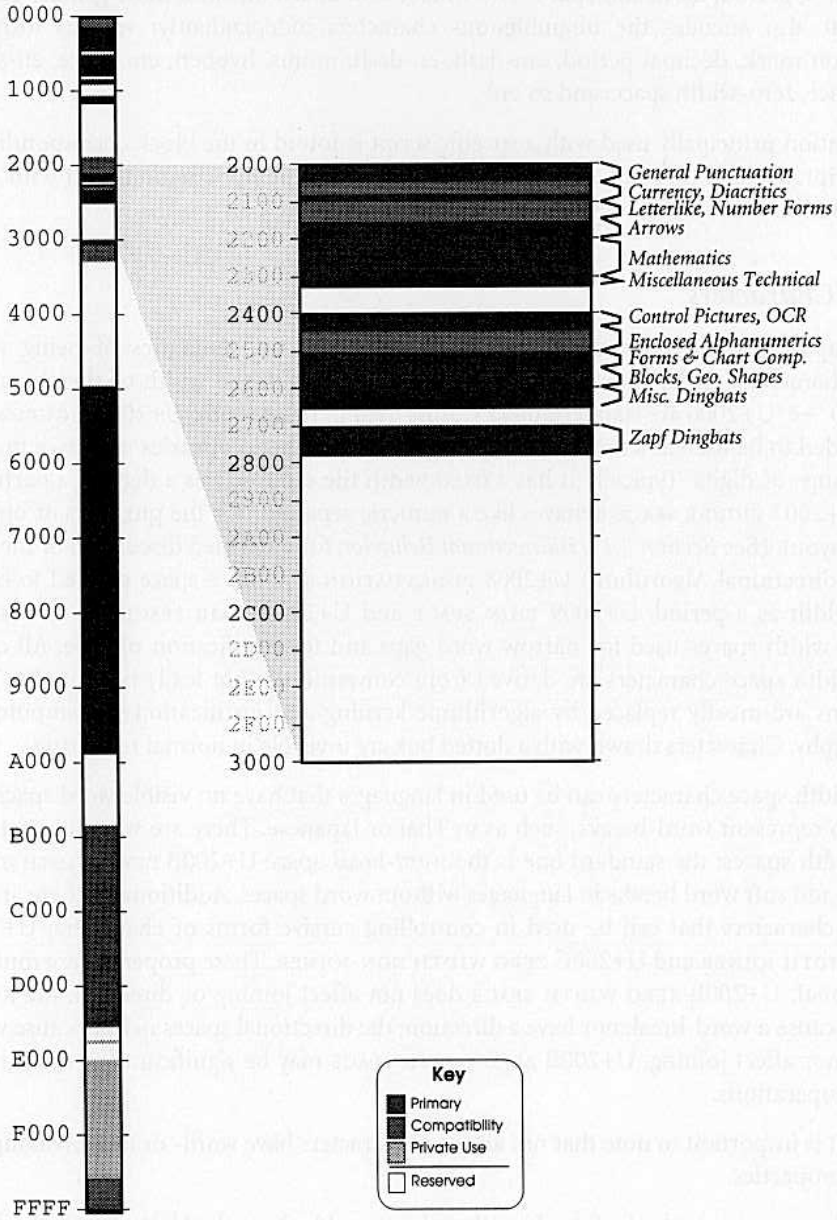
6.2 Symbols Area

The Symbols Area of the Unicode Standard includes the encoding of symbolic characters, including punctuation, numbers, pictures for control codes, and dingbats (see Figure 6-20).

Figure 6-20. Symbols



General Punctuation: U+2000—U+206F

General Punctuation combines punctuation characters and character-like elements used to achieve certain text layout effects. Some punctuation characters can be used with many different scripts. Many general punctuation characters can also be found in the Basic Latin (ASCII) and Latin-1 Supplement blocks.

In many cases, current standards include generic characters for punctuation instead of the more precisely specified characters used in printing. Examples include the single and double quotes, period, dash, and space. The Unicode Standard includes these generic characters, but also encodes the unambiguous characters independently: various forms of quotation mark, decimal period, em-dash, en-dash, minus, hyphen, em-space, en-space, hair-space, zero-width space, and so on.

Punctuation principally used with a specific script is found in the block corresponding to that script, such as U+061B ARABIC SEMICOLON "؛" or the punctuation used with ideographs in the CJK Symbols block.

Space Characters

Typographical Space Characters. Spaces generally have the semantics of being word-break characters. Other than that, the main difference is in the width of the characters. U+2000 → U+2006 are standard quad widths used in typography. U+2007 FIGURE SPACE is intended to be used as a thousands separator in cases where countries use space to separate groups of digits. Typically it has a fixed width the same size as a digit in a particular font. U+2007 FIGURE SPACE behaves like a numeric separator for the purposes of bidirectional layout. (See *Section 3.11, Bidirectional Behavior,* for a detailed discussion of the Unicode Bidirectional Algorithm.) U+2008 PUNCTUATION SPACE is a space defined to be the same width as a period. U+2009 THIN SPACE and U+200A HAIR SPACE are successively smaller-width spaces used for narrow word gaps and for justification of type. All of the fixed-width space characters are derived from conventional (hot lead) typography. Their functions are mostly replaced by algorithmic kerning and justification in computerized typography. Characters drawn with a dotted box are invisible in normal rendering.

Zero-width space characters can be used in languages that have no visible word spacing in order to represent word-breaks, such as in Thai or Japanese. There are several varieties of zero-width spaces; the standard one is the *word-break space* U+200B ZERO WIDTH SPACE, used to add soft word breaks in languages without word spaces. Additionally, there are two format characters that can be used in controlling cursive forms of characters, U+200D ZERO WIDTH JOINER and U+200C ZERO WIDTH NON-JOINER. These properties are mutually orthogonal: U+200B ZERO WIDTH SPACE does not affect joining or direction; the joiners neither cause a word-break nor have a direction; the directional spaces neither cause word-breaks nor affect joining. U+200B ZERO WIDTH SPACE may be significant for searching or sorting operations.

