Technical Reports

L2/26-185

Proposed Update Unicode® Standard Annex #14

UNICODE LINE BREAKING ALGORITHM

Version	Unicode 18.0.0 (draft 2)
Editors	Robin Leroy (eggrobin@unicode.org)
Date	2026-06-12
This Version	https://www.unicode.org/reports/tr14/tr14-56.html
Previous Version	https://www.unicode.org/reports/tr14/tr14-55.html
Latest Version	https://www.unicode.org/reports/tr14/
Latest Proposed Update	https://www.unicode.org/reports/tr14/proposed.html
Revision	56

Summary

This annex presents the Unicode line breaking algorithm along with detailed descriptions of each of the character classes established by the Unicode line breaking property. The line breaking algorithm produces a set of "break opportunities", or positions that would be suitable for wrapping lines when preparing text for display.

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*This is a **draft** document which may be updated, replaced, or superseded by other documents at any time.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 and Scope
2 Definitions
3 Introduction
  3.1 Determining Line Break Opportunities
4 Conformance
  4.1 Conformance Requirements

5 Line Breaking Properties
5.1 Description of Line Breaking Properties
5.2 Dictionary Usage
5.3 Use of Hyphen
5.4 Use of Soft Hyphen
5.5 Use of Double Hyphen
5.6 Tibetan Line Breaking
5.7 Word Separator Characters
6 Line Breaking Algorithm
6.1 Non-tailorable Line Breaking Rules
6.2 Tailorable Line Breaking Rules
7 Deleted. (Formerly was: Pair Table-Based Implementation)
8 Customization
8.1 Types of Tailoring
8.2 Examples of Customization
9 Implementation Notes
9.1 Combining Marks in Regular Expression-Based Implementations
9.2 Legacy Support for Space Character as Base for Combining Marks
10 Testing
11 History
References
Acknowledgments
Modifications

---

1 Overview and Scope

Line breaking, also known as word wrapping, is the process of breaking a section of text into lines such that it will fit in the available width of a page, window or other display area. The Unicode Line Breaking Algorithm performs part of this process. Given an input text, it produces a set of positions called "break opportunities" that are appropriate points to begin a new line. The selection of actual line break positions from the set of break opportunities is not covered by the Unicode Line Breaking Algorithm, but is in the domain of higher level software with knowledge of the available width and the display size of the text.

The text of the Unicode Standard [Unicode] presents a limited description of some of the characters with specific functions in line breaking, but does not give a complete specification of line breaking behavior. This annex provides more detailed information about default line breaking behavior, reflecting best practices for the support of multilingual texts.

For most Unicode characters, considerable variation in line breaking behavior can be expected, including variation based on local or stylistic preferences. For that reason, the line breaking properties provided for these characters are informative. Some characters are intended to explicitly influence line breaking. Their line breaking behavior is therefore expected to be identical across all implementations. As described in this annex, the Unicode Standard assigns normative line breaking properties to those characters. The Unicode Line Breaking Algorithm is a tailorable set of rules that uses these line breaking properties in context to determine line break opportunities.

This annex opens with formal definitions, a summary of the line breaking task and the context in which it occurs in overall text layout, followed by a brief section on conformance requirements. Two main sections follow:

- *Section 5, Line Breaking Properties*, contains a narrative description of the line breaking behavior of the characters in the Unicode Standard, grouping them in alphabetical order by line breaking class.
- *Section 6, Line Breaking Algorithm*, provides a set of rules listed in order of precedence that constitute a line breaking algorithm.

The next sections discuss issues of customization and implementation.

- *Section 8, Customization*, provides a discussion of how to tailor the algorithm.

- *Section 9, Implementation Notes*, provides additional information to implementers using regular expression-based techniques or requiring legacy support for combining marks.
- *Section 10, Testing*, describes the test data file that is available for checking implementations of the line breaking algorithm.
- *Section 11, History*, provides references to additional documentation for investigating changes to the algorithm across Unicode versions.

2 Definitions

The notation defined in this annex differs somewhat from the notation defined elsewhere in the Unicode Standard.

All other notation used here without an explicit definition shall be as defined elsewhere in the Unicode Standard [Unicode].

LD1**. **Line Fitting: The process of determining how much text will fit on a line of text, given the available space between the margins and the actual display width of the text.

LD2**. **Line Break: The position in the text where one line ends and the next one starts.

LD3**. **Line Break Opportunity: A place where a line is allowed to end.

- Whether a given position in the text is a valid line break opportunity depends on the context as well as the line breaking rules in force.

LD4**. **Line Breaking: The process of selecting one among several line break opportunities such that the resulting line is optimal or ends at a user-requested explicit line break.

LD5**. **Line Breaking Property: A character property with enumerated values, as listed in *Table 1*, and separated into normative and informative values.

- Line breaking property values are used to classify characters and, taken in context, determine the type of line break opportunity.

LD6**. **Line Breaking Class: A class of characters with the same line breaking property value.

- The line breaking classes are described in *Section 5.1, Description of Line Breaking Properties*.

LD7**. **Mandatory Break: A line must break following a character that has the mandatory break property.

- Such a break is also known as a *forced* break and is indicated in the rules as **B !**, where **B** is the character with the mandatory break property.

LD8**. **Direct Break: A line break opportunity exists between two adjacent characters of the given line breaking classes.

- A direct break is indicated in the rules below as **B** ÷ **A**, where **B** is the character class of the character *before* and **A** is the character class of the character *after* the break. If they are separated by one or more space characters, a break opportunity exists instead after the last space.

LD9**. **Indirect Break: A line break opportunity exists between two characters of the given line breaking classes *only* if they are separated by one or more spaces.

- For an indirect break, a break opportunity exists after the last space. No break opportunity exists if the characters are immediately adjacent.
- In the notation of the rules in *Section 6, Line Breaking Algorithm*, an indirect break is represented as two rules: **B** × **A** *and* **B** **SP+** ÷ **A** where the "+" sign means one or more occurrences.

LD10**. **Prohibited Break: No line break opportunity exists between two characters of the given line breaking classes, even if they are separated by one or more space characters.

- In the notation of the rules in *Section 6, Line Breaking Algorithm*, a prohibited break is expressed as a rule of the form: **B** SP*** × A**.

LD11**. **Hyphenation: Hyphenation uses language-specific rules to provide additional line break opportunities *within* a word.

- Hyphenation improves the layout of narrow columns, especially for languages with many longer words, such as German or Finnish. For the purpose of this annex, it is assumed that hyphenation is equivalent to inserting *soft hyphen* characters. All other aspects of hyphenation are outside the scope of this annex.

Table 1 lists all of line breaking classes by name, also giving their class abbreviation and their status as tailorable or not. The examples and brief indication of line breaking behavior in this table are merely typical, not exhaustive. *Section 5.1, Description of Line Breaking Properties*, provides a detailed description of each line breaking class, including detailed overview of the line breaking behavior for characters of that class.

Table 1. Line Breaking Classes

Class	Descriptive Name	Examples	Behavior
		Non-tailorable Line Breaking Classes	
BK	*Mandatory Break*	NL, PARAGRAPH SEPARATOR	Cause a line break (after)
CR	*Carriage Return*	CR	Cause a line break (after), except between CR and LF
LF	*Line Feed*	LF	Cause a line break (after)
CM	*Combining Mark*	Combining marks, control codes	Prohibit a line break between the character and the preceding character
NL	*Next Line*	NEL	Cause a line break (after)
SG	*Surrogate*	Surrogates	Do not occur in well-formed text
WJ	*Word Joiner*	WJ	Prohibit line breaks before and after
ZW	*Zero Width Space*	ZWSP	Provide a break opportunity
GL	*Non-breaking ("Glue")*	CGJ, NBSP, ZWNBSP	Prohibit line breaks before and after
SP	*Space*	SPACE	Enable indirect line breaks
ZWJ	*Zero Width Joiner*	Zero Width Joiner	Prohibit line breaks within joiner sequences
		Break Opportunities	
B2	*Break Opportunity Before and After*	Em dash	Provide a line break opportunity before and after the character
BA	*Break After*	Spaces, most sentence-terminal punctuation	Generally provide a line break opportunity after the character
BB	*Break Before*	Punctuation used in dictionaries	Generally provide a line break opportunity before the character
HY	*Hyphen*	HYPHEN-MINUS	Provide a line break opportunity after the character, except in numeric context
HH	*Unambiguous Hyphen*	HYPHEN	Generally provide a line break opportunity after the character, except word-initially.
CB	*Contingent Break Opportunity*	Inline objects	Provide a line break opportunity contingent on additional information

Characters Prohibiting Certain Breaks

CL	*Close Punctuation*	“}”, “)”, “⟫” etc.	Prohibit line breaks before
CP	*Close Parenthesis*	“)”, “]”	Prohibit line breaks before
EX	*Exclamation/ Interrogation*	“!”, “?”, etc.	Prohibit line breaks before
IN	*Inseparable*	Leaders	Allow only indirect line breaks between pairs
NS	*Nonstarter*	“⁉”, “⁈”, “??”, “!?”, etc.	Allow only indirect line breaks before
OP	*Open Punctuation*	“(“, “[“, “{“, etc.	Prohibit line breaks after
QU	*Quotation*	Quotation marks	Act like they are opening, closing, or both

Numeric Context

IS	*Infix Numeric Separator*	. ,	Prevent breaks after any and before numeric
NU	*Numeric*	Digits	Form numeric expressions for line breaking purposes
PO	*Postfix Numeric*	%, ¢	Do not break following a numeric expression
PR	*Prefix Numeric*	$, £, ¥, etc.	Do not break in front of a numeric expression
SY	*Symbols Allowing Break After*	/	Prevent a break before, and allow a break after

Other Characters

AI	*Ambiguous (Alphabetic or Ideographic)*	Characters with Ambiguous East Asian Width	Act like **AL** when the resolved EAW is N; otherwise, act as **ID**
AK	*Aksara*	Consonants	Form orthographic syllables in Brahmic scripts
AL	*Alphabetic*	Alphabets and regular symbols	Are alphabetic characters or symbols that are used with alphabetic characters
AP	*Aksara Pre-Base*	Pre-base repha	Form orthographic syllables in Brahmic scripts
AS	*Aksara Start*	Independent vowels	Form orthographic syllables in Brahmic scripts
CJ	*Conditional Japanese Starter*	Small kana	Treat as **NS** or **ID** for strict or normal breaking.
EB	*Emoji Base*	All emoji allowing modifiers	Do not break from following Emoji Modifier
EM	*Emoji Modifier*	Skin tone modifiers	Do not break from preceding Emoji Base
H2	*Hangul LV Syllable*	Hangul	Form Korean syllable blocks
H3	*Hangul LVT Syllable*	Hangul	Form Korean syllable blocks
HL	*Hebrew Letter*	Hebrew	Special rules around hyphens and SOLIDUS; otherwise act as Alphabetic
ID	*Ideographic*	Ideographs	Break before or after, except in some numeric context
JL	*Hangul L Jamo*	Conjoining jamo	Form Korean syllable blocks
JV	*Hangul V Jamo*	Conjoining jamo	Form Korean syllable blocks
JT	*Hangul T Jamo*	Conjoining jamo	Form Korean syllable blocks
RI	*Regional Indicator*	REGIONAL INDICATOR SYMBOL LETTER A .. Z	Keep pairs together. For pairs, break before and after other classes
SA	*Complex Context Dependent (South East Asian)*	South East Asian: Thai, Lao, Khmer	Provide a line break opportunity contingent on additional, language-specific context analysis
VF	*Virama Final*	Viramas for final consonants	Form orthographic syllables in Brahmic scripts

VI	*Virama*	Conjoining viramas	Form orthographic syllables in Brahmic scripts
XX	*Unknown*	Most unassigned, private-use	Have as yet unknown line breaking behavior or unassigned code positions

3 Introduction

Lines are broken as the result of two conditions. The first is the presence of a mandatory line breaking character. The second condition results from a formatting algorithm having selected among available line break opportunities; ideally the chosen line break results in the optimal layout of the text.

Different formatting algorithms may use different methods to determine an optimal line break. For example, simple implementations consider a single line at a time, trying to find a *locally optimal* line break. A basic, yet widely used approach is to allow no compression or expansion of the intercharacter and interword spaces and consider the longest line that fits. More complex formatting algorithms often take into account the interaction of line breaking decisions for the whole paragraph. The well-known text layout system [T_EX] implements an example of such a *globally optimal* strategy that may make complex tradeoffs across an entire paragraph to avoid unnecessary hyphenation and other legal, but inferior breaks. For a description of this strategy, see [Knuth78].

When compression or expansion is allowed, a locally optimal line break seeks to balance the relative merits of the resulting amounts of compression and expansion for different line break candidates. When expanding or compressing interword space according to common typographical practice, only the spaces marked by U+0020 SPACE and U+00A0 NO-BREAK SPACE are subject to compression, and only spaces marked by U+0020 SPACE, U+00A0 NO-BREAK SPACE, and occasionally spaces marked by U+2009 THIN SPACE are subject to expansion. All other space characters normally have fixed width. When expanding or compressing intercharacter space, the presence of U+200B ZERO WIDTH SPACE or U+2060 WORD JOINER is always ignored.

Local custom or document style determines whether and to what degree expansion of intercharacter space is allowed in justifying a line. In languages, such as German, where intercharacter space is commonly used to mark e m p h a s i s (like this), allowing variable intercharacter spacing would have the unintended effect of adding random emphasis, and is therefore best avoided. In table headings that use Han ideographs, even extreme amounts of intercharacter space commonly occur as short texts are spread out across the entire available space to distribute the characters evenly from end to end.

In line breaking it is necessary to distinguish between three related tasks. The first is the determination of all legal line break opportunities, given a string of text. This is the scope of the Unicode Line Breaking Algorithm. The second task is the selection of the actual location for breaking a given line of text. This selection not only takes into account the width of the line compared to the width of the text, but may also apply an additional prioritization of line breaks based on aesthetic and other criteria. What defines an optimal choice for a given line break is outside the scope of this annex, as are methods for its selection. The third is the possible justification of lines, once actual locations for line breaking have been determined, and is also out of scope for the Unicode Line Breaking Algorithm.

Finally, text layout systems may support an emergency mode that handles the case of an unusual line that contains no otherwise permitted line break opportunities. In such line layout emergencies, line breaks may be placed with no regard to the ordinary line breaking behavior of the characters involved. The details of such an emergency mode are outside the scope of this annex, however, it is recommended that grapheme clusters be kept together.

3.1 Determining Line Break Opportunities

Four principal styles of context analysis determine line break opportunities.

1. *Western:* spaces and hyphens are used to determine breaks
2. *East Asian:* lines can break anywhere, unless prohibited

3. *South East Asian:* line breaks require morphological analysis
4. *Brahmic:* line breaks can occur at the boundaries of any orthographic syllable

The Western style is commonly used for scripts employing the space character. Hyphenation is often used with space-based line breaking to provide additional line break opportunities—however, it requires knowledge of the language and it may need user interaction or overrides.

