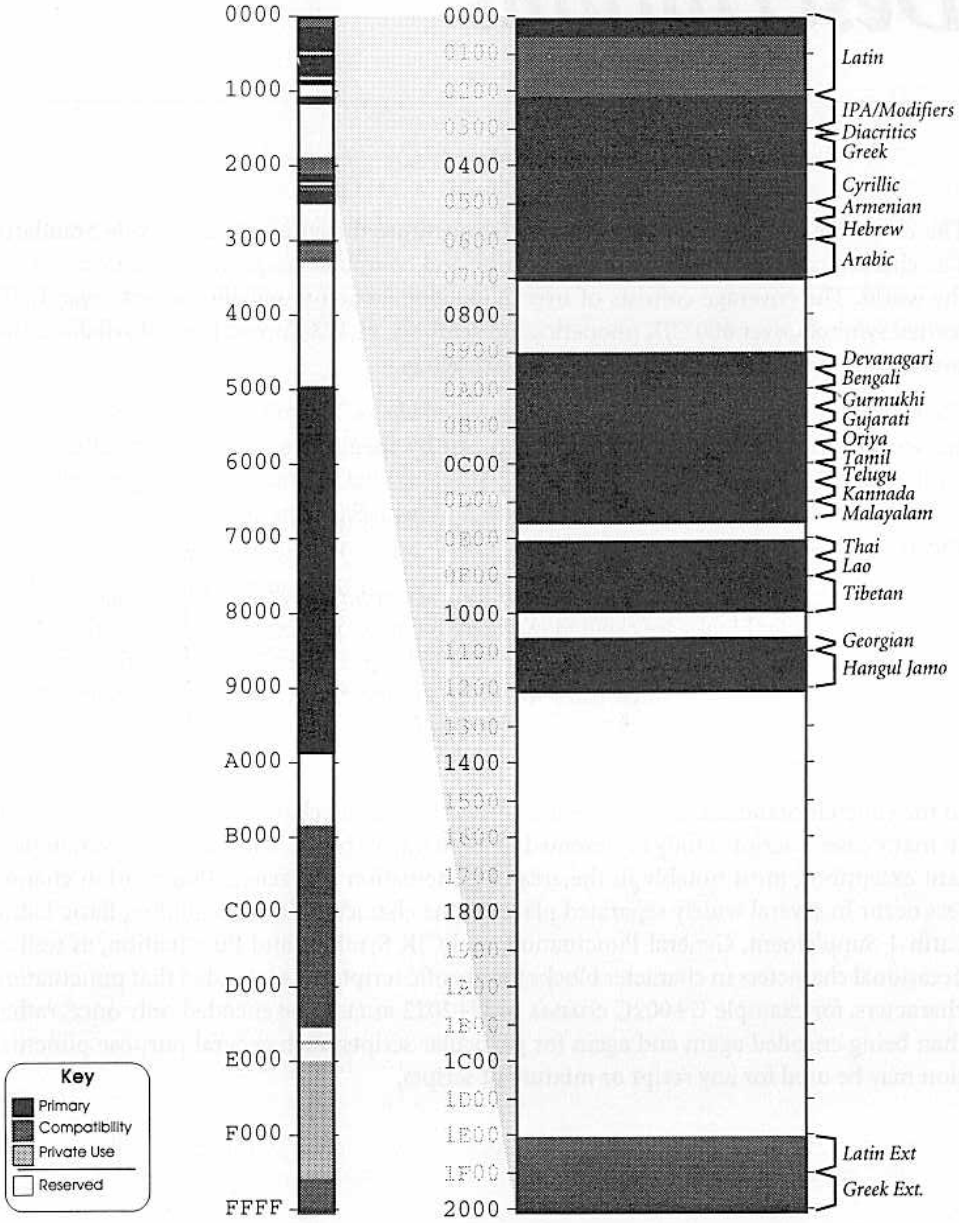
6.1 General Scripts Area

The General Scripts Area includes the encoding of all Latin and other non-ideographic script characters. This includes Greek, Cyrillic, Hebrew, Arabic, the numerous Indic scripts, Georgian, Armenian, Tibetan, Thai, and Lao (see Figure 6-1).

Figure 6-1. General Scripts



Basic Latin: U+0000—U+007F

Standards. The Unicode Standard adapts the ASCII (ISO 646) 7-bit standard by retaining the semantics and numeric code values, merely supplying enough leading zeros to convert them into 16-bit values. The content and arrangement of the ASCII standard is far from optimal in the context of a 16-bit space, but the Unicode Standard retains it without change because of its prevalence in existing usage. The ASCII (ANSI X3.4) standard is identical to ISO/IEC 646:1991-IRV.

ASCII C0 Control Codes and Delete. The Unicode Standard makes no specific use of these control codes, but it provides for the passage of the numeric code values intact, neither adding to nor subtracting from their semantics. The semantics of the C0 controls (U+0000→U+001F) and *delete* (U+007F) are generally determined by the application with which they are used. However, in the absence of specific application uses, they may be interpreted according to the semantics specified in ISO 6429. The only C0 control code that has specified semantics in the Unicode Standard is U+0009 HORIZONTAL TAB. (For more information on control codes, see *Section 2.4, Special Character and Non-Character Values.*)

There is a simple one-to-one mapping between 7-bit (and 8-bit) control codes and Unicode control codes: every 7-bit (or 8-bit) control code is simply zero-extended to a 16-bit code. For example, if LINE FEED (0A) is to be used for terminal control, then the text "WX<LF>YZ" would be transmitted in plain Unicode text as the following 16-bit values: "0057 0058 000A 0059 005A." Any interpretation of these control codes is outside the scope of the Unicode Standard; programmers should refer to a relevant standard (for example, ISO 6429) that specifies control code interpretations.

ASCII Graphic Characters. Some of the non-letter characters in this range suffer from overburdened usage as a result of the limited number of codes in a 7-bit space. Some coding consequences of this are discussed in the following subsections "Encoding Characters with Multiple Semantic Values" and "Loose versus Precise Semantics." The rather haphazard ASCII collection of punctuation and mathematical signs are isolated from the larger body of Unicode punctuation, signs, and symbols (which are encoded in ranges starting at U+2000) only because the relative locations within ASCII are so widely used in standards and software.

Encoding Characters with Multiple Semantic Values. Code values in the ASCII range are well established and used in widely varying implementations. The Unicode Standard therefore provides only minimal specifications on the typographic appearance of corresponding glyphs. For example, the value U+0024 ($) (derived from ASCII 24) has the semantic *dollar sign*, leaving open the question of whether the dollar sign is to be rendered with one vertical stroke or two. The Unicode value U+0024 refers to the *dollar sign semantic*, not to its precise appearance. Likewise, for other characters in this range that have alternative glyphs, the Unicode character is displayed with the basic or most common glyph; rendering software may present any other graphical form of that character.

Loose versus Precise Semantics. Some ASCII characters have multiple uses, either through ambiguity in the original standards or through accumulated reinterpretations of a limited codeset. For example, 27 hex is defined in ANSI X3.4 as *apostrophe (closing single quotation mark; acute accent)* and 2D hex as *hyphen minus*. In general, the Unicode Standard provides the same interpretation for the equivalent code values, without adding to or subtracting from their semantics. The Unicode Standard supplies *unambiguous* codes elsewhere for the most useful particular interpretations of these ASCII values; the corresponding unambiguous characters are cross-referenced in the character names list for this block. (For a complete list of space characters and dash characters in the Unicode Standard, see the General Punctuation subsection of *Section 6.2, Symbols Area.*)

Diacritics. ASCII contains four codes that are ambiguous regarding whether they denote combining characters or separate spacing characters. In the Unicode encoding, the corresponding code points (U+005E CIRCUMFLEX ACCENT ^; U+005F LOW LINE _; U+0060 GRAVE ACCENT `; U+007E TILDE ~) are restricted to use as spacing characters. The Unicode Standard provides unambiguous combining characters in other blocks which can be used to represent accented Latin letters by means of composed character sequences.

Semantics of Paired Punctuation. Paired punctuation marks such as parentheses (U+0028, U+0029), square brackets (U+005B, U+005D), and braces (U+007B, U+007D) are interpreted semantically rather than graphically in the context of bidirectional or vertical texts; that is, *these characters have consistent semantics but alternative glyphs depending upon the directional flow rendered by a given software program.* The software must ensure that the rendered glyph is the correct one. When interpreted semantically rather than graphically, characters containing the qualifier "LEFT" are taken to denote *opening*; characters containing the qualifer "RIGHT" are taken to denote *closing*. For example, U+0028 LEFT PARENTHESIS and U+0029 RIGHT PARENTHESIS are interpreted as opening and closing parenthesis, respectively, in the context of bidirectional or vertical texts. In a right-to-left directional flow, U+0028 is rendered as ")". In a left-to-right flow, the same character is rendered as "(".

Encoding Structure. The character block for Basic Latin characters is divided into the following ranges:

U+0000	→	U+001F	ASCII C0 control codes
U+0020	→	U+007E	ASCII graphic characters
U+007F			ASCII *delete* (also a control code)

Latin-1 Supplement: U+0080—U+00FF

Standards. ISO 8859-1, also known as Latin-1, extends ASCII by providing additional letters for major languages of Europe (listed in the next paragraph). Like ASCII, the Latin-1 set also includes a miscellaneous set of punctuation and mathematical signs. Punctuation, signs, and symbols not included in the Basic Latin and Latin-1 Supplement blocks are encoded in character blocks starting with the General Punctuation block.

Languages. The languages supported by the Latin-1 supplement include Danish, Dutch, Faroese, Finnish, Flemish, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, and Swedish. Many other languages can be written with this set of letters, including Hawaiian, Indonesian, and Swahili.

C1 Control Codes. In extending the 7-bit encoding system of ASCII to an 8-bit system, ISO/IEC 4873 (on which the 8859 family of character standards are based) introduced 32 additional control codes in the range 80–9F hex. Like the C0 control codes, the Unicode Standard makes no specific use of these C1 control codes, but provides for the passage of their numeric code values intact, neither adding to nor subtracting from their semantics. The semantics of the C1 controls (U+0080→U+009F) are generally determined by the application with which they are used. However, in the absence of specific application uses, they may be interpreted according to the semantics specified in ISO 6429.