➥ It is important to note that not all space characters have word- or line-breaking properties.

Space characters may also be found in other character blocks in the Unicode Standard. The complete list of space characters, including the joiners and directional spaces, appears in Table 6-18.

Other Punctuation

Dashes and Hyphen. U+2010 HYPHEN represents the hyphen as found in words such as "left-to-right." It differs from U+002D HYPHEN-MINUS in that the latter has an ambiguous

Table 6-18. Unicode Space Characters

Code	Name
U+0020	SPACE
U+00A0	NO-BREAK SPACE
U+2000	EN QUAD
U+2001	EM QUAD
U+2002	EN SPACE
U+2003	EM SPACE
U+2004	THREE-PER-EM SPACE
U+2005	FOUR-PER-EM SPACE
U+2006	SIX-PER-EM SPACE
U+2007	FIGURE SPACE
U+2008	PUNCTUATION SPACE
U+2009	THIN SPACE
U+200A	HAIR SPACE
U+200B	ZERO WIDTH SPACE
U+200C	ZERO WIDTH NON-JOINER
U+200D	ZERO WIDTH JOINER
U+200E	LEFT-TO-RIGHT MARK
U+200F	RIGHT-TO-LEFT MARK
U+3000	IDEOGRAPHIC SPACE
U+FEFF	ZERO WIDTH NO-BREAK SPACE

semantic value. U+2011 NON-BREAKING HYPHEN is present for compatibility with existing standards. U+2011 NON-BREAKING HYPHEN has the same semantic as U+2010 HYPHEN, but should not be broken across lines.

U+2012 FIGURE DASH is present for compatibility with existing standards; it has the same (ambiguous) semantic as the U+002D HYPHEN-MINUS, but has the same width as digits (if they are monospaced). U+2013 EN DASH is used to indicate a range of values, such as 1973–1984. It should be distinguished from the U+2122 MINUS, which is an arithmetic operator; however, typographers have typically used U+2013 EN DASH in typesetting to represent the *minus sign*. For general compatibility in interpreting formulas, U+002D HYPHEN-MINUS, U+2012 FIGURE DASH, and U+2212 MINUS SIGN should each be taken as indicating a *minus sign*, as in "x = a - b."

U+2014 EM DASH is used to make a break—like this—in the flow of a sentence. It is commonly represented with a typewriter as a double-hyphen. In older mathematical typography, U+2014 EM DASH is also used to indicate a *binary minus sign*. U+2015 HORIZONTAL BAR is used to introduce quoted text in some typographic styles.

Dashes and hyphen characters may also be found in other character blocks in the Unicode Standard. The complete list of dash and hyphen characters appears in Table 6-19.

Table 6-19. Unicode Dash Characters

Code	Name
U+002D	HYPHEN-MINUS
U+00AD	SOFT HYPHEN
U+2010	HYPHEN
U+2011	NON-BREAKING HYPHEN
U+2012	FIGURE DASH
U+2013	EN DASH
U+2014	EM DASH
U+2015	HORIZONTAL BAR (= *quotation dash*)
U+207B	SUPERSCRIPT MINUS
U+208B	SUBSCRIPT MINUS
U+2212	MINUS SIGN
U+301C	WAVE DASH
U+3030	WAVY DASH

Quotation Marks. U+201A SINGLE LOW-9 QUOTATION MARK, U+201E DOUBLE LOW-9 QUOTATION MARK, U+2039 SINGLE LEFT-POINTING ANGLE QUOTATION MARK, and U+203A SINGLE RIGHT-POINTING ANGLE QUOTATION MARK have heterogeneous semantics. They may represent opening or closing quotation marks depending on usage.

Hyphenation Point. U+2027 HYPHENATION POINT is a raised dot used to indicate correct word breaking as in dic·tion·ar·ies. This is a punctuation mark, to be distinguished from U+00B7 MIDDLE DOT, which has multiple semantics.

Fraction Slash. U+2044 FRACTION SLASH is used between digits to form numeric fractions such as 2/3, 3/9, and so on. The standard form of a fraction built using the fraction slash is defined as follows: Any sequence of one or more decimal digits, followed by the fraction slash, followed by any sequence of one or more decimal digits. Such a fraction should be displayed as a unit, such as ¾ or as $\frac{3}{4}$. The precise choice of display can depend upon additional formatting information.

If the displaying software is incapable of mapping the fraction to a unit, then it can also be displayed as a simple linear sequence as a fall-back (for example, 3/4). If the fraction is to be separated from a previous number, then a space can be used, choosing the appropriate width (normal, thin, zero-width, and so on). For example, 1 + ZERO-WIDTH SPACE + 3 + FRACTION SLASH + 4 displays as 1¾.

Spacing Overscore. U+203E OVERLINE is the above-the-line counterpart to U+005F LOW LINE. It is a spacing character, not to be confused with U+0305 COMBINING OVERLINE. As with all over- or underscores, a sequence of these characters should connect in an unbroken line. The overscoring characters also must be distinguished from U+0304 COMBINING MACRON, which does not connect horizontally in this way.

Layout Controls

The Non-Joiner and Joiner. In some fonts for some scripts, consecutive characters in a text stream may be rendered via adjacent glyphs that cursively join to each other, so as to emulate connected handwriting. For example, cursive joining is implemented in nearly all fonts for the Arabic scripts, and in a few handwriting-like fonts for the Latin script.

Cursive rendering is implemented by joining glyphs in the font, plus a process that selects the particular joining glyph to represent each individual character occurrence, based on the joining nature of its neighboring characters. This glyph selection is implemented in some combination between the rendering engine and the font itself.

In cases where cursive joining is implemented, there are occasions when an author may wish to override the normal automatic selection of joining glyphs. Typically, this is done to achieve one of the following effects:

- Cause non-default joining appearance (for example, as is sometimes required in writing Persian using the Arabic script).
- Exhibit the joining-variant glyphs themselves in isolation.

