

Evangelion Japanese Font Metric for LuaT_EX

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

西历 2023 年

概要

本文档将介绍名为 Evangelion Japanese Font Metric（下简称为“Eva-JFM”）的 JFM 文件。其适用于简体中文（以下简称为「简中」、繁体中文（以下简称为「繁中」）及日文字体的横直排。旨在提供一个充分利用 LuaT_EX-j_a 的 `priority` 特性，基于标准 [1] 的同时，支持一些罕用特性的 JFM 文件。文档暂仅使用中文撰写。

1 背景及略介：Background Knowledge and a Rough Introduction

T_EX 是高德纳教授于 20 世纪末开发的强大排版引擎，能够完全满足西文排版的需求。然因时代局限性^{*1}以及客观原因^{*2}对中日排版支持十分有限。为达成中日排版需求，在宏扩展（如 CJK 等）之外出现了引擎扩展。影响力较大的是 pT_EX 系列。

pT_EX 系列采用虚拟字体的理念，使用 TFM/VF 映射 TrueType 或 OpenType 字体完成排版。其不支持宏配置字体，也不支持直接生成 PDF 格式文件。但可以满足日本的传统横纵排版需求（工业标准）。

pdfT_EX 则是当时另一个 T_EX 的引擎扩展，支持不经 DVI 格式直接输出 PDF 格式的文件。然对 Unicode（字符编码）及 TrueType、OpenType（「现代」矢量字体格式）的支持繁琐或有限。LuaT_EX 便是基于 pdfT_EX 的引擎扩展，在原生支持 Unicode 下提供 Lua 语言扩展（使能够使用 `fontloader` 等模块）支持现代字体。宏配置字体特性由 `luaotfload` 宏集提供。它也支持直接生成 PDF 文件。

LuaT_EX-j_a 可看作是对两者的合并。这是一个由日本开发者北川弘典首倡的 LuaT_EX 下的日文支持项目，即将 pT_EX（大部分）移植到 LuaT_EX 下。由于 LuaT_EX 支持宏配置字体，故不需要 VF 文件为字体提供映射，但为标点挤压等需求保留并扩展^{*3}了 JFM 文件。

本项目就是一个 JFM 文件。使用 LuaT_EX 的 `callback`，将简中、繁中、日文及横纵方向、行间标点、悬挂标点、压缩字体等特性集中于 `jfm-eva.lua` 单个文件中。用户可按需调用特性来完成高质量的中日排版。

2 安裝及本地配置：Installation and Local Configs

本项目将源文件托管于 GitHub 平台，且已上传至 Comprehensive T_EX Archive Net（CTAN）。用户可使用

```
tlmgr install evangelion-jfm
```

或使用其他包管理器安装。用户也可使用

^{*1} 如没有事实上的统一字符编码等。

^{*2} 如中日字符集较大，以及书写方式的不同（纵书、横书），标点等。

^{*3} 如优先挤压（`priority`）特性，及一些特殊字符（如 `parbdd`、`glue`）等。

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

获取源文件，再将其放置在本地的 TEXMF 路径中，如

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

等。最后运行

```
1 mktexlsr
```

更新本地 T_EX 的 L_S-R 文件即可。

本文件一般情况下无需用户进行本地配置，但若有特殊需求可见第 5.3 节。

3 使用：Using

以下是在 L^AT_EX 下使用繁体中文字体进行直排的示例

```
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]
```

(注意需要调用支持直书的文档类或使用 `\tate` 命令)。LuaT_EX-ja 的 JFM 语法为：

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

而一般情况使用 `\setmainfont` 时则为：

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

其中，`` 自然为需要的字体名称。`<language name>` 在使用日文字体时可忽略，而使用简中、繁体中文字体时为必填，因 LuaT_EX-ja 会默认将其覆盖为 `Japanese` 选项，而这会带来灾难性的后果^{*4}。`<dir>` 选填 `TateFeatures`（直书）或 `YokoFeatures`（横书）。其后的 `<JFM name>` 为调用 JFM 的文件名^{*5}。最后的 `<JFM features>` 选项为选择使用的 JFM 特性，详细请看第 4 章。

