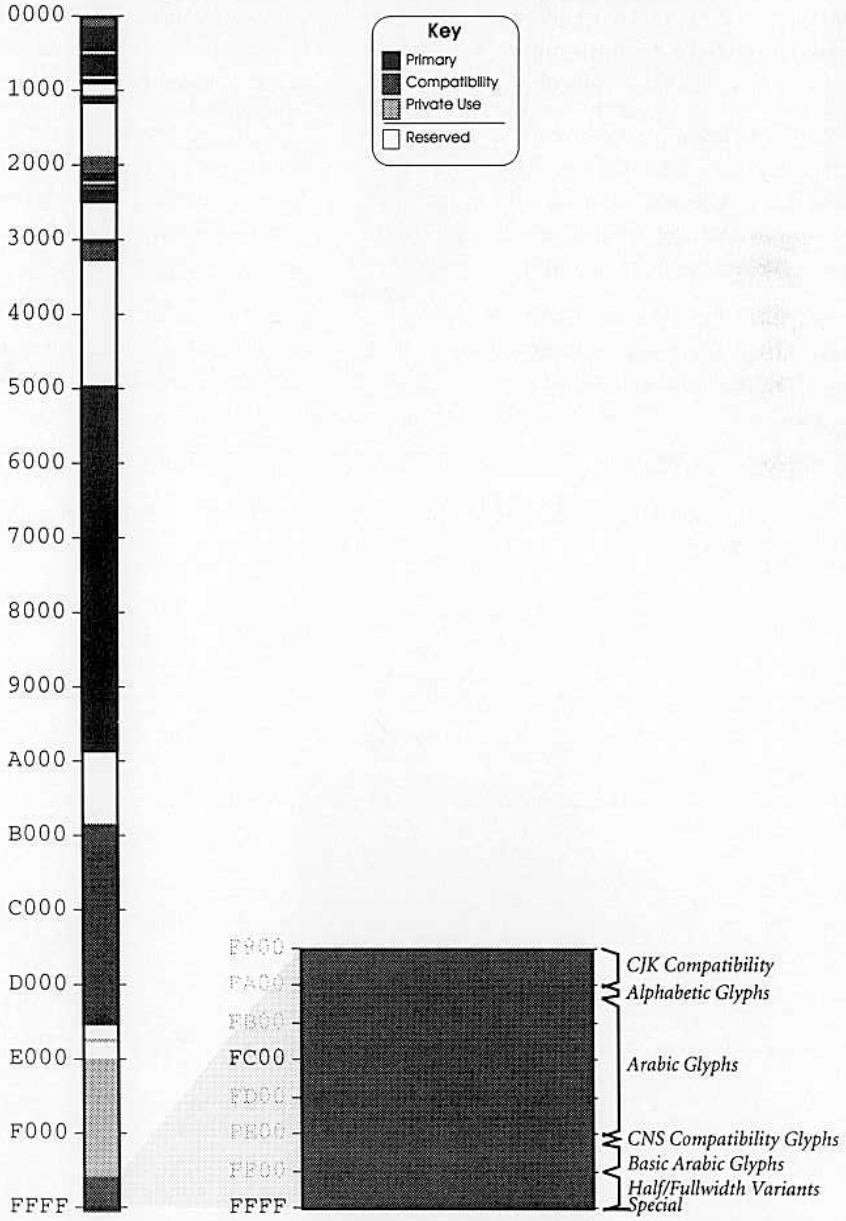
6.8 Compatibility Area and Specials

The Compatibility Area is so named because it contains miscellaneous glyphs, contextual or orientational variants, vertical forms, and width variants that can legitimately be mapped to other characters in the Unicode Standard, but which require specific Unicode values for compatibility with pre-existing character standards. Canonical mappings between Compatibility Area characters and their regular counterparts can be found in the Character Names List.

Compatibility Area characters are provided solely for backwards compatibility to existing standards. The content of the Compatibility Area includes characters that are in use in existing implementations or standards, but whose semantics or usage is fundamentally at odds with the way in which the Unicode Standard generally handles characters. (See Figure 6-33).

