

Evangelion Japanese Font Metric for LuaTeX

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

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

2023 年 4 月 26 日 黄京

概要

本ドキュメントは、高品質な中国語および日本語のドキュメントを組版するための日本語フォントメトリック「Evangelion Japanese Font Metric (以下「Eva-JFM」とする)」を紹介するものです。このメトリックは、縦書きと横書きの両方のテキストに対して、従来の中国語、簡体字中国語、および日本語のフォントとともに使用できます。これは、LuaTeX-ja で提供される優先機能を最大限に活用するフォントメトリックを提供し、標準 [1] に基づき、一部の高度な (すなわち、めったに使用されない) 機能をサポートすることを目的としています。

本ドキュメントは、完全なものではありません。文法的な (および文脈的な) エラーが多数含まれている可能性があります。

1 背景情報と簡単な紹介

TeX は「美しい本の制作」を目的とした強力な組版システムであり、英語ベースのテキストの組版には完全に対応しています。しかし、CJ テキストのサポートは限られています¹。TeX で CJ テキストを扱うために、マクロ拡張 (CJK など) やエンジン拡張が開発されました。最も影響力のあるものの 1 つが pTeX (シリーズ) です。

pTeX は仮想フォント方式を使用し、TrueType や OpenType フォントを TFM/VF ファイルを使ってマッピングします。マクロを介したフォント設定には対応せず、PDF 形式の出力にも対応していません。その利点は、伝統的な日本語のタイポグラフィレイアウト要件を扱うための証明済みの能力にあります。pdfTeX は、PDF ファイルを直接出力できる TeX エンジンの拡張です (その名の通り)。ただし、Unicode や現代のフォント形式 (TrueType や OpenType ベクターフォント形式) に対するサポートは限定的です。

LuaTeX は pdfTeX をベースにしています。Lua の組み込みにより、読み込みモジュールを使用して Unicode をサポートし、fontloader を使用して現代のフォントをサポートすることができます。マクロベースのフォント設定機能は luaotfload によって提供されます。

LuaTeX-ja は pTeX と LuaTeX の移植版と見なすことができます。LuaTeX を使用する場合、マクロによるフォント設定がサポートされるため、pTeX の VF ファイルを保持する必要はありません。しかし、一部の高度な機能 (「優先度」機能や仮想的な文字など) は、いわゆる JFM ファイルで残され、拡張されました。

このドキュメントでは、高度な JFM ファイルである Eva-JFM を説明します。LuaTeX のコールバックを使用して、Eva-JFM.lua に CJ テキスト組版に必要な機能を埋め込みます (おそらく)。現在サポートされている機

¹ おそらく普遍的に認められた CJ 文字セット標準やエンコーディングシステムが存在しなかったためかもしれません

能は、「繁体字中国語」、「簡体字中国語」、「日本語」、「縦組」、「ぶら下げ」、「行間句読点」、「全角欧文」、「半角欧文」、「非標準」。

2 インストールとローカル設定

ソースファイルは Github にホストされ、CTAN にもアップロードされています。ユーザーは単純に

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

(または他のパッケージマネージャーを使用することもできます) を使用してインストールできます。ただし、CTAN ブランチが常に更新されているわけではないことに注意してください。開発者はまた

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

を使用して最新バージョンを抽出し、それを TEXMF ディレクトリに移動することもできます。たとえば

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

あなたの TeX 配布が Ls-R データベースを更新するためにを

```
1 mktexlsr
```

を必要とする場合は、それを実行してください。

Eva-JFM はほとんどの場合、ローカル設定を必要としませんが、特別な要件がある場合は、セクション 5.3 を参照してください。

3 使用

```
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]
```

上記は、縦書きの伝統的な中国語フォントを使用して縦書きのテキストを組版する例です。注意が必要ですが、縦書きのテキストをサポートするドキュメントクラスを読み込むか、`\tate` コマンドを使用する必要があります。LuaTeX-ja の JFM 構文は、次のようになります。

```
1 jfm = <JFM name>/<JFM features>
```

LaTeX の場合、`\setmainfont` を使用する場合、最も一般的なケースは、以下のようになります

```
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 LuaTeX-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 $\langle JFM\ features \rangle$ オプションで指定します^{*2}。最後に、 $\langle JFM\ features \rangle$ 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。上級ユーザーの場合、以下を使用することをお勧めします