其他情况下设置 JFM 及其更多信息请看 LuaT_EX-ja 文档 [2]。

4 支持特性：Supported Features

本章节将介绍 Eva-JFM 的所有特性，分别为：语言特性、方向特性、扩展特性及私有特性。

4.1 語言特性：Language Features

本区特性必填且只可填一个。不然则会报错。

^{*4} 比如错误的标点位置：日文为冒号及分号中置、其余偏靠，简中是全部偏靠，而繁体中则是统统中置。

^{*5} LuaT_EX-ja 会依 `jfm-<JFM name>.lua` 的格式来查找该文件。

jp → (*JaPanese*)

日本語特性。当使用日文字体时需调用该特性。其与简中、繁中区别在于问号及感叹号后插入的伸缩胶量。影响特性 lgp，且对内部分组有影响。

trad → (*TRADitional chinese*)

繁体中特性。当使用繁体中文字体时需调用。与简中、日本語特性的区别源于中置的标点。故，对于全部标点左右插入的伸缩胶的量都与简中、日本語不同。针对句点紧挨闭括号、标点位于句末时等皆有优化。

simpl → (*SiMPLified chinese*)

简体中特性，使用简体中文字体排版时调用。与日本語、繁体中特性区别源于分号及冒号等全部偏靠从而影响其左右插入伸缩胶的量。Eva-JFM 对一些（不该出现的）神奇情况（如两个句号同时出现、开括号后出现问号等）进行优化。对问号、感叹号等作了特殊处理。

4.2 方向特性：(Writing) Direction Features

本分区特性与全部其他特性兼容，可同时调用。

vert → (*VERTical writing*)

直书特性。对标点挤压、分组有影响。直书时必须调用。

4.3 扩展特性：Extended Features

本区特性 hgp 不依赖 vert 特性，其余需同 vert 特性同时调用。否则报错。

extd → (*EXTenDed font*)

压缩字体特性。目前仅支持横比纵为 100 比 80 的字体压缩^{*6}。需同 extend (luaotfload) 或 FakeStretch (fontspec) 同时使用。

lgp → (*LineGap Punctuations*)

行间标点特性。该特性将部分标点「悬挂」至行间。日文字体时与繁、简中字体时会有区别。详见第 5 章。

hgp → (*HanGing Punctuations*)

悬挂标点特性。该特性将部分标点「悬挂」于行末。仅简中、日文字体拥有该特性。

4.4 私有特性：Dark Features

使用本区特性前请先确保你清楚地知道你在做什么。

nstd → (*Non STandard*)

忽略标准特性。字体排印标准 [1] 认为逗号的压缩权重应比句号要低。本特性将句号的压缩优先级与逗号交换，使逗号被优先压缩^{*7}。仅在使用 luatexja-adjust 宏集时有效。

5 行間標點特性：More About Linegap Punctuations

本章节将提供更多详细的关于行间标点特性的信息，以及可能出现的问题及其解决方案。

^{*6} 日本新闻字体，如每日新闻明朝体。

^{*7} 考虑逗号、句号在文字系统中占的重量，以及「开明」压缩风格。

5.1 關於「懸掛」：About “Hanging”

行間標點可見於古籍之中，是將標點符號與直書結合妥協的產物。

傳統上懸掛句號與逗號。而 Eva-JFM 特性在繁中、簡中特性下會懸掛句號、逗號、頓號、冒號及分號，日文字體下則不懸掛冒號及分號。原因在於日本習慣上將冒號與分號看作「中點類」，直書時橫置處理。

本 JFM 將全部標點懸掛於字體右下位置。詳見下一節。

5.2 懸掛的位置：Hanging Position

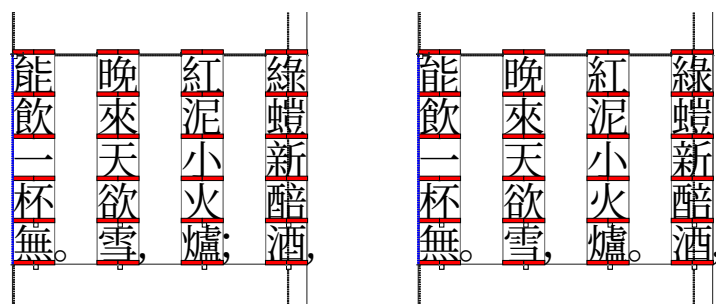


圖 1 行間標點特性示意圖

標點懸掛的位置有以下考量，可參照圖 1。若有特殊需求請看第 5.3 節。優先級由上至下。

- 三種字體風格統一，位置原則上一致（故，繁中字體也懸掛於右下、而非居中）；
- 不同標點中的相同（似）元素位置相同；
- 繁中、簡中、日文字體標點觸字框右邊線；
- 不同標點符號因形狀不同可於字框底線略下沉或上浮；
- 不同標點符號因大小不同可靠近或遠離字框右邊線；
- 三種字體可分別因字符設計的差異而位置略微區別。

