Evangelion Japanese Font Metric for LuaTeX

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 LuaTeX-ja 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

TeX 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 pTeX 系列。

pTeX 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）。

pdfTeX 则是当时另一个 TeX 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。LuaTeX 便是基于 pdfTeX 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。

LuaTeX-ja 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 LuaTeX 下的日文支持项目，即将 pTeX（大部分）移植到 LuaTeX 下。由于 LuaTeX 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 LuaTeX 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　安装及本地配置：Installation and Local Configs

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

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

[*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。

获取源文件，再将其放置在本地的 `TEXMF` 路径中，如

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

等。最后运行

```
mktexlsr
```

更新本地 TeX 的 `Ls-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 LaTeX 下使用繁中字体进行直排的示例

```
\usepackage{luatexja-fontspec, luatexja-adjust}
\setmainjfont{Source Han Serif TC}[Language = Chinese Traditional, TateFeatures = {JFM = eva/{vert, trad,
    nstd}}]
\ltjenableadjust[priority = true]
```

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

```
jfm = ⟨JFM name⟩/{⟨JFM features⟩}
```

而一般情况使用`\setmainjfont` 时则为：

```
\setmainjfont{⟨font name⟩}[Language = ⟨language name⟩, ⟨dir⟩ = {JFM = ⟨JFM name⟩/{⟨JFM features⟩}}]
```

其中，⟨*font name*⟩ 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、繁中字体时为必填，因 LuaTeX-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 LuaTeX-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。

`jp` ⟶ (*JaPanese*)

日本语特性。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。

[*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，而繁中则是统统中置。

[*5] LuaTeX-ja 会依 `jfm-`⟨*JFM name*⟩`.lua` 的格式来查找该文件。

`trad` ⟶ (*TRADitional chinese*)

繁中特性。当使用繁体中文字体时需调用。与简中、日本语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、日本语不同。针对句点紧挨闭括号、标点位于句末时等皆有优化。

`smpl` ⟶ (*SiMPLified*)

简中特性，使用简体中文字体排版时调用。与日本语、繁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 擴展特性：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”

行间标点可见于古籍之中，是将标点符号与直书结合妥协的产物。

传统上悬挂句号与逗号。而 Eva-JFM 特性在繁中、简中特性下会悬挂句号、逗号、顿号、冒号及分号，日文字体下则不悬挂冒号及分号。原因在于日本习惯上将冒号与分号看作「中点类」，直书时横置处理。

本 JFM 将全部标点悬挂于字体右下位置。详见下一节。

5.2 懸掛的位置：Hanging Position

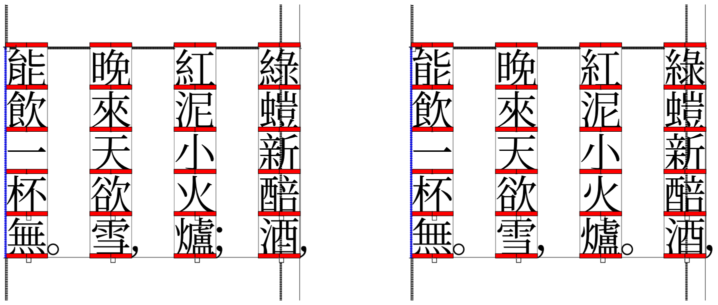

図 1　行间标点特性示意图

标点悬挂的位置有以下考量，可参照图 1。若有特殊需求请看第 5.3 节。优先级由上至下。

- 三种字体风格统一，位置原则上一致（故，繁中字体也悬挂于右下、而非居中）；
- 不同标点中的相同（似）元素位置相同；
- 繁中、简中、日文字体标点触字框右边线；
- 不同标点符号因形状不同可于字框底线略下沉或上浮；
- 不同标点符号因大小不同可靠近或远离字框右边线；
- 三种字体可分别因字符设计的差异而位置略微区别。

5.3 用戶配置：Configurations

本特性是以三套思源字体为基准设计的。而由于各字体的标点符号位置不可避免会有不同，故在某些特殊情况下会出现错位影响视觉效果的情况。或是单纯对原设定而言更偏好其他设定等原因，本节提供自定义及调整的两种方法。第一种较简单但可移植性较差，而第二种虽繁琐但一劳永逸。