Diacritics. ISO 8859-1 contains four characters that are ambiguous regarding whether they denote combining characters or separate spacing characters. In the Unicode Standard, the corresponding codepoints (U+00A8 DIAERESIS, U+00AF MACRON, U+00B4 ACUTE ACCENT, and U+00B8 CEDILLA) are restricted to use as spacing characters. The Unicode Standard provides unambiguous combining characters in the character block for Combining Diacritical Marks which can be used to represent accented Latin letters by means of composed character sequences. U+00B0 DEGREE SIGN is also occasionally used ambiguously by implementations of ISO 8859-1 to denote a spacing form of a diacritic ring above a letter; in the Unicode Standard, that spacing diacritical mark is denoted unambiguously by U+02DA RING ABOVE.

U+00AD SOFT HYPHEN indicates a hyphenation point, where a line-break is preferred when a word is to be hyphenated. Depending on the script, the visible rendering of this character when a line break occurs may differ (for example, in some scripts it is rendered as a *hyphen* -, while in others it may be invisible). See also U+2027 HYPHENATION POINT. For a complete list of dash characters in the Unicode Standard, see the General Punctuation character block description.

U+00A0 NO-BREAK SPACE is included for compatibility with existing standards. The nominal width is the same as U+0020 SPACE, but the NO-BREAK SPACE indicates that, under normal circumstances, no line breaks are permitted between it and surrounding characters, unless the preceding or following character is a line or paragraph separator. U+00A0 NO-BREAK SPACE is equivalent to the following coded character sequence: U+FEFF ZERO WIDTH NO-BREAK SPACE + U+0020 SPACE + U+FEFF ZERO WIDTH NO-BREAK SPACE. For a complete list of space characters in the Unicode Standard, see the General Punctuation character block description.

Ordinals. U+00AA FEMININE ORDINAL INDICATOR and U+00BA MASCULINE ORDINAL INDICATOR can be depicted with an underscore, but many modern fonts show them as superscripted Latin letters with no underscore. In sorting and searching these characters should be treated as weakly equivalent to their Latin character equivalents.

Quotation Marks. The characters U+00AB LEFT-POINTING DOUBLE ANGLE QUOTATION MARK « and U+00BB RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK », also known as *guillemets*, are interpreted semantically rather than graphically in the context of bidirec-

tional or vertical texts. Like the parenthesis and bracket characters of the Basic Latin block, thesecharactersareinterpretedasopeningandclosingcharacters,respectively,whenused in a bidirectional or vertical text. The rendering mechanism must select the appropriately shaped glyph form to depict the character according to its directional context.

Encoding Structure. The Latin-1 Supplement character block is divided into the following ranges:

U+0080	→	U+009F	C1 control codes
U+00A0	→	U+00FF	Latin-1 graphic characters

Latin Extended-A: U+0100—U+017F

The Latin Extended-A block contains a collection of letters which, when added to the letters contained in the Basic Latin and Latin-1 Supplement blocks, allow for the representation of most European languages that employ the Latin script. Many other languages can also be written with the characters in this block. Most of these characters are equivalent to precomposed combinations of base character forms and combining diacritical marks. These combinations may also be represented by means of composed character sequences. See *Section 2.5, Combining Characters.*

Standards. This block includes characters contained in International Standard ISO 8859 – Part 2. Latin alphabet No. 2, Part 3. Latin alphabet No. 3, Part 4. Latin alphabet No. 4, and Part 9. Latin alphabet No. 5. Many of the other graphic characters contained in these standards, such as punctuation, signs, symbols, and diacritical marks, are already encoded in the Latin-1 Supplement block. Other characters from these parts of ISO 8859 are encoded in other blocks, primarily in the Spacing Modifier Letters block (U+02B0→U+02FF) and in the character blocks starting at and following the General Punctuation block.

Languages. Most languages supported by this block also require the concurrent use of characters contained in the Basic Latin and Latin-1 Supplement blocks. When combined with these two blocks, the Latin Extended-A block supports Afrikaans, Breton, Basque, Catalan, Croatian, Czech, Esperanto, Estonian, French, Frisian, Greenlandic, Hungarian, Latin, Latvian, Lithuanian, Maltese, Polish, Provençal, Rhaeto-Romanic, Romanian, Romany, Sami, Slovak, Slovenian, Sorbian, Turkish, Welsh, and many others.

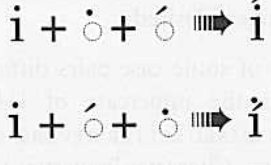
Alternative Graphics. Some characters have alternative representations, although they have a common semantic. When Czech is printed in books, letter/apostrophe forms are frequently used. In typewritten or handwritten documents, letter/hacek forms are preferred.

Exceptional Case Pairs. The characters U+0130 LATIN CAPITAL LETTER I WITH DOT ABOVE and U+0131 LATIN SMALL LETTER DOTLESS I (used primarily in Turkish) are assumed to take ASCII "i" and "I" as their case alternates, respectively. This mapping makes the corresponding reverse mapping language-specific; mapping in both directions requires special attention from the implementor (see *Section 5.15, Sorting and Searching*).

Diacritics on i. A dotted (normal) i followed by a top non-spacing mark loses the dot in rendering. Thus, in the word *naïve* the *ï* could be spelled with *i + diaeresis*. Just as Cyrillic A is not equivalent to Latin A, a *dotted-i* is not equivalent to a Turkish *dotless-i + overdot*, nor are other cases of accented *dotted-i* equivalent to accented *dotless-i* (for example, i + ¨ ≠ ı + ¨).

To express the forms sometimes used in the Baltic (where the dot is retained under a top accent), use *i + overdot + accent* (see Figure 6-2).

Figure 6-2. Diacritics on i



Encoding Structure. The characters are grouped according to their base letter without regard for diacritics. Small letters immediately follow their capital letter counterparts.

U+0100 → U+017F Latin Extended-A characters

Latin Extended-B: U+0180—U+024F

The Latin Extended-B block contains letter forms used to extend Latin scripts to represent additional languages. It also contains phonetic symbols not included in the International Phonetic Alphabet (see the IPA Extensions block, U+0250→U+02AF).

Standards. This block covers, among other things, characters in ISO 6438 Documentation—African coded character set for bibliographic information interchange, *Pinyin* Latin transcription characters from the People's Republic of China national standard GB 2312 and from the Japanese national standard JIS X 0212, and Sami characters from ISO 8859 Part 10. *Latin alphabet No. 6.*

Arrangement. The characters are arranged in a nominal alphabetical order, followed by a small collection of Latinate forms. Upper- and lowercase pairs are placed together where possible, but in many instances the other case form is encoded at some distant location and so is cross-referenced. Variations on the same base letter are arranged in the following order: turned, inverted, hook attachment, stroke extension or modification, different style (script), small cap, modified basic form, ligature, and Greek-derived.

Croatian Digraphs Matching Serbian Cyrillic Letters. Serbo-Croatian is a single language with paired alphabets: a Latin script (Croatian) and a Cyrillic script (Serbian). A set of digraph codes is provided solely for compatibility purposes. There are two potential uppercase forms for each digraph, depending on whether only the initial letter is to be capitalized (title case), or both (all uppercase). The Unicode Standard offers both forms so that software can convert one form to the other without changing font sets. The appropriate cross-references are given for the lowercase letters. For more information about canonical equivalence, see *Chapter 3, Conformance.*

Pinyin Diacritic-Vowel Combinations. The Chinese standard GB 2312, as well as the Japanese standard JIS X 0212, includes a set of codes for *Pinyin*, used for Latin transcription of Mandarin Chinese. Most of the letters used in Pinyin romanization (even those with combining diacritical marks) are already covered in the preceding Latin blocks. The group of 16 characters provided here completes the Pinyin character set specified in GB 2312 and JIS X 0212.

Case Pairs. A number of characters in this block are uppercase forms of characters whose lowercase form is part of some other grouping. Many of these came from the International Phonetic Alphabet; they acquired novel uppercase forms when they were adopted into Latinscript-basedwritingsystems.Occasionally,however,*alternative*uppercaseforms arose in this process. In some instances, research has shown that alternative uppercase forms are merely variants of the same character. If so, such variants are assigned a single Unicode value, as is the case of U+01B7 LATIN CAPITAL LETTER EZH. But when research has shown that two uppercase forms are actually used in different ways, then they are given different codes; such is the case for U+018E LATIN CAPITAL LETTER REVERSED E and U+018F LATIN CAPITAL LETTER SCHWA. In this instance, the shared lowercase form is copied: U+01DD LATIN SMALL LETTER TURNED E is a copy of U+0259 LATIN SMALL LETTER SCHWA to enable unique case-pair mappings if desired.

For historical reasons, the names of some case pairs differ. For example, U+018E LATIN CAPITAL LETTER REVERSED E is the uppercase of U+01DD LATIN SMALL LETTER TURNED E—not of U+0258 LATIN SMALL LETTER REVERSED E. (For default case mappings of Unicode characters, see *Chapter 4, Character Properties.*)

Languages. Some indication of language or other usage is given for most characters within the names lists accompanying the character charts.

Encoding Structure. The character block for Latin Extended-B is divided into the following ranges:

U+0180	→	U+01C3	General Extended Latin
U+01C4	→	U+01CC	Croatian digraphs matching Serbian Cyrillic letters
U+01CD	→	U+01DC	Pinyin diacritic-vowel combinations
U+01DD	→	U+01F5	Additional Latin characters
U+01FA	→	U+01FF	Additional Latin characters from ISO 8859-10 (for Sami)
U+0200	→	U+0217	Croatian vowels with tone marks

IPA Extensions: U+0250—U+02AF

The IPA Extensions block contains primarily the unique symbols of the International Phonetic Alphabet (IPA), which is a standard system for indicating specific speech sounds. The IPA was first introduced in 1886 and has undergone occasional revisions of content and usage since that time. The Unicode Standard covers all single symbols and all diacritics in the last published IPA revision (1989), as well as a few symbols in former IPA usage which are no longer currently sanctioned. A few symbols have been added to this block that are part of the transcriptional practices of Sinologists, Americanists, and other linguists. Some of these practices have usages independent of the IPA and may use characters from other Latin blocks rather than IPA forms. Note also that a few non-standard or obsolete phonetic symbols are encoded in the Latin Extended-B block.

An essential feature of IPA is the use of combining diacritical marks. IPA diacritical mark characters are coded in the Combining Diacritical Marks block, U+0300→U+036F. In IPA, diacritical marks can be freely applied to base form letters to indicate fine degrees of phonetic differentiation required for precise recording of different languages. In the Unicode Standard, all diacritical marks are encoded in sequence *after the base characters to which they apply.* (For more details, see the block description for Combining Diacritical Marks, and *Section 2.5, Combining Characters.*)

Standards. The characters in this block are taken from the 1989 revision of the International Phonetic Alphabet, published by the International Phonetic Association. The International Phonetic Association standard considers IPA to be a separate alphabet, so it includes the entire Latin lowercase alphabet *a–z*, a number of extended Latin letters such as U+0153 LATIN SMALL LIGATURE OE œ, and a few Greek letters and other symbols as separate and distinct characters. In contrast, the Unicode Standard does not duplicate the Latin lowercase letters *a–z*, nor other Latin or Greek letters in encoding IPA. Note that unlike other character standards referenced by the Unicode Standard, IPA constitutes an extended alphabet and phonetic transcriptional standard, rather than a character encoding standard.

