

Proposed Update Unicode® Standard Annex #11**EAST ASIAN WIDTH**

Version	Unicode 18.0.0
Editor	Ken Lunde 小林劍
Date	2026-04-26
This Version	https://www.unicode.org/reports/tr11/tr11-45.html
Previous Version	https://www.unicode.org/reports/tr11/tr11-44.html
Latest Version	https://www.unicode.org/reports/tr11/
Latest Proposed Update	https://www.unicode.org/reports/tr11/proposed.html
Revision	45

Summary

This annex introduces the concept of inherent width, and presents the specification of a normative property for Unicode characters that is useful when processing text that needs to account for inherent width. This property is of particular use when interoperating with East Asian legacy character sets, but is applicable beyond that. The annex also discusses the interaction between inherent width and variation sequences.

Status

*This is a **draft** document which may be updated, replaced, or superseded by other documents at any time. Publication does not imply endorsement by the Unicode Consortium. This is not a stable document; it is inappropriate to cite this document as other than a work in progress.*

A Unicode Standard Annex (UAX) forms an integral part of the Unicode Standard, but is published online as a separate document. The Unicode Standard may require conformance to normative content in a Unicode Standard Annex, if so specified in the Conformance chapter of that version of the Unicode Standard. The version number of a UAX document corresponds to the version of the Unicode Standard of which it forms a part.

Please submit corrigenda and other comments with the online reporting form [[Feedback](#)]. Related information that is useful in understanding this annex is found in Unicode Standard Annex #41, “[Common References for Unicode Standard Annexes](#).” For the latest version of the Unicode Standard, see [[Unicode](#)]. For a list of current Unicode Technical Reports, see [[Reports](#)]. For more information about versions of the Unicode Standard, see [[Versions](#)]. For any errata which may apply to this annex, see [[Errata](#)].

Contents

- 1 [Overview](#)
- 2 [Scope](#)
- 3 [Description](#)
- 4 [Definitions](#)
 - 4.1 [Relation to the Terms “Fullwidth” and “Halfwidth”](#)

- 4.2 [Ambiguous Characters](#)
- 4.3 [Variation Sequences](#)
- 5 [Recommendations](#)
- 6 [Classifications](#)
 - 6.1 [Unassigned and Private-Use Characters](#)
 - 6.2 [Combining Marks](#)
 - 6.3 [Data File](#)
 - 6.4 [Adding Characters](#)
- [References](#)
- [Acknowledgments](#)
- [Modifications](#)

1 Overview

When processing text, there is the concept of an *inherent* width of a character, which is of particular importance in an East Asian context. This width takes on either of two values: *narrow* or *wide*. For traditional mixed-width East Asian legacy character sets, this classification into narrow and wide corresponds with few exceptions directly to the storage size for each character: a few narrow characters use a single byte per character and all other characters (usually wide) use two or more bytes.

Layout and line breaking (to cite only two examples) in East Asian context show systematic variations depending on the value of this `East_Asian_Width` property. *Wide* characters behave like ideographs; they tend to allow line breaks after each character and remain upright in vertical text layout. *Narrow* characters are kept together in words or runs that are rotated sideways in vertical text layout.

For a traditional East Asian *fixed pitch* font, this width translates to a display width of either one half or a whole unit width. A common name for this unit width is “Em.” While an Em is customarily the *height* of the letter “M,” it is the same as the unit *width* in East Asian fonts, because in these fonts the standard character cell is square. In contrast, the character width for a fixed-pitch Latin font like Courier is generally 3/5 of an Em.

In modern practice, most alphabetic characters are rendered by variable-width fonts using narrow characters, even if their encoding in common legacy sets uses multiple bytes. In contrast, emoji characters were first developed through the use of extensions of legacy East Asian encodings, such as Shift-JIS, and in such a context they were treated as wide characters. While these extensions have been added to Unicode or mapped to standardized variation sequences, their treatment as wide characters has been retained, and extended for consistency with emoji characters that lack a legacy encoding.

