

3.0 New Character Semantics

This section provides a description of the use of new characters in Unicode 1.1 and some modifications in the semantics of Unicode 1.0 characters that were made in the process of the merger.

3.1 Double Non-Spacing Marks

[0360] COMBINING DOUBLE TILDE
[0361] COMBINING DOUBLE INVERTED BREVE

There are two double diacritics added in Unicode 1.1. These marks apply to the previous base character—just like all other non-spacing marks—but hang over the following letter as well. For example:

$$o + \tilde{\sim} \Rightarrow \tilde{o}$$

$$o + \tilde{\sim} + o \Rightarrow \tilde{oo}$$

The double diacritics always bind more loosely than other non-spacing marks, and thus sort at the end in the canonical representation. When rendering, the double diacritic will float above other diacritics (excluding surrounding diacritics).

$$o + \hat{\circ} + \tilde{\sim} + o + \ddot{o} \Rightarrow \hat{o}\tilde{\sim}\ddot{o}$$

$$o + \tilde{\sim} + \hat{\circ} + o + \ddot{o} \Rightarrow \tilde{o}\hat{\circ}\ddot{o}$$



For compatibility, character halves have been introduced for parts of these characters. They are only coded for compatibility, and their use is discouraged.

[FE22] COMBINING DOUBLE TILDE LEFT HALF
[FE23] COMBINING DOUBLE TILDE RIGHT HALF
[FE20] COMBINING LIGATURE LEFT HALF
[FE21] COMBINING LIGATURE RIGHT HALF

☞ When converting the half-characters to the double form there may be intervening characters.

Other multiple non-spacing marks, such as triple diacritics, are used in certain bibliographic cases. Their use and semantics will be discussed at the time they are introduced into Unicode.

3.2 Zero-Width Joining

U+200C  ZERO WIDTH NON-JOINER
U+200D  ZERO WIDTH JOINER

In the merger with ISO/IEC 10646-1, the semantics of these two characters have been given a narrow interpretation. This brings added precision to the explanation given in Volume 1, p. 77.

The intent of these characters is to address cursive graphical connections between the glyphs of a script, e.g. in scripts like Arabic whose printed form emulates handwriting. NON-JOINER and JOINER are best thought of as behaving like tiny letters that neighboring glyphs may connect to (JOINER) or avoid connecting to (NON-JOINER). They are thus processed as ordinary cursive letters rather than as control characters.

NON-JOINER and JOINER affect how the two neighboring glyphs connect to *them*, not to *each other*. As such, they have no direct relationship with ligature formation; in particular, JOINER does not in any way request that its two neighbors be ligatures to each other. Indeed,







both NON-JOINER and JOINER may break up ligatures by interrupting the character sequence required to form the ligature.

The precise relationship between cursive appearance and ligated appearance may differ from script to script, and therefore the precise usage of these characters is script-dependent. In the case of Latin typography, corrosiveness (handwriting emulation) and ligatures are independent. Thus the text on Volume 1, p. 77, may be clarified as follows:

f + JOINER + *i* will not form the ligature *fi*. Instead, if cursive versions of the *f* and *i* are available in the font, each will independently connect to the JOINER on the appropriate side (having the same appearance as *f* + *i*).

Usage of optional ligatures such as *fi* is not currently controlled by any codes within the Unicode standard, but is determined by protocols or resources external to the text sequence.

As further illustration, let a hyphen stand for a cursive connection to a preceding or following letter. In that case, a cursive Latin font would produce the following results:

<u>Unicode</u>	<u>Rendering</u>
f i s h	f- -i- -s- -h (optionally using a ligature: fi- -s- -h)
f  i s h	f- -i- -s- -h
f  i s h	f i- -s- -h
f   i s h	f- i- -s- -h
f   i s h	f -i- -s- -h

With regard to the Arabic script, the statements in Volume 1, p. 77, remain correct. In Volume 2, p. 390, according to Arabic rules L2 and L3 the JOINER can be used to get the appearance in parentheses.

With regard to conjuncts in Indic scripts, the statements in Volume 1, pp. 53-56, and Volume 2, pp. 399-414, remain correct. However, for clarity the term *ligature* should be replaced by the term *conjunct* throughout pp. 399-414.

3.3 Byte Order Mark

U+FEFF



ZERO WIDTH NO-BREAK SPACE

In addition to the meaning of BYTE ORDER MARK as defined in Volume 1 of the Unicode standard, the code value U+FEFF may now also be used as ZERO WIDTH NO-BREAK SPACE (ZWNBS). For convenience in discussion, it can also be referred to by this name (which is the ISO/IEC 10646-1/Unicode 1.1 name for U+FEFF).

ZWNBS behaves like a U+00A0 NO-BREAK SPACE in that it indicates the absence of word boundaries; however, ZWNBS has no width. For example, this character can be inserted after the fourth character in the text “base + delta” to indicate that there should be no line break between the “e” and the “+” (for more information, see Volume 2, pp. 6-7).

3.4 Additional Alternate Format Characters

The following format characters were introduced as a result of the merger with 10646.

- The symmetric swapping format characters can be used to control the glyphs used to represent characters such as “(“.
- The character shaping selectors could be used to control the shaping behavior of the Arabic compatibility characters.
- The numeric shape selectors codes could be used to override the normal shapes of the Western Digits.