The second style of context analysis is used with East Asian ideographic and syllabic scripts. In these scripts, lines can break anywhere, except before or after certain characters. The precise set of prohibited line breaks may depend on user preference or local custom and is commonly tailorable.

Korean makes use of both styles of line break. When Korean text is justified, the second style is commonly used, even for interspersed Latin letters. But when ragged margins are used, the Western style (relying on spaces) is commonly used instead, even for ideographs.

The third style is used for scripts such as Thai, which allow line breaks only at word boundaries, but do not mark word boundaries in any way, so that the determination of line break opportunities requires language dependent text analysis. Algorithms and data for such analysis are beyond the scope of the Unicode Standard.

The fourth style is used in some Brahmic scripts, such as Brahmi, Balinese, or Javanese, which allow line breaks to occur at the boundaries of any orthographic syllable, without restricting them to word boundaries. This style is only supported for scripts that encode orthographic syllables in primarily phonetic order.

For multilingual text, the Western, East Asian, and Brahmic styles can be unified into a single set of specifications, based on the information in this annex. Unicode characters have explicit line breaking properties assigned to them. These properties can be utilized to implement the effect of both of these two styles of context analysis for line break opportunities. Customization for user preferences or document style can then be achieved by tailoring that specification.

In bidirectional text, line breaks are determined before applying rule L1 of the Unicode Bidirectional Algorithm [UAX9]. However, line breaking is strictly independent of directional properties of the characters or of any auxiliary information determined by the application of rules of that algorithm.

4 Conformance

There is no single method for determining line breaks; the rules may differ based on user preference and document layout. The information in this annex, including the specification of the line breaking algorithm, allows for the necessary flexibility in determining line breaks according to different conventions. However, some characters have been encoded explicitly for their effect on line breaking. Because users adding such characters to a text expect that they will have the desired effect, these characters have been given required line breaking behavior.

To handle certain situations, some line breaking implementations use techniques that cannot be expressed within the framework of the Unicode Line Breaking Algorithm. Examples include using dictionaries of words for languages that do not use spaces, such as Thai; recognition of the language of the text in order to choose among different punctuation conventions; using dictionaries of common abbreviations or contractions to resolve ambiguities with periods or apostrophes; or a deeper analysis of common syntaxes for numbers or dates, and so on. The conformance requirements permit variations of this kind.

Processes which support multiple modes for determining line breaks are also accommodated. This situation can arise with marked-up text, rich text, style sheets, or other environments in which a higher-level protocol can carry formatting instructions that prevent or force line breaks in positions that differ from those specified by the Unicode Line Breaking Algorithm. The approach taken here requires that such processes have a conforming default line break behavior, and to disclose that they also include overrides or optional behaviors that are invoked via a higher-level protocol.

The methods by which a line layout process chooses optimal line breaks from among the available break opportunities is outside the scope of this specification. The behavior of a line layout process in

situations where there are no suitable break opportunities is also outside of the scope of this specification.

Note: Locale-sensitive line break specifications can be expressed in LDML [UTS35]. Tailorings are available in the Common Locale Data Repository [CLDR].

4.1 Conformance Requirements

***UAX14-C1**. A process that determines line breaks in Unicode text, and that purports to implement the Unicode Line Breaking Algorithm, shall do so in accordance with the specifications in this annex. In particular, the following three subconditions shall be met:*

1. *The sets of mandatory break positions and of break opportunities which the implementation produces include all of those specified by the rules in Section 6.1, Non-tailorable Line Breaking Rules.*
2. *There exist no break opportunities or mandatory breaks produced by the implementation that fall on a "non-break" position specified by the rules in Section 6.1, Non-tailorable Line Breaking Rules.*
3. *If the implementation tailors the behavior of Section 6.2, Tailorable Line Breaking Rules, that fact must be disclosed.*

***UAX14-C2**. If an implementation has a default line breaking operation which conforms to UAX14-C1, but also has overrides based on a higher-level protocol, that fact must be disclosed and any behavior that differs from that specified by the rules of Section 6.1, Non-tailorable Line Breaking Rules, must be documented.*

> Example: An XML format provides markup which disables all line breaking over some span of text. When the markup is not in place, the default behavior is in conformance according to UAX14-C1. As long as the existence of the option is disclosed, that format can be said to conform to the Unicode Line Breaking Algorithm according to UAX14-C2.

As is the case for all other Unicode algorithms, this specification is a logical description—particular implementations can have more efficient mechanisms as long as they produce the same results. See C18 in *Chapter 3, Conformance*, of [Unicode]. While only disclosure of tailorings is required in the conformance clauses, documentation of the differences in behaviors is strongly encouraged.

5 Line Breaking Properties

This section provides detailed narrative descriptions of the line breaking behavior of many Unicode characters. Many descriptions in this section provide additional informative detail about handling a given character at the end of a line, or during line layout, which goes beyond the simple determination of line breaks. In some cases, the text also gives guidance as to preferred characters for achieving a particular effect in line breaking.

This section also summarizes the membership of character classes corresponding to each value of the line breaking property. Note that the mnemonic names for the line break classes are intended neither as exhaustive descriptions of their membership nor as indicators of their entire range of behaviors in the line breaking process. Instead, their main purpose is to serve as unique, yet broadly mnemonic labels. In other words, as long as their line breaking behavior is identical, otherwise unrelated characters will be grouped together in the same line break class.

The classification by property values defined in this section and in the data file is used as input into the algorithm defined in *Section 6, Line Breaking Algorithm*. That section describes a workable default line breaking method. *Section 8, Customization*, discusses how the default line breaking behavior can be tailored to the needs of specific languages or for particular document styles and user preferences. Permitted customizations can include changing the classification of characters for certain classes.

In addition to the line breaking properties defined in this section, the algorithm defined in *Section 6, Line Breaking Algorithm* also makes use of East_Asian_Width property values, defined in Unicode Standard Annex #11, *East Asian Width* [UAX11], as well as the General_Category and

Extended_Pictographic properties. Note that for purposes of the line breaking algorithm, those property values are tailorable, as are the rules of the line breaking algorithm which use them. (See rules **LB15a**, **LB15b**, **LB19**, **LB19a**, **LB21a**, **LB30**, and **LB30b**.)

Data File

The full classification of all Unicode characters by their line breaking properties is available in the file LineBreak.txt [Data14] in the Unicode Character Database [UCD]. This is a semicolon-delimited, two-column, plain text file, with code position and line breaking class. A comment at the end of each line indicates the character name.

The same data, but with a more explicit listing of code point ranges with complex default values, is available in the file DerivedLineBreak.txt [Data14Derived].

The line break property assignments from the data file are normative. The descriptions of the line break classes in this UAX include examples of representative or interesting characters for each class, but for the complete list always refer to the data file.

Future Updates

As scripts are added to the Unicode Standard and become more widely implemented, line breaking classes may be added or the assignment of line breaking class may be changed for some characters. Implementers must not make any assumptions to the contrary. Any future updates will be reflected in the latest version of the data file. (See the Unicode Character Database [UCD] for any specific version of the data file.)

5.1 Description of Line Breaking Properties

Line breaking classes are listed alphabetically. For each line breaking class, the rules that explicitly reference that class are listed in italics above the description of the class. Note that characters in these classes may be involved in other rules; for instance, rule LB31 can apply to characters with almost any line breaking class, but it does not list any line breaking class explicitly.

AI: Ambiguous (Alphabetic or Ideograph)

LB1

Some characters that ordinarily act like alphabetic characters are treated like ideographs (line breaking class **ID**) in certain East Asian legacy contexts. Their line breaking behavior therefore depends on the context. In the absence of appropriate context information, they are treated as class **AL**; see the note at the end of this description.

As originally defined until Unicode Version 3.1.0, the line break class **AI** contained *all* characters with East_Asian_Width value A (ambiguous width) that would otherwise be **AL** in this classification. For more information on East_Asian_Width and how to resolve it, see Unicode Standard Annex #11, *East Asian Width* [UAX11].

The original definition included many Latin, Greek, and Cyrillic characters. Since Unicode Version 4.0.1, these characters are classified by default as **AL** because use of the **AL** line breaking class better corresponds to modern practice. Where strict compatibility with older legacy implementations is desired, some of these characters need to be treated as **ID** in certain contexts. This can be done by always tailoring them to **ID** or by continuing to classify them as **AI** and resolving them to **ID** where required.

As part of the same revision, the set of ambiguous characters has been extended to completely encompass the enclosed alphanumeric characters used for numbering of bullets.

In Unicode Version 4.0.1, the **AI** line breaking class therefore included all characters with East Asian Width A that are outside the range U+0000..U+1FFF, plus the following characters:

24EA	CIRCLED DIGIT ZERO

2780..2793	DINGBAT CIRCLED SANS-SERIF DIGIT ONE..DINGBAT NEGATIVE CIRCLED SANS-SERIF NUMBER TEN

Since that time, the East_Asian_Width and Line_Break properties have been maintained independently, with the latter being based on the need for language-specific line-breaking behavior rather than compatibility with legacy encodings. In particular, all vulgar fractions have Line_Break=AI.

Characters with the line break class **AI** with East_Asian_Width value A typically take the **AL** line breaking class when their resolved East_Asian_Width is N (narrow) and take the line breaking class **ID** when their resolved width is W (wide). The remaining characters are then resolved to **AL** or **ID** in a consistent fashion. The details of this resolution are not specified in this annex. The line breaking rules in *Section 6, Line Breaking Algorithm* merely require that all ambiguous characters be resolved appropriately as part of assigning line breaking classes to the input characters.

> *Note:* The canonical decompositions of characters of class **AI** are not necessarily of class **AI** themselves. The East_Asian_Width property A on which the definition of **AI** is largely based, does not preserve canonical equivalence. In the context of line breaking, the fact that a character has been assigned class **AI** means that the line break implementation must resolve it to either **AL** or **ID**, in the absence of further tailoring. If preserving canonical equivalence is desired, an implementation is free to make sure that the *resolved* line break classes preserve canonical equivalence. Unless compatibility with particular legacy behavior is important, it may be sufficient to map all such characters to **AL**. This achieves a canonically equivalent resolution of line breaking classes, and is compatible with emerging modern practice that treats these characters increasingly like regular alphabetic characters.

AK: Aksara

LB28a

The **AK** line break class is used for scripts that use the Brahmic style of context analysis and have a virama of Indic syllabic category Virama or Invisible_Stacker. It contains characters that can occur as the bases of orthographic syllables and can also follow a virama of Indic syllabic category Virama or Invisible_Stacker within the same orthographic syllable. Depending on the script, this may include characters with the Indic syllabic categories Consonant, Vowel_Independent, or Number.

1B05..1B33	BALINESE LETTER AKARA..BALINESE LETTER HA
1B45..1B4C	BALINESE LETTER KAF SASAK..BALINESE LETTER ARCHAIC JNYA
A984..A9B2	JAVANESE LETTER A..JAVANESE LETTER HA
11005..11037	BRAHMI LETTER A..BRAHMI LETTER OLD TAMIL NNNA
11071..11072	BRAHMI LETTER OLD TAMIL SHORT E..BRAHMI LETTER OLD TAMIL SHORT O
11075	BRAHMI LETTER OLD TAMIL LLA
11305..1130C	GRANTHA LETTER A..GRANTHA LETTER VOCALIC L
1130F..11310	GRANTHA LETTER EE..GRANTHA LETTER AI
11313..11328	GRANTHA LETTER OO..GRANTHA LETTER NA
1132A..11330	GRANTHA LETTER PA..GRANTHA LETTER RA
11332..11333	GRANTHA LETTER LA..GRANTHA LETTER LLA
11335..11339	GRANTHA LETTER VA..GRANTHA LETTER HA
11360..11361	GRANTHA LETTER VOCALIC RR..GRANTHA LETTER VOCALIC LL
11F04..11F10	KAWI LETTER A..KAWI LETTER O
11F12..11F33	KAWI LETTER KA..KAWI LETTER JNYA

AL: Ordinary Alphabetic and Symbol Characters

LB1, LB10, LB20a, LB23, LB24, LB28, LB29, LB30

Ordinary characters require other characters to provide break opportunities; otherwise, no line breaks are allowed between pairs of them. However, this behavior is tailorable. In some Far Eastern documents, it may be desirable to allow breaking between pairs of ordinary characters—particularly Latin characters and symbols.

Note: Use ZWSP as a manual override to provide break opportunities around alphabetic or symbol characters.

This class contains alphabetic or symbolic characters not explicitly assigned to another line breaking class. These are primarily characters of the following categories:

Category	General_Category Values
Alphabetic	Lu, Ll, Lt, Lm, and Lo
Symbols	Sm, Sk, and So
Non-decimal Numbers	Nl and No
Punctuation	Pc, Pd, and Po

Line break class **AL** also contains several format characters, including:

0600..0604	ARABIC NUMBER SIGN..ARABIC SIGN SAMVAT
06DD	ARABIC END OF AYAH
070F	SYRIAC ABBREVIATION MARK
2061..2064	FUNCTION APPLICATION..INVISIBLE PLUS
110BD	KAITHI NUMBER SIGN

These format characters occur in the middle or at the beginning of words or alphanumeric or symbol sequences. However, when alphabetic characters are tailored to allow breaks, these characters should not allow breaks after.

Major exceptions to the general pattern of alphabetic and symbolic characters having line break class **AL** include:

HL for Hebrew letters
AI or ID, based on the East Asian Width property of the character
ID for certain pictographic symbols
CJ for small hiragana and katakana
SA for complex context scripts
JL, JV, JT, H2 or H3 for Hangul characters

AP: Aksara Pre-Base

LB28a

The **AP** line break class is only used for scripts that use the Brahmic style of context analysis. It contains the characters of such scripts that are part of an orthographic syllable but in logical order precede the base or any half-forms. This includes characters with the Indic syllabic categories Consonant_Preceding_Repha, Consonant_With_Stacker, and Consonant_Prefixed.

11003..11004	BRAHMI SIGN JIHVAMULIYA..BRAHMI SIGN UPADHMANIYA
11F02	KAWI SIGN REPHA

AS: Aksara Start

LB28a

The **AS** line break class is only used for scripts that use the Brahmic style of context analysis. It contains characters that can occur as the bases of orthographic syllables, but cannot follow a virama of Indic syllabic category Virama or Invisible_Stacker within the same orthographic syllable. Depending on the script, this may include characters with the Indic syllabic categories Consonant, Vowel_Independent,, and several others. This class also contains all digits of scripts that use the Brahmic style of line breaking; in some of these scripts, such as Brahmi or Kawi, digits can occur as bases of orthographic syllables.