```
\def\ltj@stdyokojfm{eva/{\langle JFM features \rangle}}
```

または NFSS を使用することもできます。

他の場合で JFM を設定するには、LuaTeX-ja のドキュメント [2] を参照してください。

4 サポートされる機能

このセクションでは、Eva-JFM に組み込まれたすべての機能を概観します。それらは 5 つのグループに分けられ、それぞれ次の 5 つのサブセクションで説明されています。

4.1 言語機能

このセクションから 1 つの機能を指定する必要があります。そうしない場合、TeX はエラーを出します。

`jp` → (*JaPanese*)

日本語フォント機能。日本語フォントを使用する場合、この機能を指定する必要があります。伝統的な中国語と簡体字中国語の機能とは異なり、縦書き時に疑問符や感嘆符の後に挿入されるグルー、および句読点の位置が異なります。これは、`lgp` 機能および内部グルーピングに影響します。

`trad` → (*TRADitional chinese*)

伝統的な中国語機能。伝統的な中国語フォントを使用して組版する場合は、この機能を指定する必要があります。他の 2 つとの違いは、中央に配置された句読点です。そのため、それに隣接するグルー、行末の調整、および句読点間のカーニングは特別なものです。

`smpl` → (*SiMPLified chinese*)

簡体字中国語機能。簡体字中国語フォント用です。すべての句読点は横に配置されています。したがって、その位置は注意して扱われます。Eva-JFM はいくつかの珍しい状況も考慮しています。なお、疑問符と感嘆符の後の空きは日本語フォント機能とは異なります。

4.2 方向機能

このセクションの機能は、すべての他の機能と互換性があります。

`vert` → (*VERTical writing*)

縦組機能。カーニング、内部グルーピングなどに影響します。縦書き時には、これを指定する必要があります。

4.3 拡張機能

ぶら下げ `hgp` 機能を除くすべての機能は、`vert` 機能に依存して動作します (縦書きテキストで必要なため)。

^{*2} LuaTeX-ja は `jfm- $\langle JFM\ name \rangle$.lua` という方法で JFM ファイルを検索します。

extd → (*EXTenDed font*)

拡張フォント機能。デフォルトの比率は、 $x:y=100:80$ で、 x は幅、 y は高さを表します。extd=*(ratio)* (デフォルトの *(ratio)* は 1.25) を使用してカスタマイズできます。extend (luaotfload) または FakeStretch (fontspec) と。

lgp → (*LineGap Punctuations*)

ラインギャップの句読点機能。これは、いくつかの句読点をラインギャップに掛けます。jp 機能で使用する、いくつかの違いが発生します。詳細については、セクション 5 を参照してください。

hgp → (*HanGing Punctuations*)

行末に句読点を「ハング」する句読点機能をぶら下げます (少し目立つようにする)。繁体字中国語フォント結果がやや (むしろ) 奇妙であるため、この機能はサポートしていません。

4.4 欧文機能

このセクションの機能を使用する前に、JAchar 範囲を `\ltjsetparameter` で設定する必要があります。また、対応する OpenType 機能と併用することが推奨されています。

hwid → (*Half WIDTH*)