The Unicode Standard provides a means to influence joining glyph selection, by means of the two characters U+200C *zero width non-joiner* and U+200D *zero width joiner*. Logically, these characters do not modify the contextual selection process itself, but rather they *change the context* of a particular character occurrence. By providing a non-joining neighbor character where otherwise the neighbor would be joining, or vice-versa, they deceive the rendering process into selecting a different joining glyph.

(1) Prevent joining appearance. For example,

ص U+0635 ARABIC LETTER SAD

ZW NJ U+200C ZERO WIDTH NON-JOINER

ل U+0644 ARABIC LETTER LAM

would be rendered as صل (that is, the normal cursive joining of the interior *sad* and *lam* is overridden). Without the ZERO WIDTH NON-JOINER it would be rendered as ص ل.

(2) Exhibit joining glyphs in isolation. For example,

ZW J U+200D ZERO WIDTH JOINER

غ U+063A ARABIC LETTER GHAIN

ZW J U+200D ZERO WIDTH JOINER

would be rendered as غ. (That is, the medial glyph form of the *ghain* appears in isolation. Without the ZERO WIDTH JOINER before and after, it would be rendered as غ.)

➥ The preceding examples are adapted from the Iranian national coded character set standard, ISIRI 3342, which defines these characters as "pseudo space" and "pseudo connection," respectively.

The function of the ZERO WIDTH JOINER may also have a particular interpretation in specific scripts. For example, in Indic scripts it provides an invisible neighbor to which a dead consonant may join in order to induce a half-consonant form (see the Devanagari character block description). It also has specially defined usage in Tibetan (see the Tibetan character block description).

ZERO WIDTH NON-JOINER or ZERO WIDTH JOINER are format control characters. As with other such characters, they should be ignored by processes that analyze text content. For example, a spell-checker or find/replace operation should filter them out. (See *Section 2.4, Special Character and Non-Character Values,* for a general discussion of format control characters.)

The effect of these characters in display depends on the context in which they are found. Adding a ZERO WIDTH JOINER between characters that are already cursively connected will have no effect. Adding a ZERO WIDTH NON-JOINER between characters that are unconnected will also have no effect. For example, any number of ZERO-WIDTH NON-JOINER or ZERO-WIDTH JOINER characters sprinkled into an English text stream will have no effect on its appearance when rendered in a typical non-cursive Latin font.

Although ZERO WIDTH JOINER and ZERO-WIDTH NON-JOINER should not affect ligating behavior, in some systems they may break up ligatures by interrupting the character sequence required to form the ligature. As illustration, a cursive Latin font would produce the following results:

Memory Representation	Rendering	
f i s h	f- -i- -s- -h	
	fi- -s- -h	optionally using a ligature
f ZW J i s h	f- -i- -s- -h	
	fi- -s- -h	optionally using a ligature
f ZW NJ i s h	f -i- -s- -h	
f ZW J ZW NJ i s h	f- i- -s- -h	
f ZW NJ ZW J i s h	f -i- -s- -h	

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.

Bidirectional Ordering Codes. These codes are used in the Bidirectional Algorithm, described in *Chapter 3, Conformance.* Systems that handle bidirectional scripts (Arabic and Hebrew) should be sensitive to these codes. The codes appear in Table 6-20.

Table 6-20. Bidirectional Ordering Codes

Code	Name	Abbrev.
U+200E	LEFT-TO-RIGHT MARK	(LRM)
U+200F	RIGHT-TO-LEFT MARK	(RLM)
U+202A	LEFT-TO-RIGHT EMBEDDING	(LRE)
U+202B	RIGHT-TO-LEFT EMBEDDING	(RLE)
U+202C	POP DIRECTIONAL FORMATTING	(PDF)
U+202D	LEFT-TO-RIGHT OVERRIDE	(RLO)
U+202E	RIGHT-TO-LEFT OVERRIDE	(LRO)

As with the other zero-width character codes, except for their effect on the layout of the text in which they are contained, the bidirectional ordering characters can be ignored by the processing software. For non-layout text processing, such as sorting, searching and so on, the zero-width layout characters may be ignored. However, when modifying text, care should be taken to maintain these characters correctly, since the matching pairs of zero-width formatting characters must be coordinated (see *Chapter 3, Conformance*).

U+200E LEFT-TO-RIGHT MARK and U+200F RIGHT-TO-LEFT MARK have the semantics of a an invisible character of zero width, except that while a normal space is directionally neutral, these characters have strong directionality. These characters are intended to be used to resolve cases of ambiguous directionality in the context of bidirectional texts. Unlike U+200B ZERO WIDTH SPACE, these characters carry no word break semantics. (See *Section 3.11, Bidirectional Behavior,* for more information.)

Line and Paragraph Separator. For historical reasons, carriage-return and line-feed are not used consistently across different systems. The Unicode Standard provides (and encourages use of) the *line* and *paragraph separator* characters to provide clear information about where line and paragraph boundaries occur. This is also required for use with the Bidirectional Algorithm (see *Chapter 3, Conformance*).

A paragraph separator indicates where a new paragraph should start. This could cause, for example, the line to be broken, the interparagraph line spacing to be applied, and indentation of the first line. A line separator indicates that a line-break should occur at this point; although the text continues on the next line, it does not start a new paragraph: no interparagraph line spacing nor paragraphic indentation is applied. Since these are separator codes, it is not necessary to start the first line or paragraph, nor end the last line or paragraph with them.

Alternate Format Characters

There are three pairs of alternate format characters encoded in this block:

- Symmetric swapping format characters used to control the glyphs which depict characters such as "(". (The default state is *activated.*)
- Character shaping selectors used to control the shaping behavior of the Arabic compatibility characters. (The default state is *inhibited.*)

- Numeric shape selectors codes used to override the normal shapes of the Western Digits. (The default state is *nominal.*)

The use of these character shaping selectors and digit shapes codes is *strongly* discouraged in the Unicode Standard. Instead, the appropriate character codes should be used with the default state. For example, if contextual forms for Arabic characters are desired, then the nominal characters should be used, and not the presentation forms with the shaping selectors. Similarly, if the Arabic digit forms are desired then the explicit characters should be used, such as U+0660 ARABIC-INDIC DIGIT ZERO.