Unifications. The IPA symbols are unified as much as possible with other letters (though not with non-letter symbols like U+222B INTEGRAL ∫.) The IPA symbols have also been adopted into the Latin-based alphabets of many written languages (such as in Africa). It is futile to attempt to distinguish a transcription from an actual alphabet in such cases. Therefore, many IPA symbols are found outside the IPA Extensions block. IPA symbols that are not found in the IPA Extensions block are listed as cross-references at the beginning of the character names list for this block.

IPA Alternates. In a few cases IPA practice has, over time, produced alternate forms, such as U+0269 LATIN SMALL LETTER IOTA "ɪ" versus U+026A LATIN LETTER SMALL CAPITAL I "ɪ." The Unicode Standard provides separate encodings for the two alternate forms due to the fact that they are used in a meaningfully distinct fashion.

Case Pairs. IPA does not sanction case distinctions; in effect, its phonetic symbols are all lowercase. When IPA symbols are adopted into a particular alphabet as used by a given written language (as has occurred for example, in Africa) they acquire uppercase forms. Since these uppercase forms are not themselves IPA symbols, they are generally encoded in the Latin Extended-B block (or other Latin extension blocks) and are cross-referenced with the IPA names list.

Typographic Variants. IPA includes typographic variants of certain Latin and Greek letters that would ordinarily be considered variations of font style rather than of character identity, such as SMALL CAPITAL letter forms. Examples include a typographic variant of the Greek letter *phi* ɸ, as well as the borrowed letter Greek *iota* ɩ, which has a unique Latin uppercase form. These forms are encoded as separate characters in the Unicode Standard because they have distinct semantics in plain text.

Affricate Digraph Ligatures. IPA officially sanctions six digraph ligatures used in transcription of coronal affricates. These are encoded at U+02A3→U+02A8. The IPA digraph ligatures are explicitly defined in IPA and also have possible semantic values that make them not simply rendering forms. Thus, for example, while U+02A6 LATIN SMALL LETTER TS DIGRAPH is a transcription for the sounds that could also be transcribed in IPA as U+0074 U+0073, "ts," the choice of the digraph ligature may be the result of a deliberate distinction made by the transcriber regarding the systematic phonetic status of the affricate. It cannot be a choice left up to rendering software whether to ligate or not based on the font available. This ligature also differs in typographical design from the ts ligature found in some old-style fonts.

Encoding Structure. The IPA Extensions block is arranged in approximate alphabetical order according to the Latin letter that is graphically most similar to each symbol. This has nothing to do with a phonetic arrangement of the IPA letters. This block may be divided into the following ranges:

U+0250	→	U+0298	Pre-1989 IPA characters and other phonetic symbols
U+0299	→	U+02A2	Post-1989 IPA extensions and older IPA symbols
U+02A3	→	U+02A8	IPA digraph ligatures

Spacing Modifier Letters: U+02B0—U+02FF

Modifier letters are an assorted collection of small signs that are generally used to indicate modifications of a preceding letter. A few may modify the following letter, and some may serve as independent letters. These signs are distinguished from diacritical marks in that modifier letters are treated as free-standing, spacing characters. They are distinguished from similar or identical appearing punctuation or symbols by the fact that the members of this block are considered to be letter characters that do not break up a word. They have the "letter" character property (see *Chapter 4, Character Properties*). The majority of these signs are phonetic modifiers, including the characters required for coverage of the International Phonetic Alphabet (IPA).

Phonetic Usage. Modifier letters have relatively well-defined phonetic interpretations. Their usage is generally to indicate a specific articulatory modification of a sound represented by another letter, or to convey a particular level of stress or tone. In phonetic usage, the modifier letters are sometimes called "diacritics," which is correct in the logical sense that they are modifiers of the preceding letter. However, in the Unicode Standard, the term diacritical marks refers specifically to non-spacing marks, whereas the codes in this block specify *spacing characters*. For this reason, many of the modifier letters in this block correspond to separate diacritical mark codes, which are cross-referenced in *Section 7.1, Character Names List Entries.*

Encoding Principles. This block includes characters that may have different semantic values attributed to them in different contexts. It also includes multiple characters that may represent the same semantic values—there is no necessary one-to-one relationship. The intention of the Unicode encoding is not to resolve the variations in usage, but merely to supply implementers with a set of useful forms to choose from. The list of usages given for each modifier letter should not be considered exhaustive. For example, the glottal stop (Arabic *hamza*) in Latin transliteration has been variously represented by the characters U+02BC MODIFIER LETTER APOSTROPHE, U+02BE MODIFER LETTER RIGHT HALF RING, and U+02C0 MODIFER LETTER GLOTTAL STOP. Conversely, an apostrophe can have several uses; for a list, see the entry for U+02BC MODIFIER LETTER APOSTROPHE in the character names list. There are also instances where an IPA modifier letter is explicitly equated in semantic value to an IPA non-spacing diacritic form.

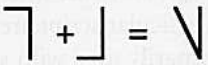
Latin Superscripts. Graphically, some of the phonetic modifier signs are raised or superscripted, some are lowered or subscripted, and some are vertically centered. Only those few forms that have specific usage in IPA or other major phonetic systems are encoded.

Spacing Clones of Diacritics. Some corporate standards explicitly specify spacing and non-spacing forms of combining diacritical marks, and the Unicode Standard provides matching codes for these interpretations when practical. A number of the spacing forms are covered in the Basic Latin and Latin-1 Supplement blocks. The six common European diacritics that do not have encodings there are added as spacing characters in the current block. These forms can have multiple semantics, such as U+02D9 DOT ABOVE, used as an indicator of the Mandarin Chinese fifth tone.

Rhotic Hook. U+02DE MODIFIER LETTER RHOTIC HOOK is defined in IPA as a free-standing modifier letter. However, in common usage it is treated as a ligated hook on a baseform letter. Hence, U+0259 LATIN SMALL LETTER SCHWA + U+02DE MODIFIER LETTER RHOTIC HOOK may be treated as equivalent to U+025A LATIN SMALL LETTER SCHWA WITH HOOK.

Tone Letters. U+02E5→U+02E9 comprise a set of basic tone letters, defined in IPA and commonly used in detailed tone transcription of African and other languages. Each tone letter refers to one of five distinguishable tone levels. In order to represent contour tones, the tone letters are used in combinations. The rendering of contour tones follows a regular set of ligation rules that result in a graphic image of the contour (see Figure 6-3).

Figure 6-3. Tone Letters



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

U+02B0	→	U+02B8,	Phonetic modifiers derived from Latin letters
U+02E0	→	U+02E4	
U+02B9	→	U+02D7,	Miscellaneous phonetic modifiers
U+02DE			
U+02D8	→	U+02DD	Spacing clones of non-spacing diacritic marks
U+02E5	→	U+02E9	IPA tone letters

Combining Diacritical Marks: U+0300—U+036F

The combining diacritical marks in this block are intended for general use with any script. Diacritical marks specific to some particular script are encoded with the alphabet for that script. Diacritical marks that are primarily used with symbols are defined in the Combining Diacritical Marks for Symbols character block (U+20D0→U+20FF).

Standards. The combining diacritical marks are derived from a variety of sources, including IPA, ISO 5426, and ISO 6937.

Sequence of Base Letters and Diacritics. In the Unicode character encoding, all non-spacing marks, including diacritics, are encoded *after* the base character. For example, the Unicode character sequence U+0061 "a" LATIN SMALL LETTER A, U+0308 "¨" COMBINING DIAERESIS, U+0075 "u" LATIN SMALL LETTER U unambiguously encodes "äu", *not* "aü".

The Unicode Standard convention is consistent with the logical order of other non-spacing marks in Semitic and Indic scripts, the great majority of which follow the base characters with respect to which they are positioned. This convention is also in line with the way modern font technology handles the rendering of non-spacing glyphic forms, so that mapping from character memory representation to rendered glyphs is simplified. (For more information on the use of diacritical marks, see *Chapter 2, General Structure,* and *Chapter 3, Conformance.*)

Diacritics Positioned Over Two Base Characters. IPA and a few languages such as Tagalog use diacritics that are applied to two base form characters. These marks apply to the previous base character—just like all other combining non-spacing marks—but hang over the following letter as well. The two characters U+0360 COMBINING DOUBLE TILDE and U+0361 COMBINING DOUBLE INVERTED BREVE are intended to be displayed as depicted in Figure 6-4.

Figure 6-4. Double Diacritics

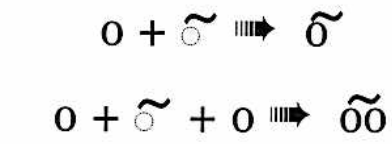
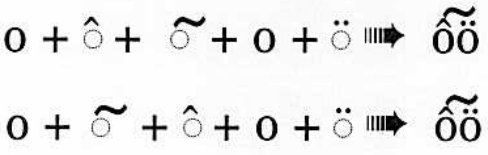


Figure 6-5. Ordering of Double Diacritics



Marks as Spacing Characters. By convention, combining marks may be exhibited in (apparent) isolation by applying them to U+0020 SPACE or to U+00A0 NO-BREAK SPACE. This might be done, for example, when referring to the diacritical mark itself as a mark, rather than using it in its normal way in text. The use of U+0020 SPACE versus U+00A0 NO-BREAK SPACE affects line-breaking behavior.

In charts and illustrations in this standard, the combining nature of these marks is illustrated by applying them to U+25CC DOTTED CIRCLE, as shown in the examples throughout this standard.

The Unicode Standard separately encodes clones of many common European diacritical marks as spacing characters. These related characters are cross-referenced in the character names list.