Except for a few characters, which are explicitly called out as *fullwidth* or *halfwidth* in the Unicode Standard, characters are not duplicated based on distinction in width. Some characters, such as the ideographs, are always wide; others are always narrow; and some can be narrow or wide, depending on the context. The Unicode character property *East_Asian_Width* provides a default classification of characters, which an implementation can use to decide at runtime whether to treat a character as narrow or wide.

The `East_Asian_Width` property does not preserve canonical equivalence, because the base characters of canonical decompositions almost always have a different `East_Asian_Width` property value than the precomposed characters. Decomposing a character, and applying the `East_Asian_Width` property to a base character and combining marks separately does not yield the expected values. For examples, see Section 4, [Definitions](#).

2 Scope

`East_Asian_Width` is a normative property and provides a useful concept for implementations that

- Have to interwork with East Asian legacy character encodings
- Support both East Asian and Western typography and line layout
- Need to associate fonts with unmarked text runs containing East Asian characters

This annex gives general guidelines how to use this property. It does not provide rules or specifications of how this property might be used in font design or line layout, because, while a useful property for this purpose, it is only one of several character properties that would need to be considered. While the specific assignments of property values for given characters may change over time, it is generally not intended to reflect evolving practice for existing characters. In particular some alphabetic and symbol characters are treated as wide in certain East Asian legacy character set implementations, and as narrow in all other cases. Instead, the guidelines on use of this property should be considered recommendations based on a particular legacy practice that may be overridden by implementations as necessary.

Note: The `East_Asian_Width` property is not intended for use by modern terminal emulators without appropriate tailoring on a case-by-case basis. Such terminal emulators need a way to resolve the halfwidth/fullwidth dichotomy that is necessary for such environments, but the `East_Asian_Width` property does not provide an off-the-shelf solution for all situations. The growing repertoire of the Unicode Standard has long exceeded the bounds of East Asian legacy character encodings, and terminal emulations often need to be customized to support edge cases and for changes in typographical behavior over time.

3 Description

By convention, 1/2 Em wide characters of East Asian legacy encodings are called “halfwidth” (or *hankaku* characters in Japanese); the others are called correspondingly “fullwidth” (or *zenkaku*) characters. Legacy encodings often use a single byte for the halfwidth characters and two bytes for the fullwidth characters. In the Unicode Standard, no such distinction is made, but understanding the distinction is often necessary when interchanging data with legacy systems, especially when fixed-size buffers are involved.

Some character blocks in the compatibility zone contain characters that are explicitly marked “halfwidth” and “fullwidth” in their character name, but for all other characters the width property must be implicitly derived. Some characters behave differently in an East Asian context than in a non-East Asian context. Their default width property is considered ambiguous and needs to be resolved into an actual width property based on context.

The Unicode Character Database [[UCD](#)] assigns to each Unicode character as its default width property one of six values: *Ambiguous*, *Fullwidth*, *Halfwidth*, *Narrow*, *Wide*, or *Neutral* (= *Not East Asian*). For any given operation, these six default property values resolve into only two property values, *narrow* and *wide*, depending on context.

4 Definitions

All terms not defined here shall be as defined elsewhere in the Unicode Standard.

ED1. *East_Asian_Width*: In the context of interoperating with East Asian legacy character encodings and implementing East Asian typography, `East_Asian_Width` is a categorization of character. It can take on two abstract values, *narrow* and *wide*.

In legacy implementations, there is often a corresponding difference in encoding length (one or two bytes) as well as a difference in displayed width. However, the *actual* display width of a glyph is given by the font and may be further adjusted by layout. An important class of fixed-width legacy fonts contains glyphs of just two widths, with the wider glyphs twice as wide as the narrower glyphs.

Note: For convenience, the classification further distinguishes between explicitly and implicitly wide and narrow characters.

ED2. *East Asian Fullwidth (F)*: All characters that are defined as Fullwidth in the Unicode Standard [[Unicode](#)] by having a compatibility decomposition of type <wide> to characters elsewhere in the Unicode Standard that are implicitly narrow but unmarked.