5.3 用戶配置：Configurations

本特性是以三套思源字體為基準設計的。而由於各字體的標點符號位置不可避免會有不同，故在某些特殊情況下會出現錯位影響視覺效果的情況。或是單純對原設定而言更偏好其他設定等原因，本節提供自定義及調整的兩種方法。第一種較簡單但可移植性較差，而第二種雖繁瑣但一勞永逸。

5.3.1 修改原程式碼：Changing Parameters

在 Eva-JFM 中，控制行間標點的分區分別為

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

只需調整其中 left 和 down 鍵的值即可。其中 left 為向右移動，down 為向下移動。具體可參照第 6 章。

5.3.2 使用外掛符號字體：Using Seperate Punctuation Font

該方法的原理就是使用特殊的僅包含（標點）符號的字體來替換原有字體中的標點符號，從而穩定其表現。符號字體可使用 <https://github.com/Buernia/Zhudou-Sans> 處的開源字體。其基於思源黑體，還添加了许多其他符號（但這裡我們只會用到六個符號）及標點擠壓等特性。

將其放入 TEXMF 並更新 Ls-R 文件後即可使用 Lua_T_EX-jā 提供的 AltFont 鍵進行替換，例元：

```
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>}
```

其中首個 <language> 可選填 Japanese、Chinese Traditional 或 Chinese Simplified，第二個則填語言特性分區的對應 jp、trad 及 simpl 特性。<utf-8 code> 則為需要替換的標點符號的 Unicode 編碼，如需替換句號 (ideographic full stop, U+3002)⁸ 則填 3002⁸ 即可。<symbol font> 以及 <main font> 填符號字體名稱、正文字體名稱即可。具體語法及示例可看 [2]。

6 啟發：Inspiration

Eva-JFM 的內部分組受 min10.tfm [5] 的啟發，支持的 priority 特性則取自阿部紀行氏的 jlreq.lua [6] 文件。其餘可見參考文獻。

參考文獻

- [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] Lua_T_EX-jā プロジェクトチーム. Lua_T_EX-jā 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 \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] Noriyuki Abe. Jlreq Document Class, 2022. <https://github.com/abenori/jlreq>.

程式碼：Implementation

以下為 jfm-eva.lua 文件內容，供參考。

⁸ 編碼可至 <https://www.unicode.org/charts/unihanrsindex.html> 查詢。

```

1 ---- Evangelion Japanese Font Metric for LuaTeX
2 ---- Current Version: 1.0.0 (c)
3 ---- Dev URL: https://github.com/RadioNoiseE/Evangelion-JFM
4 ---- © Copyright 2023, RadioNoiseE
5
6
7 -- 初始化
8 local lang_jp, lang_tc, lang_sc, dir_vt, font_extd, punc_lg, punc_hg, std_nil
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 end
20
21 -- 預處理及容錯
22 if font_extd == true and dir_vt == false then
23     tex.error('JFM feature "extd" only works with feature "vert".\n' ..
24         'For now I\'ll ignore it.')
25 end
26
27 if punc_lg == true and dir_vt == false then
28     tex.error('JFM feature "lgp" only works with feature "vert".\n' ..
29         'For now I\'ll ignore it.')
30 end
31
32 if not ((lang_jp and not (lang_tc or lang_sc)) or
33     (lang_tc and not (lang_jp or lang_sc)) or
34     (lang_sc and not (lang_jp or lang_tc))) then
35     tex.error('Specify one and only one feature from three language specific features\n' ..
36         '"jp", "trad" or "smpl"\n' ..
37         'is required.\n' ..
38         'For now I\'ll use "lang_jp" for japanese by default.')
39 end
40
41 -- 定義函數宏
42 local function logic_anif(f1, f2, r1, r2)
43     local rta = f1 and (f2 and r1) or r2
44     return rta
45 end
46
47 local function logic_if(f1, r1, r2)
48     local rti = f1 and r1 or r2
49     return rti