Encoding Principles. Because non-spacing marks have such a wide variety of applications, the characters in this block may have multiple semantic values. For example, U+0308 = *diaeresis* = *umlaut* = *double derivative.* There are also cases of several different Unicode characters for equivalent semantic values; variants of CEDILLA include at least U+0312 COMBINING TURNED COMMA ABOVE, U+0326 COMBINING COMMA BELOW, and U+0327 COMBINING CEDILLA. (For more information about the difference between non-spacing marks and combining characters, see *Chapter 2, General Structure.*)

Encoding Structure. The character block for general combining diacritical marks is divided into the following ranges:

U+0300	→	U+0333,	Ordinary diacritics
U+0339	→	U+033F	
U+0334	→	U+0338	Overstruck diacritics
U+0340	→	U+0341	Vietnamese tone mark diacritics (usage strongly discouraged)
U+0342	→	U+0345	Greek diacritics
U+0360	→	U+0361	Double diacritics

Greek: U+0370—U+03FF

The Greek script is used for writing the Greek language and (in an extended variant) the Coptic language. The Greek script had a strong influence in the development of the Latin and Cyrillic scripts.

The Greek script is written in linear sequence from left to right with the occasional use of non-spacing marks. Greek letters come in upper- and lowercase pairs.

Standards. The Unicode encoding of Greek is based on ISO 8859-7, which is equivalent to the Greek national standard ELOT 928. The Unicode Standard encodes Greek characters in the same relative positions as in ISO 8859-7. A number of variant and archaic characters are taken from the bibliographic standard ISO 5428.

Polytonic Greek. Polytonic Greek, used for ancient Greek (classical and Byzantine), may be encoded using either composite character sequences or precomposed base plus diacritic combinations. For the latter, see the Greek Extended character block (U+1F00→U+1FFF).

Non-spacing Marks. Several non-spacing marks commonly used with the Greek script are found in the Combining Diacritical Marks range (see Table 6-1).

Table 6-1. Non-Spacing Marks Used with Greek

Code	Name	Alternate
U+0300	COMBINING GRAVE ACCENT	*varia*
U+0301	COMBINING ACUTE ACCENT	*oxia*
U+0302	COMBINING CIRCUMFLEX ACCENT	
U+0303	COMBINING TILDE	
U+0304	COMBINING MACRON	
U+0306	COMBINING BREVE	
U+0308	COMBINING DIAERESIS	*dialytika*
U+030D	COMBINING VERTICAL LINE ABOVE	*tonos*
U+0313	COMBINING COMMA ABOVE	*psili*
U+0314	COMBINING REVERSED COMMA ABOVE	*dasia*
U+0342	COMBINING GREEK PERISPOMENI	
U+0343	COMBINING GREEK KORONIS	
U+0344	COMBINING GREEK DIALYTIKA TONOS	
U+0345	COMBINING GREEK YPOGEGRAMMENI	

Since the marks in that range are encoded by shape, not by meaning, they are appropriate for use in Greek where applicable. Multiple non-spacing marks applied onto the same baseform character are to be spelled as the base form character followed by the non-spacing mark characters in sequence. The order of non-spacing marks is from the base form outward. (See the general rules for applying non-spacing marks in *Section 2.5, Combining Characters.*)

U+0342 COMBINING GREEK PERISPOMENI may appear as either a circumflex or a tilde: ᾶ versus α̃. Because of this variation in form, the perispomeni was encoded distinctly from U+0303 COMBINING TILDE.

U+0313 COMBINING COMMA ABOVE and U+0343 COMBINING GREEK KORONIS both take the form of a raised comma over a baseform letter. U+0343 COMBINING GREEK KORONIS was included for compatibility reasons; U+0313 COMBINING COMMA ABOVE is the preferred form for general use.

The non-spacing mark *ypogegrammeni* (also known as *iota-subscript* in Engish) can be applied to the vowels *alpha*, *eta*, and *omega* to represent historic diphthongs. This mark appears as a small *iota* below the vowel. When applied to uppercase vowels, it can also be rendered as a small *iota* at the lower right-hand corner of the vowel.

Archaic representations of Greek words (which did not have lowercase or accents) use the Greek capital letter *iota* following the vowel for these diphthongs. Forms of the *iota* that follow the vowel are called *prosgegrammeni* (also known as *iota-adscript* in English). Such archaic representations require special case mapping.

Variant Letterforms. Variant forms of certain Greek letters are encoded as separate characters in ISO 8859-7 and ISO 5428; therefore, these forms are also included in the Unicode character set. These include U+03C2 GREEK SMALL LETTER FINAL SIGMA and U+03D0 GREEK BETA SYMBOL.

Greek Letters as Symbols. For compatibility purposes, a few Greek letters are separately encoded as symbols in other character blocks. Examples include U+00B5 MICRO SIGN μ in the Latin-1 Supplement character block and U+2126 OHM SIGN Ω in the Letterlike Symbols character block. Characters from the Greek block may be used for these symbols.

Punctuation-like Characters. The question of which punctuation-like characters are uniquely Greek and which ones can be unified with generic Western punctuation has no definitive answer. The Greek question mark U+037E GREEK QUESTION MARK *erotimatiko* ";" is encoded for compatible use by systems that treat it as a sentence-final punctuation distinct from the semicolon.

Historic Letters. Historic Greek letters have been retained from ISO 5428.

Coptic-Unique Letters. The Coptic script is regarded primarily as a stylistic variant of the Greek alphabet. The letters unique to Coptic are encoded in a separate range at the end of the Greek character block. Those characters may be used together with the basic Greek characters to represent the complete Coptic alphabet. Coptic text may be rendered using a font that contains the Coptic style of depicting the characters it shares with the Greek alphabet. Texts that mix Greek and Coptic languages together must employ appropriate font style associations.

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

U+0374	→	U+037E	Greek punctuation and *ypogegrammameni* not from ISO 8859-7, coded in relative positions where there are uncoded gaps in ISO 8859-7
U+0384	→	U+03CE	Greek letters, punctuation, and diacritical marks from ISO 8859-7 (except for those characters unified into other blocks)
U+03D0	→	U+03D6,	Variant Greek letterforms
U+03F0	→	U+03F3	
U+03DA	→	U+03E1	Archaic letters
U+03E2	→	U+03EF	Coptic-unique letters

Cyrillic: U+0400—U+04FF

The Cyrillic script is a member of the family of scripts strongly influenced by the Greek script. Cyrillic has traditionally been used for writing various Slavic languages, among which Russian is predominant. In the 19th and early 20th centuries, Cyrillic was extended to write the non-Slavic minority languages of the former Soviet Union. The Cyrillic script is written in linear sequence from left to right with the occasional use of non-spacing marks. Cyrillic letters come in upper- and lowercase pairs.

Standards. The Cyrillic block of the Unicode Standard is based on ISO 8859-5. The Unicode Standard encodes Cyrillic characters in the same relative positions as in 8859-5.

Unifications. Latin characters included in those alphabets that use both Latin and Cyrillic letters are not given duplicate Cyrillic encodings. Examples include *q* and *w* for Kurdish and U+0292 LATIN SMALL LETTER EZH for Abkhasian.

Historic Letters. The historic form of the Cyrillic alphabet is treated as a font style variation of modern Cyrillic because the historic forms are relatively close to the modern appearance and because some of them are still in modern use in languages other than Russian (for example, U+0406 CYRILLIC CAPITAL LETTER I "I" is used in modern Ukrainian and Byelorussian). Since the historic Cyrillic characters encoded in Unicode (U+0460 → U+0486) rarely occur in modern form, these letters are shown in the charts in an archaic font. A complete Old Cyrillic set would be obtained by rendering the whole Cyrillic section (that is, U+0400 → U+0486) in that same style.

Extended Cyrillic. These are the letters used in alphabets for minority languages of the former Soviet Union. The scripts of some of these languages have often been revised in the past; the Unicode Standard includes only the alphabets in current use, not the rejected old letterforms.

Glagolitic. The history of the creation of the Slavic scripts and their relationship has been lost. The Unicode Standard regards Glagolitic as a *separate* script from Cyrillic, not as a font change from Cyrillic. This is primarily because Glagolitic appears unrecognizably different from Cyrillic, and secondarily because Glagolitic has not grown to match the expansion of Cyrillic. The Glagolitic script is not currently supported by the Unicode Standard.

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

U+0400	→	U+045F	Cyrillic characters from ISO 8859-5 (except for those characters unified into other blocks as specified above)
U+0460	→	U+0481	Historic Cyrillic letters
U+0482	→	U+0486	Historic miscellaneous signs and diacritics
U+0490	→	U+04CC	Extended Cyrillic letters
U+04D0	→	U+04F9	Cyrillic letter with diacritic combinations and other compatibility additions

Armenian: U+0530—U+058F

The Armenian script is used primarily for writing the Armenian language. The script is written from left to right and generally does not use diacritics (except for the modifier letters specified below). It does have upper- and lowercase pairs.

Modifier Letters. In modern Armenian typography, the small marks in the group called Armenian modifier letters are placed above and to the right of other letters so that they occupy a letter position of their own. Therefore, in the Unicode Standard they are treated as spacing letters rather than as non-spacing marks.

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

U+0531	→	U+0556	Uppercase letters
U+0559	→	U+055F	Modifier letters
U+0561	→	U+0586	Lowercase letters
U+0589			Punctuation

Hebrew: U+0590—U+05FF

The Hebrew script is used for writing the Hebrew language as well as Yiddish, Judezmo (Ladino), and a number of other languages. Vowels and various other marks are written as *points,* which are applied to consonantal base letters; these marks are usually omitted in Hebrew, except for liturgical texts and other special applications. Five Hebrew letters assume a different graphic form when last in a word.

The Hebrew script is written from right to left. (For a general discussion of character ordering including right-to-left scripts, see *Section 3.11, Bidirectional Behavior.*)

Standards. ISO 8859-8—Part 8. *Latin/Hebrew Alphabet.* The Unicode Standard encodes the Hebrew alphabetic characters in the same relative positions as in ISO 8859-8; however, there are no points or Hebrew punctuation characters in ISO 8859-8.

Vowels and Other Marks of Pronunciation. These combining marks, generically called *points* in the context of Hebrew, indicate vowels or other modifications of consonantal letters. General rules for applying combining marks are given in *Section 2.5, Combining Characters.* Hebrew-specific behavior is described here.

Hebrew points can be separated into four classes: *Dagesh*, *Shin Dot* and *Sin Dot*, *vowels*, and *diacritics*. Each class has its own positioning rules.

Dagesh, U+05BC HEBREW POINT DAGESH, has the form of a dot that appears inside the letter that it affects. Dagesh is not a vowel, but a diacritic that affects the pronunciation of a consonant. The same base consonant can also have a vowel and/or other diacritics. *Dagesh* is the only element that goes inside a letter.

