

Evangelion Japanese Font Metric for Lua \TeX

<https://github.com/RadioNoiseE/Evangelion-JFM>

<https://www.ctan.org/pkg/evangelion-jfm>

2023, Jing Huang (黄京)

Abstract

This documentation is going to introduce Evangelion Japanese Font Metric (hereinafter referred to as “Eva-JFM”), a Japanese Font Metric for typesetting high quality Chinese and Japanese documents. It can be used with Traditional Chinese, Simplified Chinese and Japanese fonts for both vertically and horizontally typesetted texts. It aims to provide a font metric which makes full use of the priority feature (provided by Lua \TeX -ja), bases on the standard [1], and supports some advanced (a.k.a., rarely-used) features. The documentation is now written in both Chinese and English.

This documentation is far from complete. It may have many grammatical (and contextual) errors.

Contents

1	Background Information and a Rough Introduction	1
2	Installation and Local Configurations	2
3	Using	2
4	Supported Features	3
4.1	Language Features	3
4.2	Direction Features	3
4.3	Extended Features	3
4.4	English Features	4
4.5	Dark Features	4
5	Linegap Punctuation Feature	4
5.1	About “Hanging”	4
5.2	Hanging Position	4
5.3	User Configs	5
5.3.1	Changing Parameters	5
5.3.2	Using Extra Font	5
6	Inspiration	6
	References	6

1 Background Information and a Rough Introduction

TeX is a powerful typesetting system “intended for the creation of beautiful books”, it has full support for typesetting English based texts. However, its support for CJ text is limited¹. For handling CJ texts in TeX, both macro extensions (i.e., CJK) and engine extensions were developed. One of the most influential one is (the) pTeX (series).

pTeX uses a virtual font scheme, by mapping TrueType or OpenType fonts using TFM/VF files. It doesn't support font configuration through macros, and has no support for PDF format output. Its advantage is the proven ability for dealing with traditional Japanese typographic layout requirements.

pdfTeX is a TeX engine extension which can directly output PDF files (just as its name). But it has limited support to Unicode as well as modern font formats (TrueType and OpenType vector font formats).

LuaTeX is based on pdfTeX. The inclusion of Lua enables it to support Unicode with the reader module, and modern fonts by using fontloader. Its macro based font setup feature is provided by luaotfload.

LuaTeX-ja can be seen as a porting of pTeX and LuaTeX. It's a macro package for typesetting high quality Japanese documents when using LuaTeX. LuaTeX supports font configuring by macros, therefore there's no need to keep pTeX's VF file. But for advanced features it left and extended² the so-called JFM file.

This document describes Eva-JFM, an advanced JFM file. By using LuaTeX's callback, it embeds features (maybe) needed in CJ text typesetting in Eva-JFM.lua. The features supported now are “Traditional Chinese”, “Simplified Chinese”, “Japanese”, “Vertical Typesetting”, “Linegap Punctuations”, “Hanging Punctuations”, “Extended Font”, and “Non Standard”.

2 Installation and Local Configurations

The sourcefiles are hosted on Github while it's also uploaded to CTAN. Users can simply use

```
1 tlmgr install evangelion-jfm
```

(or maybe using other package managers) to install. (But note that the CTAN branch is not always updated.) Developers can also use

```
1 mkdir Evangelion-JFM [ && ] cd Evangelion-JFM
2 git clone https://github.com/RadioNoiseE/Evangelion-JFM
```

to extract the latest version, then move it to the TEXMF directory, for instance

```
1 ~/Library/texlive/2023/texmf-dist/tex/luatex/eva-jfm
```

If your TeX distribution requires

```
1 mktexlsr
```

to update the Ls-R database, make it so.

Eva-JFM doesn't require any local configuration in most cases, but if you have some special requirements, have a look at section 5.3.

3 Using

The above is an example of typesetting vertical text using Traditional Chinese fonts

```
1 \usepackage{luatexja-fontspec, luatexja-adjust}
2 \setmainfont{Source Han Serif TC}[Language = Chinese Traditional, TateFeatures = {JFM = eva/{vert, trad, nstd}}]
```

¹Maybe because there was no universally recognized or accepted CJ character set standard as well as an encoding system.

²The priority feature and some imaginary characters as well.

```
3 \ltjenableadjust[priority = true]
```

(and be aware that you need to load a document class which supports vertical typesetting or use the `\tate` command. Lua \TeX -ja's JFM syntax is the above

```
1 jfm = <JFM name>/{\JFM features}
```

while under \LaTeX the most common case while using `\setmainfont` is most likely

```
1 \setmainfont{<font name>}[Language = <language name>, <dir> = {JFM = <JFM name>/{\JFM features}}]
```

Option `` is the font (that you'd like to specify as the main font for your document)'s name. When using Japanese fonts, simply ignore the `<language name>` since Lua \TeX -ja will automatically fill it for you. In this case, filling Chinese Traditional for Traditional Chinese fonts and Chinese Simplified for Simplified Chinese fonts is necessary³. `<dir>` should be `TateFeatures` when typeset vertically and `YokoFeatures` for typesetting horizontally accordingly. The JFM's name is specified by the `<JFM name>` option⁴. Finally, for the `<JFM features>` key, fill in the JFM features. They are described in section 4.