Figure 6-33. Compatibility, Specials



Note that not all compatibility characters are actually placed in this area; many characters outside of this area are also compatibility characters. These include superscript and subscript digits, precomposed combinations of Latin and Greek letterforms with one or more combining marks, certain letterlike symbols, non-conjoining Hangul Jamo, Hangul syllables, squared kana words and alphabetic units of measure, Han ideograph typeface (Z-axis) variants introduced by the source separation rule, and many others.

Conversely, the Compatibility Area *does* contain a small number of characters that are not compatibility characters. These include, for example, the Specials Block, U+FEFF ZERO WIDTH NO-BREAK SPACE, U+FB1E HEBREW POINT JUDEO-SPANISH VARIKA, and the ornate parentheses in the Arabic Presentation Forms-A Block.

The Specials block of the Compatibility Area contains code values that are interpreted neither as control nor graphic characters and which are provided to facilitate current software practices.

CJK Compatibility Ideographs: U+F900—U+FAFF

The Korean national standard KS C 5601-1987, which served as one of the primary source sets for the Unified CJK Ideograph Repertoire and Ordering 2.0, contains 268 duplicate encodings of identical ideograph forms in order to denote alternative pronunciations. That is, in certain cases, that standard encoded a single character multiple times in order to denote different linguistic uses. This is like encoding the letter 'a' five times to denote the different pronunciations it has in the words *hat*, *able*, *art*, *father*, *adrift*. Due to the source separation rule, these Korean characters could not be unified as required. Since they are in all ways identical in shape to their nominal counterparts, they are encoded separately from the primary CJK Unified Ideographs block.

In addition, another 34 ideographs that were duplicated in various regional and industry standards were encoded in this block in order to achieve round-trip conversion compatibility.

Encoding Structure. The CJK Compatibility Ideographs block is divided into the following ranges:

U+F900	→	U+FA0B	Pronunciation variants from KS C 5601-1987
U+FA0C	→	U+FA0D	Duplicates from Taiwan BIG5 Character Set
U+FA0E	→	U+FA2D	Duplicates from industry standards.

Alphabetic Presentation Forms: U+FB00—U+FB4F

This block is composed of a number of presentation forms commonly encoded with Latin, Armenian, and Hebrew texts. Each character in this block has a preferred encoding consisting of its components (in the case of ligatures and Hebrew pointed letters) or of its nominal counterpart (in the case of the wide Hebrew variants, U+FB1E HEBREW POINT JUDEO-SPANISH VARIKA, U+FB20 HEBREW LETTER ALTERNATIVE AYIN, and U+FB29 HEBREW LETTER ALTERNATIVE PLUS SIGN).

Encoding Structure. The Alphabetic Presentation Forms block is divided into the following ranges:

U+FB00	→	U+FB06	Latin ligatures
U+FB13	→	U+FB17	Armenian ligatures
U+FB1E			Variant of 05BF HEBREW POINT RAFE
U+FB1F			Hebrew ligature (Yiddish)
U+FB20			Variant of 05E2 HEBREW LETTER AYIN
U+FB21	→	U+FB28	Wide Hebrew letter variants
U+FB29			Hebrew variant of 002B PLUS SIGN
U+FB2A	→	U+FB36,	Hebrew pointed letters
U+FB38	→	U+FB3C,	
U+FB3E,			
U+FB40	→	U+FB41,	
U+FB43	→	U+FB44,	
U+FB46	→	U+FB4E	
U+FB4F			Hebrew ligature

Arabic Presentation Forms-A: U+FB50—U+FDFF

This block contains a list of presentation forms (glyphs) encoded as characters for compatibility. At the time of publication, there are no known implementations of all of these presentation forms. As with all other compatibility encodings, these characters have a preferred encoding that makes use of non-compatibility characters.

The presentation forms in this block consist of contextual (positional) variants of Extended Arabic letters, contextual variants of Arabic letter ligatures, spacing forms of Arabic diacritic combinations, contextual variants of certain Arabic letter/diacritic combinations, and Arabic phrase ligatures. (See also Arabic Presentation Forms-B U+FE70 → U+FEFE.)

The alternate (ornate) forms of parentheses for use with the Arabic script are not compatibility characters.