半角英語文字機能。各アルファベットを CJ 文字の幅のちょうど 0.5 倍の箱に配置します。これはグリフを伸ばしたり縮めたりするわけではなく、スペーシングを調整するだけであることに注意してください。したがって、OpenType 機能 hwid が設定されていない場合、英語の文字は単に重なり合います。すべてのカーニングとイタリックの補正も失われます (これは将来のバージョンで修正されるかもしれません)、xkanjiskip パラメータは無視されます。注意して使用してください。

fwid → (*Full WIDTH*)

全角英語文字機能。スペーシングが逆に広がりますが、機能 hwid と同様です。

4.5 ダーク機能

以下の機能を使用する前に、説明を注意深く読んでください。

nstd → (*Non STandard*)

この機能は、句読点のカーニングに対する標準的な優先ルールを無視します。日本語のテキストレイアウト要件 [1] では、ピリオドの優先度がカンマよりも高いとされているため (つまり、ピリオドは伸ばしやすい)、この機能によってカンマの優先度がピリオドよりも高くなります。luatexja-adjust の priority 機能が有効になっている場合にのみ機能します (true に設定)。

5 行間句読点機能

ここでは、行間句読点に関する詳細な情報、発生する可能性のある問題、および解決策が提供されます。

5.1 「垂直懸垂」とは

垂直懸垂は、中国の古典的な書籍に見られる、句読点と伝統的な縦書き組版方法を組み合わせたものである。

Eva-JFM では、ピリオドとカンマだけが懸垂されますが、Eva-JFM ではコロンとセミコロンの向きのため、これらの文字は懸垂できません。日本語フォントにはこの点に関して異なる特徴があります。

これらの文字は、グリフの下右に懸垂されます。詳細については次のサブセクションを参照してください。

5.2 懸垂位置

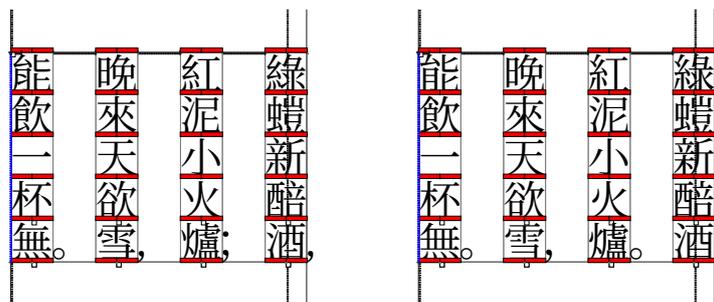


図1 垂直懸垂の機能

図1に示すように、これらの懸垂句読点の位置は、以下のルールに従って決定されます。カスタマイズについては、サブセクション 5.3 を参照してください。ルールの優先順位は早い順に高くなります。

- 3つのフォントのスタイルは統一されています；
- 異なる句読点における同じ要素の位置は同じであるべきです；
- 句読点のグリフは漢字の境界に接触するべきです；
- 異なる句読点の位置は、それぞれのグリフの形状、サイズ、デザインに応じて異なる場合があります。

5.3 ユーザー設定

この機能は Source Han フォントシリーズ（思源系列）に設計されています。異なるフォントによって異なる句読点があるため、出力が誤った結果になる可能性があります（オーバーラップ、揃っていないなど）。そのため、懸垂句読点の位置をカスタマイズするための2つの方法が提供されています。

5.3.1 パラメータの変更

Eva-JFM では、これらの懸垂句読点の位置に関するパラメータを含むテーブルは以下のようになります

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

上下のパラメータを調整して、出力を修正してください。また、最後のセクション『実装』も参照してください。

5.3.2 追加フォントの使用

句読点のグリフを抽出して、新しいフォントにパッケージ化し、後で行頭の句読点に使用することが、2番目の解決策です（Fontforge などのプログラムを使用できます）。また、句読点だけのために別のフォントをロードすることもできますが、CJ フォントを TeX のメモリにロードするコストは高くなります。

このフォントをインストールした後、LuaTeX-ja で提供される AltFont キーを使用して、句読点を置き換えることができます。実際のコードは上記に示されています：

