

Evangelion Japanese Font Metric for LuaTeX

<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 LuaTeX-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.

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
```

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.

(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 T_EX 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}}]
3 \ltjenableadjust[priority = true]
```

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

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

while under L^AT_EX the most common case while using `\setmainfont` is most likely

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

オプションの `` は、文書のメインフォントとして指定したいフォントの名前です。日本語のフォントを使用する場合、`<language name>` は自動的に LuaT_EX-ja によって補完されるため、無視してください。この場合、伝統的な中国語フォントには `Chinese Traditional`、簡体字中国語フォントには `Chinese Simplified` を指定する必要があります（これを指定しないと、出力が間違った細部、例えば句読点の形状と方向が変わってしまう可能性があります）。`<dir>` は、縦書きの場合は `<TateFeatures>`、横書きの場合は `YokoFeatures` に設定します。JFM の名前は `<JFM features>` オプションで指定します³。最後に、`<JFM features>` キーに JFM の特徴を入力してください。これらは、セクション 4 で説明されています。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 LuaT_EX-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 necessary⁴. `<dir>` should be `TateFeatures` when typeset vertically and `YokoFeatures` for typesetting horizontally accordingly. The JFM's name is specified by the `<JFM name>` option⁵. 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 LuaT_EX-ja document [2].

³LuaT_EX-ja は `jfm-<JFM name>.lua` という方法で JFM ファイルを検索します。

⁴Without this, your output may result in wrong details, for instance wrong punctuation shape & direction.

⁵LuaT_EX-ja searches for a JFM file following the method `jfm-<JFM name>.lua`.

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.

4.2 Direction Features

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

`vert` \rightarrow (*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` \rightarrow (*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 `<ratio>` is 1.25). It should be used with `extend` (luaotfload) or `FakeStretch` (fontspec).

`lgp` \rightarrow (*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` \rightarrow (*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 `\ljsetparameter` 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 WIDTH*)

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 WIDTH*)

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 period's. Only works when `luatexja-adjust's` priority feature is enabled (set to true).

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

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.

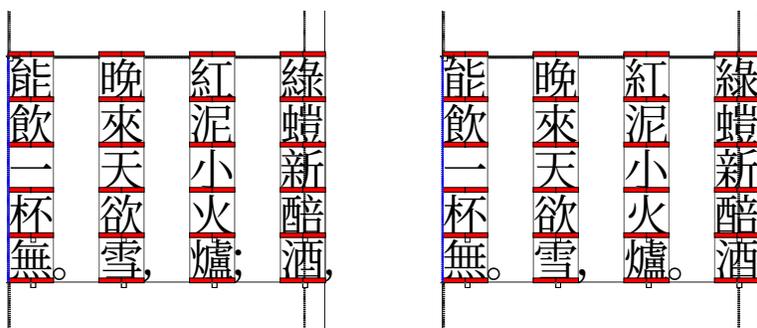


Figure 1: The linegap punctuations feature

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 \TeX 's memory has an expensive cost).

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

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

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"⁶. 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

⁶You can search <https://www.unicode.org/charts/unihanrsindex.html> for their unicodes representations.

```
1 \DeclareAlternateKanjiFont{<base encoding>}{<base family>}{<base series>}{<base shape>}{<alt encoding>}{<alt
   family>}{<alt series>}{<alt shape>}{<range>}
```

Option `<base>` and `<alt>` stands for main font and “punctuation font”.

Refer to the LuaTeX-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] LuaTeX-ja プロジェクトチーム. LuaTeX-ja パッケージ, 2022, 2023.
- [3] The Unicode Consortium. The Unicode Standard Version 15.0 - Core Specification, 2022.
- [4] Victor Eijkhout. TeX by Topic, A TeXnician’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.