Symmetric Swapping. 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 *Section 4.7, Mirrored.* They indicate whether the interpretation of the term LEFT or RIGHT in the character names should be interpreted as meaning *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 6429. In the absence of such a protocol, the default state is *activated.*

From the point of encountering U+206A INHIBIT SYMMETRIC SWAPPING format character up to a subsequent U+206B ACTIVATE SYMMETRIC SWAPPING (if any), the symmetric characters will be interpreted and rendered as left and right.

From the point of encountering U+206B ACTIVATE SYMMETRIC SWAPPING format character up to a subsequent U+206A INHIBIT SYMMETRIC SWAPPING (if any), the symmetric characters will be interpreted and rendered as opening and closing. This state (*activated*) is the default state in the absence of any symmetric swapping code or a higher level protocol.

Character Shaping Selectors. The character shaping selector format characters are used in conjunction with Arabic presentation forms. During the presentation process, certain letterforms may be joined together in cursive connection or ligatures. The shaping selector codes indicate that the character shape determination (glyph selection) process used to achieve this presentation effect is to be either activated or inhibited. The shaping selector codes do not nest.

From the point of encountering a U+206C INHIBIT ARABIC FORM SHAPING format character up to a subsequent U+206D ACTIVATE ARABIC FORM SHAPING (if any), the character shaping determination process should be inhibited. If the backing store contains Arabic presentation forms (for example, U+FE80 → U+FEFC), then these forms should be presented without shape modification. This state (*inhibited*) is the default state in the absence of any character shaping selector or a higher level protocol.

From the point of encountering a U+206D ACTIVATE ARABIC FORM SHAPING format character up to a subsequent U+206C INHIBIT ARABIC FORM SHAPING (if any), any Arabic presentation forms which appear in the backing store should be presented with shape modification by means of the character shaping (glyph selection) process.

The shaping selectors have no effect on nominal Arabic characters (U+0660 → U+06FF) which are always subject to character shaping (glyph selection) and which are unaffected by these formatting codes.

Numeric Shape Selectors. The numeric shape selector format characters allow the selection of the shapes in which the digits U+0030 → U+0039 are to be rendered. These format characters do not nest.

From the point of encountering a U+206E NATIONAL DIGIT SHAPES format character up to a subsequent U+206F NOMINAL DIGIT SHAPES (if any), the European digits (U+0030 → U+0039) should be depicted using 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 (U+0660 → U+0669). The actual character shapes (glyphs) used to display national digit shapes is not specified by the Unicode Standard.

From the point of encountering a U+206F NOMINAL DIGIT SHAPES format character up to a subsequent U+206E NATIONAL DIGIT SHAPES (if any), the European digits (U+0030 → U+0039) should be depicted using glyphs that represent the nominal digit shapes shown in the code tables for these digits. This state (*nominal*) is the default state in the absence of any numeric shape selector or a higher level protocol.

Encoding Structure. The character block for General Punctuation is divided into the following ranges:

U+2000	→	U+200A	Typographical space characters
U+200B			Zero-width space
U+200C	→	U+200F	Zero-width layout characters
U+2010	→	U+2027	Printing punctuation characters
U+2028	→	U+2029	Line and paragraph separators
U+202A	→	U+202E	Bidirectional ordering codes
U+2030	→	U+2046	Printing punctuation characters
U+206A	→	U+206F	Alternate format characters (usage strongly discouraged)

Superscripts and Subscripts: U+2070—U+209F

Superscripts and subscripts have been included in the Unicode Standard only to provide compatibility with existing character sets. In general, the Unicode character encoding does not attempt to describe the positioning of a character above or below the baseline in typographical layout. The superscript digits one, two, and three are coded in the Latin-1 Supplement block in order to obtain code point compatibility with ISO 8859-1.

Standards. The characters in this block are from sets registered with ECMA under ISO 2374 for use with ISO 2022.

Encoding Structure. The character block for Superscripts and Subscripts is divided into the following ranges:

U+2070	→	U+2079	Superscript numbers (gapped where the superscript is encoded in the Latin-1 Supplement block)
U+207A	→	U+207F	Superscript mathematical operators, punctuation, and letter
U+2080	→	U+2089	Subscript numbers
U+208A	→	U+208E	Subscript mathematical operators and punctuation

Currency Symbols: U+20A0—U+20CF

This block contains currency symbols not encoded in other blocks. Where the Unicode Standard follows the layout of an existing standard, such as for the ASCII, Latin 1, and Thai blocks, the currency symbols are encoded in those blocks, rather than here.

Unification. The Unicode Standard does not duplicate encodings where more than one currency is expressed with the same symbol. Many currency symbols are overstruck letters. There are therefore many minor variants, such as the U+0024 DOLLAR SIGN $, with one or two vertical bars, or other graphical variation. The Unicode Standard considers these variants to be typographical and provides a single encoding.

Claims that glyph variants of a certain currency symbol are used consistently to indicate a particular currency could not be substantiated upon further research. (See ISO/IEC DIS 10367, Annex B (informative) for an example of multiple renderings for U+00A3 POUND SIGN.)

The following table lists common currency symbols encoded in other blocks.

Dollar, milreis, escudo	U+0024	DOLLAR SIGN
Cent	U+00A2	CENT SIGN
Pound	U+00A3	POUND SIGN
General currency	U+00A4	CURRENCY SIGN
Yen or yuan	U+00A5	YEN SIGN
Dutch florin	U+0192	LATIN SMALL LETTER F WITH HOOK
Baht	U+0E3F	THAI CURRENCY SYMBOL BAHT

Encoding Structure. The character block for currency symbols is divided into the following ranges:

U+20A0 → U+20AB Currency symbols

Combining Marks for Symbols: U+20D0—U+20FF

Diacritical marks for symbols are generally applied to mathematical or technical symbols. These can be used to extend the range of the symbol set. For example, U+20D3 COMBINING SHORT VERTICAL LINE OVERLAY can be used to express negation. Its presentation may change in those circumstances, changing length or slant. That is, U+2261 IDENTICAL TO, followed by U+20D3 is equivalent to U+2262 NOT IDENTICAL TO. In this case, there was a precomposed form for the negated symbol. However, this is not always true, and U+20D3 can be used with other symbols to form the negation. For example, U+2258 CORRESPONDS TO followed by U+20D3 can be used to express *does not correspond to*, without requiring that a precomposed form be part of the Unicode Standard.

