Japanese LaTeX

Christian Külker

2021-07-09

Contents

1 Abstract

This document describes the usage of the Japanese writing system with LaTeX in general and with some flavors of LaTeX namely pdfLaTeX in particular. It also tries to highlight the difference between some LaTeX flavours as well as give examples for Japanese LaTeX documents. This is not a legal advice.

2 History

Originally LaTeX had been developed in times prior to Unicode (and UTF-8). TeX79 had 7 bit, TeX82 had 8 bit and are not capable handling Unicode via UTF-8 directly. Japanese at that time required encodings like ISO/IEC-2022 (ECMA-35, ANSI X3.41 JIS X 0202), with its variations ISO-2022-JP/ISO-2022-JP-2 and Shift JIS (SJIS, JIS X 0208 Appendix 1) 1997.

Some early approaches tried to overcome the problem that LaTeX was not possible to be used with Japanese by rewriting or changing the TeX/LaTeX engine to support Japanese letters. One of this approaches was `pTeX/pLaTeX` supporting LaTeX2e on Unix (BSD, Linux) from the ASCII Corporation or `jTeX` from NTT which was popular at the turn of the millennium. Later pLaTeX was incorporated into TeX Live in 2010. In 2011 pLaTeX switched from pTeX to εTeX engine but pLaTeX is relative unchanged since 2006, which led to a community fork from ptex-texmf-2.5. pLaTeX was updated to support LaTeX2e from 2015.

The Japanese `pTeX/pLaTeX` used the Yen symbol instead of the backslash and the input encoding of texts at that time was usually ISO-2022-JP and JIS or Shift-JIS for other applications. Together with special document classes like `jbook`, `jreport`, `jarticle` writing Japanese texts have been quite straight forward. While it might technical still be possible to use this 2006 engines some modern LaTeX packages might not work well with them and this approaches are usually restricted to 2 byte old encoding formats. Using TeX Live `pTeX/pLaTeX` or even better community edition might be advised. The conversion to PDF via ghostscript's DVI needed by `pTeX` seems a little bit cumbersome theses days. And finally theses systems do not support UTF-8 with its new characters. This approaches worked best with Japanese text that did not included special characters from other languages and supported Japanese language features as vertical text.

For the West the way to use LaTeX with European languages (like German with its umlauts) and the Japanese language together was to include special packages like `CJK` to declare certain areas of a LaTeX document to allow to contain different (Japanese) letters. While this made it possible to include Japanese in European texts or even mix them, it required a lot of extra markup added to the text and not all cases like the size of European characters compared to Japanese characters are dealt with properly. Also some environments did not work on Japanese text parts. Word wrapping and other issues arose, as Japanese is second class citizen to LaTeX with this packages.

Modern approaches are the porting of ASCII pTeX to εTeX or the upTeX (Unicode-implementation of pTeX) or εpTeX (Merge of pTeX and εTeX) try to fill the gaps of modern Engines to deal with Japanese support. LuaTeX with its ability to hook functions to the TeX kernel went a different way by implementing pTeX features as Lua callbacks in the luatexja package. However, implementing all feature of pTEX was impossible, Hence luatexja is not a just porting of pTeX.

3 Contemporary LaTeX

While modernized versions of the above engines exists nowadays, modern engines are added to the mix. To many as to be counted in this document. Therefore this document will focus on the modern TeX Live 2021 engines: pLaTeX, pdfLaTeX, XeLaTeX and LuaLaTeX.

While the modern TeX Live 2021 pLaTeX supports Japanese UTF-8 directly and XeLaTeX or LuaLaTeX might support Japanese directly, usually an external package, like CJK (with its flavours) or others are used in conjunction with XeLaTeX, LuaLaTeX or pdfLaTeX. The CJK project and package nowadays includes CJKutf8 for pdfLaTeX, xeCJK for XeLaTeX. LuLa-TeX uses the luatexja package.

While it might be possible to use Japanese without additional packages like CJK having this packages usually comes with the perk/burden of defining a secondary font set. One set for western languages and one set for CJK languages. When using characters that are not available in the Japanese font this might be the only solution (However UTF–8 fonts do not suffer from character limitation as the fonts in the past). Switching the secondary font however is relative easy with CJK while the set of available fonts differs between flavor of LaTeX's.