Encoding Structure. The Arabic Presentation Forms A block is divided into the following ranges:

U+FB50	→	U+FBB1,	Contextual variants of Extended Arabic letters
U+FBD3	→	U+FBE9	
U+FBFC	→	U+FBFF	Contextual variants of Extended Arabic letters (Farsi)
U+FBEA	→	U+FBFB,	Contextual variants of Extended Arabic ligatures
U+FC00	→	U+FC5D,	
U+FC64	→	U+FD3B	
U+FC5E	→	U+FC63	Spacing forms of Arabic diacritic combinations
U+FD3C	→	U+FD3D	Contextual variants of Arabic letter/diacritic combinations
U+FD3E			Ornate form of *left parenthesis*
U+FD3F			Ornate form of *right parenthesis*
U+FD50	→	U+FD8F,	Contextual variants of Arabic ligatures
U+FD92	→	U+FDC7	
U+FDF0	→	U+FDF1	Arabic phrase ligatures (Koranic stops)
U+FDF2	→	U+FDFB	Arabic phrase ligatures

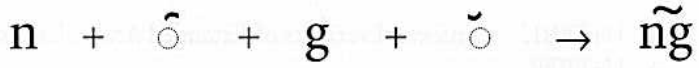
Combining Half Marks: U+FE20—U+FE2F

This block consists of a number of presentation form (glyph) encodings which may be used to visually encode certain combining marks that apply to multiple base letterforms. These characters are intended to facilitate the support of such combining marks in simple implementations.

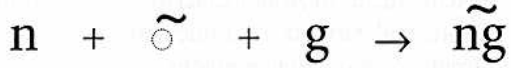
Unlike the other compatibility characters, these characters do not correspond to a single nominal character or a sequence of nominal characters; rather, a discontiguous sequence of these combining half marks corresponds to a single combining mark, as depicted in Figure 6-34.

Figure 6-34. Combining Half-Marks

Combining Half Marks



Single Combining Mark



Encoding Structure. The Combining Half Marks block is divided into the following ranges:

U+FE20	→	U+FE21	Two halves of U+0361 COMBINING DOUBLE INVERTED BREVE
U+FE22	→	U+FE23	Two halves of U+0360 COMBINING DOUBLE TILDE

CJK Compatibility Forms: U+FE30—U+FE4F

A number of presentation forms are encoded in this block which are found in the Republic of China (Taiwan) national standard CNS 11643. These forms are often explicitly encoded when Chinese text is being set in vertical rather than horizontal lines. The preferred Unicode encoding is to encode the nominal characters that correspond to these vertical variants. Then, at display time, the appropriate glyph is selected according to the line orientation.

Encoding Structure. The CJK Compatibility Forms block is divided into the following ranges:

U+FE30 → U+FE4F Vertical punctuation variants from CNS 11643

Small Form Variants: U+FE50—U+FE6F

The Republic of China (Taiwan) national standard CNS 11643 also encodes a number of small variants of ASCII punctuation. The preferred Unicode encoding is to use the corresponding punctuation characters found in the ASCII block and use rich-text mechanisms (for example, font or font style bindings) to select the appropriate size and/or position of the displayed glyphs.

The characters of this block, while construed as fullwidth characters, are nevertheless depicted using small forms that are set in a fullwidth display cell. (See the discussion in the character block description for Halfwidth and Fullwidth Forms.)

Unifications. Two small form variants from CNS 11643/plane 1 were unified with other characters outside the ASCII block: 2131_{16} was unified with U+00B7 MIDDLE DOT, and 2261_{16} was unified with U+2215 DIVISION SLASH.

Encoding Structure. The Small Form Variants block is divided into the following ranges:

U+FE50 → U+FE6B Small punctuation variants from CNS 11643

Arabic Presentation Forms-B: U+FE70—U+FEFF

This block contains additional Arabic presentation forms comprised of spacing or *tatweel* forms of Arabic diacritics, contextual variants of primary Arabic letters, and the obligatory LAM-ALEF ligature. They are included here for compatibility with pre-existing standards and implementations that use these forms as characters. They can be replaced by letters from the Arabic block (U+0600 → U+06FF). Implementations can handle contextual glyph shaping by rendering rules when accessing glyphs from fonts, rather than by encoding contextual shapes as characters.