However, the use of the character shaping selectors and digit shapes codes from ISO/IEC 10646-1 is strongly discouraged in Unicode. Instead, the appropriate character codes should be used. For example, if the Arabic digit forms are desired then the explicit characters should be used, such as U+0660 ARABIC-INDIC DIGIT ZERO. Similarly, if contextual forms for Arabic characters are desired, then the nominal characters should be used, and not the presentation forms with the shaping selectors.

3.4.1 Symmetric swapping format characters

The symmetric swapping format characters are used in conjunction with the class of left/right handed pairs of characters (symmetric characters) such as parentheses. The characters thus affected are listed in Appendix G (Symmetric Swapping Characters). They indicate whether the interpretation of the term LEFT or RIGHT in the character names should be OPENING or CLOSING respectively. They do not nest. The default state of SYMMETRIC SWAPPING may be set by a higher level protocol or standard, such as ISO/IEC 6429. In the absence of such a protocol, the default state is ACTIVATE SYMMETRIC SWAPPING.

INHIBIT SYMMETRIC SWAPPING (U+206A)

Between this character and the following ACTIVATE SYMMETRIC SWAPPING format character (if any), the symmetric characters will be interpreted and rendered as LEFT and RIGHT.

ACTIVATE SYMMETRIC SWAPPING (U+206B)

Between this character and the following INHIBIT SYMMETRIC SWAPPING format character (if any), the symmetric characters are rendered as OPENING and CLOSING characters. *This is the default state.*

3.4.2 Character shaping selectors

The character shaping selector characters are used in conjunction with Arabic presentation forms. During the presentation process, certain characters may be joined together in cursive connection or ligatures. The following characters indicate that the character shape determination process used to achieve this presentation effect is to be either activated or inhibited. The following characters do not nest.

INHIBIT ARABIC FORM SHAPING (U+206C)

Between this character and the following ACTIVATE ARABIC FORM SHAPING format character (if any), the character shaping determination process is to be inhibited. The stored Arabic presentation forms will be presented without shape modification. *This is the default state.*

ACTIVATE ARABIC FORM SHAPING (206D)

Between this character and the following INHIBIT ARABIC FORM SHAPING format character (if any), the stored Arabic presentation forms should be presented with shape modification by means of the character shaping determination process.

☞ *These characters have no effect on characters that are not presentation forms: in particular, Arabic nominal characters as from U+0600 to U+06FF are always subject to character shaping, and are unaffected by these formatting characters.*

3.4.3 Numeric shape selectors

The following characters allow the selection of the shapes in which the digits from U+0030 to U+0039 are to be rendered. The following characters do not nest.

NATIONAL DIGIT SHAPES (U+206E)

Between this character and the following NOMINAL DIGIT SHAPES format character (if any), digits from U+0030 to U+0039 are rendered with the appropriate national digit shapes as specified by means of appropriate agreements. For example, they could be displayed with shapes such as the ARABIC-INDIC digits from U+0660 to U+0669.

NOMINAL DIGIT SHAPES (U+206F)

Between this character and the following NATIONAL DIGIT SHAPES format character (if any), the digits from U+0030 to U+0039 will be rendered with the shapes as those shown in the code tables for those digits. *This is the default state.*

3.5 Other New Characters

The certain other non-decomposable characters were added in the process of merging with 10646. These characters are listed in Appendix H: New Characters, p. 34, including nominal glyph, code and name.

The characters that do have a decomposition are listed and marked in Appendix I: Unicode 1.1 Character List, p. 43.³ Two of the listed characters require cross-references:

```
[017F] LATIN SMALL LETTER LONG S
      ->non-final form of LATIN SMALL LETTER S
      x 0283 LATIN SMALL LETTER ESH
      x 222B INTEGRAL
[0342] COMBINING GREEK PERISPOMENI
      ->may have form of either tilde or inverted breve
```

3.6 Combining Marks vs. Non-spacing Marks

It was the original intention in the merger with 10646 to identify the terms non-spacing marks (from Unicode) and combining marks (from 10646). In the final editing of 10646 it became clear that there are some differences between these two terms that must be made clear.

All non-spacing marks are combining marks; however, there are some combining marks that are not non-spacing marks. The difference between the terms is that combining marks include Indic matras as well as non-spacing marks. The Unicode Consortium views this as a defect in the preparation of the document, and is filing a defect report. However, in the meantime, implementors of Indic characters should be aware of some subtle differences in terminology:

1. The derived terms such as *composed character sequence* (Unicode) are not identical with the similar terms such as *composite sequence* (10646). *Composite sequence* also includes any base character followed by any sequence of non-spacing marks and Indic matras (possibly mixed).
2. In 10646, “if a combining character is to be regarded as a composite sequence in its own right, it shall be coded as a composite sequence by association with SPACE.” This implies that an independent Indic matra (without preceding SPACE) is not to be regarded as a composite sequence; that is, a stand-alone Indic matra is not a composite sequence. This is compatible with the Unicode interpretation, which does not require a SPACE before a matra.
3. In 10646, combined marks “are intended to be positioned relative to the preceding base character in some manner.” In practice, this condition cannot be met by implementations of Indic, since the ordering relationships among matras are more complex than for simple non-spacing marks. However, since the standard only indicates the intention, the Unicode interpretation is also compatible with the requirements of 10646.

³ The APL symbols could all have been decomposed; however, they are treated as a closed set of non-extensible operators, and it did not appear worthwhile to provide decompositions at this time.