From a Japanese standpoint the need for handling a second set of fonts and switching sections of text via markup might look cumbersome and since it works with a modern pTeX and UTF-8 without, CJK seems obsolete.

4 Differences of XeLaTeX, LuaLaTeX and PDFLaTeX

LuaTeX and XeTeX are UTF-8 engines. This means that the input files can contain characters that with pdfTeX are difficult to use directly and need to be used indirect. LuaTeX and XeTeX can also use system fonts, again in contrast to pdfTeX where additional steps are needed.

XeTeX uses system-specific libraries and it is very easy to use for loading system fonts and other UTF-8 tasks. Traditional TeX struggles with this. This makes XeLaTeX an easy to use engine for end users, particularly if the `fontspec` package is used. As the OS is used there is a trade-off in flexibility and reliability.

LuaTeX has the scripting language (Lua) added that opens the internal of TeX to scripts. A lot more is possible. In contrast to XeLaTeX `fontspec2` has to be used to deal with system fonts.

5 Babel and CJK Is Not Enough

Typesetting the Japanese language require a complete different approach than Western languages. Setting some parameters and new patterns for hyphenation is not enough.

The `babel` package offers only translations, like 目次 for “Table of Contents” or “Contents”, it does not care about Japanese typesetting. Also it might seem OK to switch to Japanese fonts with CJK via Unicode, but also here the details for Japanese typesetting are not implemented. A LaTeX engine that is well used, documented and battle proven is pLaTeX. However it is somewhat old and difficult to use with UTF-8 (in the original distribution at least). The more modern approach would be upLaTeX and LuaTeX.

The W3C Working Group Note Requirements for Japanese Text Layout described the issues of typesetting Japanese. Just to name some examples. In Western texts a new line is replaced with a space, while in Japanese text this additional space is considered incorrect. While the Japanese bracket usually has the width of a full character, if the bracket follows a truncation mark, like a comma, the bracket need to be a half space bracket. Or to put in different words: the space between a bracket after a truncation mark should contain roughly on charter width and not 1.5 width. And the final example: before inserting Roman letters into Japanese text usually there will be quarter space inserted, so called /textit{shibu aki}. Not a full space or a half space.

In LuaLaTeX a jlreq packages exists that implements parts of Requirements for Japanese Text Layout.

6 Japanese Language Packages

6.1 CJK

The CJK project maintains several packages for different flavours of LaTeX.

UTF-8 (Unicode Transformation Format 8), also called UTF-2 or in the LaTeX world FSS-UTF, is a special representation of Unicode ISO 10 646. It uses multi-byte sequences of different length, but for CJK only 2-byte and 3-byte sequences are implemented. ASCII characters are the same to be UTF-8 with TeX.

6.2 LuaTeX-ja

TODO

7 Fonts

The number of fonts for LaTeX (and Linux) is steadily increasing. With the exception of pLaTeX and pdfLaTeX this fonts can be used easily. A small list of open source fonts for CJK is maintained on Wikipedia.

For choosing a font 2 cases need to be separated: LaTeX flavors that do understand fontspec and the traditional way which do not.

Fontspec	Traditional
XeLaTeX	pLaTeX
LuaLaTeX	pdfLaTeX
upLaTeX	

7.1 Selecting a Font With Fontspec

With fontspec it is possible to chose OS fonts: True Type, Type-1 and others. The fontspec package needs to be used and provides specific commands. One challenge is to find the font name to be used with this command as this name can be different from what the OS uses. In most cases the difference is in spaces, capitalization and the like.

7.2 Selecting a Font Without Fontspec

The traditional and well controlled way to select a font is to load a package, like for example `\usepackage{tgbonum}`. This packages used commands to select various aspects, sized and corrections from a font that has to be in a TeX friendly format.

In case a font needs to be used that is not distributed with TeX, tools can be used to convert the font to a TeX font and to create additionally character map files to be used with TeX. This process will give the font a short name like `min` for Wadalab Mincho that can be used either directly with TeX markup or better should be accessed by a new package (that has to be written) to be includes as the example above.

When using CJK and derivate packages the font selection is handled via CJK, but it has to be a TeX font.

7.2.1 YuGothic (游ゴシック)

The YuGothic font from Microsoft should be easy usable on Windows via XeLaTeX or LuaLaTeX (not tested). And if the font is copied to Linux also there it should be possible to used it (not tested), but it might be illegal without an additional license from the font foundry that created the font. As for YuGothic it was created by IYUKOBO Ltd and is copyrighted 2018.