Other non-spacing characters can also be used in mathematical expressions, of course. For example, a U+0304 COMBINING MACRON is commonly used in propositional logic to indicate logical negation.

Enclosing Diacritics. These non-spacing characters are supplied for compatibility with existing standards, allowing individual base characters to be enclosed in several ways. For example, U+2460 CIRCLED DIGIT ONE ① can be expressed as U+0030 DIGIT ONE "1" + U+20DD COMBINING ENCLOSING CIRCLE ⃝. As with other combining characters, this one can also be applied productively; *circled letter alef* can be produced by the sequence: U+05D0 HEBREW LETTER ALEF א + U+20DD COMBINING ENCLOSING CIRCLE ⃝. The combining enclosing diacritics cannot be used to enclose a sequence of base characters in plain text. For example, there is no way to represent U+246A CIRCLED NUMBER ELEVEN with the ENCLOSING CIRCLE, since there is no single character NUMBER ELEVEN.

Encoding Structure. The character block for Combining Marks for Symbols is divided into the following ranges:

U+20D0 → U+20E1 Symbol Diacritics

Letterlike Symbols: U+2100—U+214F

Letterlike symbols are symbols derived in some way from ordinary letters of an alphabetic script. This block includes symbols based on Latin, Greek, and Hebrew letters. These symbols are encoded for compatibility. In general, the usage of distinct codes for letterlike symbols that are merely font variants or alternative representations of other characters is strongly discouraged. When using letters as symbols in equations and formulae, as well as in other contexts, use normal alphabetic forms in the appropriate styles. For example, to represent degrees Celsius "°C", use a sequence of U+00B0 DEGREE SIGN + U+0043 LATIN CAPITAL LETTER C, rather than U+2103 DEGREE CELSIUS. For searching, treat these two sequences as identical.

Where the letterlike symbols have alphabetic equivalents, they collate in alphabetic sequence; otherwise, they should be treated as neutral symbols. The letterlike symbols may have different directional properties than normal letters; for example, the four transfinite cardinal symbols (U+2135 → U+2138) are used in ordinary mathematical text and do not share the strong right-to-left directionality of the Hebrew letters they are derived from.

Styles. The letterlike symbols constitute one of the few instances in which the Unicode Standard encodes stylistic variants of letters as distinct characters. For example, there are instances of black letter, double-struck, and script styles for certain Latin letters used as mathematical symbols. The choice of these stylistic variants for encoding reflects their common use as distinct symbols. It is recognized that a particular style can be applied to any Latin letter with a resulting semantic distinction in mathematical or logical text; applications that require such systematic stylistic semantics should achieve them by using styles directly, rather than by seeking to extend the character-by-character encoding of such variants in the Unicode Standard.

The black-letter style is often referred to as *Fraktur* or *Gothic* in various sources. Technically, Fraktur and Gothic typefaces are distinct designs from black letter, but no encoding distinctions are implied in the various symbol sources. The Unicode Standard simply uses black letter forms as the archetypes.

A similar consideration applies to the double-struck style. This style is not literally double-struck, but is instead an open outline design that gives the visual appearance of being struck twice with a horizontal shift. For encoding purposes this style can be considered equivalent to letterlike symbols rendered in outlined or shadowed typefaces to carry conventional semantic distinctions.

Standards. The Unicode Standard encodes letterlike symbols from many different national standards and corporate collections.

Encoding Structure. The character block for Letterlike Symbols is divided into the following ranges:

U+2100 → U+2138 Letterlike symbols

Number Forms: U+2150—U+218F

Number Form characters are encoded solely for compatibility with existing standards. The same considerations with respect to compatibility apply as noted in the discussion of Letterlike Symbols.

Fractions. The vulgar fraction characters encoded in this block can be equivalently represented using U+2044 FRACTION SLASH.

Roman Numerals. The Roman numerals can be composed of sequences of the appropriate Latin letters. U+2180 ROMAN NUMERAL ONE THOUSAND C D and U+216F ROMAN NUMERAL ONE THOUSAND are actually variants of the same glyph but are distinguished because of existing standards; similarly, the upper- and lowercase variants of Roman numerals have been separately encoded. U+2181 ROMAN NUMERAL FIVE THOUSAND, and U+2182 ROMAN NUMERAL TEN THOUSAND are distinct characters used in Roman Numerals, since they represent characters used in Roman numerals that do not have decompositions in the Unicode Standard.

Encoding Structure. The character block for Number Forms is divided into the following ranges:

U+2153	→	U+215F	Vulgar fractions
U+2160	→	U+2182	Roman numerals and small Roman numerals

Arrows: U+2190—U+21FF

Arrows are used for a variety of purposes: to imply directional relation, logical derivation or implication, or to represent the cursor control keys.

The Unicode Standard attempts to provide fairly complete encodings for generic arrow shapes, especially where there are established usages with well-defined semantics; the Unicode Standard does not attempt to encode every possible stylistic variant of arrows separately, especially where their use is mainly decorative. For most arrow variants, the Unicode Standard provides encodings in the two horizontal directions, often in the four cardinal directions. For the single and double arrows the Unicode Standard provides encodings in eight directions.

Standards. The Unicode Standard encodes arrows from many different national standards and corporate collections.

Unifications. Arrows expressing mathematical relations have been encoded in the arrows block. An example is U+21D2 RIGHTWARDS DOUBLE ARROW ⇒ may be the equivalent of *implies.*

Long and short arrow forms encoded in glyph standards or typesetting systems such as TEX are not represented by separate Unicode values.