Implementation

The above is the implementation of this font metric. Can be used for reference.

```
1 ---- Evangelion Japanese Font Metric for LuaTeX
2 ---- Current Version: 1.0.3 (c)
3 ---- Dev URL: https://github.com/RadioNoiseE/Evangelion-JFM
4 ---- Copyright 2023, RadioNoiseE ©
5
6
7 -- 初期化
8 local lang_jp, lang_tc, lang_sc, dir_vt, font_extd, punc_lg, punc_hg, std_nil, al_hw, al_fw
9
10 if luatexja.jfont.jfm_feature then
11     lang_jp = luatexja.jfont.jfm_feature.jp
12     lang_tc = luatexja.jfont.jfm_feature.trad
13     lang_sc = luatexja.jfont.jfm_feature.smpl
14     dir_vt = luatexja.jfont.jfm_feature.vert
15     font_extd = luatexja.jfont.jfm_feature.extd
16     punc_lg = luatexja.jfont.jfm_feature.lgp
17     punc_hg = luatexja.jfont.jfm_feature.hgp
18     std_nil = luatexja.jfont.jfm_feature.nstd
19     al_hw = luatexja.jfont.jfm_feature.hwid
20     al_fw = luatexja.jfont.jfm_feature.fwid
```

```

21 end
22
23 -- 預處理及容錯
24 if font_extd == true and dir_vt == false then
25     tex.error('JFM feature "extd" only works with feature "vert".\n' ..
26             'For now I\'ll ignore it.')
27 end
28
29 if punc_lg == true and dir_vt == false then
30     tex.error('JFM feature "lgp" only works with feature "vert".\n' ..
31             'For now I\'ll ignore it.')
32 end
33
34 if al_hw == true and al_fw == true then
35     tex.error('JFM feature "hwid" cannot be used with "fwid".')
36 end
37
38 if not ((lang_jp and not (lang_tc or lang_sc)) or
39         (lang_tc and not (lang_jp or lang_sc)) or
40         (lang_sc and not (lang_jp or lang_tc))) then
41     tex.error('Specify one and only one feature from three language specific features\n' ..
42             '"jp", "trad" or "smp1"\n' ..
43             'is required.\n' ..
44             'For now I\'ll use "lang_jp" for japanese by default.')
45 end
46
47 -- 壓縮比例設定
48 if font_extd == true then
49     local extd_ratio = (type(font_extd) == 'string') and tonumber(font_extd) or 1.25
50 end
51
52 -- 行間標點字間距補足
53 local lgp_kanjiskip = {kanjiskip_natural = 0, kanjiskip_stretch = 1, kanjiskip_shrink = 1}
54
55 -- 定義函數宏
56 local function logic_anif(f1, f2, r1, r2)
57     local rta = f1 and (f2 and r1) or r2
58     return rta
59 end
60
61 local function logic_if(f1, r1, r2)
62     local rti = f1 and r1 or r2
63     return rti
64 end
65
66 local function context_height()
67     local rth = dir_vt and (font_extd and extd_ratio/2 or 0.5) or 0.88
68     return rth
69 end
70
71 local function context_depth()
72     local rtd = dir_vt and (font_extd and extd_ratio/2 or 0.5) or 0.12
73     return rtd
74 end
75
76 -- 主體

```

```

77 local eva = {
78     version = 3,
79     dir = logic_if(dir_vt, 'tate', 'yoko'),
80     zw = 1,
81     zh = logic_anif(dir_vt, font_extd, extd_ratio, 1),
82     kanjiskip = {0, 0.25, 0},
83     xkanjiskip = {0.25, 0.125, 0.125},
84
85     [0] = { -- 缺省類
86         width = 1,
87         height = context_height(),
88         depth = context_depth(),
89         italic = 0,
90         left = 0,
91         down = 0,
92         align = 'middle',
93         glue = {
94             [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, 0},
95             {-1, -2})}, {priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
96             [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, -2},
97             {-1, 0})}, {priority = logic_if(std_nil, {-1, -2}, {-1, 0})}),
98             [3] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1,
99             priority = {-1, -1}}, {priority = {-1, -1}})),
100             [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
101             [9] = {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}}
102         },
103         round_threshold = 0.01
104     },
105
106     [1] = { -- 読点類
107         chars = logic_anif(dir_vt, punc_lg, {}, {'\ ', ' '}),
108         width = 0.5,
109         height = context_height(),
110         depth = context_depth(),
111         italic = 0,
112         left = 0,
113         down = 0,
114         align = logic_if(lang_tc, 'middle', 'left'),
115         glue = {
116             [0] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0},
117             {-1, -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
118             [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
119             [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
120             [3] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
121             {0.5, 0, 0.25, priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0,
122             0.25, ratio = 0, priority = {0, -1}})),
123             [4] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0,
124             -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2})}),
125             [5] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0,
126             -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2})}),
127             [6] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0},
128             {-1, -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
129             [7] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0},
130             {-1, -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
131             [8] = logic_if(lang_tc, {0.25, 0, 0.125}, {}),