Shin dot, U+05C1 HEBREW POINT SHIN DOT, and *sin dot*, U+05C2 HEBREW POINT SIN DOT, have the form of dots that appear respectively on the upper right and upper left side of the base letter *shin*, U+05E9 HEBREW LETTER SHIN. These two dots are mutually exclusive and occur only after the base letter *shin*. The same base letter can also have a *dagesh*, a vowel, and other diacritics. *Shin* and *sin* are two Hebrew consonants that are differentiated only by the adjunction of a *shin dot* or a *sin dot* to the base character *shin*.

Points representing vowels all appear below the base character that they affect, except for *holam*, U+05B9 HEBREW POINT HOLAM, which appears above left. There is never more than one vowel for a base character, and a base charcter may have no vowel at all. The following points represent vowels: U+05B0 → U+05B9, U+05BB.

Three points are diacritics: U+05BD, U+05BF, U+FB1E. *Metag* goes below the base character it affects; *rafe* and *varika* go above.

Shin and Sin. Separate characters for the dotted letters *shin* and *sin* are not included in this block. When it is necessary to distinguish between the two forms, they should be encoded as U+05E9 HEBREW LETTER SHIN followed by the appropriate dot (U+05C1 or U+05C2). This is consistent with Israeli standard encoding. For compatibility purposes, presentation forms of shin are available in the compatibility zone at U+FB2A HEBREW LETTER SHIN WITH SHIN DOT and U+FB2B HEBREW LETTER SHIN WITH SIN DOT.

Cantillation Marks. Cantillation marks are used in publishing liturgical texts including the Bible. There are various historical schools of cantillation marking; the set of marks included in the Unicode Standard follow the pre-publication version of Israeli national standard IS 1311.2.

Positioning. Marks may combine with vowels and other points, and there are complex typographic rules for positioning these combinations.

The latitudinal placement (meaning above, below, or inside) of points and marks is very well defined and has no exceptions. The longitudinal placement (meaning left, right or

center) of points is very well defined and has no exceptions. The longitude of marks is not well defined, and convention allows for the different placement of marks relative to their base character.

When points and marks are located below the same base letter, the point always comes first (on the right) and the mark after it (on the left), except for the marks *yetiv*, U+059A HEBREW ACCENT YETIV, and *dehi*, U+05AD HEBREW ACCENT DEHI, which come first (on the right) and are followed (on the left) by the point.

These rules are followed when points and marks are located above the same base letter:

- If the point is *holam*, all cantillation marks precede it (on the right), except *pashta*, U+0599 HEBREW ACCENT PASHTA.
- *Pashta* always follows (goes to the left of) points.
- *Holam* on a sin consonant (*shin* base + *sin dot*) follows (goes to the left of) the *sin dot*.
- *Shin dot* and *sin dot* are generally represented closer vertically to the base letter than other points and marks which go above.

Punctuation. Most punctuation marks used with the Hebrew script are not given independent codes (that is, they are unified with Latin punctuation), except for the few cases where the mark has a unique form in Hebrew, namely: U+05BE HEBREW PUNCTUATION MAQAF, U+05C0 HEBREW PUNCTUATION PASEQ, U+05C3 HEBREW PUNCTUATION SOF PASUQ, U+05F3 HEBREW PUNCTUATION GERESH, and U+05F4 HEBREW PUNCTUATION GERSHAYIM. See also U+FB1E HEBREW POINT JUDEO-SPANISH VARIKA. Note that for paired punctuation such as parentheses, the glyphs chosen to represent U+0028 LEFT PARENTHESIS and U+0029 RIGHT PARENTHESIS will depend upon the direction of the rendered text.

Final (Contextual Variant) Letterforms. Variant forms of five Hebrew letters are encoded as separate characters in all Hebrew standards; therefore this practice is followed in the Unicode Standard. These five variant forms are encoded in this block rather than the compatibility zone in order to retain structural consistency between this block and ISO 8859-8.

Other Presentation Forms. For compatibility purposes, a number of additional presentation forms of Hebrew letters and letter combinations are encoded in the compatibility zone in the range U+FB1F→U+FB4F. These presentation forms include wide and alternative variant forms of certain letters, precomposed combinations of letters or positional letter forms and points, and ligatures.

Yiddish Digraphs. These are considered to be independent characters in Yiddish. The Unicode Standard has included them as separate characters in order to distinguish certain letter combinations in Yiddish text; for example, to distinguish the digraph *double vav* from an occurrence of a consonontal *vav* followed by a vocalic *vav*. The use of digraphs is consistent with standard Yiddish orthography. Other letters of the Yiddish alphabet, such as *pasekh alef*, can be composed from other characters.

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

U+0591	→	U+05AF	Cantillation marks and accents
U+05B0	→	U+05C4,	Points and punctuation
U+05F3	→	U+05F4	
U+05D0	→	U+05EA	Hebrew letters
U+05F0	→	U+05F2	Yiddish digraphs

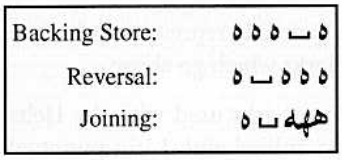
Arabic: U+0600—U+06FF

The Arabic script is used for writing the Arabic language and has been extended for representing a number of other languages, such as Persian, Urdu, Pashto, Sindhi, and Kurdish. Some languages, such as Indonesian/Malay, Turkish, and Ingush, formerly used the Arabic script and now employ the Latin or Cyrillic scripts.

The Arabic script is cursive, even in its printed form (see Figure 6-6). As a result, the same letter may be written in different forms depending on how it joins with its neighbors. Vowels and various other marks may be written as combining marks called *harakat*, which are applied to consonantal base letters. In normal writing, however, these *harakat* are omitted.

The Arabic script is written from right to left. (For a general discussion of character ordering including right-to-left scripts, see *Section 3.11, Bidirectional Behavior.*)

Figure 6-6. Reversal and Cursive Connection



Standards. ISO 8859-6—Part 6. *Latin/Arabic Alphabet.* The Unicode Standard encodes the basic Arabic characters in the same relative positions as in ISO 8859-6. ISO 8859-6, in turn, is based on ECMA-114, which was based on ASMO 449.

Encoding Principles. The basic set of Arabic letters is well defined. Each letter receives only one Unicode character value in the basic Arabic block, no matter how many different contextual appearances it may exhibit in text. Each Arabic letter in the Unicode Standard may be said to represent the inherent semantic identity of the letter. A word is spelled as a sequence of these letters. The graphic form (glyph) shown in the Unicode character chart for an Arabic letter (usually the form of the letter when standing by itself) is not the identity of that character. (See also *Section 6.8, CompatibilityArea and Specials.*)

Punctuation. Most punctuation marks used with the Arabic script are not given independent codes (that is, they are unified with Latin punctuation), except for the few cases where the mark has a significantly different appearance in Arabic, namely: U+060C ARABIC COMMA, U+061B ARABIC SEMICOLON, U+061F ARABIC QUESTION MARK, and U+066A ARABIC PERCENT SIGN. Note that for paired punctuation such as parentheses, the glyphs chosen to represent U+0028 LEFT PARENTHESIS and U+0029 RIGHT PARENTHESIS will depend upon the direction of the rendered text.

The Non-Joiner and the Joiner. The Unicode Standard provides two user-selectable zero-width formatting codes: U+200C ZERO WIDTH NON-JOINER and U+200D ZERO WIDTH JOINER (see Figure 6-7, Figure 6-8, and Figure 6-9). The use of a non-joiner between two letters prevents them from forming a cursive connection with each other when rendered. Examples include the Persian plural suffix, some Persian proper names, and Ottoman Turkish vowels. For further discussion of joiners and non-joiners, see the General Punctuation block description.

Figure 6-7. Using Joiner

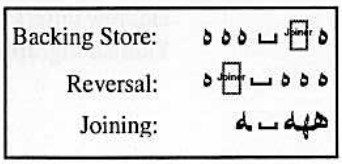


Figure 6-8. Using Non-Joiner

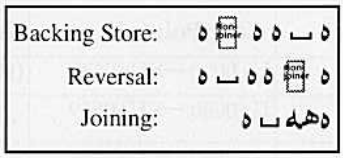
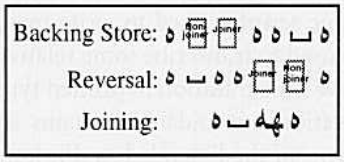


Figure 6-9. Combinations of Joiner and Non-Joiner



Harakat (Vowel) Non-Spacing Marks. *Harakat* are marks that indicate vowels or other modifications of consonant letters. The occurrence of a character in the *harakat* range and its depiction in relation to a dashed circle constitute an assertion that this character is intended to be applied via some process *to the character that precedes it* in the text stream, the base character. General rules for applying non-spacing marks are given in the Combining Diacritical Marks block description section. The few marks that are placed after (to the left of) the base character are treated as ordinary spacing characters in the Unicode Standard. The Unicode Standard does not specify a sequence order in case of multiple *harakat* applied to the same Arabic base character since there is no possible ambiguity of interpretation. (For more information about the canonical ordering of non-spacing marks, see *Chapter 2, General Structure,* and *Chapter 3, Conformance.*)

Arabic-Indic Digits. The names for the forms of decimal digits vary widely across different languages. The decimal numbering system originated in India (Devanagari ०१२३...) and was subsequently adopted in the Arabic world with a different appearance (Arabic ٠١٢٣...). The Europeans adopted decimal numbers from the Arabic world, although once again the forms of the digits changed greatly (European 0123...). The European forms were later adopted widely around the world and are used even in many Arabic-speaking countries in North Africa. In each case, the interpretation of decimal numbers remained the same. However, the forms of the digits changed to a degree that they are no longer recognizably the same characters. Because of the origin of these characters, the European decimal numbers are widely known as "Arabic numerals" or "Hindi-Arabic numerals," while the decimal numbers in use in the Arabic world are widely known there as "Hindi numbers."

The Unicode Standard includes both *Indic* digits (including forms used with different Indic scripts), *Arabic* digits (with forms used in most of the Arabic world), and *European* digits (now used internationally). Because of this, the traditional names could not be retained without confusion. In addition, there are two main variants of the Arabic digits—those used in Iran and Pakistan (here called *Eastern Arabic-Indic*) and those used in other parts of the Arabic world. The Persian and Urdu variant digits are given separate codes in the Unicode Standard to account for the differences in appearance and directional treatment when rendering them. (For a complete discussion of directional formatting in the Unicode Standard, see *Section 3.11, Bidirectional Behavior.*)

In summary, the Unicode Standard uses the names shown in Table 6-2. These names have been chosen to reduce the confusion involved in the use of the decimal number forms. They do not have any normative content; as with the choice of any other names, they are meant to be unique distinguishing labels and should not be viewed as favoring one culture over another.