Encoding Principles. Because the arrows have such a wide variety of applications, there may be several semantic values for the same Unicode character value: for example, U+21B5 DOWNWARDS ARROW WITH CORNER LEFTWARDS ↵ may be the equivalent of *carriage return*; U+2191 UPWARDS ARROW ↑ may be the equivalent of *increases* or *exponent.*

Encoding Structure. The character block for arrows is divided into the following ranges:

U+2190 → U+21EA Arrows

Mathematical Operators: U+2200—U+22FF

The Mathematical Operators block includes character encodings for operators, relations, geometric symbols, and a few other symbols with special usages confined largely to mathematical contexts.

In addition to the characters in this block, mathematical operators are also found in the Basic Latin (ASCII) and Latin-1 Supplement blocks. A few of the symbols from the Miscellaneous Technical block, and characters from General Punctuation are also used in mathematical notation. Latin letters in special font styles and used as mathematical operators, such as U+2118 SCRIPT CAPITAL P ℘, as well as the Hebrew letter *alef* used as the operator first transfinite cardinal encoded by U+2135 ALEF SYMBOL ℵ, are encoded in the block for letterlike symbols.

Standards. Many national standards' mathematical operators are covered by the characters encoded in this block. These standards include such special collections as ANSI Y10.20, ISO DIS 6862.2, ISO 8879, and the collection of the American Mathematical Society, as well as the original repertoire of TEX.

Encoding Principles. Mathematical operators often have more than one meaning. Therefore the encoding of this block is intentionally rather shape-based, with numerous instances in which several semantic values can be attributed to the same Unicode value. For example, U+2218 RING OPERATOR ∘ may be the equivalent of *white small circle* or *composite function* or *apl jot*. The Unicode Standard does not attempt to distinguish all the possible semantic values which may be applied to mathematical operators or relation symbols.

On the other hand, mathematical operators, and especially relation symbols, may appear in various standards, handbooks, and fonts with a large number of purely graphical variants. Where variants were recognizable as such from the sources, they were not encoded separately.

Unifications. Mathematical operators such as *implies* ⇒ and *if and only if* ↔ have been unified with the corresponding arrows (U+21D2 RIGHTWARDS DOUBLE ARROW and U+2194 LEFT RIGHT ARROW, respectively) in the Arrows block.

The operator U+2208 ELEMENT OF is occasionally rendered with a taller shape than shown in the code charts. Mathematical handbooks and standards consulted treat these as variants of the same glyph. U+220A SMALL ELEMENT OF is separately encoded, because some existing standards distinguish it from U+2208.

The operators U+226B MUCH GREATER-THAN and U+226A MUCH LESS-THAN are sometimes rendered in a nested shape. Since no semantic distinction applies, the Unicode Standard provides a single encoding for each of these operators.

A large class of unifications applies to variants of relation symbols involving equality, similarity, and/or negation. Variants involving one- or two-barred *equal signs*, one- or two-*tilde similarity signs*, and vertical or slanted *negation slashes* and *negation slashes* of different lengths are not separately encoded. Thus, for example, U+2288 NEITHER A SUBSET OF NOR EQUAL TO, is the archetype for at least six different glyph variants noted in various collections.

There are two instances in which essentially stylistic variants are separately encoded: U+2265 GREATER-THAN OR EQUAL TO is distinguished from U+2267 GREATER-THAN OVER EQUAL TO; the same distinction applies to U+2264 LESS-THAN OR EQUAL TO and U+2266 LESS-THAN OVER EQUAL TO. This exception to the general rule regarding variation results from requirements for character mapping to some Asian standards which distinguish the two forms.

Greek-Derived Symbols. Several mathematical operators derived from Greek characters have been given separate encodings to match usage in existing standards. These operators may occasionally occur in context with Greek-letter variables. These operators include U+2206 INCREMENT Δ, U+220F N-ARY PRODUCT ∏, and U+2211 N-ARY SUMMATION Σ.

Other duplicated Greek characters are those for U+00B5 MICRO SIGN µ in the Latin-1 Supplement block, U+2126 OHM SIGN Ω in Letterlike symbols, and several characters among the APL functional symbols in the Miscellaneous Technical block. All other Greek characters with special mathematical semantics are found in the Greek block since duplicates were not required for compatibility.

Miscellaneous Symbols. U+2212 MINUS SIGN − is a mathematical operator, to be distinguished from the ASCII-derived U+002D HYPHEN-MINUS -, which may look the same as minus sign, or may be shorter in length. (For a complete list of dashes in the Unicode Standard, see the General Punctuation character block description.) U+22EE → U+22F1 are a set of ellipses used in matrix notation.

Math Property. A list of characters with the math property are listed in *Chapter 4, Character Properties.*

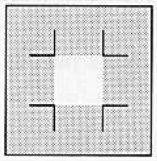
Encoding Structure. The character block for Mathematical Operators is divided into the following ranges:

U+2200 → U+22F1 Mathematical operators

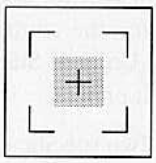
Miscellaneous Technical: U+2300—U+23FF

This block encodes technical symbols including keytop labels such as U+232B ERASE TO THE LEFT. Excluded from consideration were symbols that are not normally used in one-dimensional text but are intended for two-dimensional diagrammatic use, such as symbols for electronic circuits. An unusually large expansion space is provided since it is anticipated that there are a large number of technical symbols that could be considered for addition to the Unicode Standard.

Crops and Quine Corners. Crops and quine corners are most properly used in two-dimensional layout but may be referred to in plain text. The usage of crops and quine corners is as indicated in this diagram:



Use of crops



Use of quine corners

APL Functional Symbols. APL (A Programming Language) makes extensive use of functional symbols constructed by composition with other, more primitive functional symbols. It made extensive use of backspace and overstrike mechanisms in early computer implementations. While, in principle, functional composition is productive in APL, in practice, a relatively small set of composed functional symbols have become standard operators in APL. This relatively small set is encoded in entirety in this block. All other APL extensions can be encoded by composition of other Unicode characters. For example, the APL symbol *a underbar* can be represented by U+0061 LATIN SMALL LETTER A + U+0332 COMBINING LOW LINE.

