Vertical layout rendering

109:2024, Version 1

109:2024, Version 1

Ribose

Contents

Ribose Asia Limited

Suite 1, 8/F, New Henry House
10 Ice House Street
Central
Hong Kong

copyright@ribose.com
www.ribose.com

Foreword

This document is part of the Metanorma specifications series that defines requirements for document rendering.

Introduction

East Asian languages, particularly Chinese, Japanese, and Korean (CJK), have a long history of vertical text layout where text flows from top to bottom and columns are arranged from right to left. This traditional layout remains important in modern publishing, especially for literary works, traditional documents, and designs emphasizing cultural aesthetics.

While modern CJK publishing often uses horizontal text layout similar to Western languages, vertical layout continues to be essential for:

- Traditional literary works
- Classical texts
- Newspaper layouts
- Artistic and design applications
- Cultural and historical documents
- Religious texts
- Certificates and formal documents

Metanorma, as a standards document processing system, supports both horizontal and vertical text layouts to meet the needs of international standardization communities. This capability is particularly important for standards organizations in East Asian regions where vertical text layout is frequently required.

1. Scope

This document specifies the requirements for Metanorma content renderers to implement character conversion and transformation when rendering text in vertical layout. It defines:

- Character mappings between horizontal and vertical forms
- Transformation rules for characters without dedicated vertical forms
- Implementation requirements for vertical text rendering
- Conformance criteria for Metanorma vertical layout support

2. Normative references

There are no normative references in this document.

3. Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1. vertical text layout PREFERRED

text layout where characters are laid out vertically, typically from top to bottom with multiple columns arranged from right to left

3.2. glyph transformation PREFERRED

modification of a character's glyph when displayed in vertical text layout

3.3. vertical presentation form PREFERRED

dedicated Unicode character representing the vertical form of a horizontal character

3.4. typographic transformation PREFERRED

modification of a character's appearance without using a dedicated vertical form character

4. Technical background

4.1. Vertical text layout in CJK publishing

Vertical text layout has been the traditional format for East Asian documents for over two millennia. Modern digital publishing systems must support both vertical and horizontal layouts to maintain cultural continuity while embracing contemporary publishing needs.

4.2. Metanorma vertical layout support

Metanorma implements vertical layout support through:

- Automatic detection of document language and script
- Application of appropriate layout rules based on document type and context

- Character transformation according to Unicode standards
- Support for mixed horizontal and vertical content

NOTE the default type of layout applied depends on the flavor and document type.

5. Implementation requirements

5.1. General principles

A conforming Metanorma renderer shall:

1. Support automatic conversion between horizontal and vertical forms of characters
2. Implement correct character transformations as specified in this document
3. Maintain proper text flow and alignment in vertical layout
4. Handle mixed-direction text appropriately
5. Support vertical forms of punctuation marks and symbols

5.2. Character transformation categories

5.2.1. Characters with dedicated vertical forms that can't be achieved through rotation the character

TABLE 1: Brackets and parentheses transformation

Horizontal Half-width	Horizontal Full-width	Vertical Form	Implementation Instructions
-	U+3001 IDEOGRAPHIC COMMA (、)	U+FE11 PRESENTATION FORM FOR VERTICAL IDEOGRAPHIC COMMA ( )	-
-	U+FE50 SMALL COMMA (﹐)	U+FE10 PRESENTATION FORM FOR VERTICAL COMMA ( )	-
-	U+FE51 SMALL IDEOGRAPHIC COMMA (﹑)	U+FE11 PRESENTATION FORM FOR VERTICAL IDEOGRAPHIC COMMA ( )	-

Horizontal Half-width	Horizontal Full-width	Vertical Form	Implementation Instructions
-	U+FF0C FULLWIDTH COMMA (，)	U+FE10 PRESENTATION FORM FOR VERTICAL COMMA (﹐)	-
-	U+FF1A FULLWIDTH COLON (：)	U+FE13 PRESENTATION FORM FOR VERTICAL COLON (﹕)	-
-	U+FF1B FULLWIDTH SEMICOLON (；)	U+FE14 PRESENTATION FORM FOR VERTICAL SEMICOLON (﹔)	-
-	U+FF01 FULLWIDTH EXCLAMATION MARK (！)	U+FE15 PRESENTATION FORM FOR VERTICAL EXCLAMATION MARK (﹖)	-
-	U+FF1F FULLWIDTH QUESTION MARK (？)	U+FE16 PRESENTATION FORM FOR VERTICAL QUESTION MARK (﹗)	-