1B50..1B59	BALINESE DIGIT ZERO..BALINESE DIGIT NINE
1BC0..1BE5	BATAK LETTER A..BATAK LETTER U
A9D0..A9D9	JAVANESE DIGIT ZERO..JAVANESE DIGIT NINE
AA00..AA28	CHAM LETTER A..CHAM LETTER HA
AA50..AA59	CHAM DIGIT ZERO..CHAM DIGIT NINE
11066..1106F	BRAHMI DIGIT ZERO..BRAHMI DIGIT NINE
11350	GRANTHA OM
1135E..1135F	GRANTHA LETTER VEDIC ANUSVARA..GRANTHA LETTER VEDIC DOUBLE ANUSVARA
11950..11959	DIVES AKURU DIGIT ZERO..DIVES AKURU DIGIT NINE
11EE0..11EF1	MAKASAR LETTER KA..MAKASAR LETTER A
11F50..11F59	KAWI DIGIT ZERO..KAWI DIGIT NINE

BA: Break After

LB12a, LB21

Like SPACE, the characters in this class provide a break opportunity; unlike SPACE, they do not take part in determining indirect breaks. They can be subdivided into several categories.

Breaking Spaces

Breaking spaces are a subset of characters with General_Category Zs. Examples include:

1680	OGHAM SPACE MARK
2000	EN QUAD
2001	EM QUAD
2002	EN SPACE
2003	EM SPACE
2004	THREE-PER-EM SPACE
2005	FOUR-PER-EM SPACE
2006	SIX-PER-EM SPACE
2008	PUNCTUATION SPACE
2009	THIN SPACE
200A	HAIR SPACE
2012	FIGURE DASH
2013	EN DASH
205F	MEDIUM MATHEMATICAL SPACE
3000	IDEOGRAPHIC SPACE

All of these space characters have a specific width, but otherwise behave as breaking spaces. In setting a justified line, none of these spaces normally changes in width, except for THIN SPACE when used in mathematical notation. See also the **SP** property.

The OGHAM SPACE MARK may be rendered visibly between words but it is recommended that it be elided at the end of a line. For more information, see *Section 5.7, Word Separator Characters*.

For a list of all space characters in the Unicode Standard, see *Section 6.2, General Punctuation*, in [Unicode].

Tabs

0009	TAB

Except for the effect of the location of the tab stops, the tab character acts similarly to a space for the purpose of line breaking.

~~***Conditional Hyphens***~~

~~00AD~~	~~SOFT HYPHEN (SHY)~~

~~SHY is an invisible format character with no width. It marks the place where an optional line break may occur inside a word. It can be used with all scripts. If a line is broken at an optional line break position marked by a SHY, the text at that line break position often has a modified appearance as described in *Section 5.4, Use of Soft Hyphen*.~~

Visible Word Dividers

The following are examples of other forms of visible word dividers that provide break opportunities:

0F0B	TIBETAN MARK INTERSYLLABIC TSHEG
1361	ETHIOPIC WORDSPACE
17D8	KHMER SIGN BEYYAL
17DA	KHMER SIGN KOOMUUT

The Tibetan *tsheg* is a visible mark, but it functions effectively like a space to separate words (or other units) in Tibetan. It provides a break opportunity after itself. For additional information, see *Section 5.6, Tibetan Line Breaking*.

The ETHIOPIC WORDSPACE is a visible word delimiter and is kept on the previous line. In contrast, U+1360 ETHIOPIC SECTION MARK is typically used in a sequence of several such marks on a separate line, and separated by spaces. As such lines are typically marked with separate hard line breaks (**BK**), the section mark is treated like an ordinary symbol and given line break class **AL**.

2027	HYPHENATION POINT

A hyphenation point is a raised dot, which is mainly used in dictionaries and similar works to visibly indicate syllabification of words. Syllable breaks frequently also are potential line break opportunities in the middle of words. When an actual line break falls inside a word containing hyphenation point characters, the hyphenation point is usually rendered as a regular hyphen at the end of the line.

007C	VERTICAL LINE

In some dictionaries, a vertical bar is used instead of a hyphenation point. In this usage, U+0323 COMBINING DOT BELOW is used to mark stressed syllables, so all breaks are marked by the vertical bar. For an actual line break the vertical bar is rendered as a hyphen at the end of the line.

Historic Word Separators

Historic texts, especially ancient ones, often do not use spaces, even for scripts where modern use of spaces is standard. Special punctuation was used to mark word boundaries in such texts. For modern text processing it is recommended to treat these as line break opportunities by default. **WJ** can be used to override this default, where necessary.

Examples of Historic Word Separators include:

16EB	RUNIC SINGLE PUNCTUATION
16EC	RUNIC MULTIPLE PUNCTUATION
16ED	RUNIC CROSS PUNCTUATION
2056	THREE DOT PUNCTUATION
2058	FOUR DOT PUNCTUATION
2059	FIVE DOT PUNCTUATION
205A	TWO DOT PUNCTUATION
205B	FOUR DOT MARK
205D	TRICOLON

205E	VERTICAL FOUR DOTS
2E19	PALM BRANCH
2E2A	TWO DOTS OVER ONE DOT PUNCTUATION
2E2B	ONE DOT OVER TWO DOTS PUNCTUATION
2E2C	SQUARED FOUR DOT PUNCTUATION
2E2D	FIVE DOT MARK
2E30	RING POINT
10100	AEGEAN WORD SEPARATOR LINE
10101	AEGEAN WORD SEPARATOR DOT
10102	AEGEAN CHECK MARK
1039F	UGARITIC WORD DIVIDER
103D0	OLD PERSIAN WORD DIVIDER
1091F	PHOENICIAN WORD SEPARATOR
12470	CUNEIFORM PUNCTUATION SIGN OLD ASSYRIAN WORD DIVIDER

Dandas

DEVANAGARI DANDA is similar to a full stop. The *danda* or historically related symbols are used with several other Indic scripts. Unlike a full stop, the *danda* is not used in number formatting. DEVANAGARI DOUBLE DANDA marks the end of a verse. It also has analogues in other scripts.

Examples of dandas include:

0964	DEVANAGARI DANDA
0965	DEVANAGARI DOUBLE DANDA
0E5A	THAI CHARACTER ANGKHANKHU
0E5B	THAI CHARACTER KHOMUT
104A	MYANMAR SIGN LITTLE SECTION
104B	MYANMAR SIGN SECTION
1735	PHILIPPINE SINGLE PUNCTUATION
1736	PHILIPPINE DOUBLE PUNCTUATION
17D4	KHMER SIGN KHAN
17D5	KHMER SIGN BARIYOOSAN
1B5E	BALINESE CARIK SIKI
1B5F	BALINESE CARIK PAREREN
A8CE	SAURASHTRA DANDA
A8CF	SAURASHTRA DOUBLE DANDA
AA5D	CHAM PUNCTUATION DANDA
AA5E	CHAM PUNCTUATION DOUBLE DANDA
AA5F	CHAM PUNCTUATION TRIPLE DANDA
10A56	KHAROSHTHI PUNCTUATION DANDA
10A57	KHAROSHTHI PUNCTUATION DOUBLE DANDA

Tibetan

0F34	TIBETAN MARK BSDUS RTAGS
0F7F	TIBETAN SIGN RNAM BCAD
0F85	TIBETAN MARK PALUTA
0FBE	TIBETAN KU RU KHA
0FBF	TIBETAN KU RU KHA BZHI MIG CAN
0FD2	TIBETAN MARK NYIS TSHEG

For additional information, see *Section 5.6, Tibetan Line Breaking*.

Other Terminating Punctuation

Termination punctuation stays with the line, but otherwise allows a break after it. This is similar to **EX**, except that the latter may be separated by a space from the preceding word without allowing a break, whereas these marks are used without spaces. Terminating punctuation includes:

1804	MONGOLIAN COLON
1805	MONGOLIAN FOUR DOTS
1B5A	BALINESE PANTI
1B5B	BALINESE PAMADA
1B5D	BALINESE CARIK PAMUNGKAH
1B60	BALINESE PAMENENG
1C3B	LEPCHA PUNCTUATION TA-ROL
1C3C	LEPCHA PUNCTUATION NYET THYOOM TA-ROL
1C3D	LEPCHA PUNCTUATION CER-WA
1C3E	LEPCHA PUNCTUATION TSHOOK CER-WA
1C3F	LEPCHA PUNCTUATION TSHOOK
1C7E	OL CHIKI PUNCTUATION MUCAAD
1C7F	OL CHIKI PUNCTUATION DOUBLE MUCAAD
2CFA	COPTIC OLD NUBIAN DIRECT QUESTION MARK
2CFB	COPTIC OLD NUBIAN INDIRECT QUESTION MARK
2CFC	COPTIC OLD NUBIAN VERSE DIVIDER
2CFF	COPTIC MORPHOLOGICAL DIVIDER
2E0E..2E15	EDITORIAL CORONIS..UPWARDS ANCORA
A60D	VAI COMMA
A60F	VAI QUESTION MARK
A92E	KAYAH LI SIGN CWI
A92F	KAYAH LI SIGN SHYA
10A50	KHAROSHTHI PUNCTUATION DOT
10A51	KHAROSHTHI PUNCTUATION SMALL CIRCLE
10A52	KHAROSHTHI PUNCTUATION CIRCLE
10A53	KHAROSHTHI PUNCTUATION CRESCENT BAR
10A54	KHAROSHTHI PUNCTUATION MANGALAM
10A55	KHAROSHTHI PUNCTUATION LOTUS
11EF7..11EF8	MAKASAR PASSIMBANG..MAKASAR END OF SECTION

Letters Attached to Orthographic Syllables

In scripts that use the Brahmic style of line breaking, most characters that attach to the initial consonant cluster of an orthographic syllable and are part of that syllable are encoded as combining marks. These have line break class **CM**. Sometimes, however, additional characters with general category Lo or Lm, such as final consonants or vowel lengtheners, should remain attached to the preceding orthographic syllable. They are then assigned line break class **BA**.

A9CF	JAVANESE PANGRANGKEP
AA40..AA42	CHAM LETTER FINAL K..CHAM LETTER FINAL NG
AA44..AA4B	CHAM LETTER FINAL CH..CHAM LETTER FINAL SS
1133D	GRANTHA SIGN AVAGRAHA
1135D	GRANTHA SIGN PLUTA
11EF2	MAKASAR ANGKA

BB: Break Before

LB21

Characters of this line break class move to the next line at a line break and thus provide a line break opportunity before.

Examples of **BB** characters are described in the following sections.

Dictionary Use

00B4	ACUTE ACCENT
1FFD	GREEK OXIA

In some dictionaries, stressed syllables are indicated with a spacing acute accent instead of the hyphenation point. In this case the accent moves to the next line, and the preceding line ends with a hyphen. The oxia is canonically equivalent to the acute accent.

02DF	MODIFIER LETTER CROSS ACCENT

A cross accent also appears in some dictionaries to mark the stress of the following syllable, and should be handled in the same way as the other stress marking characters in this section. The accent should not be separated from the syllable it marks by a break.

02C8	MODIFIER LETTER VERTICAL LINE
02CC	MODIFIER LETTER LOW VERTICAL LINE

These characters are used in dictionaries to indicate stress and secondary stress when IPA is used. Both are prefixes to the stressed syllable in IPA. Breaking before them keeps them with the syllable.

> *Note:* It is hard to find actual examples in most dictionaries because the pronunciation fields usually occur right after the headword, and the columns are wide enough to prevent line breaks in most pronunciations.

Tibetan and Phags-Pa Head Letters

0F01	TIBETAN MARK GTER YIG MGO TRUNCATED A
0F02	TIBETAN MARK GTER YIG MGO -UM RNAM BCAD MA
0F03	TIBETAN MARK GTER YIG MGO -UM GTER TSHEG MA
0F04	TIBETAN MARK INITIAL YIG MGO MDUN MA
0F06	TIBETAN MARK CARET YIG MGO PHUR SHAD MA
0F07	TIBETAN MARK YIG MGO TSHEG SHAD MA
0F09	TIBETAN MARK BSKUR YIG MGO
0F0A	TIBETAN MARK BKA- SHOG YIG MGO
0FD0	TIBETAN MARK BSKA- SHOG GI MGO RGYAN
0FD1	TIBETAN MARK MNYAM YIG GI MGO RGYAN
0FD3	TIBETAN MARK INITIAL BRDA RNYING YIG MGO MDUN MA
A874	PHAGS-PA SINGLE HEAD MARK
A875	PHAGS-PA DOUBLE HEAD MARK

Tibetan head letters allow a break before. For more information, see *Section 5.6, Tibetan Line Breaking*.

Mongolian

1806	MONGOLIAN TODO SOFT HYPHEN

Despite its name, this Mongolian character is not an invisible control like SOFT HYPHEN, but rather a visible character like a regular hyphen. Unlike the hyphen, MONGOLIAN TODO SOFT HYPHEN stays with the following line. Whenever optional line breaks are to be marked invisibly, SOFT HYPHEN should be used instead.

B2: Break Opportunity Before and After

LB17

2014	EM DASH

The EM DASH is used to set off parenthetical text. Normally, it is used without spaces. However, this is language dependent. For example, in Swedish, spaces are used around the EM DASH. Line breaks can occur before and after an EM DASH. Because EM DASHes are sometimes used in pairs instead of a single quotation dash, the default behavior is not to break the line between even though not all fonts use connecting glyphs for the EM DASH.

Some languages, including Spanish, use EM DASH to set off a parenthetical, and the surrounding dashes should not be broken from the contained text. In this usage there is space on the side where it can be broken. This does not conflict with symmetrical usages, either with spaces on both sides of the em-dash or with no spaces.

BK: Mandatory Break (Non-tailorable)

LB4, LB6, LB9, LB15a, LB15b, LB20a

Explicit breaks act independently of the surrounding characters. No characters can be added to the **BK** class as part of tailoring, but implementations are not required to support the VT character.

000B	LINE TABULATION (VT)
000C	FORM FEED (FF)

FORM FEED separates pages. The text on the new page starts at the beginning of the line. In some layout modes there may be no visible advance to a new “page”.

2028	LINE SEPARATOR

The text after the LINE SEPARATOR starts at the beginning of the line. This is similar to HTML <BR>.

2029	PARAGRAPH SEPARATOR

The text of the new paragraph starts at the beginning of the line. This character defines a paragraph break, causing suitable formatting to be applied, for example, interparagraph spacing or first line indentation. LINE SEPARATOR, FF, VT as well as **CR**, **LF** and **NL** do not define a paragraph break.

Newline Function (NLF)

Newline Functions are defined in the Unicode Standard as providing additional mandatory breaks. They are not individual characters, but are encoded as sequences of the control characters NEL, LF, and CR. If a character sequence for a Newline Function contains more than one character, it is kept together. The particular sequences that form an NLF depend on the implementation and other circumstances as described in *Section 5.8, Newline Guidelines*, of [Unicode].

This specification defines the NLF implicitly. It defines the three character classes **CR**, **LF**, and **NL**. Their line break behavior, defined in rule **LB5** in *Section 6.1, Non-tailorable Line Breaking Rules*, is to break after **NL**, **LF**, or **CR**, but not between **CR** and **LF**.