Table 6-2. Digit Names

Name	Code Points	Forms
European	U+0030 → U+0039	0123456789
Arabic-Indic	U+0660 → U+0669	٠١٢٣٤٥٦٧٨٩
Eastern Arabic-Indic	U+06F0 → U+06F9	۰۱۲۳۴۵۶۷۸۹
Indic (Devanagari)	U+0966 → U+096F	०१२३४५६७८९

Extended Arabic Letters. Arabic script is used to write major languages, such as Persian and Urdu, but it has also been used to transcribe some relatively obscure languages, such as Baluchi and Lahnda, which have little tradition in printed typography. As a result, the set of characters encoded in this section unavoidably contains spurious forms. The Unicode Standard encodes multiple forms of the Extended Arabic letters and variant digits because the character forms and usages are not well documented for a number of languages. This approach was felt to be the most practical in the interest of minimizing the risk of omitting valid characters.

Languages. The languages using a given character are occasionally indicated, even though this information is incomplete. When such an annotation ends with an ellipsis (...), then the languages cited are merely the known principal ones among many.

Minimum Rendering Requirements. The cursive nature of the Arabic script imposes special requirements on display or rendering processes that are not typically found in Latin script-based systems. A display process must convert between the logical order in which Arabic characters are placed in backing store and the visual (or physical) order required by the display device. (See *Section 3.11, Bidirectional Behavior,* for a description of the conversion between logical and visual orders.)

At a minimum, a display process must also select an appropriate glyph to depict each Arabic letter according to its immediate *joining* context; furthermore, it must substitute certain ligature glyphs for sequences of Arabic characters. The remainder of this section specifies a minimum set of rules that provide legible Arabic joining and ligature substitution behavior.

Joining Classes. Each Arabic letter must be depicted by one of a number of possible contextual glyph forms. The appropriate form is determined on the basis of its *joining class* and the joining class of adjacent characters. Each Arabic character falls into one of the classes shown in Table 6-3 (see the end of this block description for a complete list).

Table 6-3. Arabic Joining Classes

Joining Class	Members
Right-joining:	ALEF, DAL, THAL, RA, ZAIN ,
Left-joining:	*none*
Dual-joining:	BAA, TAA, THAA, JEEM ...
Join-causing:	ZERO WIDTH JOINER, TATWEEL (*kashida*)
Non-joining:	ZERO WIDTH NON-JOINER and all spacing characters (other than the above), including HAMZAH, HIGH HAMZA, spaces, digits, punctuation, non-Arabic letters, and so on.
Transparent:	All combining marks and format marks, including FATHATAN, DAMMATAN, FATHAH, DAMMAH, KASRAH, SHADDAH, SUKUN, ALEF ABOVE, RIGHT-LEFT MARK, and so on.

In addition to the above classes, two superset classes will be employed as follows: a *right-join causing* character is either a dual-joining, right-joining or join-causing character; a *left-*

join causing character is either a dual-joining, left-joining or join-causing character. Here *right* and *left* refer to visual order.

Joining Rules. The following rules describe the joining behavior of Arabic letters in terms of their display (visual) order. In other words, the positions of letterforms in the included examples are presented as they would appear on the screen *after* the bidirectional algorithm has reordered the characters of a line of text.

➥ An implementation may choose to restate the following rules according to logical order so as to apply *before* the bidirectional algorithm's reordering phase. In this case, the words *right* and *left* as used in this section would become *preceding* and *following*.

In the following rules, if *X* refers to a character, then various glyph types representing that character are referred to as show in Table 6-4.

Table 6-4. Arabic Glyph Types

Glyph Types	Description
X_n	Nominal glyph form as it appears in the code charts.
X_r	Right-joining glyph form (both right-joining and dual-joining characters may employ this form).
X_l	Left-joining glyph form (both left-joining and dual-joining characters may employ this form).
X_m	Dual-joining (medial) glyph form which joins on both left and right (only dual-joining characters employ this form).

R1 ***Transparent characters do not affect the joining behavior of base (spacing) characters. For example:***

MEEM.N + SHADDAH.N + LAM.N ➞ MEEM.R + SHADDAH.N + LAM.L

م + ◌ّ + ل ➞ مـ + ◌ّ + ل ➞ لّم

R2 ***A right-joining character X that has a right join-causing character on the right will adopt the form X_r.***
For example:

ALEF.N + TATWEEL.N ➞ ALEF.R + TATWEEL.N

ا + ـ ➞ ـا + ـ ➞ ـا

R3 ***A left-joining character X that has a join-causing character on the left will adopt the form X_l.***

R4 ***A dual-joining character X that has a join-causing character on the right and a join-causing character on the left will adopt the form X_m. For example:***

TATWEEL.N + MEEM.N + TATWEEL.N ➞ TATWEEL.N + MEEM.M + TATWEEL.N

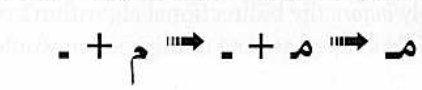
ـ + م + ـ ➞ ـ + مـ + ـ ➞ ـمـ

R5 ***A dual-joining character X that has a join-causing character on the right and no join-causing character on the left will adopt the form X_r. For example:***

MEEM.N + TATWEEL.N → MEEM.R + TATWEEL.N

R6 ***A dual-joining character X that has a join-causing character on the left and no join-causing character on the right will adopt the form X_l. For example:***

TATWEEL.N + MEEM.N → TATWEEL.N + MEEM.L



R7 ***If none of the above rules apply to a character X, then it will adopt the nominal form X_n.***

As just noted, the ZERO WIDTH NON-JOINER may be used to prevent joining, as in the Persian (Farsi) plural suffix or Ottoman Turkish vowels.

Ligature Classes. Certain types of ligatures are obligatory in Arabic script regardless of font design. Many other optional ligatures are possible, depending on font design. Since they are optional, those ligatures are not covered.

For the purpose of describing the obligatory Arabic ligatures, certain Unicode characters fall into the following classes (see the end of this block description for a complete list):

Alef-types: MADDAH-ON-ALEF, HAMZAH ON ALEF ...

Lam-types: LAM, LAM WITH SMALL V, LAM WITH DOT ABOVE ...

These two classes are designated below as *ALEF* and *LAM*, respectively.

Ligature Rules. The following rules describe the formation of ligatures. They are applied after the preceding joining rules. Like the joining rules just discussed, the following rules describe ligature behavior of Arabic letters in terms of their display (visual) order.

In the following rules, if *X* and *Y* refer to characters, then various glyph types representing combinations of these characters are referred to as as shown in Table 6-5.

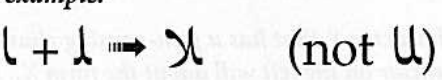
Table 6-5. Ligature Notation

Symbol	Description
$(X.Y)_n$	Nominal ligature glyph form representing a combination of an X_r form and a Y_l form.
$(X.Y)_r$	Right-joining ligature glyph form representing a combination of an X_r form and a Y_m form.
$(X.Y)_l$	Left-joining ligature glyph form representing a combination of an X_m form and a Y_l form.
$(X.Y)_m$	Dual-joining (medial) ligature glyph form representing a combination of an X_m form and a Y_m form.

R1 ***Transparent characters do not affect the ligating behavior of base (non-transparent) characters. For example:***

ALEF.R + FATHAH.N + LAM.L → LAM-ALEF.N + FATHAH.N

R2 ***Any sequence with $ALEF_r$ on the left and LAM_m on the right will form the ligature $(ALEF.LAM)_l$. For example:***



R3 ***Any sequence with $ALEF_r$ on the left and LAM_l on the right will form the ligature $(ALEF.LAM)_n$. For example:***

ل + ا ➟ لا (not ∪)

➥ From the perspective of logical (or reading) order, the preceding *(ALEF.LAM)* ligature forms are referred to as LAM-ALEF forms. The difference is due to the use of visual order rather than logical order to state ligature rules.

Optional Features. There are many other ligatures and contextual forms that are optional—depending on the font and application—such as the following:

تحـ, لـحـ, بـمـ, نـنـ

In addition, the context-sensitive placement of non-spacing vowels such as FATHA can greatly improve the appearance of Arabic text.

Arabic Character Joining Types. The Tables 6-6, 6-7, 6-8, and 6-9 provide a detailed list of the Arabic characters that are either right-joining or dual-joining. All other Arabic characters (aside from TATWEEL) are non-joining, including U+06D5 ARABIC LETTER AE. For brevity in the names, the words TWO, THREE, and FOUR are abbreviated using digits.

Most of the extended Arabic characters are merely variations on the basic Arabic shapes, with additional or different marks. For compatibility, many precomposed forms are included.

In some cases there are characters that occur only at the end of words in correct spelling; these are called *trailing characters*. Examples include TEH MARBUTA, ALEF MAQSURAH, and DAMMATAN. When trailing characters are joining (such as TEH MARBUTA), they are classed as right-joining (even when similarly-shaped characters are dual-joining). When trailing characters do not join or cause joining (such as DAMMATAN), they are classed as transparent, which treats all combining marks similarly.

NOTE. In the case of U+0647 HEH, the glyph HEH_l is also shown in the code chart box. This is often done to reduce the chance of misidentifying HEH as U+0665 ARABIC INDIC DIGIT FIVE, which has a very similar shape. The nominal form of HEH is the isolate form, which looks like U+06D5 ARABIC LETTER AE. In the case of U+06C1 HEH GOAL, the nominal form is not even listed; it also resembles ARABIC LETTER AE.

The characters in these tables are grouped by shape and not by standard Arabic alphabetical order.