For advanced users, it's also recommended to use the following

```
1 \def\ltj@stdyokojfm{eva/{<JFM features>}}
```

or with the NFSS.

To set up JFM in other cases, please refer to the Lua \TeX -ja document [2].

4 Supported Features

This section is going to give you a glance at all the features embedded in Eva-JFM. They are divided into 5 groups, and are described in the next 5 subsections respectively.

4.1 Language Features

You should specify one and only one feature from this section, or your \TeX is going to complain about it.

`jp` \rightarrow (*JaPanese*)

Japanese font feature. When using Japanese fonts, you are required to specify this. It's very difference from Traditional Chinese and Simplified Chinese feature, namely the glue inserted after Question Mark and Exclamation Mark, and some punctuation mark's position when typeset vertically. It affects the feature `lgp`, as well as the internal grouping.

`trad` \rightarrow (*TRADitional chinese*)

Traditional Chinese feature. You should specify this when you are typesetting using Traditional Chinese fonts. The differences from the other two is because of its middle-placed punctuations. Hence the glues inserted next to it, the line-end adjust, as well as some kernings between punctuations are special.

`smp1` \rightarrow (*SiMPLified chinese*)

Simplified Chinese feature, for Simplified Chinese fonts. Almost all the punctuations are laid down and placed aside. Therefore its position is treated with care. Eva-JFM also takes some rare conditions into consideration. Note that the *aki* after Question Mark and Exclamation Mark is different from that of the Japanese font feature.

³Without this, your output may result in wrong details, for instance wrong punctuation shape & direction.

⁴Lua \TeX -ja searches for a JFM file following the method `jfm-<JFM name>.lua`.

4.2 Direction Features

Features in this section is compatible with all the other features.

`vert` → (*VERTical writing*)

Vertical Typesetting feature. It affects kerning, internal grouping, etc. You should specify this when typesetting vertically.

4.3 Extended Features

Except the feature `hgp` doesn't rely on feature `vert`, all the other features need `vert` to work (since they should only be needed in vertical texts).

`extd` → (*EXTEnded font*)

Extended font features. The default ratio is $x:y=100:80$ while x is the width and y is the height. You can customize it using `extd=<ratio>` (the default $\langle ratio \rangle$ is 1.25). It should be used with `extend` (`luaotfload`) or `FakeStretch` (`fontspec`).

`lgp` → (*LineGap Punctuations*)

The linegap punctuations feature. This hangs some punctuations into the linegap. Some difference occurs when it's used with the `jp` feature. For more information see section 5.

`hgp` → (*HanGing Punctuations*)

Hanging punctuation feature which “hangs” some punctuation at line-end (allowing them to stick out a bit). Traditional Chinese fonts doesn't support this feature because the result is somewhat (rather) wierd.

4.4 English Features

You need to set the JAchar range using `\ltjsetParameter` before using features in this section, or they won't work properly. It's also recommended to use with the corresponding OpenType features.

`hwid` → (*Half WIDTHh*)

Half width English characters feature. This will place each alphabets into a box which width is exactly 0.5 times the CJ character's width. It's worth noting that it will not stretch or shrink the glyph, it only adjusts the spacing. Hence if the OpenType feature `hwid` is not set, English characters will simply overlap. All the kernings and italic corrections will also be lost (this may be fixed in the future versions), and will ignore the parameter `xkanjiskip`. Please use with care.

`fwid` → (*Full WIDTHh*)

Full width English characters feature. It's similar from feature `hwid` above except that the spacing will be stretched out on the contrary.

4.5 Dark Features

Before using the following features, please make sure that you have carefully read the descriptions.

`nstd` → (*Non STandard*)

This one ignores the standard priority rules for punctuation kerning. While Japanese text layout requirement [1] suggests that the priority for the period should be higher than the comma (which means the period is easier to stretch), this makes the comma's priority higher than the preiod's. Only works when `luatexja-adjust's` priority feature is enabled (set to `true`).

plain → (PLAIN punct kerning style)

Disables the performance of spacing adjustment for any of the puncts. Designed to be used in the verbatim environment. Developers can use the hook provided by that environment to specify this feature.

5 Linegap Punctuation Feature

Here more detailed information about linegap punctuations are provided, as well as the issues may occur and the possible solution.

5.1 About “Hanging”

Linegap punctuations can be seen in Chinese ancient books, it’s a combination of the punctuations marks and the traditional vertical typesetting method.

Only periods and commas should be hanged but Eva-JFM hangs three more punctuations in addition. Japanese font is different in this aspect however, since the direction of colon and semicolon makes it impossible to be hanged.

They are all hanged to the lower right of the glyph. See the next subsection for more details.