5.2.2. Brackets and parentheses for vertical layout

TABLE 2: Brackets and parentheses transformation

Horizontal Half-width	Horizontal Full-width	Vertical Form	Implementation Instructions
U+0028 LEFT PARENTHESIS (()	U+FF08 FULLWIDTH LEFT PARENTHESIS (（)	U+FE35 PRESENTATION FORM FOR VERTICAL LEFT PARENTHESIS (﹁)	Rotate 90° clockwise
U+0029 RIGHT PARENTHESIS ())	U+FF09 FULLWIDTH RIGHT PARENTHESIS (）)	U+FE36 PRESENTATION FORM FOR VERTICAL RIGHT PARENTHESIS (﹂)	Rotate 90° clockwise
-	U+FE59 SMALL LEFT PARENTHESIS (﹙)	-	Rotate 90° clockwise
-	U+FE5A SMALL RIGHT PARENTHESIS (﹚)	-	Rotate 90° clockwise
-	U+FF5F FULLWIDTH LEFT WHITE PARENTHESIS (⦅)	-	Rotate 90° clockwise
-	U+FF60 FULLWIDTH RIGHT WHITE PARENTHESIS (⦆)	-	Rotate 90° clockwise
U+007B LEFT CURLY BRACKET ({)	U+FF5B FULLWIDTH LEFT CURLY BRACKET (｛)	U+FE37 PRESENTATION FORM FOR VERTICAL LEFT CURLY BRACKET (﹃)	Rotate 90° clockwise

Horizontal Half-width	Horizontal Full-width	Vertical Form	Implementation Instructions
U+007D RIGHT CURLY BRACKET (})	U+FF5D FULLWIDTH RIGHT CURLY BRACKET (｝)	U+FE38 PRESENTATION FORM FOR VERTICAL RIGHT CURLY BRACKET (︸)	Rotate 90° clockwise
-	U+FE5B SMALL LEFT CURLY BRACKET (﹛)	-	Rotate 90° clockwise
-	U+FE5C SMALL RIGHT CURLY BRACKET (﹜)	-	Rotate 90° clockwise
-	U+3014 LEFT TORTOISE SHELL BRACKET (〔)	U+FE39 PRESENTATION FORM FOR VERTICAL LEFT TORTOISE SHELL BRACKET (︹)	Rotate 90° clockwise
-	U+3015 RIGHT TORTOISE SHELL BRACKET (〕)	U+FE3A PRESENTATION FORM FOR VERTICAL RIGHT TORTOISE SHELL BRACKET (︺)	Rotate 90° clockwise
-	U+3018 LEFT WHITE TORTOISE SHELL BRACKET (〘)	-	Rotate 90° clockwise
-	U+3019 RIGHT WHITE TORTOISE SHELL BRACKET (〙)	-	Rotate 90° clockwise
-	U+FE5D SMALL LEFT TORTOISE SHELL BRACKET (﹝)	-	Rotate 90° clockwise
-	U+FE5E SMALL RIGHT TORTOISE SHELL BRACKET (﹞)	-	Rotate 90° clockwise
-	U+3010 LEFT BLACK LENTICULAR BRACKET (【)	U+FE3B PRESENTATION FORM FOR VERTICAL LEFT BLACK LENTICULAR BRACKET (︻)	Rotate 90° clockwise
-	U+3011 RIGHT BLACK LENTICULAR BRACKET (】)	U+FE3C PRESENTATION FORM FOR VERTICAL RIGHT BLACK LENTICULAR BRACKET (︼)	Rotate 90° clockwise
-	U+300A LEFT DOUBLE ANGLE BRACKET (《)	U+FE3D PRESENTATION FORM FOR VERTICAL LEFT DOUBLE ANGLE BRACKET (︽)	Rotate 90° clockwise
-	U+300B RIGHT DOUBLE ANGLE BRACKET (》)	U+FE3E PRESENTATION FORM FOR VERTICAL	Rotate 90° clockwise