Table 6-6. Dual-Joining Arabic Characters

Group	CHAR .N	CHAR .R	CHAR .M	CHAR .L	Other Characters with Similar Shaping Behavior
BAA	ب	ب	بـ	ـب	TAA, THAA, TAA WITH SMALL TAH, TAA WITH 2 DOTS VERTICAL ABOVE, BAA WITH 2 DOTS VERTICAL BELOW, TAA WITH RING, TAA WITH 3 DOTS ABOVE DOWNWARD, TAA WITH 3 DOTS BELOW, TAA WITH 4 DOTS ABOVE, BAA WITH 4 DOTS BELOW
NOON	ن	ن	نـ	ـن	DOTLESS NOON, DOTLESS NOON WITH SMALL TAH, NOON WITH RING, NOON WITH 3 DOTS ABOVE

Table 6-6. Dual-Joining Arabic Characters (Continued)

Group	CHAR .N	CHAR .R	CHAR .M	CHAR .L	Other Characters with Similar Shaping Behavior
YA	ي	ي	ـيـ	ـي	HIGH HAMZAH YA, HAMZAH ON YA, DOTLESS YEH, YA WITH SMALL V, YA WITH 2 DOTS VERTICAL BELOW, YA WITH 3 DOTS BELOW
HAA	ج	ج	ـحـ	ـح	JEEM, KHAA, HAMZAH ON HAA, HAA WITH 2 DOTS VERTICAL ABOVE, HAA WITH MIDDLE 2 DOTS, HAA WITH MIDDLE 2 DOTS VERTICAL, HAA WITH 3 DOTS ABOVE, HAA WITH MIDDLE 3 DOTS DOWNWARD, HAA WITH MIDDLE 4 DOTS
SEEN	س	س	ـشـ	ـس	
SAD	ص	ص	ـصـ	ـص	DAD, SAD WITH 2 DOTS BELOW, SAD WITH 3 DOTS ABOVE
TAH	ط	ط	ـطـ	ـط	DHAH, TAH WITH 3 DOTS ABOVE
AIN	ع	ع	ـعـ	ـع	GHAIN, AIN WITH 3 DOTS ABOVE
FA	ف	ف	ـفـ	ـف	DOTLESS FA, FAH WITH DOT MOVED BELOW, FA WITH DOT BELOW, FA WITH 3 DOTS ABOVE, FA WITH 3 DOTS BELOW, FA WITH 4 DOTS ABOVE
QAF	ق	ق	ـقـ	ـق	QAF WITH DOT ABOVE, QAF WITH 3 DOTS ABOVE
MEEM	م	م	ـمـ	ـم	
KNOTTED HA	ه	ه	ـهـ | ـه	ـه	
HA	ه	ه	ـهـ | ـه	ـه	
HA GOAL	ه	ہ	ـهـ | ـه	ـه	HA GOAL, HAMZAH ON HA GOAL
CAF	ك	ك	ـكـ	ـك	CAF WITH DOT ABOVE, CAF WITH 3 DOTS ABOVE, CAF WITH 3 DOTS BELOW
SWASH CAF	ک	ک	ـکـ	ـک	
GAF	گ	گ	ـگـ	ـگ	OPEN CAF, CAF WITH RING, GAF WITH RING, GAF WITH 2 DOTS ABOVE, GAF WITH 2 DOTS BELOW, GAF WITH 2 DOTS VERTICAL BELOW, GAF WITH 3 DOTS ABOVE
LAM	ل	ل	ـلـ	ـل	LAM WITH SMALL V, LAM WITH DOT ABOVE, LAM WITH 3 DOTS ABOVE

Table 6-7. Right-Joining Arabic Characters

Group	CHAR.N	CHAR.R	Other Characters with Similar Shaping Behavior
ALEF	ا	ـا	ALEF, HAMZAH ON ALEF, MADDAH ON ALEF, HAMZAH UNDER ALEF, WAVY HAMZAH ON ALEF, HIGH HAMZAH ALEF
WAW	و	ـو	HAMZAH ON WAW, HAMZAH ON WAW, HIGH HAMZAH WAW, HIGH HAMZAH WAW WITH DAMMAH, WAW WITH RING, WAW WITH BAR, WAW WITH SMALL V, WAW WITH DAMMAH, WAW WITH ALEF ABOVE, WAW WITH INVERTED SMALL V, WAW WITH 2 DOTS ABOVE, WAW WITH 3 DOTS ABOVE
DAL	د	ـد	THAL, DAL WITH SMALL TAH, DALL WITH RING, DALL WITH DOT BELOW, DAL WITH DOT BELOW AND SMALL TAH, DAL WITH 2 DOTS, DAL WITH 2 DOTS BELOW, DALL WITH 3 DOTS ABOVE, DALL WITH 3 DOTS ABOVE DOWNWARD, DAL WITH 4 DOTS ABOVE,
RA	ر	ـر	ZAIN, RA WITH SMALL TAH, RA WITH SMALL V, RA WITH RING, RA WITH DOT BELOW, RA WITH SMALL V BELOW, RA WITH DOT BELOW AND DOT ABOVE, RA WITH 2 DOTS ABOVE, RA WITH 3 DOTS ABOVE, RA WITH 4 DOTS ABOVE
TAA MARBUTAH	ة	ـة	
TAA MARBUTAH GOAL	ة	ـة	HAMZAH ON HA
ALEF MAQSURAH	ى	ـى	YA WITH TAIL
YA BAREE	ے	ـے	HAMZAH ON YA BARREE

Table 6-8. Other Arabic Character Joining Classes

Class	Members
Link-causing	ZERO WIDTH JOINER, TATWEEL
Non-linking	ZERO WIDTH NON-JOINER, and all spacing characters (other than those mentioned above) are non-linking, including: HAMZAH, HIGH HAMZAH, HAMZAT WASL ON ALEF, spaces, digits, punctuation, non-Arabic letters, and so on).
Transparent	All combining marks and format marks are irrelevant, including: FATHATAN, DAMMATAN, FATHAH, DAMMAH, KASRAH, SHADDAH, SUKUN, ALEF ABOVE, RIGHT-LEFT MARK, and so on.

Table 6-9. Indexed Arabic Character Joining Classes

Unic.	Name	Link	Link Group
0622	MADDAH ON ALEF	R	ALEF
0623	HAMZAH ON ALEF	R	ALEF
0624	HAMZAH ON WAW	R	WAW
0625	HAMZAH UNDER ALEF	R	ALEF
0626	HAMZAH ON YA	D	YA
0627	ALEF	R	ALEF
0628	BAA	D	BAA
0629	TAA MARBUTAH	R	TAA MARBUTAH
062A	TAA	D	BAA
062B	THAA	D	BAA
062C	JEEM	D	HAA

Table 6-9. Indexed Arabic Character Joining Classes

Unic.	Name	Link	Link Group
062D	HAA	D	HAA
062E	KHAA	D	HAA
062F	DAL	R	DAL
0630	THAL	R	DAL
0631	RA	R	RA
0632	ZAIN	R	RA
0633	SEEN	D	SEEN
0634	SHEEN	D	SEEN
0635	SAD	D	SAD
0636	DAD	D	SAD
0637	TAH	D	TAH
0638	DHAH	D	TAH
0639	AIN	D	AIN
063A	GHAIN	D	AIN
0640	TATWEEL	C	<no shaping>
0641	FA	D	FA
0642	QAF	D	QAF
0643	CAF	D	CAF
0644	LAM	D	LAM
0645	MEEM	D	MEEM
0646	NOON	D	NOON
0647	HA	D	HA
0648	WAW	R	WAW
0649	ALEF MAQSURAH	R	ALEF MAQSURAH
064A	YA	D	YA
0671	HAMZAT WASL ON ALEF	U	<no shaping>
0672	WAVY HAMZAH ON ALEF	R	ALEF
0673	WAVY HAMZAH UNDER ALEF	R	ALEF
0674	HIGH HAMZAH	U	<no shaping>
0675	HIGH HAMZAH ALEF	R	ALEF
0676	HIGH HAMZAH WAW	R	WAW
0677	HIGH HAMZAH WAW WITH DAMMAH	R	WAW
0678	HIGH HAMZAH YA	D	YA
0679	TAA WITH SMALL TAH	D	BAA
067A	TAA WITH 2 DOTS VERTICAL ABOVE	D	BAA
067B	BAA WITH 2 DOTS VERTICAL BELOW	D	BAA
067C	TAA WITH RING	D	BAA
067D	TAA WITH 3 DOTS ABOVE DOWNWARD	D	BAA
067E	TAA WITH 3 DOTS BELOW	D	BAA
067F	TAA WITH 4 DOTS ABOVE	D	BAA
0680	BAA WITH 4 DOTS BELOW	D	BAA
0681	HAMZAH ON HAA	D	HAA
0682	HAA WITH 2 DOTS VERTICAL ABOVE	D	HAA
0683	HAA WITH MIDDLE 2 DOTS	D	HAA
0684	HAA WITH MIDDLE 2 DOTS VERTICAL	D	HAA
0685	HAA WITH 3 DOTS ABOVE	D	HAA
0686	HAA WITH MIDDLE 3 DOTS DOWNWARD	D	HAA
0687	HAA WITH MIDDLE 4 DOTS	D	HAA
0688	DAL WITH SMALL TAH	R	DAL
0689	DAL WITH RING	R	DAL
068A	DAL WITH DOT BELOW	R	DAL
068B	DAL WITH DOT BELOW AND SMALL TAH	R	DAL
068C	DAL WITH 2 DOTS ABOVE	R	DAL
068D	DAL WITH 2 DOTS BELOW	R	DAL
068E	DAL WITH 3 DOTS ABOVE	R	DAL
068F	DAL WITH 3 DOTS ABOVE DOWNWARD	R	DAL
0690	DAL WITH 4 DOTS ABOVE	R	DAL
0691	RA WITH SMALL TAH	R	RA
0692	RA WITH SMALL V	R	RA
0693	RA WITH RING	R	RA
0694	RA WITH DOT BELOW	R	RA