```

```

50 end
51
52 local function context_height()
53     local rth = dir_vt and (font_extd and 0.625 or 0.5) or 0.88
54     return rth
55 end
56
57 local function context_depth()
58     local rtd = dir_vt and (font_extd and 0.625 or 0.5) or 0.12
59     return rtd
60 end
61
62 -- 主體
63 local eva = {
64     version = 3,
65     dir = logic_if(dir_vt, 'tate', 'yoko'),
66     zw = 1,
67     zh = logic_anif(dir_vt, font_extd, 1.25, 1),
68     kanjiskip = {0, 0.25, 0},
69     xkanjiskip = {0.25, 0.25, 0.125},
70
71     [0] = { -- 缺省類
72         width = 1,
73         height = context_height(),
74         depth = context_depth(),
75         italic = 0,
76         left = 0,
77         down = 0,
78         align = 'middle',
79         glue = {
80             [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}, {}),
81             [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, -2}, {-1, 0})}, {}),
82             [3] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}}, {priority = {0, -1}})),
83             [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
84             [9] = {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}}
85         },
86         round_threshold = 0.01
87     },
88
89     [1] = { -- 読点類
90         chars = logic_anif(dir_vt, punc_lg, {}, {'\ ', ' '}),
91         width = 0.5,
92         height = context_height(),
93         depth = context_depth(),
94         italic = 0,
95         left = 0,

```

```

96     down = 0,
97     align = logic_if(lang_tc, 'middle', 'left'),
98     glue = {
99         [0] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1,
100             -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
101         [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
102         [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
103         [3] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5,
104             0, 0.25, priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5,
105             0, 0.25, ratio = 0, priority = {0, -1}})),
106         [4] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0,
107             -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2})}),
108         [5] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0,
109             -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2})}),
110         [6] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1,
111             -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
112         [7] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1,
113             -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
114         [8] = logic_if(lang_tc, {0.25, 0, 0.125}, {}),
115         [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
116             priority = {0, -1}})
117     },
118     end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0}, {0, 0}))
119 },
120
121 [101] = { -- 読点類 (行間a)
122     chars = logic_anif(dir_vt, punc_lg, {'\'}, {}),
123     width = 0,
124     height = context_height(),
125     depth = context_depth(),
126     italic = 0,
127     left = 0.38,
128     down = -0.34,
129     align = 'left'
130 },
131
132 [102] = { -- 読点類 (行間b)
133     chars = logic_anif(dir_vt, punc_lg, {'\'}, {}),
134     width = 0,
135     height = context_height(),
136     depth = context_depth(),
137     italic = 0,
138     left = logic_if(lang_tc, 0.62, 0.40),
139     down = logic_if(lang_tc, -0.58, -0.26),
140     align = 'left'
141 },
142
143 [2] = { -- 句点類
144     chars = logic_anif(dir_vt, punc_lg, {}, {'.', '。'}),

```



```

137     width = 0.5,
138     height = context_height(),
139     depth = context_depth(),
140     italic = 0,
141     left = 0,
142     down = 0,
143     align = logic_if(lang_tc, 'middle', 'left'),
144     glue = {
145         [0] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1,
146             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
147         [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
148         [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
149         [3] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5,
150             0, 0.25, ratio = 0, priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5,
151             0, 0.25, ratio = 0, priority = {0, -1}})),
152         [4] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0,
153             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0, 0}})),
154         [5] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0,
155             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0, 0}})),
156         [6] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1,
157             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
158         [7] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1,
159             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
160         [8] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0}, {}),
161         [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
162             priority = {0, -1}})
163     },
164     end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0}, {0, 0}))
165 },
166
167 [201] = { -- 句點類 (行間a)
168     chars = logic_anif(dir_vt, punc_lg, {'.'}, {}),
169     width = 0,
170     height = context_height(),
171     depth = context_height(),
172     italic = 0,
173     left = logic_if(lang_tc, 0.68, 0.34),
174     down = logic_if(lang_tc, -0.58, -0.28),
175     align = 'left'
176 },
177
178 [202] = { -- 句點類 (行間b)
179     chars = logic_anif(dir_vt, punc_lg, {'.'}, {}),
180     width = 0,
181     height = context_height(),
182     depth = context_height(),
183     italic = 0,
184     left = 0.42,
185     down = -0.35,