Note: The Unicode property value aliases drop the common prefix East Asian for this and the following property values.

ED3. East Asian Halfwidth (H): All characters that are explicitly defined as Halfwidth in the Unicode Standard by having a compatibility decomposition of type <narrow> to characters elsewhere in the Unicode Standard that are implicitly wide but unmarked, plus U+20A9 ₩ WON SIGN.

Note: Unlike U+00A5 ¥ YEN SIGN, U+20A9 ₩ WON SIGN has an explicit `East_Asian_Width` property value of *East Asian Halfwidth (H)*. What makes U+00A5 different is that this character was included in a very common—and non-East Asian—character set standard, specifically ISO/IEC 8859-1, and encoded at 0xA5. Almost all legacy Latin fonts supported ISO/IEC 8859-1 in its entirety, using variable-width glyphs. By contrast, most legacy font implementations used an explicit half-width glyph for the won sign, whose source is the standard KS X 1003, and encoded at 0x5C. The assignment of the *East Asian Halfwidth (H)* property value does not preclude font developers from using a variable-width glyph for U+20A9, and doing so has become a common practice.

ED4. East Asian Wide (W): All other characters that are *always* wide. These characters occur only in the context of East Asian typography where they are wide characters (such as the Unified Han Ideographs or Squared Katakana Symbols). This category includes characters that have explicit halfwidth counterparts, along with characters that have the [UTS51] property *Emoji_Presentation*, with the exception of characters that have the [UCD] property *Regional_Indicator*

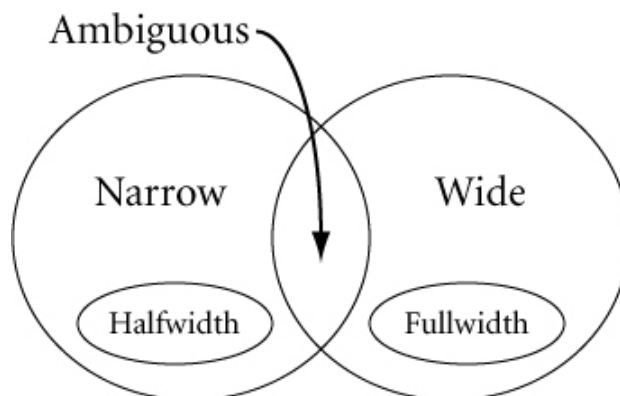
ED5. East Asian Narrow (Na): All other characters that are *always* narrow and have explicit fullwidth or wide counterparts. These characters are implicitly narrow in East Asian typography and legacy character sets because they have explicit fullwidth or wide counterparts. All of ASCII is an example of East Asian Narrow characters.

It is useful to distinguish characters explicitly defined as halfwidth from other narrow characters. In particular, halfwidth punctuation behaves in some important ways like ideographic punctuation, and knowing a character is a halfwidth character can aid in font selection when binding a font to unstyled text.

ED6. East Asian Ambiguous (A): All characters that can be sometimes wide and sometimes narrow. Ambiguous characters require additional information not contained in the character code to further resolve their width. See Section 4.2, [Ambiguous Characters](#), for more details.

Ambiguous characters occur in East Asian legacy character sets as *wide* characters, but as *narrow* (i.e., normal-width) characters in non-East Asian usage. (Examples are the basic Greek and Cyrillic alphabet found in East Asian character sets, but also some of the mathematical symbols.) Private-use characters are considered ambiguous by default, because additional information is required to know whether they should be treated as wide or narrow.

Figure 1. Venn Diagram Showing the Set Relations for Five of the Six Categories



When they are treated as *wide* characters, ambiguous characters would typically be rendered upright in vertical text runs.

Because East Asian legacy character sets do not always include complete case pairs of Latin characters, two members of a case pair may have different East_Asian_Width properties:

Ambiguous:	01D4	LATIN SMALL LETTER U WITH CARON
Neutral:	01D3	LATIN CAPITAL LETTER U WITH CARON

Canonical equivalents of ambiguous characters may not be ambiguous themselves. For example, U+212B Å ANGSTROM SIGN is Ambiguous, while its decomposition, U+00C5 Å LATIN CAPITAL LETTER A WITH RING ABOVE, is Neutral.