8 Examples

The following examples will use a modern LaTeX system: TeX Live 2021 from March 25th 3. The modern distribution of TeX Live includes at least `LuaTeX, pdfTeX, XeTeX` and its LaTeX counter parts. The examples assume the installation from DVD 4. Also for the example using UTF-8 is preferred.

To make sure a file is UTF-8 use `file  -bi FILE.tex` that should output something like this: `text/x-tex; charset=utf-8`.

8.1 pdfLaTeX

In the following example pdfTeX version 3.141592653-2.6-1.40.22 (TeX Live 2021) is used.

To see possible candidates for UTF-8 fonts to be used with pdfLaTeX you might use this command:

```
locate c70|grep 2021|grep fd|grep -v fdx
```

```
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/contrib/wadalab/c70goth.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/contrib/wadalab/c70maru.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/contrib/wadalab/c70min.fd
```

```
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/texinput/UTF8/c70bkai.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/texinput/UTF8/c70bsmi.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/texinput/UTF8/c70gbsn.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/texinput/UTF8/c70gkai.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/texinput/UTF8/c70mj.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/cjk/texinput/UTF8/c70song.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/ctex/fd/c70rm.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/ctex/fd/c70sf.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/ctex/fd/c70tt.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/ipaex-type1/c70ipxg.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/ipaex-type1/c70ipxga.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/ipaex-type1/c70ipxm.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/ipaex-type1/c70ipxma.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/nanumtype1/c70nanumgt.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/nanumtype1/c70nanummj.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/nanumtype1/c70uhcmj.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70fs.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70hei.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70kai.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70li.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70you.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70zhfs.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70zhhei.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70zhkai.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70zhli.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70zhsong.fd
/usr/local/texlive/2021/texmf-dist/tex/latex/zhmetrics/c70zhyou.fd
```

Keep in min that this are font definition files. The name of the fonts are different.

Font definition file(s)	TeX font names	font file name(s)
c70min.fd, (a)	udmj00,… (b)	dmjsy.pfb,…

(a) Font definition files for the Wadalab fonts are in the directory `contrib/wadalab`.
(b) These are virtual fonts.

At least in a default installation of TeX Live 2021 IPAex and Wadalab fonts are Japanese fonts to be used with CJKutf8 and pdfLaTeX.

Traditional font setting with pdfLaTeX is done by loading the font package. This sets internally all fonts (sans, serif, monospace). With CJK in addition to this a font parameter can be used

- CJKutf8, used PAex明朝/ゴシック or Wadalab
- bxcjkjatype, uses IPAex明朝/ゴシック, used by arXiv
- CJK

8.1.1 Fonts

CJK uses NFSS (New Font Selection Scheme, part of LaTeX 2e). The local documentation on how to add new fonts to CJK and pdflatex is located in HOWTO.txt.

ttf2tfm wadalab-gothic.ttf -P 3 -E 2 -w wdgt@SJIS@ ttf2tfm watanabe-mincho.ttf -P 3 -E 2 -w wnmc@SJIS@

ttf2tfm TakaoGothic.ttf -P 3 -E 2 -w tkgo@SJIS@ This is ttf2tfm version 2.0 ttf2tfm: ERROR: Invalid platform and/or encoding ID. Valid PID/EID pairs are: (0,3) Apple + x (3,1) Win + unicode 2 (evtl.) (3,10) Win + UCS-4 ttf2tfm TakaoGothic.ttf -P 3 -E 1 -w tkgo@SJIS,JIS,Unicode@

8.1.2 pdfLaTeX Example 04 CJKutf8

The `CJKutf8` package merges LaTeX's and CJK's UTF-8 support. If a Unicode character (within a CJK environment) corresponds to a glyph from the selected LaTeX font encoding, it is used, otherwise it is taken from the selected CJK unicode font.

For details look into the CJKutf8 documentation (especially for cmap and hyperref). Together with CJKutf8 you might want to used the `cmap` package.

- Source
- PDF

8.2 XeLaTeX

XeLaTeX accepts UTF-8, UTF-16 and UTF-32 as input and can use xeCJK, zxjatype (used xcCJK internally), bxjsclasses.

