issue(at)texjp.org)

- If you prefer classes not for Japanese, you can load `luatexja` directly:

```
1 \documentclass{book}
2 \usepackage{luatexja}
3 \begin{document}
4 ...
```

* Note that using non-Japanese classes will lead to unnatural page layouts for Japanese books, especially regarding line gaps and line lengths.

2 Relatively stable alternative: up \LaTeX

Lua \LaTeX is relatively new, and Lua \TeX -ja is under active development. As a more stable alternative, you can use up \LaTeX . It is a somewhat legacy implementation, which requires TFM files for typesetting Japanese characters (called JFM).

The `jlreq` class also supports up \LaTeX , but here we provide an example using an relatively old class `jsclasses`, which has been widely used for over a decade:

```
1 \documentclass[uplatex]{jsbook}% or, \documentclass{ujbook}
2 \begin{document}
3 ...
```

Save the above as `jp2.tex` and run the following command.

```
$ uplatex jp2
```

You will get `jp2.dvi`.

up \LaTeX and p \LaTeX (explained later) are often referred to as (u)p \LaTeX . They always output DVI files. To convert DVI containing Japanese characters into PDF, the easiest way is to use `dvipdfmx`. If you have `jp2.dvi`, running the following command

```
$ dvipdfmx jp2
```

will generate `jp2.pdf`.

Since the default DVI driver in \TeX Live is set to `dvips`,¹⁾ it is recommended to add a global driver option to be passed to all driver-dependent packages (e.g. `graphicx`, `color`, `hyperref`):

```
1 \documentclass[uplatex,dvipdfmx]{jsbook}
2 \usepackage{graphicx}
3 \begin{document}
4 ...
```

1) Though `dvips` supports DVI with Japanese fonts, additional settings for Ghostscript are required to convert the resulting PostScript file into PDF.

3 Using legacy p \LaTeX

Some old Japanese journal classes may require p \LaTeX . It is a legacy implementation which was born in 1980s, before the release of Unicode 1.0. p \LaTeX only supports a limited character set, JIS X 0208 (6879 characters).

```
1 \documentclass{jsbook}% or, \documentclass{jbook}
2 \begin{document}
3 ...
```

Save the above as jp3.tex and run the following command.

```
$ platex jp3
```

You will get jp3.dvi. Again, the DVI file can be converted into PDF using dvipdfmx.

Note for users who know the CJK package: (u)p \LaTeX does *not* support the CJK package! The reasons are the followings:

- Both the (u)p \LaTeX kernel and CJK.sty redefine the command `\selectfont`, and those definitions are incompatible.
- The input handling of (u)p \TeX engine is different from that of Western \TeX engines, and Japanese tokens are no longer treated as active characters.

Anyway, you will not need the CJK package at all when using (u)p \LaTeX .

4 Differences between Western and Japanese typesetting

The sections above described some practical ways to achieve “acceptable” Japanese typesetting results. So, what are the minimum requirements for “acceptable” results for native Japanese readers?

The W3C Working Group Note “Requirements for Japanese Text Layout” [1] is one of the comprehensive documentation which describes issues of text composition in the Japanese writing system. However, they are too complicated to explain here thoroughly,²⁾ so we focus on the main issues regarding simple and short horizontal writing within a few lines.

Consider the following input, Listing 1. Using Lua \LaTeX , it is allowed to include any Unicode character out-of-the-box, though a suitable Japanese font should be declared.

2) They include specification of basic page layout (*kihon-hanmen*; 基本版面), vertical alignment of Japanese text (*tate-gumi*; 縦組), small-sized supplementary text (*ruby*; ルビ), inline cutting note (*warichu*; 割注) etc.

Listing 1 Example input using Lua^LA_TE_X

[illegible]

However, the result (Figure 1) is not acceptable, especially for the following reasons:

- Japanese text has no interword spaces, but line break is allowed between any Japanese characters except before some punctuation marks (*kinsoku*; 禁則). The overfull rules that appeared in Figure 1 come from not knowing such line breaking rules.
- Similarly, it is common to insert a new line character after arbitrary Japanese characters in the input source. Such a new line character should not be treated as a blank space.