```

```

122     [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
priority = {0, -1}})
123     },
124     end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0.5, 0}, {0, 0}))
125 },
126
127 [101] = { -- 読点類 (行間a)
128     chars = logic_anif(dir_vt, punc_lg, {''}, {}),
129     width = 0,
130     height = context_height(),
131     depth = context_depth(),
132     italic = 0,
133     left = 0.38,
134     down = -0.34,
135     align = 'left',
136     glue = {
137         [0] = lgp_kanjiskip
138     }
139 },
140
141 [102] = { -- 読点類 (行間b)
142     chars = logic_anif(dir_vt, punc_lg, {''}, {}),
143     width = 0,
144     height = context_height(),
145     depth = context_depth(),
146     italic = 0,
147     left = logic_if(lang_tc, 0.62, 0.40),
148     down = logic_if(lang_tc, -0.58, -0.26),
149     align = 'left',
150     glue = {
151         [0] = lgp_kanjiskip
152     }
153 },
154
155 [2] = { -- 句点類
156     chars = logic_anif(dir_vt, punc_lg, {}, {'.', '。'}),
157     width = 0.5,
158     height = context_height(),
159     depth = context_depth(),
160     italic = 0,
161     left = 0,
162     down = 0,
163     align = logic_if(lang_tc, 'middle', 'left'),
164     glue = {
165         [0] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2},
{-1, 0})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0})}),
166         [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
167         [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
168         [3] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
{0.5, 0, 0.25, priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0,
0.25, ratio = 0, priority = {0, -1}})),
169         [4] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0,
0})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0, 0})}),
170         [5] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0,
0})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0, 0})}),

```

```

171     [6] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2},
172     {-1, 0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
173     [7] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2},
174     {-1, 0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
175     [8] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0}, {}),
176     [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
177     priority = {0, -1}})
178     },
179     end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0.5, 0}, {0, 0}))
180 },
181
182 [201] = { -- 句點類 (行間a)
183     chars = logic_anif(dir_vt, punc_lg, {'.'}, {}),
184     width = 0,
185     height = context_height(),
186     depth = context_height(),
187     italic = 0,
188     left = logic_if(lang_tc, 0.68, 0.34),
189     down = logic_if(lang_tc, -0.58, -0.28),
190     align = 'left',
191     glue = {
192         [0] = lgp_kanjiskip
193     }
194 },
195
196 [202] = { -- 句點類 (行間b)
197     chars = logic_anif(dir_vt, punc_lg, {'。'}, {}),
198     width = 0,
199     height = context_height(),
200     depth = context_height(),
201     italic = 0,
202     left = 0.42,
203     down = -0.35,
204     align = 'left',
205     glue = {
206         [0] = lgp_kanjiskip
207     }
208 },
209
210 [3] = { -- 兩點類
211     chars = logic_if(lang_jp, {}, (logic_anif(dir_vt, punc_lg, {}, {':', ';'}))),
212     width = logic_if(dir_vt, 1, 0.5),
213     height = context_height(),
214     depth = context_depth(),
215     italic = 0,
216     left = 0,
217     down = 0,
218     align = logic_if(lang_tc, 'middle', 'left'),
219     glue = {
220         [0] = logic_if(dir_vt, {priority = {-1, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
221     priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
222         [1] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {
223     priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25, ratio =
224     0, priority = {0, -1}})),
225         [2] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {
226     priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25, ratio =

```

```

0, priority = {0, -1})),
220     [3] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.5, 0, 0.25, priority = {0,
-1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
221     [4] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
222     [5] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
223     [6] = logic_if(dir_vt, {priority = {-1, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
224     [7] = logic_if(dir_vt, {priority = {-1, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
225     [8] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5, 0, 0.25,
ratio = 0, priority = {0, -1}}),
226     [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
priority = {0, -1}})
227     }
228 },
229
230 [301] = { -- 兩點類 (行間a)
231     chars = logic_if(lang_jp, {}, logic_anif(dir_vt, punc_lg, {' :'}, {})),
232     width = 0,
233     height = context_height(),
234     depth = context_depth(),
235     italic = 0,
236     left = logic_if(lang_tc, 0.94, 0.72),
237     down = logic_if(lang_tc, -0.58, -0.34),
238     align = 'left',
239     glue = {
240         [0] = lgp_kanjiskip
241     }
242 },
243
244 [302] = { -- 兩點類 (行間b)
245     chars = logic_if(lang_jp, {}, logic_anif(dir_vt, punc_lg, {' ;'}, {})),
246     width = 0,
247     height = context_height(),
248     depth = context_depth(),
249     italic = 0,
250     left = logic_if(lang_tc, 0.96, 0.78),
251     down = logic_if(lang_tc, -0.58, -0.34),
252     align = 'left',
253     glue = {
254         [0] = lgp_kanjiskip
255     }
256 },
257
258 [4] = { -- 小書きの仮名類
259     chars = {
260         ' あ', ' い', ' う', ' え', ' お', ' つ', ' や', ' ゆ', ' よ', ' わ', ' か',
261         ' け', ' っ', ' ざ', ' ア', ' イ', ' ウ', ' エ', ' オ', ' ツ', ' ヤ', ' ユ',
262         ' ヨ', ' ワ', ' カ', ' ケ', ' っ', ' ざ', ' ク', ' シ', ' ス', ' ト', ' ヌ',
263         ' ハ', ' ヒ', ' フ', ' ヘ', ' ホ', ' ム', ' ラ', ' リ', ' ル', ' レ', ' ロ'
264     },
265     width = 1,
266     height = context_height(),
267     depth = context_depth(),