5.2 Hanging Position

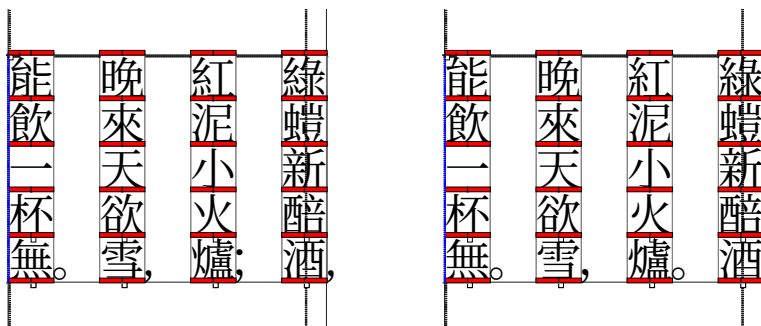


Figure 1: The linegap punctuations feature

The position of these hanged punctuations is decided according to the following rules as shown in the figure 1. For customizing, see subsection 5.3. The rules which occurs more early have the higher priorities.

- The style of the three fonts are unified;
- The position of the similar elements in different punctuations should be the same;
- The glyph of the punctuations should touch the *kanji*'s boundary;
- Different punctuations' position can vary considering their glyphs' shapes, sizes, design respectively.

5.3 User Configs

This feature is designed for the Source Han font series (思源系列). Due to different fonts' different punctuation marks, the output may be wrong (overlap, not aligned, etc). Also you may prefer your own settings. Therefore, two methods of customizing the positions of hanged punctuations is provided here.

5.3.1 Changing Parameters

In Eva-JFM, the tables which contains the parameters for the positions of these hanged punctuations is

```
1 [101,2] ==> [1]; [201,2] ==> [2]; [301,2] ==> [3].
```

Kindly modify left (dir right) and down (dir down) until the output is fine. You can also refer to the last section (*Implementing*).

5.3.2 Using Extra Font

Extracting the glyphs for punctuation marks and package them into a new font (you can use programs like *fontforge*) and use them for hanging punctuations later is the second solution. You can also load another font just for its punctuations (but loading a CJ font into T_EX's memory has an expensive cost).

After installing that font, you can use the AltFont key provided by LuaT_EX-ja to replace the punctuations. The actual code is shown above.

```
1 \setmainfont[
2   Language =  $\langle language \rangle$ ,
3   TateFeatures = {
4     JFM = eva/{vert, lgp,  $\langle language \rangle$ },
5     AltFont = {
6       {Range = " $\langle utf-8 code \rangle$ ", Font =  $\langle symbol font \rangle$ }
7     }
8   }
9 ]{ $\langle main font \rangle$ }
```

One of Japanese, Chinese Traditional or Chinese Simplified should be filled in the first $\langle language \rangle$ option, the other one is for the corresponding JFM features. $\langle utf-8 code \rangle$ selects the punctuations you'd like to replace with the “punctuation font”⁵. Finally, it's obvious that the $\langle symbol font \rangle$ and the $\langle main font \rangle$ options are for the “punctuation font” and the main font.

It's also recommended for the developers to use the NFSS with

```
1 \DeclareAlternateKanjiFont{ $\langle base encoding \rangle$ }{ $\langle base family \rangle$ }{ $\langle base series \rangle$ }{ $\langle base shape \rangle$ }{ $\langle alt encoding \rangle$ }{ $\langle alt$ 
    $\langle family \rangle$ }{ $\langle alt series \rangle$ }{ $\langle alt shape \rangle$ }{ $\langle range \rangle$ }
```

Option $\langle base \rangle$ and $\langle alt \rangle$ stands for main font and “punctuation font”.

Refer to the LuaT_EX-ja document [2] for more detailed syntax and usage as well as some examples.

6 Inspiration

Eva-JFM's internal grouping is inspired by min10.tfm [5], while its priority feature's data partly comes from Noriyuki Abe's jlreq.lua [6].

This JFM's name comes from the animation *Neon Genesis Evangelion* by Hideaki Anno.

References

[1] W3C Japanese Layout Task Force (ed). Requirements for Japanese Text Layout (W3C Working Group Note), 2022, 2023. <https://www.w3.org/TR/jlreq/>.

[2] LuaT_EX-ja プロジェクトチーム. LuaT_EX-ja パッケージ, 2022, 2023.

⁵You can search <https://www.unicode.org/charts/unihanrsindex.html> for their unicodes representations.

- [3] The Unicode Consortium. The Unicode Standard Version 15.0 - Core Specification, 2022.
- [4] Victor Eijkhout. \TeX by Topic, A \TeX nician's Reference, Addison-Wesley, 1992.
- [5] 乙部巖己. min10 フォントについて. <http://argent.shinshu-u.ac.jp/~otobe/tex/files/min10.pdf>.
- [6] 阿部紀行. Jlreq Document Class, 2022. <https://github.com/abenori/jlreq>.
- [7] 庵野秀明. 新世紀エヴァンゲリオン.