Wrong: 改行後空白が入る。... Correct: 改行後空白が入る。...

- Ordinary Japanese characters are designed to have virtual full-width (*zenkaku*) bounding box. However, spacing between adjacent punctuation marks must be truncated.

Wrong: 句読点と、(括弧)の間... Correct: 句読点と、(括弧)の間...

- It is preferred to insert a quarter em space (*shibu aki*) between Japanese and Latin characters.³⁾

Wrong: 日本語とLatinの間... Correct: 日本語とLatinの間...

Figure 2 shows an acceptable output of the same text. You will see that choosing a Japanese font on Unicode-enabled engine is not enough for obtaining properly formatted result.

All of the methods explained in the previous sections, namely

1. Lua^AT_EX with LuaT_EX-ja package,
2. up^AT_EX,

3) Such a space is often called “xkanjiskip” glue in Japanese TeX world.

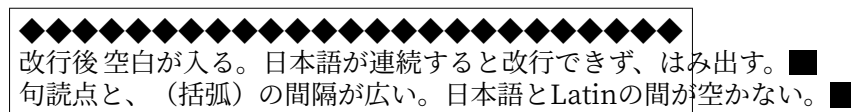


Figure1 Output from Listing 1 (not acceptable)

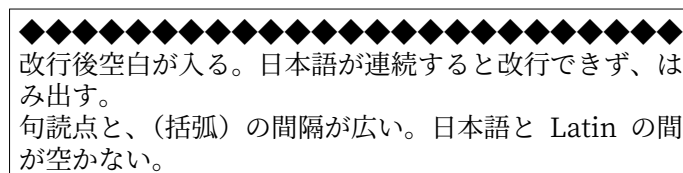


Figure2 Acceptable output of the above text

3. p_LA_TE_X,

are developed to support (most of) the requirements mentioned above, and their output is most likely to be acceptable. Therefore, we recommend to use one of those platforms.

5 Using multi-weight typefaces for Japanese typesetting

Latin font families used in Western typesetting consist of many typefaces:

- Serif: Upright, *Italic*, **Bold**, ***BoldItalic***
- Sans Serif: Upright, *Italic*, **Bold**, ***BoldItalic***
- Typewriter: Upright, *Italic*, **Bold**, ***BoldItalic***

In contrast, Japanese font families provide limited typefaces. Most Japanese font families do not provide italic or slanted shapes. Rather, these shapes look strange and have rarely been used in traditional typesetting. Also, there is no typewriter font; ordinary Japanese typefaces are inherently designed to have a fixed width.

- 明朝 (*Mincho*; Japanese Serif)
- ゴシック (*Gothic*; Japanese Sans Serif)

Therefore, the typeface used to emphasize in Japanese text is different from those often used in Western text:

- Normal text (in upright shape) and *Emphasized text (in italic shape)*
- 通常の文 (in *Mincho*) と, 強調された文 (in *Gothic*) か強調された文 (with *Kenten*; 圈点)

Most of common Japanese classes, including those used in the previous sections, use

Mincho and *Gothic* with a single weight each. Nowadays, some multi-weight Japanese fonts are available for free, so using them is a good way to overcome the situation. The [deluxe] option of packages `luatexja-preset` (on `Lua \LaTeX`) or `otf` (on `(u)p \LaTeX`) supports maximum of 7 typefaces:

- 細明朝, 明朝, 太明朝 (*Mincho* Light, Medium and Bold)
- ゴシック, 太ゴシック, 極太ゴシック (*Gothic* Medium, Bold and Extra Bold)
- 丸ゴシック (*Maru Gothic* Medium)⁴⁾

With the help of these packages, you can emphasize Japanese text by changing to bold, instead of changing Serif to Sans Serif.