Horizontal Half-width	Horizontal Full-width	Vertical Form	Implementation Instructions
		RIGHT DOUBLE ANGLE BRACKET (﹀)	
U+FF62 HALFWIDTH LEFT CORNER BRACKET (｢)	U+300C LEFT CORNER BRACKET (「)	U+FE41 PRESENTATION FORM FOR VERTICAL LEFT CORNER BRACKET (﹁)	Rotate 90° clockwise
U+FF63 HALFWIDTH RIGHT CORNER BRACKET (｣)	U+300D RIGHT CORNER BRACKET (」)	U+FE42 PRESENTATION FORM FOR VERTICAL RIGHT CORNER BRACKET (﹂)	Rotate 90° clockwise
-	U+300E LEFT WHITE CORNER BRACKET (『)	U+FE43 PRESENTATION FORM FOR VERTICAL LEFT WHITE CORNER BRACKET (﹃)	Rotate 90° clockwise
-	U+300F RIGHT WHITE CORNER BRACKET (』)	U+FE44 PRESENTATION FORM FOR VERTICAL RIGHT WHITE CORNER BRACKET (﹄)	Rotate 90° clockwise
U+005B LEFT SQUARE BRACKET ([)	U+FF3B FULLWIDTH LEFT SQUARE BRACKET ([)	U+FE47 PRESENTATION FORM FOR VERTICAL LEFT SQUARE BRACKET (﹇)	Rotate 90° clockwise
U+005D RIGHT SQUARE BRACKET (])	U+FF3D FULLWIDTH RIGHT SQUARE BRACKET (])	U+FE48 PRESENTATION FORM FOR VERTICAL RIGHT SQUARE BRACKET (﹈)	Rotate 90° clockwise
-	U+2329 LEFT-POINTING ANGLE BRACKET (〈)	-	Rotate 90° clockwise
-	U+232A RIGHT-POINTING ANGLE BRACKET (〉)	-	Rotate 90° clockwise
-	U+3008 LEFT ANGLE BRACKET (〈)	U+FE3F PRESENTATION FORM FOR VERTICAL LEFT ANGLE BRACKET (﹈)	Rotate 90° clockwise
-	U+3009 RIGHT ANGLE BRACKET (〉)	U+FE40 PRESENTATION FORM FOR VERTICAL RIGHT ANGLE BRACKET (﹀)	Rotate 90° clockwise
-	U+3016 LEFT WHITE LENTICULAR BRACKET (〖)	U+FE17 PRESENTATION FORM FOR VERTICAL LEFT WHITE LENTICULAR BRACKET (﹇)	Rotate 90° clockwise

Horizontal Half-width	Horizontal Full-width	Vertical Form	Implementation Instructions
-	U+3017 RIGHT WHITE LENTICULAR BRACKET (〗)	U+FE18 PRESENTATION FORM FOR VERTICAL RIGHT WHITE LENTICULAR BRACKET (﹘)	Rotate 90° clockwise
-	U+301A LEFT WHITE SQUARE BRACKET (〚)	-	Rotate 90° clockwise
-	U+301B RIGHT WHITE SQUARE BRACKET (〛)	-	Rotate 90° clockwise

