Vertical writing

Everything below the rule is set with one option: -direction ttb. The columns run down the page and, being Japanese, they are placed right to left - which the caller does, one [text] call per column, because tclpdf sets one column and does not break a paragraph into several.

秋の田のかりほの庵の苫をあらみ
わが衣手は露にぬれつつ
春過ぎて夏来にけらし白妙の
衣ほすてふ天の香具山

同じ書体、横書き
the same face, set horizontally

The metric is a different one

U+3031, the vertical kana repeat mark, is 1000 units wide and 2000 units tall. Below, the same two characters are set once across and once down, both at 14 pt. [textWidth] answers the extent ALONG the line in both directions, so it says 28 pt for the row and 42 pt for the column - and the mark takes two cells there, which is what vmtx says and what no reading of hmtx could have told.

〈日

〈
日

down: 42.0 pt - across: 28.0 pt
the mark takes two cells going down and one going across

Some of the glyphs are other glyphs

The same string twice, across and down. The brackets turn, the ideographic comma and full stop move from the bottom left of their square to the top right, and the ellipsis stands up. None of it is a rotation applied by tclpdf - each is a different glyph in the face, reached through the GSUB feature "vert".

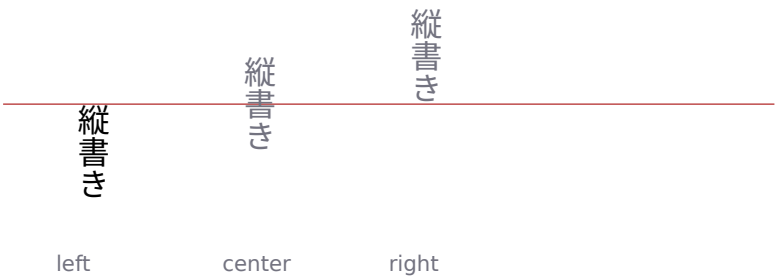
「引用」、…。

「
引
用
」
、
…
。

down at the left, across beside it
- same characters, same face

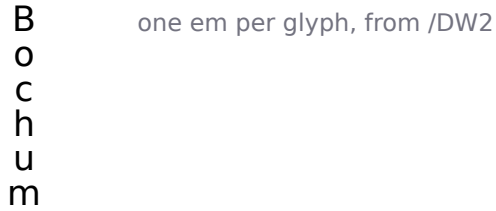
Where -at sits on the column

-align says where the given point falls ON the line, and for a vertical line that is its own axis: left begins at -at, right ends there, center is centred on it. The three columns below are drawn at the same y, marked by the rule, and differ only in -align.



A face that has no vertical metric

DejaVu Sans carries neither vhea nor vmtx, and a vertical line in it is not refused for that. ISO 32000-2 defines the case and gives it a default of one em per glyph, so the letters come out in a column of square cells - which is the only defined answer, and the file then says /DW2 and no /W2 at all, because every glyph in it wants exactly the default.



What a vertical line will not do

Both messages below were caught from this very page. The first is the block road: -width breaks lines by width and steps them down by the leading, which is the direction a vertical line already runs in, so it is refused by name rather than drawn. The second is the writing mode itself - it lives in the Type 0 font's CMap, and a standard face is addressed through WinAnsiEncoding, which has none.

tclpdf: -direction ttb sets ONE vertical line, and the block road PLACES its lines down the page - which is the direction a vertical line already writes in. Breaking such a text into columns is a different question and is answered: [$doc textLines $text -width $columnHeight -direction ttb] returns the columns, and each is set with its own [text] call at its own -at, moving leftwards. Or drop -direction ttb for this block

tclpdf: -direction ttb needs a TrueType or OpenType face embedded with [font embed] - "helvetica" is addressed through WinAnsiEncoding, which has no vertical writing mode; embed a face with [font embed] for this text

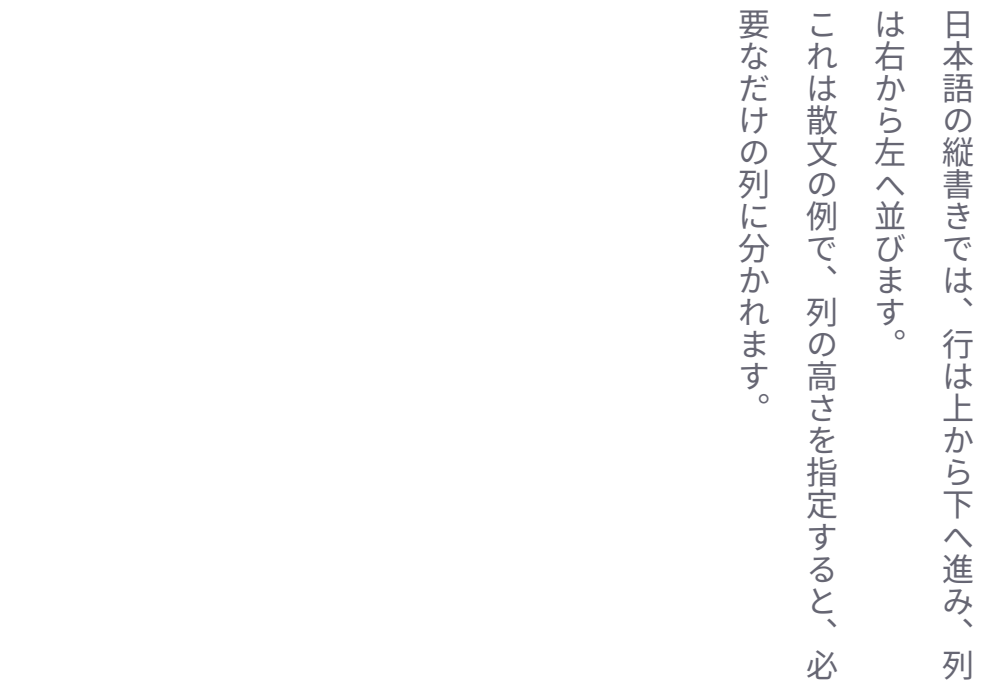
Check it yourself

The first command shows the two Type 0 fonts: one Identity-H and one Identity-V over the SAME descendant, the same descriptor and one embedded font file - the writing mode is a property of the CMap, not a second copy of the face. The second shows /DW2 at the standard's own default and the /W2 entries for the glyphs that differ from it, three numbers each. The third is the one that matters for a reader: the columns have to come back in the order they were written, which is what says the ToUnicode map was not turned round with the glyphs. The fourth renders the page, because a column whose characters are each right and whose spacing is wrong looks correct until it is looked at.

```
qpdf --qdf --object-streams=disable 02.15-vertical-writing.pdf - | grep -B4 Identity-
qpdf --qdf --object-streams=disable 02.15-vertical-writing.pdf - | grep -A12 DW2
pdftotext -f 1 -l 1 02.15-vertical-writing.pdf - | sed -n 1,6p
pdftoppm -f 1 -l 1 -r 150 -png 02.15-vertical-writing.pdf page
```

Prose, broken into columns

One call breaks it, one loop places it. -width is the column height; the columns come back in reading order and are set from the right. What is deliberately NOT built is the placing - see the manual under "Vertical writing" for why half of it would be worse than this loop.



4 columns at a column height of 90 mm. The same text at half that height needs 8 - which is the measurement that says -width is being read along the writing direction and not across it.