ED7. Neutral (Not East Asian): All other characters. Neutral characters do not occur in legacy East Asian character sets. By extension, they also do not tend to occur in East Asian typography. For example, there is no traditional Japanese way of typesetting Devanagari. Canonical equivalents of narrow and neutral characters may not themselves be narrow or neutral respectively. For example, U+00C5 Å LATIN CAPITAL LETTER A WITH RING ABOVE is Neutral, but its decomposition starts with a Narrow character.

Strictly speaking, it makes no sense to talk of narrow and wide for neutral characters, but because for all practical purposes they behave like Na, they are treated as narrow characters (the same as Na) under the recommendations below.

In a broad sense, *wide characters* include W, F, and A (when in East Asian context), and *narrow characters* include N, Na, H, and A (when not in East Asian context).

Figure 2. Examples for Each Character Class and Their Resolved Widths



4.1 Relation to the Terms “Fullwidth” and “Halfwidth”

When converting a DBCS mixed-width encoding to and from Unicode, the fullwidth characters in such a mixed-width encoding are mapped to the fullwidth compatibility characters in the FFxx block, whereas the corresponding halfwidth characters are mapped to ordinary Unicode characters (for example, ASCII in U+0021..U+007E, plus a few other scattered characters).

In the context of interoperability with DBCS character encodings, this restricted set of Unicode characters in the General Scripts area can be construed as halfwidth, rather than fullwidth. (This applies only to the restricted set of characters that can be paired with the fullwidth compatibility characters.)

In the context of interoperability with DBCS character encodings, all other Unicode characters that are not explicitly marked as halfwidth can be construed as fullwidth.

In any other context, Unicode characters not explicitly marked as being either fullwidth or halfwidth compatibility forms are neither halfwidth nor fullwidth.

Seen in this light, the “halfwidth” and “fullwidth” properties are not unitary character properties in the same sense as “space” or “combining” or “alphabetic.” They are, instead, relational properties of a pair

of characters, one of which is explicitly encoded as a halfwidth or fullwidth form for compatibility in mapping to DBCS mixed-width character encodings.

What is “fullwidth” by default today could in theory become “halfwidth” tomorrow by the introduction of another character on the SBCS part of a mixed-width code page somewhere, requiring the introduction of another fullwidth compatibility character to complete the mapping. However, because the single byte part of mixed-width character sets is limited, there are not going to be many candidates and neither the Unicode Technical Committee [UTC] nor WG2 has any intention to encode additional compatibility characters for this purpose.

4.2 Ambiguous Characters

Ambiguous width characters originated from characters that can occur as fullwidth characters in any of a number of East Asian legacy character encodings. They have a “resolved” width of either narrow or wide depending on the context of their use. As originally defined, the width of ambiguous characters resolves to narrow, if they are not used in the context of the specific legacy encoding to which they belong. Otherwise, it resolves to fullwidth or halfwidth. The term *context* as used here includes extra information such as explicit markup, knowledge of the source code page, font information, or language and script identification. For example:

- Greek characters resolve to narrow when used with a standard Greek font, because there is no East Asian legacy context.
- Private-use character codes and the replacement character have ambiguous width, because they may stand in for characters of any width.
- Ambiguous quotation marks are generally resolved to wide when they enclose and are adjacent to a wide character, and to narrow otherwise.

The `East_Asian_Width` property does not preserve canonical equivalence, because the base characters of canonical decompositions almost always have a different `East_Asian_Width` than the precomposed characters. The `East_Asian_Width` property is designed for use with legacy character sets so the property value is not designed to respect canonical equivalence.