CB: Contingent Break Opportunity

LB1, LB20, LB20a

By default, there is a break opportunity both *before* and *after* any inline object. Object-specific line breaking behavior is implemented in the associated object itself, and where available can override the default to prevent either or both of the default break opportunities. Using U+FFFC OBJECT REPLACEMENT CHARACTER allows the object anchor to take a character position in the string.

FFFC	OBJECT REPLACEMENT CHARACTER

Object-specific line break behavior is best implemented by querying the object itself, not by replacing the **CB** line breaking class by another class.

CJ: Conditional Japanese Starter

LB1

This character class contains Japanese small hiragana and katakana. Characters of this class may be treated as either NS or ID.

CSS Text Level 3 (which supports Japanese line layout) defines three distinct values for its line-break behavior:

- strict, typically used for long lines
- normal, the behavior typically used for books and documents
- loose, typically used for short lines such as in newspapers

These have different sets of “kinsoku” characters which cannot be at the beginning or end of a line; strict has the largest set, while loose has the smallest. The motivation for the smaller number of kinsoku characters is to avoid triggering justification that puts characters off the grid position.

Treating characters of class CJ as class NS will give CSS strict line breaking; treating them as class ID will give CSS normal breaking.

The CJ line break class includes

3041, 3043, 3045, etc.	Small hiragana
30A1, 30A3, 30A5, etc.	Small katakana
30FC	KATAKANA-HIRAGANA PROLONGED SOUND MARK
FF67..FF70	Halfwidth variants

CL: Close Punctuation

LB13, LB15b, LB16, LB25

The closing character of any set of paired punctuation should be kept with the preceding character, and the same applies to all forms of wide comma and full stop. This is desirable, even when there are intervening space characters, to prevent the appearance of a bare closing punctuation mark at the head of a line.

The class CL is closely related to the class CP (Close Parenthesis). They differ only in that CP will not introduce a break when followed by a letter or number, which prevents breaks within constructs like “(s)he”.

The CL line break class contains characters of General_Category Pe in the Unicode Character Database, but excludes any characters included in the class CP. It also contains certain non-paired punctuation characters, including:

3001..3002	IDEOGRAPHIC COMMA..IDEOGRAPHIC FULL STOP
FE10	PRESENTATION FORM FOR VERTICAL COMMA
FE11	PRESENTATION FORM FOR VERTICAL IDEOGRAPHIC COMMA
FE12	PRESENTATION FORM FOR VERTICAL IDEOGRAPHIC FULL STOP
FE50	SMALL COMMA
FE52	SMALL FULL STOP
FF0C	FULLWIDTH COMMA
FF0E	FULLWIDTH FULL STOP
FF61	HALFWIDTH IDEOGRAPHIC FULL STOP
FF64	HALFWIDTH IDEOGRAPHIC COMMA

CM: Combining Mark (Non-tailorable)

LB1, *LB9*, *LB10*

Combining Characters

Combining character sequences are treated as units for the purpose of line breaking. The line breaking behavior of the sequence is that of the base character.

The preferred base character for showing combining marks in isolation is U+00A0 NO-BREAK SPACE. If a line break before or after the combining sequence is desired, U+200B ZERO WIDTH SPACE can be used. The use of U+0020 SPACE as a base character is deprecated.

For most purposes, combining characters take on the properties of their base characters, and that is how the **CM** class is treated in rule **LB9** of this specification. As a result, if the sequence <0021, 20E4> is used to represent a triangle enclosing an exclamation point, it is effectively treated as **EX**, the line break class of the exclamation mark. If U+26A0 WARNING SIGN had been used, which also looks like an exclamation point inside a triangle, it would have the line break class of **AL**. Only the latter corresponds to the line breaking behavior expected by users for this symbol. To avoid surprising behavior, always use a base character that is a symbol or letter (Line Break **AL**) when using enclosing combining marks (General_Category Me).

The **CM** line break class includes all combining characters with General_Category Mc, Me, and Mn, unless listed explicitly elsewhere. This includes *viramas* that don't have line break class **VI** or **VF**.

In particular, line breaking class **CM** includes the character U+034F COMBINING GRAPHEME JOINER. This character is used for specialized collation or display; see Unicode Technical Standard #10, "Unicode Collation Algorithm" [UTS10], and Unicode Standard Annex #53, "Unicode Arabic Mark Rendering" [UAX53]. It functions as an invisible combining mark; it should be ignored outside of the few processes that ascribe meaning to it. Assigning it class **CM** means the line breaking algorithm ignores it.

Control and Formatting Characters

Most control and formatting characters are ignored in line breaking and do not contribute to the line width. By giving them class **CM**, the line breaking behavior of the last preceding character that is not of class **CM** affects the line breaking behavior.

> *Note:* When control codes and format characters are rendered visibly during editing, more graceful layout might be achieved by treating them as if they had the line break class of the visible symbols instead, that is **AL** or **ID**. Such visible modes do not violate the constraint on tailorability, because they are logically equivalent to having temporarily substituted symbol *characters*, such as the characters from the Control Pictures block, or in some cases, character sequences, for the actual control characters.

The **CM** line break class includes all characters of General_Category Cc and Cf, unless listed explicitly elsewhere.

The **CM** class also includes U+3035 VERTICAL KANA REPEAT MARK LOWER HALF. This character is normally preceded by either U+3033 VERTICAL KANA REPEAT MARK UPPER HALF or U+3034 VERTICAL KANA REPEAT WITH VOICED SOUND MARK UPPER HALF, and should not be separated from them.

CP: Closing Parenthesis

LB13, *LB15b*, *LB16*, *LB25*, *LB30*

This class contains two common characters, U+0029 RIGHT PARENTHESIS and U+005D RIGHT SQUARE BRACKET. It also contains closing brackets used in phonetic notations. Characters of class **CP** differ from those of the **CL** (Close Punctuation) class in that they will not cause a break opportunity when appearing in contexts like "(s)he." In all other respects the breaking behavior of **CP** and **CL** are the same.

0029	RIGHT PARENTHESIS
005D	RIGHT SQUARE BRACKET
2E56	RIGHT SQUARE BRACKET WITH STROKE
2E58	RIGHT SQUARE BRACKET WITH DOUBLE STROKE
2E5A	TOP HALF RIGHT PARENTHESIS
2E5C	BOTTOM HALF RIGHT PARENTHESIS

CR: Carriage Return (Non-tailorable)

LB5, LB6, LB9, LB15a, LB15b, LB20a

000D	CARRIAGE RETURN (CR)

A **CR** indicates a mandatory break after, unless followed by a **LF**. See also the discussion under **BK**.

> *Note:* On some platforms the character sequence <CR, CR, LF> is used to indicate the location of actual line breaks, whereas <CR, LF> is treated like a hard line break. As soon as a user edits the text, the location of all the <CR, CR, LF> sequences may change as the new text breaks differently, while the relative position of any <CR, LF> to the surrounding text stays the same. This convention allows an editor to return a buffer and the client to tell which text is displayed on which line by counting the number of <CR, CR, LF> and <CR, LF> sequences. This convention is essentially equivalent to markup that captures the result of applying the line break algorithm, not a tailoring of the CR character. The <CR, CR, LF> sequences are thus not considered part of the plain text content.

EB: Emoji Base

LB23a, LB30b

This class includes characters whose appearance can be modified by a subsequent emoji modifier in an emoji modifier sequence. This class directly corresponds to the Emoji_Modifier_Base property as defined in *Section 1.4.4 Emoji Modifiers* of [UTS51].

Examples include:

1F466	BOY
1F478	PRINCESS
1F6B4	BICYCLIST

Breaks within emoji modifier sequences are prevented by rule **LB30b**. In other contexts, characters of class EB behave similarly to ideographs of class **ID**, with break opportunities before and after.

EM: Emoji Modifier

LB23a, LB30b

This class includes characters that can be used to modify the appearance of a preceding emoji in an emoji modifier sequence. This class directly corresponds to the Emoji_Modifier property as defined in *Section 1.4.4 Emoji Modifiers* of [UTS51].

Breaks within emoji modifier sequences are prevented by rule **LB30b**.

Emoji modifiers include:

1F3FB..1F3FF	EMOJI MODIFIER FITZPATRICK TYPE-1-2..EMOJI MODIFIER FITZPATRICK TYPE-6

EX: Exclamation/Interrogation

LB13, LB15b

Characters in this line break class behave like closing characters, except in relation to postfix (**PO**) and non-starter characters (**NS**). Examples include:

0021	EXCLAMATION MARK
003F	QUESTION MARK
05C6	HEBREW PUNCTUATION NUN HAFUKHA
061B	ARABIC SEMICOLON
061E	ARABIC TRIPLE DOT PUNCTUATION MARK
061F	ARABIC QUESTION MARK
06D4	ARABIC FULL STOP
07F9	NKO EXCLAMATION MARK
0F0D	TIBETAN MARK SHAD
FF01	FULLWIDTH EXCLAMATION MARK
FF1F	FULLWIDTH QUESTION MARK

GL: Non-breaking (“Glue”) (Non-tailorable)

LB12, LB12a, LB15a, LB15b, LB20a

Non-breaking characters prohibit breaks on either side, but that prohibition can be overridden by **SP** or **ZW**. In particular, when NO-BREAK SPACE follows SPACE, there is a break opportunity after the SPACE and the NO-BREAK SPACE will go as visible space onto the next line. See also **WJ**. The following are examples of characters of line break class **GL**:

00A0	NO-BREAK SPACE (NBSP)
202F	NARROW NO-BREAK SPACE (NNBSP)
180E	MONGOLIAN VOWEL SEPARATOR (MVS)

NO-BREAK SPACE is the preferred character to use where two words are to be visually separated but kept on the same line, as in the case of a title and a name “Dr.<NBSP>Joseph Becker”. When SPACE follows NO-BREAK SPACE, there is no break, because there never is a break in front of SPACE.

NARROW NO-BREAK SPACE has exactly the same line breaking behavior as NO-BREAK SPACE, but with a narrow display width. The MONGOLIAN VOWEL SEPARATOR acts like a NARROW NO-BREAK SPACE in its line breaking behavior. Both of these characters are regularly used in Mongolian text, where they participate in special shaping behavior, as described in *Section 13.5, Mongolian* of [Unicode].

When NARROW NO-BREAK SPACE occurs in French text, it should be interpreted as an “espace fine insécable”.

1107F	BRAHMI NUMBER JOINER
13430..13436	EGYPTIAN HIEROGLYPH VERTICAL JOINER..EGYPTIAN HIEROGLYPH OVERLAY MIDDLE
13439..1343B	EGYPTIAN HIEROGLYPH INSERT AT MIDDLE..EGYPTIAN HIEROGLYPH INSERT AT BOTTOM
16FE4	KHITAN SMALL SCRIPT FILLER

These characters participate in shaping behavior. Together with the characters on either side, they form a ligature, quadrat, or cluster, within which there can be no line break. See *Section 14.1, Brahmi*, *Section 11.4, Egyptian Hieroglyphs*, and *Section 18.12, Khitan Small Script*, respectively, of [Unicode].

2007	FIGURE SPACE

This is the preferred space to use in numbers. It has the same width as a digit and keeps the number together for the purpose of line breaking.

2011	NON-BREAKING HYPHEN

This is the preferred character to use where words need to be hyphenated but may not be broken at the hyphen. Because of its use as a substitute for ordinary hyphen, the appearance of this character should match that of U+2010 HYPHEN.

0F08	TIBETAN MARK SBRUL SHAD
0F0C	TIBETAN MARK DELIMITER TSHEG BSTAR
0F12	TIBETAN MARK RGYA GRAM SHAD

The TSHEG BSTAR looks exactly like a Tibetan *tsheg*, but can be used to prevent a break like *no-break space*. It inhibits breaking on either side. For more information, see *Section 5.6, Tibetan Line Breaking*.

035C..0362	COMBINING DOUBLE BREVE BELOW..COMBINING DOUBLE RIGHTWARDS ARROW BELOW

These diacritics span two characters, so no word or line breaks are possible on either side.

FE20	COMBINING LIGATURE LEFT HALF
FE22	COMBINING DOUBLE TILDE LEFT HALF
FE24	COMBINING MACRON LEFT HALF
FE27	COMBINING LIGATURE LEFT HALF BELOW
FE29	COMBINING TILDE LEFT HALF BELOW
FE2B	COMBINING MACRON LEFT HALF BELOW
FE2E	COMBINING CYRILLIC TITLO LEFT HALF
FE26	COMBINING CONJOINING MACRON
FE2D	COMBINING CONJOINING MACRON BELOW

The left half diacritics are part of a legacy representation of the double diacritics; they occur between the two characters spanned by the double diacritic. Preventing breaks on either side therefore achieves the same line breaking behavior as when using the preferred representation U+035C..U+0362.

In addition, the conjoining macrons above and below, together with left and right half marks, form marks spanning more than two characters; likewise no line break occurs within such spans.

H2: Hangul LV Syllable

LB26, LB27

This class includes all characters of Hangul Syllable Type LV.

Together with conjoining jamos, Hangul syllables form Korean Syllable Blocks, which are kept together; see Unicode Standard Annex #29, “Unicode Text Segmentation” [UAX29]. Korean uses space-based line breaking in many styles of documents. To support these, Hangul syllables and conjoining jamos need to be tailored to use class **AL**. The default in this specification is class **ID**, which supports the case of Korean documents not using space-based line breaking. See *Section 8.1, Types of Tailoring*. See also **JL**, **JT**, **JV**, and **H3**.

H3: Hangul LVT Syllable

LB26, LB27

This class includes all characters of Hangul Syllable Type LVT. See also **JL**, **JT**, **JV**, and **H2**.

HH: Unambiguous Hyphen

LB12a, LB20a, LB21, LB21a

This class consists of breaking hyphens. These characters establish explicit break opportunities immediately after each occurrence, unless they occur word-initally, as when referring to a suffix such

as *-ing*. The hyphens become non-breaking between Hebrew and non-Hebrew.

058A	ARMENIAN HYPHEN
05BE	HEBREW PUNCTUATION MAQAF
1400	CANADIAN SYLLABICS HYPHEN
2010	HYPHEN
~~2012~~	~~FIGURE DASH~~
~~2013~~	~~EN DASH~~
2E17	DOUBLE OBLIQUE HYPHEN
2E40	DOUBLE HYPHEN
2E5D	OBLIQUE HYPHEN
10D6E	GARAY HYPHEN
10EAD	YEZIDI HYPHENATION MARK

Hyphens are graphic characters with width. Because, unlike spaces, they are visible, they are included in the measured part of the preceding line, except where the layout style allows hyphens to hang into the margins. For additional information about how to format line breaks resulting from the presence of hyphens, see *Section 5.3, Use of Hyphen*.