Encoding Structure. The character block for Miscellaneous Technical symbols is divided into the following ranges:

U+2300	→	U+2307,	Miscellaneous symbols
U+2310	→	U+231B,	
U+2322	→	U+2323	
U+2308	→	U+230B	Ceilings and floors
U+230C	→	U+230F	Crops
U+231C	→	U+231F	Quine corners
U+2320	→	U+2321	Partial math symbols for compatibility
U+2324	→	U+2328,	Keyboard symbols
U+232B			
U+2329	→	U+232A	Bra and Ket
U+232C			Benzene ring
U+232D	→	U+2335	Technical drafting symbols
U+2336	→	U+237A	APL functional symbols

Control Pictures: U+2400—U+243F

The need to show the presence of the C0 control codes and the SPACE unequivocally when data is displayed has led to conventional representations for these non-graphic characters.

By definition, control codes themselves are manifested only by their action. However, it is sometimes necessary to show the position of a control code within a data stream. Conventional illustrations for the ASCII C0 control codes have been developed.

By definition, the SPACE is a blank graphic. Conventions have also been established for the visible representation of the space.

Standards. The CNS 11643 standard encodes characters for pictures of control codes. Standard representations for control characters have been defined, for example, in ANSI X3.32 and ISO 2047, but for the control code graphics U+2400 → U+241F only the semantic is encoded in the Unicode Standard. This allows a particular application to use the graphic representation it prefers.

Pictures for ASCII Space. Two specific glyphs are provided that may be used to represent the ASCII space character (U+2420 and U+2422).

Code Points for Pictures for Control Codes. The remaining code points in this block are not associated with specific glyphs, but rather are available to encode *any* desired pictorial representation of the given control code. The assumption is that the particular pictures used to represent control codes are often specific to different systems, and are not often the subject of text interchange between systems.

Encoding Structure. The character block for Control Pictures is divided into the following ranges:

U+2400 → U+241F U+0000→U+001F	Code points for pictures for control codes
U+2420, U+2422 → U+2423	Pictures for the ASCII space character
U+2421	Picture for *delete*
U+2424	Picture for *new line*

Optical Character Recognition: U+2440—U+245F

This block includes those symbolic characters of the OCR-A character set that do not correspond to ASCII characters, and magnetic ink character recognition (MICR) symbols used in check processing.

Standards. Both sets of symbols are specified in ISO 2033.

Encoding Structure. The character block for Optical Character Recognitions is divided into the following ranges:

U+2440	→	U+2445	OCR-A Symbols
U+2446	→	U+244A	MICR symbols

Enclosed Alphanumerics: U+2460—U+24FF

The enclosed numbers and Latin letters of this block come from several sources, chiefly East Asian standards, and are provided for compatibility with them.

Standards. Enclosed letters and numbers occur in the Korean National Standard, KS C 5601, and in the Chinese national standard, GB 2312, as well as in various East Asian industry standards.

The Zapf Dingbats character set in widespread industry use contains four sets of encircled numbers (including encircled zero). The black on white set that has numbers with serifs is encoded here (U+2460 → U+2468, and U+24EA). The other three sets are encoded in the range U+2776 → U+2793 in the Dingbats block.

Decompositions. The parenthesized letters or numbers may be decomposed to a sequence of opening parenthesis, letter or digit(s), closing parenthesis. The numbers with a period may be decomposed to digit(s), followed by a period. The encircled letters and single-digit numbers may be decomposed to letter or digit followed by U+20DD COMBINING ENCLOSING CIRCLE. Decompositions for the encircled numbers 10 through 20 are not supported in Unicode plain text. (For more information, see *Chapter 2, General Structure* and *Chapter 3, Conformance.*)

Encoding Structure. The character block for Enclosed Alphanumerics is divided into the following ranges:

U+2460	→	U+2473	Encircled numbers 1–20
U+2474	→	U+2487	Parenthesized numbers 1–20
U+2488	→	U+249B	Numbers with period 1–20
U+249C	→	U+24B5	Parenthesized small Latin a–z
U+24B6	→	U+24CF	Encircled capital Latin A–Z
U+24D0	→	U+24E9	Encircled small Latin a–z
U+24EA			Encircled number 0

Box Drawing: U+2500—U+257F

The characters in the Box Drawing block are encoded solely for compatibility with existing standards.

Standards. GB 2312, KS C 5601, and industry standards.

Encoding Structure. The character block for Box Drawing is divided into the following ranges:

U+2500	→	U+254F	Single-line box and line-drawing elements
U+2550	→	U+256C	Line-box drawing elements with double-line segments
U+256D	→	U+2570	Curved corner segments
U+2571	→	U+2573	Diagonal line segments and miscellaneous
U+2574	→	U+257F	Line end pieces and connectors

Block Elements: U+2580—U+259F

The Block Elements block represents a graphic compatibility zone in the Unicode Standard. A number of existing national and vendor standards, including IBM PC Code Page 437, contain a number of characters intended to enable a simple kind of display cell graphics by filling some fraction of each cell, or by filling each display cell by some degree of shading. The Unicode Standard does not encourage this kind of character-based graphics model but includes a minimal set of such characters for backward compatibility with the existing standards.

Half-block fill characters are included for each half of a display cell, plus a graduated series of vertical and horizontal fractional fills based on one-eighth parts. Also included is a series of shades based on one-quarter shadings. The fractional fills do not form a logically complete set but are intended only for backward compatibility.

Encoding Structure. The character block for Block Elements is divided into the following ranges:

U+2580	→	U+2590,	Display cell fractional fill characters
U+2594	→	U+2595	
U+2591	→	U+2593	Percent shade characters

Geometric Shapes: U+25A0—U+25FF

The Geometric Shapes are a collection of characters intended to encode prototypes for various commonly used geometrical shapes—mostly squares, triangles, and circles. The collection is somewhat arbitrary in scope; it is a compendium of shapes from various character and glyph standards. The typical distinctions more systematically encoded include black versus white, large versus small, basic shape (square versus triangle versus circle), orientation, and top versus bottom or left versus right part.