5.3.1 修改原程式碼：Changing Parameters

在 Eva-JFM 中，控制行间标点的分区分别为

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

只需调整其中 `left` 和 `down` 键的值即可。其中 `left` 为向右移动，`down` 为向下移动。具体可参照第 6 章。

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

该方法的原理就是使用特殊的仅包含（标点）符号的字体来替换原有字体中的标点符号，从而稳定其表现。符号字体可使用 `https://github.com/Buernia/Zhudou-Sans` 处的开源字体。其基于思源黑体，还添加了许多其他符号（但这里我们只会用到六个符号）及标点挤压等特性。

将其放入 `TEXMF` 并更新 `Ls-R` 文件后即可使用 LuaTeX-ja 提供的 `AltFont` 键进行替换，例元：

```
\setmainjfont[
    Language = ⟨language⟩,
    TateFeatures = {
        JFM = eva/{vert, lgp, ⟨language⟩},
        AltFont = {
            {Range = "⟨utf-8 code⟩, Font = ⟨symbol font⟩}
        }
    }
]{⟨main font⟩}
```

其中首个 ⟨*language*⟩ 可选填 `Japanese`、`Chinese Traditional` 或 `Chinese Simplified`，第二个则填语言特性分区的对应 `jp`、`trad` 及 `smpl` 特性。⟨*utf-8 code*⟩ 则为需要替换的标点符号的 Unicode 编码，如需替换句号（ideographic full stop，`U+3002`）则填 `3002`[*8]即可。⟨*symbol font*⟩ 以及 ⟨*main font*⟩ 填符号字体名称、正文字体名称即可。具体语法及示例可看 [2]。

6 启發：Inspiration

Eva-JFM 的内部分组受 `min10.tfm` [5] 的启发，支持的 `priority` 特性则取自阿部紀行氏的 `jlreq.lua` [6] 文件。其余可见参考文献。

参考文献

[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] 乙部厳己. 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