Conditional Hyphens

00AD	SOFT HYPHEN (SHY)

SHY is an invisible format character with no width. It marks the place where an optional line break may occur inside a word. It can be used with all scripts. If a line is broken at an optional line break position marked by a SHY, the text at that line break position often has a modified appearance as described in *Section 5.4, Use of Soft Hyphen*.

HY: Hyphen

LB12a, LB20a, LB21, LB21a, LB25

002D	HYPHEN-MINUS

Some additional context analysis is required to distinguish usage of this character as a hyphen from its usage as a minus sign (or indicator of numerical range). If used as hyphen, it acts like U+2010 HYPHEN, which has line break class **HH**.

> *Note:* Some typescript conventions use runs of HYPHEN-MINUS to stand in for longer dashes or horizontal rules. If actual character code conversion is not performed and it is desired to treat them like the characters or layout elements they stand for, line breaking needs to support these runs explicitly.

ID: Ideographic

LB23a

Characters with this property do not require other characters to provide break opportunities; lines can ordinarily break before and after and between pairs of ideographic characters. Examples of characters with the ID line break class include most assigned characters in the ranges listed below. Note that this class also includes characters other than Han ideographs.

2E80..2FFF	CJK, Kangxi Radicals, Ideographic Description Symbols
3040..309F	Hiragana (except small characters)
30A2..30FA	Katakana (except small characters)
3400..4DBF	CJK Unified Ideographs Extension A
4E00..9FFF	CJK Unified Ideographs

F900..FAFF	CJK Compatibility Ideographs

See the data file LineBreak.txt [Data14] or the data file DerivedLineBreak.txt [Data14Derived] for the complete list of characters with the **ID** line break class.

> *Note:* Use U+2060 WORD JOINER as a manual override to prevent break opportunities around characters of class **ID**.

Unassigned code points in blocks or regions of the Unicode codespace that have been reserved for CJK scripts are also assigned this line break class. These assignments anticipate that future characters assigned in these ranges will have the class **ID**. Once a character is assigned to one of these code points, the property value could change.

For example, all of the undesignated code points in Planes 2 (20000..2FFFD) and 3 (30000..3FFFD) default to **ID**. See the data file DerivedLineBreak.txt for the complete list of code point ranges which default to the **ID** line break class.

Korean

Korean is encoded with conjoining jamos, Hangul syllables, or both. See also **JL**, **JT**, **JV**, **H2**, and **H3**. The following set of compatibility jamo is treated as **ID** by default.

3130..318F	HANGUL COMPATIBILITY JAMO

Symbols

Certain pictographic symbols of General Category So are also included in this line break class.

HL: Hebrew Letter

LB20a, LB21a, LB21b, LB23, LB24, LB28, LB29, LB30

This class includes all Hebrew letters.

When a Hebrew letter is separated from following non-Hebrew text by a hyphen, there is no break on either side of the hyphen. In this context a hyphen is any character of class **HY** or class **BA**. There is also no break between a solidus and a Hebrew letter. In other respects, Hebrew letters behave the same as characters of class **AL**.

Included in this class are all characters of General Category Letter that have Script=Hebrew.

IN: Inseparable Characters

LB22

Leaders

These characters are intended to be used consecutively. There is never a line break between two characters of this class.

Examples include:

2024	ONE DOT LEADER
2025	TWO DOT LEADER
2026	HORIZONTAL ELLIPSIS
FE19	PRESENTATION FORM FOR VERTICAL HORIZONTAL ELLIPSIS

HORIZONTAL ELLIPSIS can be used as a three-dot leader.

IS: Infix Numeric Separator

LB15b, LB15c, LB15d, LB25, LB29

Characters that usually occur inside a numerical expression may not be separated from the numeric characters that follow, unless a space character intervenes. For example, there is no break in “100.00” or “10,000”, nor in “12:59”.

Examples include:

002C	COMMA
002E	FULL STOP
003A	COLON
003B	SEMICOLON
037E	GREEK QUESTION MARK (canonically equivalent to 003B)
0589	ARMENIAN FULL STOP
060C	ARABIC COMMA
060D	ARABIC DATE SEPARATOR
07F8	NKO COMMA
2044	FRACTION SLASH

When not used in a numeric context, infix separators are sentence-ending punctuation. Therefore they always prevent breaks before.

Note: FIGURE SPACE, not being a punctuation mark, has been given the line break class **GL**.

JL: Hangul L Jamo

LB26, LB27

The **JL** line break class consists of all characters of Hangul Syllable Type L.

Conjoining jamos form Korean Syllable Blocks, which are kept together; see Unicode Standard Annex #29, “Unicode Text Segmentation” [UAX29]. Korean uses space-based line breaking in many styles of documents. To support these, Hangul syllables and conjoining jamos need to be tailored to use class **AL**. The default in this specification is class **ID**, which supports the case of Korean documents not using space-based line breaking. See *Section 8.1, Types of Tailoring*. See also **JT**, **JV**, **H2**, and **H3**.

JT: Hangul T Jamo

LB26, LB27

The **JT** line break class consists of all characters of Hangul Syllable Type T. See also **JL**, **JV**, **H2**, and **H3**.

JV: Hangul V Jamo

LB26, LB27

The **JV** line break class consists of all characters of Hangul Syllable Type V. See also **JL**, **JT**, **H2**, and **H3**.

LF: Line Feed (Non-tailorable)

LB5, LB6, LB9, LB15a, LB15b, LB20a

000A	LINE FEED (LF)

There is a mandatory break after any LF character, but see the discussion under **BK**.

NL: Next Line (Non-tailorable)

LB5, *LB6*, *LB9*, *LB15a*, *LB15b*, *LB20a*

0085	NEXT LINE (NEL)

The **NL** class acts like **BK** in all respects (there is a mandatory break after any NEL character). It cannot be tailored, but implementations are not required to support the NEL character; see the discussion under **BK**.

NS: Nonstarters

LB1, *LB16*, *LB21*

Nonstarter characters cannot start a line, but unlike **CL** they may allow a break in some contexts when they follow one or more space characters. Nonstarters include:

17D6	KHMER SIGN CAMNUC PII KUUH
203C	DOUBLE EXCLAMATION MARK
203D	INTERROBANG
2047	DOUBLE QUESTION MARK
2048	QUESTION EXCLAMATION MARK
2049	EXCLAMATION QUESTION MARK
3005	IDEOGRAPHIC ITERATION MARK
301C	WAVE DASH
303C	MASU MARK
303B	VERTICAL IDEOGRAPHIC ITERATION MARK
309B.. 309E	KATAKANA-HIRAGANA VOICED SOUND MARK..HIRAGANA VOICED ITERATION MARK
30A0	KATAKANA-HIRAGANA DOUBLE HYPHEN
30FB	KATAKANA MIDDLE DOT
30FD..30FE	KATAKANA ITERATION MARK..KATAKANA VOICED ITERATION MARK
FE13	PRESENTATION FORM FOR VERTICAL COLON
FE54..FE55	SMALL SEMICOLON..SMALL COLON
FF1A..FF1B	FULLWIDTH COLON.. FULLWIDTH SEMICOLON
FF65	HALFWIDTH KATAKANA MIDDLE DOT
FF9E..FF9F	HALFWIDTH KATAKANA VOICED SOUND MARK..HALFWIDTH KATAKANA SEMI-VOICED SOUND MARK

> *Note:* Optionally, the **NS** restriction may be relaxed by tailoring, with some or all characters treated like **ID** to achieve a more permissive style of line breaking, especially in some East Asian document styles. Alternatively, line breaking can be tightened by moving characters that are **ID** into **NS**.

For additional information about U+30A0 KATAKANA-HIRAGANA DOUBLE HYPHEN, see *Section 5.5, Use of Double Hyphen*.

NU: Numeric

LB15c, *LB23*, *LB25*, *LB30*

These characters behave like ordinary characters (**AL**) in the context of most characters but activate the prefix and postfix behavior of prefix and postfix characters.

Numeric characters consist of decimal digits (all characters of General_Category Nd), except:

1. those with East_Asian_Width F (Fullwidth)
2. those from scripts that use the Brahmic style of context analysis

plus these characters:

066B	ARABIC DECIMAL SEPARATOR
066C	ARABIC THOUSANDS SEPARATOR

Unlike **IS** characters, the Arabic numeric punctuation does not occur as sentence terminal punctuation outside numbers.

OP: Open Punctuation

LB14, LB15a, LB25, LB30

The opening character of any set of paired punctuation should be kept with the character that follows. This is desirable, even if there are intervening space characters, as it prevents the appearance of a bare opening punctuation mark at the end of a line. The **OP** line break class consists of all characters of General_Category Ps in the Unicode Character Database, plus

00A1	INVERTED EXCLAMATION MARK
00BF	INVERTED QUESTION MARK
2E18	INVERTED INTERROBANG

> *Note:* The first two of these characters used to be in the class **AI** based on their East_Asian_Width assignment of A. Such characters are normally resolved to either **ID** or **AL**. However, the characters listed above are used as punctuation marks in Spanish, where they would behave more like a character of class **OP**.

PO: Postfix Numeric

LB23a, LB24, LB25, LB27

Characters that usually follow a numerical expression may not be separated from preceding numeric characters or preceding closing characters. For example, there is no break opportunity in “(12.00)%”.

Some of these characters—in particular, *degree sign* and *percent sign*—can appear on both sides of a numeric expression. Therefore the line breaking algorithm by default does not break between **PO** and numbers or letters on either side.

Examples of Postfix characters include

0025	PERCENT SIGN
00A2	CENT SIGN
00B0	DEGREE SIGN
060B	AFGHANI SIGN
066A	ARABIC PERCENT SIGN
2030	PER MILLE SIGN
2031	PER TEN THOUSAND SIGN
2032..2037	PRIME..REVERSED TRIPLE PRIME
20A7	PESETA SIGN
2103	DEGREE CELSIUS
2109	DEGREE FAHRENHEIT
FDFC	RIAL SIGN
FE6A	SMALL PERCENT SIGN
FF05	FULLWIDTH PERCENT SIGN
FFE0	FULLWIDTH CENT SIGN

Alphabetic characters are also widely used as unit designators in a postfix position. For purposes of line breaking, their classification as alphabetic is sufficient to keep them together with the preceding number.

PR: Prefix Numeric

LB23a, LB24, LB25, LB27

Characters that usually precede a numerical expression may not be separated from following numeric characters or following opening characters. For example, there is no break opportunity in “$(100.00)”.

Many currency signs can appear on both sides, or even the middle, of a numeric expression. Therefore the line breaking algorithm, by default, does not break between **PR** and numbers or letters on either side.

All currency symbols (General_Category Sc) except those in class **PO** have been assigned line breaking class **PR**. This class also contains all unassigned code points in the Currency Symbols block, and additional characters, including:

002B	PLUS SIGN
005C	REVERSE SOLIDUS
00B1	PLUS-MINUS SIGN
2116	NUMERO SIGN
2212	MINUS SIGN
2213	MINUS-OR-PLUS SIGN

> *Note:* Many currency symbols may be used either as prefix or as postfix, depending on local convention. For details on the conventions used, see [CLDR].

QU: Quotation

LB15a, LB15b, LB19, LB19a

Some quotation characters can be opening or closing, or even both, depending on usage. The default is to use the General_Category values Initial_Punctation and Final_Punctation as a hint, together with context, but to err on the side of treating them as both opening and closing, thus preventing breaks on either side. This will prevent some breaks that might have been legal for a particular language or usage, such as outside a Simplified Chinese quotation of Latin text, or before a German quotation of text starting with a full stop.

> *Note:* If language information is available, it can be used to determine which character is used as the opening quote and which as the closing quote. See the information in *Section 6.2, General Punctuation*, in [Unicode]. In such a case, the quotation marks could be tailored to either **OP** or **CL** depending on their actual usage.

The **QU** line break class consists of characters of General_Category Pf or Pi in the Unicode Character Database and additional characters, including:

0022	QUOTATION MARK
0027	APOSTROPHE
275B	HEAVY SINGLE TURNED COMMA QUOTATION MARK ORNAMENT
275C	HEAVY SINGLE COMMA QUOTATION MARK ORNAMENT
275D	HEAVY DOUBLE TURNED COMMA QUOTATION MARK ORNAMENT
275E	HEAVY DOUBLE COMMA QUOTATION MARK ORNAMENT
2E00..2E01	RIGHT ANGLE SUBSTITUTION MARKER..RIGHT ANGLE DOTTED SUBSTITUTION MARKER
2E06..2E08	RAISED INTERPOLATION MARKER..DOTTED TRANSPOSITION MARKER
2E0B	RAISED SQUARE

RI: Regional Indicator

LB30a

For line Breaking, the Regional Indicator characters are all those with the Unicode character property of Regional_Indicator. This includes:

1F1E6..1F1FF	REGIONAL INDICATOR SYMBOL LETTER A .. REGIONAL INDICATOR SYMBOL LETTER Z

Pairs of RI characters are used to represent a two-letter ISO 3166 region code.

Runs of adjacent RI characters are grouped into pairs, beginning at the start of the run. No break opportunity occurs within a pair; breaks can occur between adjacent pairs. When RI characters are adjacent to characters of other classes, breaks can occur before and after, except where forbidden by other rules.

SA: Complex-Context Dependent (South East Asian)

LB1

Runs of these characters require morphological analysis to determine break opportunities. This is similar to, for example, a hyphenation algorithm. For the characters that have this property, **no** break opportunities will be found otherwise. Therefore complex context analysis, often involving dictionary lookup of some form, is required to determine non-emergency line breaks. If such analysis is not available, it is recommended to treat them as **AL**.

> *Note:* These characters can be mapped into their equivalent line breaking classes by using dictionary lookup, thus permitting a logical separation of this algorithm from the morphological analysis.

The class **SA** consists of all characters of General_Category Cf, Lo, Lm, Mn, or Mc in the following blocks that are not members of another line break class.

0E00..0E7F	Thai
0E80..0EFF	Lao
1000..109F	Myanmar
1780..17FF	Khmer
1950..197F	Tai Le
1980..19DF	New Tai Lue
1A20..1AAF	Tai Tham
A9E0..A9FF	Myanmar Extended-B
AA60..AA7F	Myanmar Extended-A
AA80..AADF	Tai Viet
11700..1173F	Ahom

SG: Surrogate (Non-tailorable)

LB1

Line break class **SG** comprises all code points with General_Category Cs. The line breaking behavior of isolated surrogates is undefined. In UTF-16, paired surrogates represent non-BMP code points. Such code points must be resolved before assigning line break properties. In UTF-8 and UTF-32 surrogate code points represent corrupted data and their line break behavior is undefined.

> *Note:* The use of this line breaking class is deprecated. It was of limited usefulness for UTF-16 implementations that did not support characters beyond the BMP. The correct implementation is to resolve a *pair* of surrogates into a supplementary character before line breaking.