The hatched and cross-hatched squares at U+25A4 → U+25A9 derive from the Korean national standard (KS C 5601), in which they were probably intended as representations of fill patterns; however, since the semantics of those characters is insufficiently defined in that standard, the Unicode character encoding simply carries the glyphs themselves as geometric shapes to provide a mapping for that standard.

U+25CA LOZENGE ◊ is a typographical symbol seen in PostScript and in the Macintosh character set. It should be distinguished both from the generic U+25C7 WHITE DIAMOND and the U+2662 WHITE DIAMOND SUIT, as well as from another character sometimes called a lozenge, U+2311 SQUARE LOZENGE.

The squares and triangles at U+25E7 → U+25EE are derived from the Linotype font collection. U+25EF LARGE CIRCLE is included for compatibility with the JIS X 0208-1990 Japanese standard.

Standards. The Geometric Shapes are derived from a large range of national and vendor character standards.

Encoding Structure. The character block for Geometric Shapes is divided into the following ranges:

U+25A0	→	U+25EF	Geometric shapes based on various forms

Miscellaneous Symbols: U+2600—U+26FF

The Miscellaneous Symbols block consists of a very heterogenous collection of symbols that do not fit in any other Unicode character block and which tend to be rather pictographic in nature. The usage of these symbols is typically for text decorations, but they may also be treated as normal text characters in applications such as typesetting chess books, card game manuals, and horoscopes.

Characters in the Miscellaneous Symbols block may be rendered in more than one way, unlike characters in the Dingbats block, in which characters correspond to an explicit glyph. For example, both U+2641 EARTH and U+2645 URANUS have common alternative glyphs.

The order of the Miscellaneous Symbols is completely arbitrary, but an attempt has been made to keep like symbols together and to group subsets of them into meaningful orders. Some of these subsets include weather and astronomical symbols, pointing hands, religious and ideological symbols, the I Ching trigrams, planet and zodiacal symbols, chess pieces, card suits, and musical dingbats. (For other moon phases, see the circle-based shapes in the Geometric Shapes block.)

Corporate logos and collections of pictures of animals, vehicles, foods, and so on are not included since they tend either to be very specific in usage (logos, political party symbols) or nonconventional in appearance and semantic interpretation (pictures of cows or of cats; fizzing champagne bottles), and hence are inappropriate for encoding as characters. The Unicode Standard recommends that such items be incorporated in text via higher protocols that allow intermixing of graphic images with text, rather than by indefinite extension of the number of Miscellaneous Symbols encoded as characters. However, a large unassigned space has been set aside in this block with the expectation that other conventional sets of such symbols may be found appropriate for character encoding in the future.

No attempt is made to provide a complete character encoding for musical notation. The Unicode Standard considers musical notation to be a higher-order text format that requires two-dimensional layout control and complex structures.

Standards. The Miscellaneous Symbols are derived from a large range of national and vendor character standards.

Encoding Structure. The character block for Miscellaneous Symbols is divided into the following ranges:

U+2600	→	U+2603	Weather symbols
U+2604	→	U+262F,	Miscellaneous symbols
U+2638	→	U+263C,	
U+2645	→	U+2647,	
U+2668			
U+2630	→	U+2637	I-Ching symbols
U+263D	→	U+2644	Moon phases and planets
U+2648	→	U+2653	Signs of the zodiac
U+2654	→	U+265F	Chess pieces
U+2660	→	U+2667	Card suits
U+2669	→	U+266F	Musical symbols

Dingbats: U+2700—U+27BF

The Dingbats are a well-established set of symbols comprising the industry standard "Zapf Dingbat" font—currently available in most laser printers. Other series of Dingbats also exist but are not encoded in the Unicode Standard because they are not widely implemented in existing hardware and software as character-encoded fonts. Dingbats that are part of other standards have been encoded in the Geometrical Shapes, Enclosed Alphanumerics, and Miscellaneous Symbols blocks. The order of the remaining dingbats follows the PostScript encoding.

The Dingbats differ in their treatment in the Unicode Standard from all other characters. They are encoded as specific glyph shapes, rather than as glyphic archetypes for abstract characters that can be represented in different faces and styles. Thus, it would be incorrect to arbitrarily replace U+279D TRIANGLE-HEADED RIGHTWARDS ARROW → with any other right arrow dingbat or with any of the generic arrows from the Arrows block (U+2190 → U+21FF). In other words, since the Zapf Dingbats refer to glyphs from a specific typeface, their semantic value *is* their shape.

Unifications. A number of the Dingbats represent shapes that overlap with regular Unicode symbol characters. Instead of coding both a Zapf Dingbat glyph shape and a separate character whose glyphic representation is normally indistinguishable from that shape, the Unicode Standard unifies the two. The characters in question include card suits, BLACK STAR, BLACK TELEPHONE, and BLACK RIGHT-POINTING INDEX (see "Miscellaneous Symbols"); BLACK CIRCLE and BLACK SQUARE (see "Geometric Shapes"); white encircled numbers 1 to 10 (see "Enclosed Alphanumerics"); and several generic arrows (see "Arrows"). These four entries appear elsewhere in this section.

The positions of these unified characters are left unassigned in the Dingbats block and are cross-referenced to the assigned positions in the other blocks. Applications may use alternative glyphs for representing those characters (as for any normal Unicode characters), including, of course, the exact shapes required for rendering them in the Zapf Dingbat font on an imaging device.

To illustrate this distinction, an application encoding an encircled digit one with U+2460 ① CIRCLED DIGIT ONE may render that encircled digit in any appropriate typeface—serif or sans serif, roman or italic, and with the circle rendered in different thicknesses. On the other hand, an application encoding an encircled digit one with the Dingbat U+2780 SANS SERIF CIRCLED DIGIT ONE ➊ requires an explicit sans serif glyph from the Zapf Dingbat font for rendering.

Encoding Structure. The character block for Dingbats is divided into the following ranges:

U+2701 → U+27BE ITC Zapf Dingbats, series 100 (with gaps for unifications)