以下为 `jfm-eva.lua` 文件内容，供参考。

[*8] 编码可至 `https://www.unicode.org/charts/unihanrsindex.html` 查询。

```lua
---- Evangelion Japanese Font Metric for LuaTeX
---- Current Version: 1.0.0 (b)
---- Dev URL: https://github.com/RadioNoiseE/Evangelion-JFM
---- © Copyright 2023, RadioNoiseE


-- 初始化
local lang_jp, lang_tc, lang_sc, dir_vt, font_extd, punc_lg, punc_hg, std_nil

if luatexja.jfont.jfm_feature then
    lang_jp = luatexja.jfont.jfm_feature.jp
    lang_tc = luatexja.jfont.jfm_feature.trad
    lang_sc = luatexja.jfont.jfm_feature.smpl
    dir_vt = luatexja.jfont.jfm_feature.vert
    font_extd = luatexja.jfont.jfm_feature.extd
    punc_lg = luatexja.jfont.jfm_feature.lgp
    punc_hg = luatexja.jfont.jfm_feature.hgp
    std_nil = luatexja.jfont.jfm_feature.nstd
end

-- 預處理及容錯
if font_extd == true and dir_vt == false then
    tex.error('JFM feature "extd" only works with feature "vert".\n' ..
              'For now I\'ll ignore it.')
end

if punc_lg == true and dir_vt == false then
    tex.error('JFM feature "lgp" only works with feature "vert".\n' ..
              'For now I\'ll ignore it.')
end

if not ((lang_jp and not (lang_tc or lang_sc)) or
    (lang_tc and not (lang_jp or lang_sc)) or
    (lang_sc and not (lang_jp or lang_tc))) then
    tex.error('Specify one and only one feature from three language specific features\n' ..
              '"jp", "trad" or "smpl"\n' ..
              'is required.\n' ..
              'For now I\'ll use "lang_jp" for japanese by default.')
end

-- 定義函數宏
local function logic_anif(f1, f2, r1, r2)
     local rta = f1 and (f2 and r1) or r2
     return rta
end

local function logic_if(f1, r1, r2)
     local rti = f1 and r1 or r2
     return rti
```

```
end

local function context_height()
    local rth = dir_vt and (font_extd and 0.625 or 0.5) or 0.88
    return rth
end

local function context_depth()
    local rtd = dir_vt and (font_extd and 0.625 or 0.5) or 0.12
    return rtd
end

-- 主體
local eva = {
    version = 3,
    dir = logic_if(dir_vt, 'tate', 'yoko'),
    zw = 1,
    zh = logic_anif(dir_vt, font_extd, 1.25, 1),
    kanjiskip = {0, 0.25, 0},
    xkanjiskip = {0.25, 0.25, 0.125},

    [0] = { -- 缺省類
        width = 1,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = 'middle',
        glue = {
            [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}, {}),
            [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, -2}, {-1, 0})}, {}),
            [3] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}}, {priority = {0, -1}})),
            [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
            [9] = {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}}
        },
        round_threshold = 0.01
    },

    [1] = { -- 読点類
        chars = logic_anif(dir_vt, punc_lg, {}, {'、', ','}),
        width = 0.5,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = logic_if(lang_tc, 'middle', 'left'),
        glue = {
            [0] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
            [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
            [2] = logic_if(lang_tc, {0.5, 0, 0.5}, {0.5, 0, 0.25}),
            [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}})),
            [4] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2})}),
            [5] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2})}),
            [6] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
            [7] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2})}),
            [8] = logic_if(lang_tc, {0.25, 0, 0.125}, {}),
            [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3, priority = {0, -1}})
        },
        end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0}, {0, 0}))
    },

    [101] = { -- 読点類（行間a）
        chars = logic_anif(dir_vt, punc_lg, {'、'}, {}),
        width = 0,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0.38,
        down = -0.34,
        align = 'left'
    },

    [102] = { -- 読点類（行間b）
        chars = logic_anif(dir_vt, punc_lg, {'，'}, {}),
        width = 0,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = logic_if(lang_tc, 0.62, 0.40),
        down = logic_if(lang_tc, -0.58, -0.26),
        align = 'left'
    },

    [2] = { -- 句點類
        chars = logic_anif(dir_vt, punc_lg, {}, {'．', '。'}),
        width = 0.5,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = logic_if(lang_tc, 'middle', 'left'),
        glue = {
            [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})}),
            [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
            [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
            [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}})),
            [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})}),
            [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})}),
            [6] = 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})}),
            [7] = 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})}),
            [8] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0}, {}),
            [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3, priority = {0, -1}})
        },
        end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0}, {0, 0}))
    },

    [201] = { -- 句點類（行間a）
        chars = logic_anif(dir_vt, punc_lg, {'.'}, {}),
        width = 0,
        height = context_height(),
        depth = context_height(),
        italic = 0,
        left = logic_if(lang_tc, 0.68, 0.34),
        down = logic_if(lang_tc, -0.58, -0.28),
        align = 'left'
    },

    [202] = { -- 句點類（行間b）
        chars = logic_anif(dir_vt, punc_lg, {'。'}, {}),
        width = 0,
        height = context_height(),
        depth = context_height(),
        italic = 0,
        left = 0.42,
        down = -0.35,
        align = 'left'
    },

    [3] = { -- 兩點類
        chars = logic_if(lang_jp, {}, (logic_anif(dir_vt, punc_lg, {}, {'：', '；'}))),
        width = logic_if(dir_vt, 1, 0.5),
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = logic_if(lang_tc, 'middle', 'left'),
        glue = {
            [0] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
            [1] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {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}})),
            [2] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {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}})),