Spacing and Tatweel Forms of Arabic Diacritics. For compatibility with certain implementations, a set of spacing forms of the Arabic diacritics are provided here. The *tatweel* forms are combinations of the joining connector *tatweel* and a diacritic.

Zero Width No-Break Space. This character (U+FEFF), which is not an Arabic presentation form, is described in the Specials block (U+FFF0 → U+FFFF).

Encoding Structure. The Arabic Presentation Forms B block is divided into the following ranges:

U+FE70	→	U+FE72	Spacing/Tatweel forms of Arabic diacritics
U+FE74			Spacing forms of Arabic diacritic
U+FE76	→	U+FE7F	Spacing/Tatweel forms of Arabic diacritics
U+FE80	→	U+FEF4	Contextual variants of Arabic letters
U+FEF5	→	U+FEFC	Contextual variants of Arabic LAM-ALEF ligature
U+FEFF			*See Specials block description.*

Halfwidth and Fullwidth Forms: U+FF00—U+FFEF

In the context of East Asian coding systems, a double-byte character set (DBCS) such as JIS X 0208-1990 or KS C 5601-1987 is generally used together with a single-byte character set (SBCS), such as ASCII or a variant of ASCII. Text that is encoded with both a DBCS and SBCS is typically displayed such that the glyphs representing DBCS characters occupy two display cells where a display cell is defined in terms of the glyphs used to display the SBCS (ASCII) characters. In these systems, the two-display-cell width is known as the *fullwidth* or *zenkaku* form, while the one-display-cell width is known as the *halfwidth* or *hankaku* form.

Because of this mixture of display widths, certain characters often appear twice, once in fullwidth form in the DBCS repertoire and once in halfwidth form in the SBCS repertoire. In order to achieve round-trip conversion compatibility with such mixed encoding systems, it is necessary to encode both fullwidth and halfwidth forms of certain characters. This block consists of the additional forms needed to support conversion for existing texts which employ both forms.

In the context of conversion to and from such mixed width encodings, all characters in the General Scripts area should be construed as halfwidth (*hankaku*) characters. All characters in the CJK Phonetics and Symbols area and the Unified CJK Ideograph area, along with the characters in the CJK Compatibility Ideographs, CJK Compatibility Forms, and Small Form Variants blocks, should be construed as fullwidth (*zenkaku*) characters. Other Compatibility Area characters outside of the current block should be construed as halfwidth characters. The characters of the Symbols Area are neutral regarding their width semantics.

The characters in this block consist of fullwidth forms of the ASCII block (except SPACE), certain characters of the Latin-1 Supplement, and some currency symbols. In addition, this block contains halfwidth forms of the Katakana and Hangul Compatibility Jamo characters. Finally, a number of characters from the Symbols Area are replicated here (U+FFE8 → U+FFEE) with explicit halfwidth semantics.

As with other compatibility characters, the preferred Unicode encoding is to use the nominal counterparts of these characters and use rich text font or style bindings to select the appropriate glyph size and width.

Unifications. The fullwidth form of U+0020 SPACE is unified with U+3000 IDEOGRAPHIC SPACE.

Encoding Structure. The Halfwidth and Fullwidth Forms block is divided into the following ranges:

U+FF01	→	U+FF5E	Fullwidth ASCII
U+FF61	→	U+FF64	Halfwidth CJK punctuation
U+FF65	→	U+FF9F	Halfwidth Katakana
U+FFA0			Halfwidth Hangul Jamo filler
U+FFA1	→	U+FFDC	Halfwidth Hangul Jamo
U+FFE0	→	U+FFE6	Fullwidth punctuation and currency signs
U+FFE8	→	U+FFEE	Halfwidth forms, arrows, and shapes

Specials: U+FEFF, U+FFF0—U+FFFF

The fourteen Unicode values from U+FFF0 → U+FFFD are reserved for special character definitions. The only special character currently defined is U+FFFD REPLACEMENT CHARACTER, which is the general substitute character in the Unicode Standard. That character can be substituted for any "unknown" character in another encoding which cannot be mapped in terms of known Unicode values (see *Section 5.4, Unknown and Missing Characters*). In addition to these fourteen positions, two code values are specified here for use not as characters but as special signaling devices (described below).