```
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>}
```

最初の `<language>` オプションには、日本語、繁体字中国語、簡体字中国語のいずれかを入力し、もう一方は対応する JFM の機能のためのものです。 `<utf-8 code>` は、置き換えたい句読点を「句読点フォント」で選択します³。最後に、`<symbol font>` と `<main font>` オプションは、「句読点フォント」とメインフォントのためのものであることが明らかです。

開発者は、以下のように NFSS を使用することを推奨します

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

オプション `<base>` と `<alt>` は、メインフォントと「句読点フォント」のためのものです。

詳しい構文や使用方法、例については、LuaTeX-ja のドキュメント [2] を参照してください。

6 インスピレーション

Eva-JFM の内部グループ化は、`min10.tfm` [5] に触発されています。また、優先度の特徴のデータは、阿部紀行氏の `jlreq.lua` [6] から一部引用しています。

この JFM の名前は、庵野秀明氏のアニメーション『新世紀エヴァンゲリオン』。

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.

```
1 ---- Evangelion Japanese Font Metric for LuaTeX
```

³ Unicode のコードポイントは <https://www.unicode.org/charts/unihanrsindex.html> で確認できます。

```

2 ---- Current Version: 1.0.3 (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, 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 "smpl"\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}, {-1,
-2}}), {priority = logic_if(std_nil, {-1, 0}, {-1, -2}})},
95             [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, -2}, {-1,
0}}), {priority = logic_if(std_nil, {-1, -2}, {-1, 0}})},
96             [3] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1,
priority = {-1, -1}}, {priority = {-1, -1}})),

```

```

97     [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
98     [9] = {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}}
99     },
100     round_threshold = 0.01
101 },
102
103 [1] = { -- 読点類
104     chars = logic_anif(dir_vt, punc_lg, {}, {'、', ',', '。'}),
105     width = 0.5,
106     height = context_height(),
107     depth = context_depth(),
108     italic = 0,
109     left = 0,
110     down = 0,
111     align = logic_if(lang_tc, 'middle', 'left'),
112     glue = {
113         [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})}),
114         [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
115         [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
116         [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}})),
117         [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})}),
118         [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})}),
119         [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})}),
120         [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})}),
121         [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,
166         0})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0})}),
167         [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
168         [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
169         [3] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5,
170         0, 0.25, priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25,
171         ratio = 0, priority = {0, -1}})),
172         [4] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0,
173         0})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0, 0})}),
174         [5] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0,
175         0})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0, 0})}),
176         [6] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1,
177         0})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0})}),
178         [7] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1,
179         0})}, {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0})}),
180         [8] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0}, {}),
181         [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
182         priority = {0, -1}})
183     },
184     end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0.5, 0}, {0, 0}))
185 },

```

```

179 [201] = { -- 句點類 (行間a)
180     chars = logic_anif(dir_vt, punc_lg, {'.'}, {}),
181     width = 0,
182     height = context_height(),
183     depth = context_height(),
184     italic = 0,
185     left = logic_if(lang_tc, 0.68, 0.34),
186     down = logic_if(lang_tc, -0.58, -0.28),
187     align = 'left',
188     glue = {
189         [0] = lgp_kanjiskip
190     }
191 },
192
193 [202] = { -- 句點類 (行間b)
194     chars = logic_anif(dir_vt, punc_lg, {'。'}, {}),
195     width = 0,
196     height = context_height(),
197     depth = context_height(),
198     italic = 0,
199     left = 0.42,
200     down = -0.35,
201     align = 'left',
202     glue = {
203         [0] = lgp_kanjiskip
204     }
205 },
206
207 [3] = { -- 兩點類
208     chars = logic_if(lang_jp, {}, (logic_anif(dir_vt, punc_lg, {}, {':', ';'}))),
209     width = logic_if(dir_vt, 1, 0.5),
210     height = context_height(),
211     depth = context_depth(),
212     italic = 0,
213     left = 0,
214     down = 0,
215     align = logic_if(lang_tc, 'middle', 'left'),
216     glue = {
217         [0] = 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}})),
218         [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}})),
219         [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}})),
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', ' >', ' ズ', ' ア', ' イ', ' ウ', ' エ', ' オ', ' ツ', ' ヤ', ' ユ',
262         ' ヨ', ' ワ', ' カ', ' ケ', ' \', ' ヴ', ' 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(),
268     italic = 0,
269     left = 0,
270     down = 0,
271     align = 'middle',
272     glue = {
273         [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
274         [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1}, {}),
275         [3] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {priority = {0, -1}}),
276         [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
277         [9] = {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}
278     }
279 },
280
281 [5] = { -- 疑問感嘆類
282     chars = {'!', '?', '!!', '!', '?!', '??'},
283     width = logic_if(dir_vt, 1, logic_if(lang_sc, 0.5, 1)),
284     height = context_height(),
285     depth = context_depth(),
286     italic = 0,
287     left = 0,
288     down = 0,
289     align = logic_if(dir_vt, 'middle', logic_if(lang_sc, 'left', 'middle')),
290     glue = {
291         [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}})),
292         [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}, {})),
293         [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}, {})),
294         [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}})),
295         [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})),
296         [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
297         [8] = logic_anif(not dir_vt, lang_sc, {0.5, 0, 0.25, ratio = 0}, {}),
298         [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}})
299     }
300 },
301
302 [6] = { -- 分離禁止類
303     chars = {'—', '—', '…', '…', '☒', '/', '^', '\\'},
304     width = 1,
305     height = context_height(),
306     depth = context_depth(),
307     italic = 0,

```



```

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, priroity = {-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)

```