```

```

178     align = 'left'
179 },
180
181 [3] = { -- 兩點類
182     chars = logic_if(lang_jp, {}, (logic_anif(dir_vt, punc_lg, {}, {':', ';'}))),
183     width = logic_if(dir_vt, 1, 0.5),
184     height = context_height(),
185     depth = context_depth(),
186     italic = 0,
187     left = 0,
188     down = 0,
189     align = logic_if(lang_tc, 'middle', 'left'),
190     glue = {
191         [0] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
192             {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
193         [1] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {
194             priority = {0, -1}}, logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25,
195             ratio = 0, priority = {0, -1}})),
196         [2] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {
197             priority = {0, -1}}, logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25,
198             ratio = 0, priority = {0, -1}})),
199         [3] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25,
200             ratio = 0, priority = {0, -1}})),
201         [4] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
202             {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
203         [5] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
204             {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
205         [6] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
206             {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
207         [7] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
208             {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
209         [8] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5, 0, 0.25, ratio =
210             0, priority = {0, -1}},
211         [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
212             priority = {0, -1}})
213     }
214 },
215
216 [301] = { -- 兩點類 (行間a)
217     chars = logic_if(lang_jp, {}, logic_anif(dir_vt, punc_lg, {':', {}}, {})),
218     width = 0,
219     height = context_height(),
220     depth = context_depth(),
221     italic = 0,
222     left = logic_if(lang_tc, 0.94, 0.72),
223     down = logic_if(lang_tc, -0.58, -0.34),
224     align = 'left'
225 },

```

```

215 [302] = { -- 兩點類 (行間b)
216     chars = logic_if(lang_jp, {}, logic_anif(dir_vt, punc_lg, {';', {}}, {})),
217     width = 0,
218     height = context_height(),
219     depth = context_depth(),
220     italic = 0,
221     left = logic_if(lang_tc, 0.96, 0.78),
222     down = logic_if(lang_tc, -0.58, -0.34),
223     align = 'left'
224 },
225
226 [4] = { -- 小書きの仮名類
227     chars = {
228         'あ', 'い', 'う', 'え', 'お', 'つ', 'や', 'ゆ', 'よ', 'わ', 'か',
229         'け', 'く', 'ぎ', 'ア', 'イ', 'ウ', 'エ', 'オ', 'ツ', 'ヤ', 'ユ',
230         'ヨ', 'ワ', 'カ', 'ケ', 'ハ', 'ベ', 'ク', 'シ', 'ス', 'ト', 'ヌ',
231         'ハ', 'ヒ', 'フ', 'ヘ', 'ホ', 'ム', 'ラ', 'リ', 'ル', 'レ', 'ロ'
232     },
233     width = 1,
234     height = context_height(),
235     depth = context_depth(),
236     italic = 0,
237     left = 0,
238     down = 0,
239     align = 'middle',
240     glue = {
241         [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
242         [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
243         [3] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {priority = {0, -1}}),
244         [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
245         [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
246     }
247 },
248
249 [5] = { -- 疑問感嘆類
250     chars = {'!', '?', '!!', 'VS', '!? VS', '?!', '??'},
251     width = logic_if(dir_vt, 1, logic_if(lang_sc, 0.5, 1)),
252     height = context_height(),
253     depth = context_depth(),
254     italic = 0,
255     left = 0,
256     down = 0,
257     align = logic_if(dir_vt, 'middle', logic_if(lang_sc, 'left', 'middle')),
258     glue = {
259         [0] = logic_if(dir_vt, logic_if(lang_jp, {1, 0, 0.5, ratio = 0, priority = {-1, 0}}, {priority =
                {-1, 0}}, logic_if(lang_tc, {priority = {-1, 0}}, {0.5, 0, 0.25, ratio = 0, priority =
                {-1, 0}})),
260         [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, logic_anif(not dir_vt, lang_sc, {0.5, 0,
                0.25, ratio = 0}, {})),

```



```

350         [8] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
351         [9] = {0.5, 0, 0.25, priority = {0, -1}}
352     }
353 },
354
355 [10] = { -- 行頭
356     chars = {'boxbdd', 'parbdd'},
357     glue = {
358         [7] = {0, 0, 0}
359     }
360 },
361
362 [11] = { -- 伸縮膠
363     chars = {'glue'}
364 }
365 }
366
367 luatexja.jfont.define_jfm(eva)

```