- 通常の文 (in *Mincho*) と, 強調された文 (in *Mincho* Bold)
- 通常の文 (in *Gothic*) と, 強調された文 (in *Gothic* Bold)

6 Further readings

- “The `Lua \TeX -ja` package” (`luatexja-en.pdf`, in English)
 - For beginners, refer to Part I “User’s manual.”
 - For more interested users, refer to Part II “Reference.”
 - For developers, refer to Part III “Implementations.”
 - Japanese edition: `luatexja-ja.pdf`
- “Guide to `p \TeX` for developers unfamiliar with Japanese” (`ptex-guide-en.pdf`, in English)
 - For developers who aim to support `p \TeX` /`p \LaTeX` and its variants `up \TeX` /`up \LaTeX` .
- “About `up \LaTeX 2 ϵ` ” (`uplatex-en.pdf`, in English)
 - For users of `up \LaTeX` , but a somewhat legacy document.
 - Japanese edition: `uplatex-en.pdf`
- “About `p \LaTeX 2 ϵ` ” (`platex-en.pdf`, in English)
 - For users of `p \LaTeX` , but a somewhat legacy document.
 - Japanese edition: `platex-en.pdf`
- “Babel-Option `japanese`” (`babel-japanese.pdf`, in both English and Japanese)
 - Basic introduction of `japanese` option of `babel`.
 - Provides translated replacement texts for keywords and date.

4) The HaranoAji family used in this documentation does not provide *Maru Gothic* (rounded *Gothic*) typefaces.

- Note: the translations provided by this package may not become effective on most of Western classes, due to lack of appropriate placeholders inside captions.

7 Japanese deliverables in T_EX Live

There are many packages developed specially for Japanese typesetting in T_EX Live; most of these are installed as part of collection-langjapanese. Only a subset of these useful packages are shown below. For more details, please refer to individual documentation.

7.1 Document classes

Japanese text can be aligned in two directions. One is horizontal direction, called *yoko-gumi* (横組), the other is vertical direction, called *tate-gumi* (縦組). The list below shows *tate* classes in green and *yoko/tate* classes in blue (default is *yoko*, and the class option [tate] switches to *tate* mode).

All the major frameworks for Japanese typesetting provide “standard classes”:

- LuaT_EX-ja (luatexja by LuaT_EX-ja project team)
 - Bundle: ltjsclasses (ltjsarticle, ltjsbook, ltjsreport)
 - Bundle: ltjclasses (ltjarticle, ltjbook, ltjreport, ltjtarticle, ltjtbook, ltjtreport)
- upL_AT_EX (uplatex by Japanese T_EX Development Community)
 - Bundle: ujclasses (ujarticle, ujbook, ujreport, utarticle, utbook, utreport)*
- pL_AT_EX (platex by Japanese T_EX Development Community)
 - Bundle: jclasses (jarticle, jbook, jreport, tarticle, tbook, treport)*

* Due to historical reasons, ujclasses and jclasses are left with some inconvenient behaviors. You may need to adjust some layout parameters by hand, or use other classes mentioned below.

Other major classes designed for Japanese:

- Class: jlreq (by Noriyuki Abe)
 - Supports LuaL_AT_EX, upL_AT_EX and pL_AT_EX.
 - This class aims to implement “Requirements for Japanese Text Layout” (JLReq, 日本語組版処理の要件, [1]). The class file and necessary JFM (Japanese font metric) files are included.
- Bundle: jsclasses (by Haruhiko Okumura & Japanese T_EX Development Community)

- Class: `jsarticle`, `jsbook`, `jsreport`
 - * Supports only $\text{up}\text{\LaTeX}$ (with option `[uplatex]`) and $\text{p}\text{\LaTeX}$.
 - * In addition to layout adjustments, these classes use newly developed JFM set (jis metrics) to avoid strange behaviors of standard $\text{p}\text{\TeX}$ JFM set (min10 etc). For more details, please refer to [2].
 - * Note: when font size other than 10pt is specified, the engine primitive `\mag` is employed. Conflicts with other packages can lead to inconsistent results in graphics and/or page size. To avoid this, specify `nomag` or `nomag*` option.