SP: Space (Non-tailorable)

LB7, LB8, LB9, LB12a, LB14, LB15a, LB15b, LB15c, LB16, LB17, LB18, LB20a

The space characters are used as explicit break opportunities; they allow line breaks before most other characters. However, spaces at the end of a line are ordinarily not measured for fit. If there is a sequence of space characters, and breaking after any of the space characters would result in the

same visible line, then the line breaking position after the last space character in the sequence is the locally most optimal one. In other words, when the last character measured for fit is *before* the space character, any number of space characters are kept together invisibly on the previous line and the first non-space character starts the next line.

0020	SPACE (SP)

Note: By default, SPACE, but none of the other breaking spaces, is used in determining an indirect break. For other breaking space characters, see **BA**.

SY: Symbols Allowing Break After

LB13, LB15b, LB21b, LB25

The **SY** line breaking property is intended to provide a break opportunity after, except in front of digits, so as to not break “1/2” or “06/07/99”.

002F	SOLIDUS

URLs are now so common in regular plain text that they need to be taken into account when assigning general-purpose line breaking properties. Slash (*solidus*) is allowed as an additional, limited break opportunity to improve layout of Web addresses. As a side effect, some common abbreviations such as “w/o” or “A/S”, which normally would not be broken, acquire a line break opportunity. The recommendation in this case is for the layout system not to utilize a line break opportunity allowed by **SY** unless the distance between it and the next line break opportunity exceeds an implementation-defined minimal distance.

Note: Normally, symbols are treated as **AL**. However, symbols can be added to this line breaking class or classes **BA**, **BB**, and **B2** by tailoring. This can be used to allow additional line breaks—for example, after “=”. Mathematics requires additional specifications for line breaking, which are outside the scope of this annex.

VF: Virama Final

LB28a

The **VF** line break class is only used for scripts that use the Brahmic style of context analysis. It contains the viramas of Indic syllabic category Pure_Killer in scripts where the final consonant of a phonological syllable is expressed as a sequence of a consonant and such a virama, and the final consonant needs to be kept together with the preceding orthographic syllable. This includes:

1BF2..1BF3	BATAK PANGOLAT..BATAK PANONGONAN

Viramas of Indic syllabic category Pure_Killer that don’t meet the conditions for line break class **VF** use the line break class **CM**.

VI: Virama

LB28a

The **VI** line break class is only used for scripts that use the Brahmic style of context analysis. It contains the viramas of Indic syllabic categories Virama and Invisible_Stacker of such scripts.

1B44	BALINESE ADEG ADEG
A9C0	JAVANESE PANGKON
11046	BRAHMI VIRAMA
1134D	GRANTHA SIGN VIRAMA
11F42	KAWI CONJOINER

WJ: Word Joiner (Non-tailorable)

LB11, LB15b

These characters glue together left and right neighbor characters such that they are kept on the same line.

2060	WORD JOINER (WJ)
FEFF	ZERO WIDTH NO-BREAK SPACE (ZWNBSP)

The word joiner character is the preferred choice for an invisible character to keep other characters together that would otherwise be split across the line at a direct break. The character FEFF has the same effect, but because it is also used in an unrelated way as a *byte order mark,* the use of the WJ as the preferred interword glue simplifies the handling of FEFF.

By definition, WJ and ZWNBSP take precedence over the action of **SP**, but not **ZW**.

XX: Unknown

LB1

The **XX** line break class consists of all characters with General_Category Co as well as those unassigned code points that are not within a CJK block. Unassigned characters in blocks or ranges of the Unicode codespace that have been reserved for CJK scripts default to the class **ID**, and are listed in the description of that class.

Unassigned code positions, private-use characters, and characters for which reliable line breaking information is not available are assigned this line breaking property. The default behavior for this class is identical to class **AL**. Users can manually insert ZWSP or WORD JOINER around characters of class **XX** to allow or prevent breaks as needed.

In addition, implementations can override or tailor this default behavior—for example, by assigning characters the property **ID** or another class. Doing so may give better default behavior for their users. There are other possible means of determining the desired behavior of private-use characters. For example, one implementation might treat any private-use character in ideographic context as **ID**, while another implementation might support a method for assigning specific properties to specific definitions of private-use characters. The details of such use of private-use characters are outside the scope of this standard.

For supplementary characters, a useful default is to treat characters in the range 10000..1FFFD as **AL** and characters in the ranges 20000..2FFFD and 30000..3FFFD as **ID**, until the implementation can be revised to take into account the actual line breaking properties for these characters.

For more information on handling default property values for unassigned characters, see the discussion on default property values in *Section 5.3, Unknown and Missing Characters*, of [Unicode].

The line breaking rules in *Section 6, Line Breaking Algorithm* assume that all unknown characters have been assigned one of the other line breaking classes, such as **AL**, as part of assigning line breaking classes to the input characters.

Implementations that do not support a given character should also treat it as unknown (**XX**).

ZW: Zero Width Space (Non-tailorable)

LB7, LB8, LB9, LB15a, LB15b, LB20a

200B	ZERO WIDTH SPACE (ZWSP)

This character is used to enable additional (invisible) break opportunities wherever SPACE cannot be used. As its name implies, it normally has no width. However, its presence between two characters does not prevent increased letter spacing in justification.

ZWJ: Zero Width Joiner (Non-tailorable)

LB8a, LB9, LB10

200D	ZERO WIDTH JOINER (ZWJ)

A ZWJ prevents breaks between most pairs of characters that would otherwise break. It has various uses, including as a connector in emoji zwj sequences and as a joiner in complex scripts.

Emoji zwj sequences are defined by *ED-16, emoji zwj sequence*, in [UTS51] and implemented for line breaking by rule **LB8a**. In other respects, the line breaking behavior of ZWJ is that of a combining character of class **CM**.

5.2 Dictionary Usage

Dictionaries follow specific conventions that guide their use of special characters to indicate features of the terms they list. Marks used for some of these conventions may occur near line break opportunities and therefore interact with line breaking. For example, in one dictionary a natural hyphen in a word becomes a tilde dash when the word is split. Section 6.2.8, *Hyphenation Point and Dictionary Syllabification*, of [Unicode] illustrates the use of marks whose line breaking classes have been assigned to accomodate various dictionary usages.

5.3 Use of Hyphen

The rules for treating hyphens in line breaking vary by language. In many instances, these rules are not supported as such in the algorithm, but the correct appearance can be realized by using a *non-breaking hyphen.*

Some languages and some transliteration systems use a hyphen at the first position in a word. For example, the Finnish orthography uses a hyphen at the start of a word in certain types of compounds of the form xxx yyy -zzz (where xxx yyy is a two-word expression that acts as the first part of a compound noun, with zzz as the second part). Line break after the hyphen is not allowed here by rule LB20a.

There are line breaking conventions that modify the appearance of a line break when the line break opportunity is based on an explicit hyphen. In standard Polish orthography, explicit hyphens are always promoted to the next line if a line break occurs at that location in the text. For example, if, given the sentence "Tam wisi czerwono-niebieska flaga" ("There hangs a red-blue flag"), the optimal line break occurs at the location of the explicit hyphen, an additional hyphen will be displayed at the beginning of the next line like this:

> Tam wisi czerwono-
> -niebieska flaga.

The same convention is used in Portuguese, where the use of hyphens is common, because they are mandatory for verb forms that include a pronoun. Homographs or ambiguity may arise if hyphens are treated incorrectly: for example, "disparate" means "folly" while "dispara-te" means "fire yourself" (or "fires onto you"). Therefore the former needs to be line broken as

> dispara-
> te

and the latter as

> dispara-
> -te.

A recommended practice is to type <SHY, NON-BREAKING HYPHEN> instead of <HYPHEN> to achieve promotion of the hyphen to the next line. This practice is reportedly already common and supported by major text layout applications. See also *Section 5.4, Use of Soft Hyphen*.

5.4 Use of Soft Hyphen

Unlike U+2010 HYPHEN, which always has a visible rendition, the character U+00AD SOFT HYPHEN (SHY) is an invisible format character that merely indicates a preferred intraword line break position. If the line is broken at that point, then whatever mechanism is appropriate for intraword line breaks should be invoked, just as if the line break had been triggered by another hyphenation mechanism, such as a dictionary lookup. Depending on the language and the word, that may produce different visible results, for example:

- Simply inserting a hyphen glyph
- Inserting a hyphen glyph and changing spelling in the divided word parts
- Not showing any visible change and simply breaking at that point
- Inserting a hyphen glyph at the beginning of the new line

The following are a few examples of spelling changes. Each example shows the line break as " / " and any inserted hyphens. There are many other cases.

- In pre-reform German orthography, a "c" before the hyphenation point can change into a "k": "Drucker" hyphenates into "Druk- / ker".
- In modern Dutch, an *e-diaeresis* after the hyphenation point can change into a simple "e": "geërfde" hyphenates into "ge- / erfde", and "geëerd" into "ge-/ eerd".
- In Swedish, a consonant is sometimes doubled: "tuggummi"; hyphenates into "tugg- / gummi".
- In Dutch, a letter can disappear: "opaatje" hyphenates into "opa- / tje".

The inserted hyphen glyph can take a wide variety of shapes, as appropriate for the situation. Examples include shapes like U+2010 HYPHEN, U+058A ARMENIAN HYPHEN, U+180A MONGOLIAN NIRUGU, or U+1806 MONGOLIAN TODO SOFT HYPHEN.

When a SHY is used to represent a possible hyphenation location, the spelling is that of the word without hyphenation: "tug<SHY>gummi". It is up to the line breaking implementation to make any necessary spelling changes when such a possible hyphenation is actually used.

Sometimes it is desirable to encode text that includes line breaking decisions and will not be further broken into lines. If such text includes hyphenations, the spelling needs to reflect the changes due to hyphenation: "tugg<U+2010>/ gummi", including the appropriate character for any inserted hyphen. For a list of dash-like characters in Unicode, see *Section 6.2, General Punctuation*, in [Unicode].

Hyphenation, and therefore the SHY, can be used with the Arabic script. If the rendering system breaks at that point, the display—including shaping—should be what is appropriate for the given language. For example, sometimes a hyphen-like mark is placed on the end of the line. This mark looks like a *kashida*, but is not connected to the letter preceding it. Instead, the appearance of the mark is as if it had been placed—and the line divided—after the contextual shapes for the line have been determined. For more information on shaping, see [UAX9] and *Section 9.2, Arabic*, of [Unicode].

There are three types of hyphens: explicit hyphens, conditional hyphens, and dictionary-inserted hyphens resulting from a hyphenation process. There is no character code for the third kind of hyphen. If a distinction is desired, the fact that a hyphen is dictionary-inserted and not user-supplied can only be represented out of band or by using another control code instead of SHY.

The action of a hyphenation algorithm is equivalent to the insertion of a SHY. However, when a word contains an explicit SHY, it is customarily treated as overriding the action of the hyphenator for that word.

The sequence <SHY, NON-BREAKING HYPHEN> is given a particular interpretation, see *Section 5.3, Use of Hyphen*.

5.5 Use of Double Hyphen

In some fonts, notably Fraktur fonts, it is customary to use a double-stroke form of the hyphen, usually oblique. Such use is a font-based glyph variation and does not affect line breaking in any way. In texts using such a font, automatic hyphenation or SHY would also result in the display of a double-stroke, oblique hyphen.

Some modern editions of older German publications use a horizontal double hyphen to transcribe the original Fraktur hyphens, but a single hyphen for modern automatic hyphenation. Such editions can be represented using U+2E40 ⹀ DOUBLE HYPHEN for the double hyphens.

In some dictionaries, such as *Webster's 3rd New International Dictionary*, double-stroke, oblique hyphens are used to indicate an explicit hyphen at the end of the line; in other words, a hyphen that would be retained when the term shown is not line wrapped. It is not necessary to store a special character in the data to support this option; one merely needs to substitute the glyph of any ordinary hyphen that winds up at the end of a line. In this example, if the shape of the special hyphen matches an existing character, such as U+2E17 DOUBLE OBLIQUE HYPHEN, that character can be substituted temporarily for display purposes by the line formatter. With such a convention, automatic hyphenation or SHY would result in the display of an ordinary hyphen without further substitution. (See also *Section 5.3, Use of Hyphen*).

Certain linguistic notations make use of a double-stroke, oblique hyphen to indicate specific features, often contrasting with the ordinary hyphen. The U+2E17 ⸗ DOUBLE OBLIQUE HYPHEN character is used in this case.

U+30A0 ゠ KATAKANA-HIRAGANA DOUBLE HYPHEN is used in scientific notation, for example, to mark the presence of a space that would otherwise have been lost in transcribing text, such as the name of a chemical compound, into Katakana. In such notation, ordinary hyphens are retained.

5.6 Tibetan Line Breaking

The Tibetan script uses spaces sparingly, relying instead on the *tsheg*. There is no punctuation equivalent to a period in Tibetan; Tibetan *shad* characters indicate the end of a phrase, not a sentence. Phrases are often metrical—that is, written after every *N* syllables—and a new sentence can often start within the middle of a phrase. Sentence boundaries need to be determined grammatically rather than by punctuation.

Traditionally there is nothing akin to a paragraph in Tibetan text. It is typical to have many pages of text without a paragraph break—that is, without an explicit line break. The closest thing to a paragraph in Tibetan is a new section or topic starting with U+0F12 or U+0F08. However, these occur inline: one section ends and a new one starts on the same line, and the new section is marked only by the presence of one of these characters.

Some modern books, newspapers, and magazines format text more like English with a break before each section or topic—and (often) the title of the section on a separate line. Where this is done, authors insert an explicit line break. Western punctuation (full stop, question mark, exclamation mark, comma, colon, semicolon, quotes) is starting to appear in Tibetan documents, particularly those published in India, Bhutan, and Nepal. Because there are no formal rules for their use in Tibetan, they get treated generically by default. In Tibetan documents published in China, CJK bracket and punctuation characters occur frequently; it is recommended to treat these as in horizontally written Chinese.

> *Note:* The detailed rules for formatting Tibetan texts are complex, and the original assignment of line break classes was found to be insufficient. In [Unicode4.1], the assignment of line break classes for Tibetan was revised significantly in an attempt to better model Tibetan line breaking behavior. No new rules or line break classes were added.

The set of line break classes for Tibetan is expected to provide a good starting point, even though there is limited practical experience in their implementation. As more experience is gained, some modifications, possibly including new rules or additional line break classes, can be expected.

5.7 Word Separator Characters

Visible word separator characters may behave in one of three ways at line breaks. As an example, consider the text "The:quick:brown:fox:jumped.", where the colon (:) represents a visible word separator, with a break between "brown" and "fox". The desired visual appearance could be one of the following:

1. suppress the visible word separator

    The:quick:brown
    fox:jumped.

2. break before the visible word separator

    The:quick:brown
    :fox:jumped.

3. break after the visible word separator

    The:quick:brown:
    fox:jumped.

Both (2) and (3) can be expressed with the Unicode Line Breaking Algorithm by tailoring the Line Break property value for the word separator character to be Break Before or Break After, respectively.