Modern Rendering Practice. Modern practice is evolving toward rendering ever more of the ambiguous characters with proportionally spaced, narrow forms that rotate with the direction of writing, independent of their treatment in one or more legacy character sets. In other words, context information beyond the choice of font or source character set is employed to resolve the width of the character. This annex does not attempt to track such changes in practice; therefore, the set of characters with mappings to legacy character sets that have been assigned ambiguous width constitute a different set than the set of such characters that may be rendered as wide characters in a given context. In particular, an application might find it useful to treat characters from alphabetic scripts as narrow by default. Conversely, many of the symbols in the Unicode Standard have no mappings to legacy character sets, yet they may be rendered as “wide” characters if they appear in an East Asian context. An implementation might therefore elect to treat them as ambiguous even though they are classified as *neutral* here.

4.3 Variation Sequences

In some cases, variation sequences have been defined that request a character to be displayed as wide or narrow. These variation sequences are defined in `StandardizedVariants.txt` [StandardizedVariants] in the [UCD], but they are identified via the description of the intended glyph style and not via a formal property of the variation sequence. Nevertheless, the presence of the respective variation selector should be considered a sufficient condition to resolve ambiguous width to either wide or narrow, perhaps subject only to availability in the actual font used. If the font does not have a wide form, a variation sequence requesting the wide form would be ignorable and it makes no sense to treat the character as “Wide” if it is being displayed as “Narrow.”

5 Recommendations

When mapping Unicode to East Asian legacy character encodings

- Wide Unicode characters *always* map to fullwidth characters.

- Narrow (and neutral) Unicode characters *always* map to halfwidth characters.
- Halfwidth Unicode characters *always* map to halfwidth characters.
- Ambiguous Unicode characters *always* map to fullwidth characters.

When mapping Unicode to non–East Asian legacy character encodings

- Wide Unicode characters *do not* map to non–East Asian legacy character encodings.
- Narrow (and neutral) Unicode characters *always* map to regular (narrow) characters.
- Halfwidth Unicode characters *do not* map.
- Ambiguous Unicode characters *always* map to regular (narrow) characters.

When processing or displaying data

- Wide characters behave like ideographs in important ways, such as layout. Except for certain punctuation characters, they are not rotated when appearing in vertical text runs. In fixed-pitch fonts, they take up one Em of space.
- Halfwidth characters behave like ideographs in some ways, however, they are rotated like narrow characters when appearing in vertical text runs. In fixed-pitch fonts, they take up 1/2 Em of space.
- Narrow characters behave like Western characters, for example, in line breaking. They are rotated sideways, when appearing in vertical text. In fixed-pitch East Asian fonts, they take up 1/2 Em of space, but in rendering, a non–East Asian, proportional font is often substituted.
- Ambiguous characters behave like wide or narrow characters depending on the context (language tag, script identification, associated font, source of data, or explicit markup; all can provide the context). If the context cannot be established reliably, they should be treated as narrow characters by default.

Note: Some variation sequences may specify a narrow or wide presentation behaviour. (This can be true for sequences defined in StandardizedVariants.txt [StandardizedVariants] in the [UCD] or in emoji-variation-sequences.txt in [UTS51].) For such sequences, the width behaviour specified by a variation sequence can differ from the East_Asian_Width property of the base character of that variation sequence. The width indication of the variation sequence can specify some aspects of glyph appearance such as advance width, while not specifying other aspects of presentation, such as rotational behavior in vertical text.

6 Classifications

The classifications presented here are based on the most widely used mixed-width legacy character sets in use in East Asia as of this writing. In particular, the assignments of the Neutral or Ambiguous categories depend on the contents of these character sets. For example, an implementation that knows *a priori* that it needs to interchange data *only* with the Japanese Shift-JIS character set, but not with other East Asian character sets, could reduce the number of characters in the Ambiguous classification to those actually encoded in Shift-JIS. Alternatively, such a reduction could be done implicitly at runtime in the context of interoperating with Shift-JIS fonts or data sources. Conversely, if additional character sets are created and widely adopted for legacy purposes, more characters would need to be classified as ambiguous.

6.1 Unassigned and Private-Use Characters

All private-use characters are by default classified as Ambiguous, because their definition depends on context.