7.2 Packages

- Package: `plautopatch` (by Hironobu Yamashita)
 - Supports $\text{up}\text{\LaTeX}$ and $\text{p}\text{\LaTeX}$, but does no harm on other \LaTeX (silently ignored).
 - (u) $\text{p}\text{\LaTeX}$ users are recommended to load this package *always*, to resolve errors and incompatibilities with Western \LaTeX packages which are not aware of (u) $\text{p}\text{\LaTeX}$.
- Package: `pxchfon` (by Takayuki Yato)
 - Supports $\text{up}\text{\LaTeX}$ and $\text{p}\text{\LaTeX}$.
 - Helps users to declare physical fonts embedded in `dvipdfmx`'s PDF output. It utilizes `\special{pdf:mapline}` or `\special{pdf:mapfile}` syntaxes.
- Package: `otf` (japanese-otf, japanese-otf-uptex by Shuzaburo Saito & TTK)
 - Supports only $\text{up}\text{\LaTeX}$ (with option `[uplatex]`) and $\text{p}\text{\LaTeX}$.
 - Main feature 1: allows switching between seven different typefaces, compared to the standard of two typefaces. This function is realized by shipping seven “logical fonts” (= set of a JFM and a virtual font).
 - Main feature 2: provides commands for outputting many characters by specifying Unicode (`\UTF{ . . . }`) or Adobe CID (`\CID{ . . . }`), even with $\text{p}\text{\LaTeX}$. This function is realized by mapping a huge real font from lots of smaller subset virtual fonts. A series of shorthand commands (`\aj . . .`) are also available.
 - $\text{Lua}\text{\TeX}$ -ja provides `luatexja-otf` to partially emulate feature 2 of this package.

7.3 Other support files

- `ptex-fontmaps`
 - Provides map files to be used with `dvipdfmx` and `dvips`.
 - To customize Japanese fonts to be used in the output, please use the command

`kanji-config-updmap.`

- `cjk-gs-integrate`

- Supports easy configuration of Ghostscript for CJK (Chinese, Japanese and Korean) fonts, which is necessary for processing of PostScript file output from dvips.

References

- [1] W3C Working Group, *“Requirements for Japanese Text Layout”*.
<https://www.w3.org/TR/jlreq/?lang=en>
- [2] Haruhiko Okumura, *“pT_EX and Japanese Typesetting”*. The Asian Journal of T_EX, Volume 2, No. 1, 2008.
<http://ajt.ktug.org/2008/0201okumura.pdf>

A Examples of acceptable Japanese typesetting results

A.1 A simple Japanese document

Using Lua \LaTeX :

```
1 \documentclass[book]{jlreq}
2 \begin{document}
3 \chapter{日本語がメインのシンプルな文書}
4 こんにちは、日本語に少しEnglishを含めてみます。
5 \end{document}
```

Using up \LaTeX (or p \LaTeX) + dvipdfmx:

```
1 \RequirePackage{plautopatch}% recommended
2 \documentclass[book,dvipdfmx]{jlreq}% driver option
3 \begin{document}
4 \chapter{日本語がメインのシンプルな文書}
5 こんにちは、日本語に少しEnglishを含めてみます。
6 \end{document}
```

Using pdf \LaTeX or X \LaTeX : though they have only limited Japanese typesetting capabilities, bxjsbook (from bxjscls bundle by Takayuki Yato) aims to emulate jsclasses as much as possible. Due to technical constraints, manual adjustments are often required in the input.⁵⁾

```
1 \documentclass[pdflatex,ja=standard]{bxjsbook}
2 %\documentclass[xelatex,ja=standard]{bxjsbook}
3 \begin{document}
4 \chapter{日本語がメインのシンプルな文書}
5 こんにちは、日本語に少し~English~を含めてみます。
6 \end{document}
```

A.2 Changing Japanese fonts

The default Japanese font family is “HaranoAji Mincho/Gothic” which is derived from Source Han (or Noto) fonts. Suppose you want to switch to “IPAex Mincho/Gothic” ...

Using Lua \LaTeX :

```
1 \documentclass[book]{jlreq}
2 \usepackage[ipaex]{luatexja-preset}% pre-defined
3 ...
```

5) pdf \TeX and X \TeX cannot insert a quarter em space (*shibu aki*) automatically, so the active character ~ is redefined to a macro to insert it manually.