For case (1), the line break opportunity is positioned after the word separator character, as in case (3), but the visual display of the character is suppressed. The means by which a line layout and display process inhibits the visible display of the separator character are outside of the scope of the Line Break algorithm. U+1680 OGHAM SPACE MARK is an example of a character which may exhibit this behavior.

6 Line Breaking Algorithm

Unicode Standard Annex #29, “Unicode Text Segmentation” [UAX29], describes a particular method for boundary detection, based on a set of hierarchical rules and character classifications. That method is well suited for implementation of some of the advanced heuristics for line breaking.

The line breaking algorithm presented in this section can be expressed in a series of rules that take line breaking classes defined in *Section 5.1, Description of Line Breaking Properties*, as input. The title of each rule contains a mnemonic summary of the main effect of the rule. The formal statement of each line breaking rules consists either of a remap rule or of one or more regular expressions containing one or more line breaking classes and one of three special symbols indicating the type of line break opportunity:

! Mandatory break at the indicated position

× No break allowed at the indicated position

÷ Break allowed at the indicated position

In the regular expressions, parentheses may be used for grouping, and square brackets, &, -, and \p{...} may be used to compose sets of characters, as in UAX #29, *Unicode Text Segmentation* [UAX29] and in UTS #18, *Unicode Regular Expressions* [UTS18]. Use of a line break class such as **BK** is short for the property expression \p{lb=**BK**}. The symbol $EastAsian stands for the set [\p{ea=F}\p{ea=W}\p{ea=H}] of characters with Fullwidth, Wide, or Halfwidth East Asian Width.

The rules are applied in order. That is, there is an implicit “otherwise” at the front of each rule following the first. It is possible to construct alternate sets of such rules that are fully equivalent. To be equivalent, an alternate set of rules must have the same effect.

The distinction between a direct break and an indirect break as defined in *Section 2, Definitions*, is handled in rule **LB18**, which explicitly considers the effect of **SP**. Because rules are applied in order, allowing breaks following **SP** in rule **LB18** implies that any prohibited break in rules **LB19**–**LB30** is equivalent to an indirect break.

The examples for each rule use representative characters, where ‘H’ stands for an ideographs, ‘h’ for small kana, and ‘9’ for digits. Except where a rule contains no expressions, the italicized text of the rule

is intended merely as a handy summary.

The algorithm consists of a part for which tailoring is prohibited and a freely tailorable part.

6.1 Non-tailorable Line Breaking Rules

The rules in this subsection and the membership in the classes **BK**, **CM**, **CR**, **GL**, **LF**, **NL**, **SP**, **WJ**, **ZW** and **ZWJ** define behavior that is required of all line break implementations; see *Section 4, Conformance*.

Resolve line breaking classes:

LB1 *Assign a line breaking class to each code point of the input. Resolve **AI**, **CB**, **CJ**, **SA**, **SG**, and **XX** into other line breaking classes depending on criteria outside the scope of this algorithm.*

In the absence of such criteria all characters with a specific combination of original class and General_Category property value are resolved as follows:

Resolved	Original	General_Category
AL	AI, SG, XX	Any
CM	SA	Only Mn or Mc
AL	SA	Any except Mn and Mc
NS	CJ	Any

Start and end of text:

There are two special logical positions: **sot**, which occurs before the first character in the text, and **eot,** which occurs after the last character in the text. Thus an empty string would consist of **sot** followed immediately by **eot**. With these two definitions, the line break rules for start and end of text can be specified as follows:

LB2 *Never break at the start of text.*

sot ×

LB3 *Always break at the end of text.*

! eot

These two rules are designed to deal with degenerate cases, so that there is at least one character on each line, and at least one line break for the whole text. Emergency line breaking behavior usually also allows line breaks anywhere on the line if a legal line break cannot be found. This has the effect of preventing text from running into the margins.

Mandatory breaks:

A hard line break can consist of **BK** or a Newline Function (NLF) as described in *Section 5.8, Newline Guidelines*, of [Unicode]. These three rules are designed to handle the line ending and line separating characters as described there.

LB4 *Always break after hard line breaks.*

BK !

LB5 *Treat **CR** followed by **LF**, as well as **CR**, **LF**, and **NL** as hard line breaks.*

CR × LF

CR !

LF !

NL !

Note: When displaying source code, failing to support all forms of the new line function can have security implications; for instance, executable code can appear commented out. It is therefore strongly recommended that source code editors support the VT character within the BK class, and support the NEL character within the NL class, even though that support is not required for conformance. See *Unicode Technical Standard #55, Unicode Source Code Handling* [UTS55].

***LB6** Do not break before hard line breaks.*

× (BK | CR | LF | NL)

Explicit breaks and non-breaks:

***LB7** Do not break before spaces or zero width space.*

× SP

× ZW

***LB8** Break before any character following a zero-width space, even if one or more spaces intervene.*

ZW SP* ÷

***LB8a** Do not break after a **zero width joiner**.*

ZWJ ×

A **ZWJ** will prevent breaks between most pairs of characters. This behavior is used to prevent breaks within emoji zwj sequences.

Combining marks:

See also *Section 9.2, Legacy Support for Space Character as Base for Combining Marks.*

***LB9** Do not break a combining character sequence; treat it as if it has the line breaking class of the base character in all of the following rules. Treat **ZWJ** as if it were **CM**.*

Treat X (CM | ZWJ)* as if it were X.

where X is any line break class except **BK**, **CR**, **LF**, **NL**, **SP**, or **ZW**.

In subsequent rules, any **CM** or **ZWJ** characters affected by this rule are ignored. Note that despite the summary title, this rule is not limited to standard combining character sequences. For the purposes of line breaking, sequences containing most of the control codes or layout control characters are treated like combining sequences.

***LB10** Treat any remaining **combining mark** or **ZWJ** as **AL**.*

Treat any remaining CM or ZWJ as if it had the properties of U+0041 A LATIN CAPITAL LETTER A, that is, Line_Break=AL, General_Category=Lu, East_Asian_Width=Na, Extended_Pictographic=N.

This catches the case where a **CM** is the first character on the line or follows **SP**, **BK**, **CR**, **LF**, **NL**, or **ZW**.

Word joiner:

***LB11** Do not break before or after Word joiner and related characters.*

× WJ

WJ ×

Non-breaking characters:

LB12 *Do not break after NBSP and related characters.*

GL ×

6.2 Tailorable Line Breaking Rules

The following rules and the classes referenced in them provide a reasonable default set of line break opportunities. Implementations should implement them unless alternate approaches produce better results for some classes of text or applications. When using alternative rules or algorithms, implementations must ensure that the mandatory breaks, break opportunities and non-break positions determined by the algorithm and rules of *Section 6.1, Non-tailorable Line Breaking Rules*, are preserved. See *Section 4, Conformance*.

Non-breaking characters:

LB12a *Do not break before NBSP and related characters, except after spaces and hyphens.*

[^SP BA HY HH] × GL

The expression [^SP BA HY HH] designates any line break class other than **SP**, **BA**, **HY**, or **HH**. The symbol ^ is used, instead of !, to avoid confusion with the use of ! to indicate an explicit break. Unlike the case for **WJ**, inserting a **SP** overrides the non-breaking nature of a **GL**. Allowing a break after **BA** **HH** or **HY** matches widespread implementation practice and supports a common way of handling special line breaking of explicit hyphens, such as in Polish and Portuguese. See *Section 5.3, Use of Hyphen*.

Opening and closing:

These have special behavior with respect to spaces, and therefore come before rule LB18.

LB13 *Do not break before ‘]’ or ‘!’ or ‘/’, even after spaces.*

× CL

× CP

× EX

× SY

LB14 *Do not break after ‘[’, even after spaces.*

OP SP* ×

LB15a *Do not break after an unresolved initial punctuation that lies at the start of the line, after a space, after opening punctuation, or after an unresolved quotation mark, even after spaces.*

(sot | BK | CR | LF | NL | OP | QU | GL | SP | ZW) [\p{Pi}&QU] SP* ×

LB15b *Do not break before an unresolved final punctuation that lies at the end of the line, before a space, before a prohibited break, or before an unresolved quotation mark, even after spaces.*

× [\p{Pf}&QU] (SP | GL | WJ | CL | QU | CP | EX | IS | SY | BK | CR | LF | NL | ZW | eot)

LB15c *Break before a decimal mark that follows a space, for instance, in ‘subtract .5’.*

SP ÷ IS NU

LB15d Otherwise, do not break before ';', ',', or '.', even after spaces.

× IS

LB16 Do not break between closing punctuation and a nonstarter (lb=NS), even with intervening spaces.

(CL | CP) SP* × NS

LB17 Do not break within '——', even with intervening spaces.

B2 SP* × B2

Spaces:

LB18 Break after spaces.

SP ÷

Special case rules:

LB19 Do not break before non-initial unresolved quotation marks, such as ' " ' or ' " ', nor after non-final unresolved quotation marks, such as ' " ' or ' " '.

× [QU - \p{Pi}]

[QU - \p{Pf}] ×

LB19a Unless surrounded by East Asian characters, do not break either side of any unresolved quotation marks.

[^$EastAsian] × QU

× QU ([^$EastAsian] | eot)

QU × [^$EastAsian]

(sot | [^$EastAsian]) QU ×

LB20 Break before and after unresolved CB.

÷ CB

CB ÷

Conditional breaks should be resolved external to the line breaking rules. However, the default action is to treat unresolved **CB** as breaking before and after.

LB20a Do not break after a word-initial hyphen.

(sot | BK | CR | LF | NL | SP | ZW | CB | GL) (HY | HH) × (AL | HL)

LB21 Do not break before hyphen-minus, other hyphens, fixed-width spaces, small kana, and other non-starters, or after acute accents.

× BA

× HH

× HY

× NS

BB ×

LB21a *Do not break after the hyphen in Hebrew + Hyphen + non-Hebrew.*

HL (HY | HH) × [^HL]

LB21b *Do not break between Solidus and Hebrew letters.*

SY × HL

LB22 *Do not break before ellipses.*

× IN

Examples: ‘9...’, ‘a...’, ‘H...’

Numbers:

Do not break alphanumerics.

LB23 *Do not break between digits and letters.*

(AL | HL) × NU

NU × (AL | HL)

LB23a *Do not break between numeric prefixes and ideographs, or between ideographs and numeric postfixes.*

PR × (ID | EB | EM)

(ID | EB | EM) × PO

LB24 *Do not break between numeric prefix/postfix and letters, or between letters and prefix/postfix.*

(PR | PO) × (AL | HL)

(AL | HL) × (PR | PO)

In general, it is recommended to not break lines inside numbers of the form described by the following regular expression:

(**PR** | **PO**) ? (**OP** | **HY**) ? **IS** ? **NU** (**NU** | **SY** | **IS**) * (**CL** | **CP**) ? (**PR** | **PO**) ?

Examples: $(12.35) 2,1234 (12)¢ 12.54¢ .50 ₹1,00,000.00 -1/12

The default line breaking algorithm implements this with the following rule. Note that some cases have already been handled, such as ‘9,’, ‘[9’.

LB25 *Do not break numbers:*

NU (SY | IS)* CL × PO

NU (SY | IS)* CP × PO

NU (SY | IS)* CL × PR

NU (SY | IS)* CP × PR

NU (SY | IS)* × PO

NU (SY | IS)* × PR

PO × OP NU

PO × OP IS NU

PO × NU

PR × OP NU

PR × OP IS NU

PR × NU

HY × NU

IS × NU

NU (SY | IS)* × NU

Korean syllable blocks

Conjoining jamos, Hangul syllables, or combinations of both form Korean Syllable Blocks. Such blocks are effectively treated as if they were Hangul syllables; no breaks can occur in the middle of a syllable block. See Unicode Standard Annex #29, “Unicode Text Segmentation” [UAX29], for more information on Korean Syllable Blocks.

LB26 *Do not break a Korean syllable.*

JL × (JL | JV | H2 | H3)

(JV | H2) × (JV | JT)

(JT | H3) × JT

where the notation (JT | H3) means JT or H3. The effective line breaking class for the syllable block matches the line breaking class for Hangul syllables, which is **ID** by default. This is achieved by the following rule:

LB27 *Treat a Korean Syllable Block the same as **ID**.*

(JL | JV | JT | H2 | H3) × PO

PR × (JL | JV | JT | H2 | H3)

When Korean uses SPACE for line breaking, the classes in rule **LB26**, as well as characters of class **ID**, are often tailored to **AL**; see *Section 8, Customization*.

Finally, join alphabetic letters into words and break everything else.

LB28 *Do not break between alphabetics (“at”).*

(AL | HL) × (AL | HL)

LB28a *Do not break inside the orthographic syllables of Brahmic scripts.*

AP × (AK | [◌] | AS)

(AK | [◌] | AS) × (VF | VI)

(AK | [◌] | AS) VI × (AK | [◌])

(AK | [◌] | AS) × (AK | [◌] | AS) VF

Note: In the above regular expressions, the class [◌] contains the single character U+25CC DOTTED CIRCLE.

LB29 *Do not break between numeric punctuation and alphabetics (“e.g.”).*

IS × (AL | HL)

LB30 *Do not break between letters, numbers, or ordinary symbols and opening or closing parentheses.*

(AL | HL | NU) × [OP-$EastAsian]

[CP-$EastAsian] × (AL | HL | NU)

The purpose of this rule is to prevent breaks in common cases where a part of a word appears between delimiters—for example, in “person(s)”.

The excluded set ($EastAsian) refines the behavior of this rule, to enable a break before an East Asian OP or after an East Asian CP. Those cases are identified by excluding East_Asian_Width values of Fullwidth, Wide, or Halfwidth. This is illustrated by the following example, which shows East Asian corner brackets immediately following a Latin letter in Japanese text. In such a case, the preferred line break is between the Latin letter and the opening angle bracket.

Preferred	**Bad Break**
日中韓統合漢字拡張G「ユニコード」	日中韓統合漢字拡張G「ユニコード」

LB30a *Break between two regional indicator symbols if and only if there are an even number of regional indicators preceding the position of the break.*

sot (RI RI)* RI × RI

[^RI] (RI RI)* RI × RI

LB30b *Do not break between an* ***emoji base*** *(or potential emoji) and an* ***emoji modifier****.*

EB × EM

[\p{Extended_Pictographic}&\p{Cn}] × EM

LB31 *Break everywhere else.*

ALL ÷

÷ ALL

7 Deleted

Formerly was: Pair Table-Based Implementation.

8 Customization

A real-world line breaking algorithm has to be tailorable to some degree to meet user or document requirements.

In Korean, for example, two distinct line breaking modes occur, which can be summarized as breaking after each character or breaking after spaces (as in Latin text). The former tends to occur when text is set justified; the latter, when ragged margins are used. In that case, even ideographs are broken only at space characters. In Japanese, for example, tighter and looser specifications of prohibited line breaks may be used.