```



```

316         [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
317         [3] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1,
priority = {0, -1}}, {priority = {0, -1}})),
318         [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
319         [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
320     }
321 },
322
323 [7] = { -- 開括號類
324     chars = {'(', '[', '(', '[', '{', '<', '《', '「', '『', '【', '《', '〔', '〔', '≈', '‘', '‘',
“},
325     width = 0.5,
326     height = context_height(),
327     depth = context_depth(),
328     italic = 0,
329     left = 0,
330     down = 0,
331     align = 'right',
332     glue = {
333         [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
334         [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
335         [3] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1,
priority = {0, -1}}, {priority = {0, -1}})),
336         [7] = {0, 0, 0, priority = {0, 0}, kanjiskip_shrink = 1},
337         [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
338     }
339 },
340
341 [8] = { -- 閉括號類
342     chars = {')', ']', ')', '}', '>', '》', '」', '』', '】', ')', '』', '}', '≈', '’', '’'},
343     width = 0.5,
344     height = context_height(),
345     depth = context_depth(),
346     italic = 0,
347     left = 0,
348     down = 0,
349     align = 'left',
350     glue = {
351         [0] = {0.5, 0, 0.25, ratio = 0, priority = {-1, -2}},
352         [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
353         [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
354         [3] = logic_if(lang_sc, {priority = {0, -1}}, {0.25, 0, 0.125, ratio = 1, priority = {0,
-1}}),
355         [4] = {0.5, 0, 0.25, ratio = 0, priority = {0, -2}},
356         [5] = {0.5, 0, 0.25, ratio = 0, priority = {0, -2}},
357         [6] = {0.5, 0, 0.25, ratio = 0, priority = {-1, -2}},
358         [7] = {0.5, 0, 0.25, ratio = 0, priority = {-1, -2}, kanjiskip_stretch = 1},
359         [8] = {0, 0, 0, priority = {0, 0}, kanjiskip_shrink = 1},
360         [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
361     }
362 },
363
364 [9] = { -- 中點類
365     chars = logic_if(lang_jp, {'·', ':', ';'}, {'·', '·'}),
366     width = 0.5,
367     height = context_height(),

```

```

368     depth = context_depth(),
369     italic = 0,
370     left = 0,
371     down = 0,
372     align = 'middle',
373     glue = {
374         [0] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
375         [1] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.25, 0, 0.125, ratio = 0,
priority = {0, -1}}),
376         [2] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.25, 0, 0.125, ratio = 0,
priority = {0, -1}}),
377         [3] = logic_if(dir_vt, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, logic_if(lang_tc,
{0.5, 0, 0.25, priority = {0, -1}}, {0.25, 0, 0.125, priority = {0, -1}})),
378         [4] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
379         [5] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
380         [6] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
381         [7] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
382         [8] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
383         [9] = {0.5, 0, 0.25, priority = {0, -1}}
384     },
385     end_adjust = {0.25, 0}
386 },
387
388 [10] = { -- 西文
389     chars = {},
390     width = 0,
391     height = context_height(),
392     depth = context_depth(),
393     italic = 0,
394     left = 0,
395     down = 0,
396     align = 'middle',
397     glue = {}
398 },
399
400 [11] = { -- 行頭
401     chars = {'parbdd', 'boxbdd'},
402     glue = {
403         [7] = {0, 0, 0}
404     }
405 }
406 }
407
408 if al_hw == true or al_fw == true then
409     eva[10].chars = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
410         'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z',
411         'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M',
412         'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'}
413     eva[10].glue = table.fastcopy(eva[0].glue)
414     eva[10].glue[0] = {0.25, 0.125, 0.125, ratio = 0, priority = {0, -1}}
415     eva[0].glue[10] = {0.25, 0.125, 0.125, ratio = 1, priority = {0, -1}}
416     for _, catnum in ipairs({1, 2, 3, 5, 8, 9}) do
417         eva[catnum].glue[10] = table.fastcopy(eva[catnum].glue[0])
418     end
419 end
420

```

```
421 if al_hw == true and al_fw == false then
422     eva[10].width = 0.5
423 end
424
425 if al_fw == false and al_fw == true then
426     eva[10].width = 1
427 end
428
429 luatexja.jfont.define_jfm(eva)
```