Using up \LaTeX (or p \LaTeX) + dvipdfmx:

```
1 \RequirePackage{plautopatch}% recommended
2 \documentclass[book,dvipdfmx]{jlreq}% driver option
3 \usepackage[ipaex]{pxchfon}% pre-defined
4 ...
```

A.3 More typefaces

Using Lua \LaTeX :

```
1 \documentclass[article]{jlreq}
2 \usepackage{jlreq-deluxe}% multi weight
3 \begin{document}
4 \section{簡単な文章}
5 通常のテキストにはSerifと明朝体を使います。
6 \textbf{太字も使うことができます。}
7 \textsf{Sans Serifにはゴシック体が似合います。}
8 \textbf{太字にも対応します。}}
9 \end{document}
```

If you are using Lua \LaTeX with classes other than jlreq, you can also use luatexja-preset:

```
1 \documentclass{ltjsarticle}
2 \usepackage[deluxe]{luatexja-preset}% multi weight
3 \begin{document}
4 ...
```

Using up \LaTeX (or p \LaTeX) + dvipdfmx:

```
1 \documentclass[article,dvipdfmx]{jlreq}% driver option
2 \usepackage{jlreq-deluxe}% multi weight
3 \begin{document}
4 ...
```

The package jlreq-deluxe is relatively new, and it is designed to be used with the jlreq class.

Traditionally the otf package has been widely used for over a decade.

Using up \LaTeX + dvipdfmx:

```
1 \documentclass[uplatex,dvipdfmx]{jsarticle}
2 \usepackage[deluxe]{otf}% multi weight
3 \begin{document}
4 ...
```

Using p \LaTeX + dvipdfmx:

```
1 \documentclass[dvipdfmx]{jsarticle}
2 \usepackage[deluxe]{otf}% multi weight
3 \begin{document}
4 ...
```

A.4 Vertical writing

As a Japanese version of lipsum, here we use `bxjalipsum`. It provides several sample texts.

Using `LuaLTeX`:

```
1 \documentclass[article,tate]{jlreq}
2 \usepackage{bxjalipsum}
3 \begin{document}
4 \title{吾輩は猫である}% I Am A Cat
5 \author{夏目漱石}% Natsume Soseki
6 \maketitle
7 \jalipsum{wagahai}% 吾輩は猫である。名前... (33 paragraphs)
8 \end{document}
```

Using `upLTeX` (or `pLTeX`) + `dvipdfmx`:

```
1 \documentclass[article,tate,dvipdfmx]{jlreq}% driver option
2 \usepackage{bxjalipsum}
3 \begin{document}
4 % The preamble of the Constitution of Japan
5 \jalipsum{preamble}% 日本国民は、正当に選... (4 paragraphs)
6 \end{document}
```

A.5 Beamer with Japanese

Note that the class `beamer` is not designed for Japanese.

Using `LuaLTeX`:⁶⁾

```
1 \documentclass[unicode,12pt]{beamer}% ensure unicode for hyperref
2 \usepackage{luatexja}% for Japanese
3 \renewcommand{\kanjifamilydefault}{\gtdefault}% sans-serif also for Japanese
4 \usetheme{Copenhagen}% as you like!
5 \begin{document}
6 \section{日本語のサンプル}
7 \begin{frame}{日本語でBeamerを使う}
8 こんにちは、日本語に少しEnglishを含めてみます。
9 \end{frame}
10 \end{document}
```

Using `upLTeX` + `dvipdfmx`:

```
1 \RequirePackage{plautopatch}% loads (at least) the pxjahyper package
2 \documentclass[dvipdfmx,12pt]{beamer}% driver option
3 \usepackage{bxdpj-beamer}% additional settings for dvipdfmx
```

6) Since `hyperref` 2020-08-14 v7.00f, the option `unicode` is enabled by default, so the explicit option is already redundant.

```
4 \renewcommand{\kanjifamilydefault}{\gtdefault}% sans-serif also for Japanese
5 ...
```

Using p^AT_EX + dvipdfmx:

```
1 \RequirePackage{plautopatch}% loads (at least) the pxjahyper package
2 \documentclass[dvipdfmx,12pt]{beamer}% driver option
3 \usepackage{bxdpj-beamer}% additional settings for dvipdfmx
4 \usepackage{minijs}% avoid pTeX's standard JFM (min10 etc.)
5 \renewcommand{\kanjifamilydefault}{\gtdefault}% sans-serif also for Japanese
6 ...
```