Specialized text or specialized text constructs may need specific line breaking behavior that differs from the default line breaking rules given in this annex. This may require additional tailorings beyond those considered in this section. For example, the rules given here are insufficient for mathematical equations, whether inline or in display format. Likewise, text that commonly contains lengthy URLs might benefit from special tailoring that suppresses **SY** × **NU** from rule **LB25** within the scope of a URL to allow breaks after a "/" separated segment in the URL regardless of whether the next segment starts with a digit.

Notes:

- Locale-sensitive line break specifications can be expressed in LDML [UTS35]. Tailorings are available in the Common Locale Data Repository [CLDR].
- Some changes to rules and data are needed for the best segmentation behavior of emoji zwj sequences [UTS51]. Implementations are strongly encouraged to use the line break rules in the latest version of CLDR (Version 35 or later) [CLDR] and the latest emoji properties (Version 12.0 or later) [UTS51].

The remainder of this section gives an overview of common types of tailorings.

8.1 Types of Tailoring

There are two principal ways of tailoring the line breaking algorithm:

1. **Changing the line breaking class assignment for some characters**
   This is useful in cases where the line breaking properties of one class of characters are occasionally lumped together with the properties of another class to achieve a less restrictive line breaking behavior.
2. **Changing the line breaking rules**
   Adding new rules, or altering or removing existing rules, provides more flexibility in changing the line breaking behavior. This can also include introducing new character classes for use by the new or altered rules.

For example, specialized rules could be added to recognize and break common constructs, such as URLs, numeric expressions, and so on. Such open-ended customizations place no limits on possible changes, other than the requirement that non-tailorable line breaking rules be correctly implemented. This means that whatever changes are made must be equivalent to changes to the line breaking assignments of tailorable line breaking rules, and to alteration, removal, or addition of rules applied after rule LB12.

8.2 Examples of Customization

Example 1. The exact method of resolving the line break class for characters with class **SA** is not specified in the default algorithm. One method of implementing line breaks for complex scripts is to invoke context-based classification for all runs of characters with class **SA**. For example, a dictionary-based algorithm could return different classes for Thai letters depending on their context: letters at the start of Thai words would become **BB** and other Thai letters would become **AL**. Alternatively, for text consisting of, or predominantly containing characters with line breaking class **SA**, it may be useful to instead defer the determination of line breaks to a different algorithm.

Example 2. To implement terminal style line breaks, it would be necessary to allow breaks at fixed positions. These could occur inside a run of spaces or in the middle of words without regard to hyphenation. Such a modification essentially disregards the output of the line breaking algorithm, and is therefore not a conformant tailoring. For a system that supports both regular line breaking and terminal style line breaks, only some of its line break modes would be conformant.

Example 3. Depending on the nature of the document, Korean either uses implicit breaking around characters (type 2 as defined in *Section 3, Introduction*) or uses spaces (type 1). Space-based layout is common in magazines and other informal documents with ragged margins, while books, with both margins justified, use the other type, as it affords more line break opportunities and therefore leads to better justification.

Example 4. In a Far Eastern context it is sometimes necessary to allow alphabetic characters and digit strings to break anywhere. According to reference [Suign98], this can again be done in the same way as Korean. This can be implemented by adjusting rules **LB23**, **LB25** and **LB28** to allow breaks between all permutations of the character classes **AL** and **NU**.

Example 5. Some users prefer to relax the requirement that Kana syllables be kept together. For example, the syllable *kyu,* spelled with the two kanas *KI* and "small *yu*", would no longer be kept together as if *KI* and *yu* were atomic. This customization can be handled by mapping class **CJ** to be handled as class **ID** in rule **LB1**.

Example 6. Tailor to prevent line breaks from falling within default grapheme clusters, as defined by Unicode Standard Annex #29, "Unicode Text Segmentation" [UAX29]. The tailoring can be accomplished by first segmenting the text into grapheme clusters according to the rules defined in UAX #29, and then finding line breaks according to the default line break rules, as follows: After applying the mandatory line break rules, give each grapheme cluster the line breaking class of its first code point.

An example of a grapheme cluster that would be split by the default line break rules is U+0020 SPACE followed by a combining mark.

Example 7 (deleted). *Versions 4.1.0 through 15.1.0 of The Unicode Standard defined a tailoring of the line breaking of numeric expressions as Example 7. This tailoring was used in the test files provided with Unicode 5.1.0 and later. Since Unicode version 16.0, that behavior has been incorporated into the default; it no longer constitutes a tailoring.*

Example 8. Some scripts that traditionally follow the Brahmic style of context analysis are nowadays occasionally written with spaces, and word-based line breaking might be desired in that case. This can be accomplished by remapping the line break classes **AK**, **AP**, and **AS** to **AL**; and **VI** or **VF** to **CM**. In some cases other word-forming characters, such as U+A9CF JAVANESE PANGRANGKEP, also need to be remapped to **AL**. Digits, which may have line break class **AS** or **ID** in such scripts, need to be remapped to **NU**. Punctuation, which may have line break class **ID** in such scripts, need to be remapped to **AL** or **BA**.

9 Implementation Notes

This section provides additional notes on implementation issues.

9.1 Combining Marks in Regular Expression-Based Implementations

Implementations that use regular expressions cannot directly express rules **LB9** and **LB10**. However, it is possible to make these rules unnecessary by rewriting *all* the rules from **LB11** on down so that the overall result of the algorithm is unchanged. This restatement of the rules is therefore not a tailoring, but rather an equivalent statement of the algorithm that can be directly expressed as regular expressions.

To replace rule **LB9**, terms of the form

B # **A**

B SP* # **A**

B #

B SP* #

are replaced by terms of the form

B CM* # **A**

B CM* SP* # **A**

B CM* #

B CM* SP* #

where **B** and **A** are any line break class or set of alternate line break classes, such as (X |Y), and where # is any of the three operators !, ÷, or ×.

Note that because **sot**, **BK**, **CR**, **LF**, **NL**, and **ZW** are all handled by rules above **LB9**, these classes cannot occur in position **B** in any rule that is rewritten as shown here.

Replace **LB10** by the following rule:

× CM

For each rule containing AL on its left side, add a rule that is identical except for the replacement of AL by CM, but taking care of correctly handling sets of alternate line break classes. For example, for rule

(AL | NU) × OP

add another rule

CM × OP.

These prescriptions for rewriting the rules are, in principle, valid even where the rules have been tailored as permitted in *Section 4, Conformance*. However, for extended context rules such as in LB25, additional considerations apply. These are described in *Section 6.2, Replacing Ignore Rules*, of Unicode Standard Annex #29, “Unicode Text Segmentation” [UAX29].

9.2 Legacy Support for Space Character as Base for Combining Marks

As stated in *Section 7.9, Combining Marks* of [Unicode], combining characters are shown in isolation by applying them to U+00A0 NO-BREAK SPACE (NBSP). In earlier versions, this recommendation included the use of U+0020 SPACE. The use of SPACE for this purpose has been deprecated because it leads to many complications in text processing. The visual appearance is the same with both NO-BREAK SPACE and SPACE, but the line breaking behavior is different. Under the current rules, **SP CM*** will allow a break between **SP** and **CM***, which could result in a new line starting with a combining mark. Previously, whenever the base character was **SP**, the sequences **CM*** and **SP CM*** were defined to act like indivisible clusters, allowing breaks on either side like **ID**.

Where backward compatibility with documents created under the prior practice is desired, the following tailoring should be applied to those **CM** characters that have a General_Category value of Combining_Mark (M):

*Legacy-CM: In all of the rules following rule **LB8**, if a space is the base character for a combining mark, the space is changed to type **ID**. In other words, break before **SP** in the same cases as one would break before an **ID**.*

Treat **SP CM*** as if it were **ID**.

While this tailoring changes the location of the line break opportunities in the string, it is ordinarily not expected to affect the display of the text. That is because spaces at the end of the line are normally invisible and the recommended display for isolated combining marks is the same as if they were applied to a preceding SPACE or NBSP.

10 Testing

As with the other default specifications, implementations are free to override (tailor) the results to meet the requirements of different environments or particular languages as described in *Section 4, Conformance*. For those who do implement the default breaks as specified in this annex and wish to check that their implementation matches that specification, a test file has been made available in [Tests14].

These tests cannot be exhaustive, because of the large number of possible combinations; but they do provide samples that test all pairs of property values, using a representative character for each value,

plus certain other sequences.

A sample HTML file is also available for each that shows various combinations in chart form, in [Charts14]. The header cells of the chart consist of a property value, followed by a representative code point number. The body cells in the chart show the break status: whether a break occurs between the row property value and the column property value. If the browser supports tool-tips, then hovering the mouse over the code point number will show the character name, General_Category and Script property values. Hovering over the break status will display the number of the rule responsible for that status.

Note: To determine a break it is generally not sufficient to just test the two adjacent characters.

The chart is followed by some test cases. These test cases consist of various strings with the break status between each pair of characters shown by blue lines for breaks and by whitespace for non-breaks. Hovering over each character (with tool-tips enabled) shows the character name and property value; hovering over the break status shows the number of the rule responsible for that status.

Due to the way they have been mechanically processed for generation, the test rules do not match the rules in this annex precisely. In particular:

1. The rules are cast into a more regex-style.
2. ~~The rules “sot”, “eot”, and “Any” are added mechanically and have artificial numbers.~~
3. The rules are given decimal numbers without prefixes, so rules such as LB14 are given a number using tenths, such as 14.0.
4. Where a rule has multiple parts (lines), each one is numbered using hundredths, such as
    - 13.01) [^NU] × CL
    - 13.02) × EX
    - ...

The mapping from the rule numbering in this annex to the numbering for the test rules is summarized in *Table 4*.

Table 4. Numbering of Test Rules

Rule in This Annex	Test Rule	Comment
~~LB2~~	~~0.2~~	~~start of text~~
~~LB3~~	~~0.3~~	~~end of text~~
LB12a	12.0	GL ×
LB12b	12.1	[^SP, BA, HY] × GL
~~LB31~~	~~999~~	~~÷ any~~

11 History

Since its publication in 1999 as part of Unicode Version 3.0.0, the line breaking algorithm has undergone many changes. It started as a set of 29 line breaking classes involved in 23 rules which were representable as a pair table with some special handling for combining marks and spaces. It now encompasses 48 line breaking classes involved in more than 40 rules, many of which rely on extended context which may be several characters removed from the position they govern.

As the algorithm grew, rules were split, reordered, added, and removed. In Unicode Version 5.0, the rules were renumbered to reduce the number of alphabetic suffixes on the rule numbers.

Please refer to Unicode Technical Note #54, “Annotated Line Breaking Algorithm” [UTN54], for a complete history of the changes to the text of this document since Unicode Version 3.0.0, and for

additonal background on these changes.

Of particular note is the history of the line breaking assignment of U+034F COMBINING GRAPHEME JOINER. This character was originally meant to merge adjoining characters into a graphemic unit, and the character was accordingly originally documented in Version 3.2 of this annex as having line breaking class **GL**. However, this behavior of the combining grapheme joiner was made obsolete in Unicode Version 4.0, and the character was repurposed for uses where the line breaking algorithm should ignore it. From that point on, retaining the line breaking assignment **GL** was a mistake, and changing it to **CM** would have been appropriate. This was only corrected in Unicode Version 17.0, more than twenty years later. For more on the history of U+034F COMBINING GRAPHEME JOINER, including a mistake in the data files in the other direction between Unicode Version 3.2 and Unicode Version 4.1, see Section 6.3 of UTC document L2/24-224.

References

For references for this annex, see Unicode Standard Annex #41, “Common References for Unicode Standard Annexes” [UAX41].

Acknowledgments

Asmus Freytag created the initial version of this annex and maintained the text for many years. Andy Heninger maintained the text from 2008 through 2019. Christopher Chapman maintained the text from 2020 through 2022. Robin Leroy has maintained the text since September 2022.

The initial assignments of properties are based on input by Michel Suignard. Mark Davis provided algorithmic verification and formulation of the rules, and detailed suggestions on the algorithm and text. Ken Whistler, Rick McGowan, Deborah Anderson, Lorna Evans, and other members of the editorial committee provided valuable feedback. Tim Partridge enlarged the information on dictionary usage. Sun Gi Hong reviewed the information on Korean and provided copious printed samples. Eric Muller reanalyzed the behavior of the soft hyphen and collected the samples. Adam Twardoch provided the Polish example. António Martins-Tuválkin supplied information about Portuguese. Tomoyuki Sadahiro provided information on use of U+30A0. Christopher Fynn provided the background information on Tibetan line breaking. Andrew West, Kamal Mansour, Andrew Glass, Daniel Yacob, and Peter Kirk suggested improvements for Mongolian, Arabic, Kharoshthi, Ethiopic, and Hebrew punctuation characters, respectively. Kent Karlsson reviewed the line break properties for consistency. Jerry Hall reviewed the sample code. Elika J. Etemad (fantasai) reviewed the entire document in an effort to make it easier to reference from external standards. Norbert Lindenberg added the Brahmic style of line breaking and provided clarifications on the South East Asian style of line breaking. Charlotte Buff and David Corbett provided ample feedback on property assignments and ramifications of the rules. Many others provided additional review of the rules and property assignments.

Modifications

The following summarizes modifications from the previous revision of this annex.

Revision 56:

- **Proposed Update** for Unicode 18.0.
- Changed the Line_Break assignment of U+2012 – FIGURE DASH and U+2013 – EN DASH from HH to BA, and the Line_Break assignment of U+00AD SOFT HYPHEN from BA to HH. [187-C45]
- Changed rule LB12a to disallow a break between BA and GL, e.g., between EN DASH and NO-BREAK SPACE. [187-C46]

~~**Revision 55:**~~

- ~~**Reissued** for Unicode 17.0.~~

- ~~Updated the test files to more closely match the rules.~~
- ~~Split part of class BA into a new class Unambiguous_Hyphen (HH) and updated rules LB20a and LB21a to use it. Rules LB12a and LB21 treat HH like BA, preserving their old behavior. [181-C53]~~
- ~~Updated rule LB20a to treat Hebrew letters (HL) like other alphabetic characters (AL). [181-C53]~~
- ~~Updated the descriptions of classes CM and GL to reflect the change to the Line_Break property of U+034F COMBINING GRAPHEME JOINER. Added a discussion of this long-standing mistaken assignment in Section 11, History. [181-C54]~~
- ~~Section 5.3, *Use of Hyphen*: updated for LB20a added in Unicode 16.0. [179-C32]~~
- ~~Section 5.5, *Use of Double Hyphen*: Add a discussion of the actual DOUBLE HYPHEN, based on L2/11-038.~~
- ~~Section 5.5, *Use of Double Hyphen*: Corrected confusing statements about U+2E17 DOUBLE OBLIQUE HYPHEN.~~

Modifications for previous versions are listed in those respective versions.

© 1999–2026 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.