Unassigned code points in ranges intended for CJK ideographs are classified as Wide. Those ranges are:

- the CJK Unified Ideographs block, 4E00..9FFF
- the CJK Unified Ideographs Extension A block, 3400..4DBF
- the CJK Compatibility Ideographs block, F900..FAFF

- the Supplementary Ideographic Plane, 20000..2FFFFD
- the Tertiary Ideographic Plane, 30000..3FFFFD

All other unassigned code points are by default classified as Neutral.

For additional recommendations for handling the default property value for unassigned characters, see *Section 5.3, Unknown and Missing Characters*, in [Unicode].

6.2 Combining Marks

Combining marks have been classified and are given a property assignment based on their typical applicability. For example, combining marks typically applied to characters of class **N**, **Na**, or **W** are classified as **A**. Combining marks for purely non–East Asian scripts are marked as **N**, and nonspacing marks used only with wide characters are given a **W**. Even more so than for other characters, the `East_Asian_Width` property for combining marks is not the same as their display width.

In particular, nonspacing marks do not possess actual advance width. Therefore, even when displaying combining marks, the `East_Asian_Width` property cannot be related to the advance width of these characters. However, it can be useful in determining the encoding length in a legacy encoding, or the choice of font for the range of characters including that nonspacing mark. The width of the glyph image of a nonspacing mark should always be chosen as the appropriate one for the width of the base character.

6.3 Data File

The `East_Asian_Width` classification of all Unicode characters is listed in the file `EastAsianWidth.txt` [Data11] in the [Unicode Character Database \[UCD\]](#). This is a tab-delimited, two-column, plain text file, with code position and `East_Asian_Width` designator. A comment at the end of each line indicates the character name. Ideographic, Hangul, Surrogate, and Private Use ranges are collapsed by giving a range in the first column.

6.4 Adding Characters

As more characters are added to the Unicode Standard, or if additional character sets are created and widely adopted for legacy purposes, the assignment of `East_Asian_Width` may be changed for some characters. Implementations should not make any assumptions to the contrary. The sets of Narrow, Fullwidth, and Halfwidth characters are fixed for all practical purposes. New characters for most scripts will be Neutral characters; however, characters for East Asian scripts using wide characters will be classified as Wide. Symbol characters that are, or are expected to be, used both as wide characters in East Asian usage and as narrow characters in non–East Asian usage will be classified Ambiguous.

References

For references for this annex, see Unicode Standard Annex #41, “[Common References for Unicode Standard Annexes](#).”

Acknowledgments

Asmus Freytag was the author of the initial version of this annex, and served as the editor up through and including the version for Unicode 6.1.0.

Michel Suignard provided extensive input into the analysis and source material for the detail assignments of these properties. Mark Davis and Ken Whistler performed consistency checks on the data files at various times. Tomohiro Kubota reviewed the `East_Asian_Width` assignments against some common legacy encodings.

Modifications

The following summarizes modifications from the previous published version of this annex.

Revision 45

- **Proposed Update** for Unicode 18.0.0.
- The unassigned code point ranges in [Section 6.1](#) were adjusted.

Revision 44

- **Reissued** for Unicode 17.0.0.

Previous revisions can be accessed with the “Previous Version” link in the header.

© 1998–~~2025~~2026 Unicode, Inc. This publication is protected by copyright, and permission must be obtained from Unicode, Inc. prior to any reproduction, modification, or other use not permitted by the [Terms of Use](#). Specifically, you may make copies of this publication and may annotate and translate it solely for personal or internal business purposes and not for public distribution, provided that any such permitted copies and modifications fully reproduce all copyright and other legal notices contained in the original. You may not make copies of or modifications to this publication for public distribution, or incorporate it in whole or in part into any product or publication without the express written permission of Unicode.

Use of all Unicode Products, including this publication, is governed by the Unicode [Terms of Use](#). The authors, contributors, and publishers have taken care in the preparation of this publication, but make no express or implied representation or warranty of any kind and assume no responsibility or liability for errors or omissions or for consequential or incidental damages that may arise therefrom. This publication is provided “AS-IS” without charge as a convenience to users.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc., in the United States and other countries.