            [3] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
            [4] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
            [5] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
            [6] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
            [7] = logic_if(dir_vt, {}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
            [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}}),
            [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3, priority = {0, -1}})
        }
    },

    [301] = { -- 兩點類（行間a）
        chars = logic_if(lang_jp, {}, logic_anif(dir_vt, punc_lg, {'：'}, {})),
        width = 0,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = logic_if(lang_tc, 0.94, 0.72),
        down = logic_if(lang_tc, -0.58, -0.34),
        align = 'left'
    },

    [302] = { -- 兩點類（行間b）
        chars = logic_if(lang_jp, {}, logic_anif(dir_vt, punc_lg, {'；'}, {})),
        width = 0,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = logic_if(lang_tc, 0.96, 0.78),
        down = logic_if(lang_tc, -0.58, -0.34),
        align = 'left'
    },

    [4] = { -- 小書きの仮名類
        chars = {
            'ぁ', 'ぃ', 'ぅ', 'ぇ', 'ぉ', 'っ', 'ゃ', 'ゅ', 'ょ', 'ゎ', 'ゕ',
            'ゖ', 'ゝ', 'ゞ', 'ァ', 'ィ', 'ゥ', 'ェ', 'ォ', 'ッ', 'ャ', 'ュ',
            'ョ', 'ヮ', 'ヵ', 'ヶ', 'ヽ', 'ヾ', 'ㇰ', 'ㇱ', 'ㇲ', 'ㇳ', 'ㇴ',
            'ㇵ', 'ㇶ', 'ㇷ', 'ㇸ', 'ㇹ', 'ㇺ', 'ㇻ', 'ㇼ', 'ㇽ', 'ㇾ', 'ㇿ'
        },
        width = 1,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = 'middle',
        glue = {
            [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
            [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
            [3] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {priority = {0, -1}}),
            [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
            [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
        }
    },

    [5] = { -- 疑問感嘆類
        chars = {'！', '？', '‼︎', '⁉︎', '⁈', '⁇'},
        width = logic_if(dir_vt, 1, logic_if(lang_sc, 0.5, 1)),
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = logic_if(dir_vt, 'middle', logic_if(lang_sc, 'left', 'middle')),
        glue = {
            [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}})),
            [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}, {})),
            [2] = 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}, {})),
            [3] = logic_if(dir_vt, {priority = {-1, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}}, {0.75, 0, 0.25, ratio = 1/3, priority = {-1, -1}})),
            [4] = logic_if(dir_vt, logic_if(lang_jp, {1, 0, 0.5, ratio = 0}, {}), logic_if(lang_tc, {}, {0.5, 0, 0.25, ratio = 0})),
            [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
            [8] = logic_anif(not dir_vt, lang_sc, {0.5, 0, 0.25, ratio = 0}, {}),
            [9] = logic_anif(not dir_vt, lang_sc, {0.75, 0, 0.25, ratio = 1/3, priority = {-1, -1}}, {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}})
        }
    },

    [6] = { -- 分離禁止類
        chars = {'—', '―', '‥', '…', '⋯', '〳', '〴', '〵'},
        width = 1,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = 'middle',
        kern = {
            [6] = 0
        },
        glue = {
            [1] = logic_if(lang_tc, {0.25, 0, 0,125, ratio = 1}, {}),
            [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
            [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}})),
            [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
            [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
        }
    },

    [7] = { -- 開括號類
        chars = {'（', '〔', '［', '｛', '〈', '《', '「', '『', '【', '｟', '〘', '〚', '〝', '‘', '“'},
        width = 0.5,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = 'right',
        glue = {
            [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
            [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
            [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}})),
            [7] = {0, 0, 0, priority = {0, 0}, kanjiskip_shrink = 1},
            [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
        }
    },

    [8] = { -- 閉括號類
        chars = {'）', '〕', '］', '｝', '〉', '》', '」', '』', '】', '〙', '〗', '〛', '〟', '’', '”'},
        width = 0.5,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = 'left',
        glue = {
            [0] = {0.5, 0, 0.25, ratio = 0, priority = {-1, -2}},
            [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
            [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
            [3] = logic_if(lang_sc, {}, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}),
            [4] = {0.5, 0, 0.25, ratio = 0, priority = {0, -2}},
            [5] = {0.5, 0, 0.25, ratio = 0, priority = {0, -2}},
            [6] = {0.5, 0, 0.25, ratio = 0, priroity = {-1, -2}},
            [7] = {0.5, 0, 0.25, ratio = 0, priority = {-1, -2}, kanjiskip_stretch = 1},
            [8] = {0, 0, 0, priority = {0, 0}, kanjiskip_shrink = 1},
            [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
        }
    },

    [9] = { -- 中點類
        chars = logic_if(lang_jp, {'・', '：', '；'}, {'・', '·'}),
        width = 0.5,
        height = context_height(),
        depth = context_depth(),
        italic = 0,
        left = 0,
        down = 0,
        align = 'middle',
        glue = {
            [0] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
            [1] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}),
            [2] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}),
            [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}})),
            [4] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
            [5] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
            [6] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
            [7] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
            [8] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
            [9] = {0.5, 0, 0.25, priority = {0, -1}}
        }
    },

    [10] = { -- 行頭
        chars = {'boxbdd', 'parbdd'},
        glue = {
            [7] = {0, 0, 0}
        }
    },

    [11] = { -- 伸縮膠
        chars = {'glue'}
    }
}

luatexja.jfont.define_jfm(eva)
```