U+FFFE. The 16-bit unsigned hexadecimal value U+FFFE is *not* a Unicode character value. Its occurrence in a stream of Unicode data strongly suggests that the Unicode characters should be byte-swapped before interpretation. U+FFFE should be interpreted only as an incorrectly byte-swapped version of U+FEFF ZERO WIDTH NO-BREAK SPACE, also known as the byte order mark.

Byte Order Mark. The code value U+FEFF *byte order mark* may be used at the beginning of a stream of coded characters to indicate that the characters following are Unicode characters.

The byte order mark is defined to be a signal of correct byte-order polarity. An application may use this signal character to explicitly enable the "big-endian" or "little-endian" byte order to be determined in Unicode text which may exist in either byte order (for example, in networks that mix Intel and Motorola or RISC CPU architectures for data storage). U+FEFF is the correct or legal order; finding a value U+FFFE is a signal that text of the incorrect byte order for an interpreting process has been encountered.

Encoding Form Signature. A character stream starting off with bytes FE and FF is unlikely to be ASCII text. Data streams (or files) that begin with U+FEFF *byte order mark* are likely to contain Unicode values. It is recommended that applications sending or receiving untyped data streams of coded characters use this signature.

The code value FEFF is assigned a signature role in Informative Annex F to ISO 10646. As specified in that annex, the code value FEFF may be used at the beginning of a stream of coded characters to indicate that the characters following are encode in the UCS-2 or UCS-4 representation, as follows:

Unicode encoding (UCS-2) signature:	FEFF
UCS-4 signature:	0000 FEFF

Since UCS-2 in ISO 10646 terminology is equivalent to the Unicode encoding, this convention for discerning between UCS-2 and UCS-4 forms of ISO 10646 is recommended to the attention of implementers of the Unicode Standard.

Zero Width No-Break Space. In addition to the meaning of *byte order mark*, the code value U+FEFF possesses the semantics of ZERO WIDTH NO-BREAK SPACE.

As ZERO WIDTH NO-BREAK SPACE, U+FEFF behaves like U+00A0 NO-BREAK SPACE in that it indicates the absence of word boundaries; however, the former 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 "+."

The characters U+2011 NON-BREAKING HYPHEN and U+00A0 NO-BREAK SPACE may also be expressed by using their counterparts U+2010 HYPHEN and U+0020 SPACE bracketed by ZERO WIDTH NO-BREAK SPACES. The ZERO WIDTH NO-BREAK SPACE can also be used in this manner to prevent line-breaking with other characters that do not have non-breaking variants, such as U+2009 THIN SPACE or U+2015 HORIZONTAL BAR.

This character has the opposite function from the U+200B ZERO WIDTH SPACE. The latter indicates a word boundary, except that it has no width. It can be used to indicate word boundaries in scripts such as Thai which do not use visible spaces to separate words.

The ZERO WIDTH NO-BREAK SPACE is not to be confused with U+200C ZERO WIDTH NON-JOINER. U+200C ZERO WIDTH NON-JOINER and U+200D ZERO WIDTH JOINER have no effect on word boundaries, while ZERO WIDTH NO-BREAK SPACE and ZERO WIDTH SPACE have no effect on joining or linking behavior. In other words, the ZERO WIDTH NO-BREAK SPACE and the ZERO WIDTH SPACE should be ignored when determining cursive joining behavior; the ZERO WIDTH NON-JOINER and ZERO WIDTH JOINER should be ignored when determining word boundaries. (For more discussion see the General Punctuation character block description.)

U+FFFF. The 16-bit unsigned hexadecimal value U+FFFF is *not* a Unicode character value; it may be used by an application as a error code or other non-character value. The specific interpretation of U+FFFF is not defined by the Unicode Standard, so it can be viewed as a kind of private-use non-character.

Encoding Structure. The Specials block is divided into the following ranges:

U+FEFF	ZERO WIDTH NON-BREAK SPACE (and byte order mark)
U+FFFD	Replacement character
U+FFFE	Not a character; byte swap required signal value
U+FFFF	Not a character