5.2.3. Special marks and symbols for vertical layout

TABLE 3: Special marks and symbols transformation

Horizontal Half-width	Horizontal Full-width	Vertical Form	Implementation Instructions
-	U+3002 IDEOGRAPHIC FULL STOP (。)	-	Rotate 180°
-	U+FE52 SMALL FULL STOP (﹒)	-	Rotate 180°
-	U+FF0E FULLWIDTH FULL STOP (．)	-	Rotate 180°
-	U+3003 DITTO MARK (〃)	-	No rotation needed
-	U+301C WAVE DASH (〜)	-	Rotate 90° clockwise
-	U+3030 WAVY DASH (〰)	-	Rotate 90° clockwise
-	U+FF5E FULLWIDTH TILDE (～)	-	Rotate 90° clockwise
-	U+301D REVERSED DOUBLE PRIME QUOTATION MARK (〝)	-	Special positioning (see UTR #50)
-	U+301E DOUBLE PRIME QUOTATION MARK (〞)	-	Special positioning (see UTR #50)
-	U+301F LOW DOUBLE PRIME QUOTATION MARK (〟)	-	Special positioning (see UTR #50)
U+FF65 HALFWIDTH KATAKANA MIDDLE DOT (･)	U+30FB KATAKANA MIDDLE DOT (・)	-	No rotation needed
-	U+30FC KATAKANA-HIRAGANA PROLONGED SOUND MARK (ー)	-	Rotate 90° clockwise

Horizontal Half-width	Horizontal Full-width	Vertical Form	Implementation Instructions
-	U+309B KATAKANA-HIRAGANA VOICED SOUND MARK (゛)	-	Special positioning (see UTR #50)
-	U+309C KATAKANA-HIRAGANA SEMI-VOICED SOUND MARK (゜)	-	Rotate 180°
-	U+30A0 KATAKANA-HIRAGANA DOUBLE HYPHEN (゠)	-	Rotate 90° clockwise
-	U+FF3F FULLWIDTH LOW LINE (＿)	-	Rotate 90° clockwise
-	U+FF5C FULLWIDTH VERTICAL LINE (｜)	-	Rotate 90° clockwise
-	U+FFE3 FULLWIDTH MACRON (￣)	-	Rotate 90° clockwise

5.3. Rendering requirements

5.3.1. Character conversion

The renderer shall:

1. Use dedicated vertical presentation forms when available and the vertical forms can't be easily achieved through rotation the character
   NOTE: Using the glyphs that the vertical forms are easily achieved through rotation can preserve the copy-paste text into the text editor in horizontal mode, and the easy searching also.
2. Apply appropriate rotations for characters without vertical forms
3. Implement special positioning rules for specific characters
4. Maintain correct spacing and alignment in vertical layout

5.3.2. Text flow

The renderer shall:

1. Arrange text in top-to-bottom columns
2. Order columns from right to left
3. Handle mixed horizontal and vertical content appropriately
4. Maintain proper line breaks and justification

5.3.3. Special cases

5.3.3.1. Mixed-direction text

For text containing both horizontal and vertical elements, the renderer shall:

1. Maintain proper orientation for each text segment
2. Ensure correct alignment and spacing between different directions
3. Handle embedded horizontal elements within vertical text appropriately

5.3.3.2. Numerical content

For numerical content in vertical text, the renderer shall:

1. Keep multi-digit numbers in horizontal orientation
2. Maintain proper alignment with surrounding vertical text
3. Handle mixed alphanumeric content appropriately

6. Conformance criteria

6.1. Basic conformance

A conforming implementation shall:

1. Support all character transformations specified in this document
2. Implement correct text flow for vertical layout
3. Handle mixed-direction text appropriately

6.2. Extended conformance

A conforming implementation should:

1. Support advanced typography features
2. Handle complex layout scenarios
3. Provide fallback mechanisms for unsupported characters

7. Testing and verification

7.1. Character transformation testing

Implementations shall be tested against:

1. Correct mapping of characters to vertical forms
2. Proper rotation of transformed characters
3. Accurate positioning of special characters

7.2. Layout testing

Implementations shall be verified for:

1. Correct column ordering and flow
2. Proper handling of mixed-direction text
3. Accurate spacing and alignment

Bibliography

[1] UTR #50, Unicode Technical Report #50: Unicode Vertical Text Layout

[2] Unicode, The Unicode Standard, Version 15.1

[3] ISO/IEC 10646:2020, International Organization for Standardization (committee). *Information technology — Universal coded character set (UCS)*. Sixth edition. 2020. Geneva: International Organization for Standardization and International Electrotechnical Commission. https://www.iso.org/standard/76835.html.