8.3 LuaLaTeX

LuaLaTeX can use luateja, luatexja-fontspec (inlcuded in luatexja), ltjsclasses (inlcuded in luatexja)

8.4 upLaTeX

Documents can be safed in UTF-8 and upLaTeX can used ujclasses, like `ujarticle` to compile DVI.

```
uplatex foo.tex
dvipdfmx foo.dvi
```

8.5 pLaTeX

Documents sould be saved in SJIS (not tested recently) Simple tests worked also with UTF-8 when using TeX Live 2021. Using the jsclasses, like jsarticle is straight forward.

```
\renewcommand{\kanjifamilydefault}{\gtdefault}% sans-serifalsofor Japanese
```

9 Pro And Cons Using pdfLaTeX

Using pdfLaTeX has the advantage to control the PDF output in every detail. Many printing presses nowadays moved from Postscript to PDF, but the requirement for the PDF is usually strict. Amazon for example require version 1.3 while xelatex or lualatex usually generate version 1.4 or higher. Xelatex and lualatex can be advised to output 1.3 but lack the control of all aspects. pdfLaTeX uses the standard TeX methods of controlling characters, it accepts only 8 bit characters and if used with packages like CJKutf8 also multiple bit UTF-8 characters can be used in the text file, but are converted internally to meet the TeX standard while creating the PDF. CJKutf8 can be used for Japanese, but it supports basically all UTF8-characters, also German, if the font supports it, but it is also possible to use two or more fonts (one for Japanese, one for the rest of the document, for example).

Speaking of fonts, one major difference is the font handling. While pdfLaTeX uses C70 Type 1 fonts, XeLaTeX needs fontspec1 and LuLaTeX recently supports fonts using fontspec2. Fontspec has the advantage to basically making available any OS system font (type-1, ttf, otf, …) to LaTeX. While this comes with a variety of new fonts and characters even of prior unsupported languages, it has the disadvantage of the loss of control. LaTeX documents are usually dependent on the TeX distribution which changes only gradually over the years. Using system fonts in LaTeX basically means that a document rendered with the same TeX version on two different computers might actually look different. Also mixing UTF-8 code points (quasi randomly changed characters, or using Chinese instead of Japanese characters) is an issue with fontspec (at XeLaTeX or LuaLaTeX for example). The lack of control and this problems are usually not seen on pdfLaTeX which make it more suited for long term documents that might to be printed professionally. For pdfLaTeX to support a new CJK font, the font has to be converted to .fb files (maybe also .fbx) files. This step is not needed when fonspec is used with LuaLaTeX or XeLaTeX. An other issue is a legal one. While most fonts on a computer can be used for most of the tasks royalty free, it might be different to include a font with fontspec that will be included in a PDF and distributed commercially as an e-book on Amazon for example. Some of the additional installed fonts have different licenses that do not cover commercially use. This should not be a problem if one used standard TeX Live fonts with pdfLaTeX. But also here better check than sorry.

For Japanese also the newer pLateX that comes with TeX Live (not the old one from the ASCII Corporation) might be a consideration, as pLaTeX supports many Japanese typesetting conventions, including vertical typesetting and is required by some Japanese journals. However it generates *.dvi files which have to be converted with `dvipdfmx -o file.pdf file.dvi`. Packages like geometry, hyperref, bookmark, graphicx, and xcolor need special options to support PDF conversion via DVI. As the layout was already created at the DVI level the PDF control is minimal.

10 Summary

If constant output quality is needed with the option to professional print and the CJK fonts of LaTeX contain all characters, pdfLaTeX should be used. If the font lacks characters or a different shape is needed, but control is mandatory, a font conversion should be attempted. In other cases XeLaTeX should be used, unless scripting TeX is required, which would need LuaLaTeX anyway. Only if the output is intended for a special journal or Japanese text features are needed that can not be done with pdfLaTeX (or XeLaTeX, LuaLaTeX), pLaTeX should be used.

https://www.microsoft.com/en-us/download/confirmation.aspx?id=49114 https://www.wfonts.com/font/yu-gothic

11 Reference

The latest version of the document can be found under https://christian.kuelker.info/en_US/Japanese/Writing/LaTeX/japanese-latex.html and as PDF under https://christian.kuelker.info/en_US/Japanese/Writing/LaTeX/japanese-latex.pdf

11.1 History

Version	Date	Notes
0.1.0	2021-07-10	Initial release

12 Disclaimer of Warranty

OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

13 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.