Table 6-9. Indexed Arabic Character Joining Classes

Unic.	Name	Link	Link Group
0695	RA WITH SMALL V BELOW	R	RA
0696	RA WITH DOT BELOW AND DOT ABOVE	R	RA
0697	RA WITH 2 DOTS ABOVE	R	RA
0698	RA WITH 3 DOTS ABOVE	R	RA
0699	RA WITH 4 DOTS ABOVE	R	RA
069A	SEEN WITH DOT BELOW AND DOT ABOVE	D	SEEN
069B	SEEN WITH 3 DOTS BELOW	D	SEEN
069C	SEEN WITH 3 DOTS BELOW AND 3 DOTS ABOVE	D	SEEN
069D	SAD WITH 2 DOTS BELOW	D	SAD
069E	SAD WITH 3 DOTS ABOVE	D	SAD
069F	TAH WITH 3 DOTS ABOVE	D	TAH
06A0	AIN WITH 3 DOTS ABOVE	D	AIN
06A1	DOTLESS FA	D	FA
06A2	FA WITH DOT MOVED BELOW	D	FA
06A3	FA WITH DOT BELOW	D	FA
06A4	FA WITH 3 DOTS ABOVE	D	FA
06A5	FA WITH 3 DOTS BELOW	D	FA
06A6	FA WITH 4 DOTS ABOVE	D	FA
06A7	QAF WITH DOT ABOVE	D	QAF
06A8	QAF WITH 3 DOTS ABOVE	D	QAF
06A9	OPEN CAF	D	GAF
06AA	SWASH CAF	D	SWASH CAF
06AB	CAF WITH RING	D	GAF
06AC	CAF WITH DOT ABOVE	D	CAF
06AD	CAF WITH 3 DOTS ABOVE	D	CAF
06AE	CAF WITH 3 DOTS BELOW	D	CAF
06AF	GAF	D	GAF
06B0	GAF WITH RING	D	GAF
06B1	GAF WITH 2 DOTS ABOVE	D	GAF
06B2	GAF WITH 2 DOTS BELOW	D	GAF
06B3	GAF WITH 2 DOTS VERTICAL BELOW	D	GAF
06B4	GAF WITH 3 DOTS ABOVE	D	GAF
06B5	LAM WITH SMALL V	D	LAM
06B6	LAM WITH DOT ABOVE	D	LAM
06B7	LAM WITH 3 DOTS ABOVE	D	LAM
06BA	DOTLESS NOON	D	NOON
06BB	DOTLESS NOON WITH SMALL TAH	D	NOON
06BC	NOON WITH RING	D	NOON
06BD	NOON WITH 3 DOTS ABOVE	D	NOON
06BE	KNOTTED HA	D	KNOTTED HA
06C0	HAMZAH ON HA	R	TAA MARBUTAH
06C1	HA GOAL	D	HA GOAL
06C2	HAMZAH ON HA GOAL	R	HAMZAH ON HA GOAL
06C3	TAA MARBUTAH GOAL	R	HAMZAH ON HA GOAL
06C4	WAW WITH RING	R	WAW
06C5	WAW WITH BAR	R	WAW
06C6	WAW WITH SMALL V	R	WAW
06C7	WAW WITH DAMMAH	R	WAW
06C8	WAW WITH ALEF ABOVE	R	WAW
06C9	WAW WITH INVERTED SMALL V	R	WAW
06CA	WAW WITH 2 DOTS ABOVE	R	WAW
06CB	WAW WITH 3 DOTS ABOVE	R	WAW
06CC	DOTLESS YA	D	YA
06CD	YA WITH TAIL	R	ALEF MAQSURAH
06CE	YA WITH SMALL V	D	YA
06D0	YA WITH 2 DOTS VERTICAL BELOW	D	YA
06D1	YA WITH 3 DOTS BELOW	D	YA
06D2	YA BARREE	R	YA BARREE
06D3	HAMZAH ON YA BARREE	R	YA BARREE
06D5	AE	U	<no shaping>

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

U+060C	→	U+061F	Arabic-unique punctuation from 8859-6
U+0621	→	U+064A	Arabic letters and tatweel from 8859-6
U+064B	→	U+0652	Combining characters from 8859-6
U+0660	→	U+0669	Arabic-Indic digits
U+066A	→	U+066D	Arabic punctuation
U+0670			Additional combining character
U+0671	→	U+06D5	Extended Arabic characters
U+06D6	→	U+06ED	Extended Arabic characters (Koranic extensions)
U+06F0	→	U+06F9	Eastern Arabic-Indic digits

Devanagari: U+0900—U+097F

The Devanagari script is used for writing classical Sanskrit and its modern historical derivative, Hindi. Extensions to Devanagari are used to write other related languages of India (such as Marathi) and of Nepal (Nepali). In addition, the Devanagari script is used to write the following languages: Awadhi, Bagheli, Bhatneri, Bhili, Bihari, Braj Bhasha, Chhattisgarhi, Garhwali, Gondi (Betul, Chhindwara, and Mandla dialects), Harauti, Ho, Jaipuri, Kachchhi, Kanauji, Konkani, Kului, Kumaoni, Kurku, Kurukh, Marwari, Mundari, Newari, Palpa, and Santali.

All other Indian scripts, as well as the Sinhala script of Sri Lanka and the Southeast Asian scripts (Thai, Lao, Khmer, and Burmese), are historically connected with the Devanagari script as descendants of the ancient Brahmi script, and the entire family of scripts shares a large number of structural features.

The principles of the Indian scripts are covered in some detail in this introduction to the Devanagari script. The remaining introductions to the Indian scripts are abbreviated but highlight any differences from Devanagari where appropriate.

Standards. The Devanagari block of the Unicode Standard is based on ISCII-1988 (Indian Standard Code for Information Interchange). The ISCII standard of 1988 differs from and is an update of earlier ISCII standards issued in 1983 and in 1986.

The Unicode Standard encodes Devanagari characters in the same relative position as those coded in positions A0-$F4_{16}$ in the ISCII-1988 standard. The same character code layout is followed for eight other Indian scripts in the Unicode Standard: Bengali, Gurmukhi, Gujarati, Oriya, Tamil, Telugu, Kannada, and Malayalam. This parallel code layout emphasizes the structural similarities of the Brahmi scripts and follows the stated intention of the Indian coding standards to enable one-to-one mappings between analogous coding positions in different scripts in the family. Sinhala, Thai, Lao, Khmer, and Burmese depart to a greater extent from the Devanagari structural pattern, so the Unicode Standard does not attempt to provide any direct mappings for these scripts to the Devanagari order.

In November 1991, at the time the Unicode Standard, Version 1.0, was published, the Bureau of Indian Standards published a new version of ISCII in Indian Standard (IS) 13194:1991. This new version partially modified the layout and repertoire of the ISCII-1988 standard. Because of these events, the Unicode Standard does not precisely follow the layout of the current version of ISCII. Nevertheless, the Unicode Standard remains a superset of the ISCII-1991 repertoire except for a number of new Vedic extension characters defined in IS 13194:1991 *Annex G—Extended Character Set for Vedic.* Modern, non-Vedic texts encoded with ISCII-1991 may be automatically converted to Unicode code values and back to their original encoding without loss of information.

Encoding Principles. The writing systems that employ Devanagari and other Indian scripts constitute a cross between syllabic writing systems and phonemic writing systems (alphabets). The effective unit of these writing systems is the orthographic syllable, consisting of a consonant and vowel (*CV*) core and, optionally, one or more preceding consonants, with a canonical structure of ((*C*)*C*)*CV*. The orthographic syllable need not correspond exactly with a phonological syllable, especially when a consonant cluster is involved, but the writing system is built on phonological principles and tends to correspond quite closely to pronunciation.

The orthographic syllable is built up of alphabetic pieces, the actual letters of the Devanagari script. These consist of three distinct character types: consonant letters, independent vowels, and dependent vowel signs. In a text sequence, these characters are stored in logical (phonetic) order.

Rendering Devanagari Characters. Devanagari characters, like characters from many other scripts, can combine or change shape depending on their context. A character's appearance is affected by its ordering with respect to other characters, the font used to render the character, and the application or system environment. These variables can cause the appearance of Devanagari characters to be different from their nominal glyphs (used in the code charts).

Additionally, a few Devanagari characters cause a change in the order of the displayed characters. This reordering is not commonly seen in non-Indic scripts and occurs independently of any bidirectional character reordering that might be required.

Consonant Letters. The consonant letters each represent a single consonantal sound but also have the peculiarity of having an *inherent vowel*, generally the short vowel /a/ in Devanagari and the other Indian scripts. Thus, U+0915 DEVANAGARI LETTER KA represents not just /k/ but /ka/. In the presence of a dependent vowel, however, the inherent vowel associated with a consonant letter is overridden by the dependent vowel.

Consonant letters may also be rendered as *half-forms*, which are presentation forms used to depict the initial consonant in consonant clusters. These half-consonant forms do not have an inherent vowel. Their rendered forms in Devanagari often resemble the full consonant but are missing the vertical stem, which marks a syllabic core. (The stem glyph is graphically and historically related to the sign denoting the inherent /a/ vowel.)

Some Devanagari consonant letters are depicted with alternate presentation forms whose choice depends upon neighboring consonants. This is especially true of U+0930 DEVANAGARI LETTER RA, which has numerous different forms, both as the initial and as the final element of a consonant cluster. Only the nominal forms rather than the contextual alternates are depicted in the code chart. In certain cases, however, more than one nominal form is depicted for a single character, where a common stylistic alternate of a nominal form exists.

The traditional Sanskrit/Devanagari alphabetic encoding order for consonants follows articulatory phonetic principles, starting with velar consonants and moving forward to bilabial consonants, followed by liquids and then fricatives. ISCII and the Unicode Standard both observe this traditional order.

Independent Vowel Letters. The independent vowels in Devanagari are letters that stand on their own. The writing system treats independent vowels as orthographic CV syllables in which the consonant is null. The independent vowel letters are used to write syllables which start with a vowel.

Dependent Vowel Signs (Matras). The dependent vowels serve as the common manner of writing non-inherent vowels and are generally referred to as *vowel signs*, or as *matras* in Sanskrit. The dependent vowels do not stand alone; rather, they are visibly depicted in combination with a base letterform. A single consonant, or a consonant cluster, may have a dependent vowel applied to it to indicate the vowel quality of the syllable, when it is different from the inherent vowel. Explicit appearance of a dependent vowel in a syllable overrides the inherent vowel of a single consonant letter.

The greatest variation among different Indian scripts is found in the way that the dependent vowels are applied to base letterforms. Devanagari has a collection of non-spacing dependent vowel signs that may appear above or below a consonant letter, as well as spacing dependent vowel signs that may occur to the right or to the left of a consonant letter or consonant cluster. Other Indian scripts generally have one or more of these forms, but what is a non-spacing mark in one script may be a spacing mark in another. Also, some of the Indian scripts have single dependent vowels that are indicated by two or more glyph components—and those glyph components may *surround* a consonant letter both to the left and right or may occur both above and below it.

The Devanagari script has only one character denoting a left-side dependent vowel sign: U+093F DEVANAGARI VOWEL SIGN I. Other Indian scripts either have no such vowel signs (Telugu and Kannada) or as many as three of these signs (Bengali, Tamil, and Malayalam).

A one-to-one correspondence exists between the independent vowels and the dependent vowel signs. Independent vowels are sometimes represented by a sequence consisting of the independent form of the vowel /a/ followed by a dependent vowel sign. For example, Figure 6-10 illustrates this relationship (see the notation formally described in the "Rules for Rendering" later in this section).

Figure 6-10. Dependent versus Independent Vowels

/a/ + Dependent Vowel				Independent Vowel
$A_n + I_{lvs}$	→	$I_{lvs} + A_n$	≈	I_n
अ + ि	→	िअ	≈	इ
$A_n + U_{vs}$	→	$A_n + U_{vs}$	≈	U_n
अ + ु	→	अु	≈	उ

The combination of the independent form of the default vowel /a/ (in the Devanagari script, U+0905 DEVANAGARI LETTER A) with a dependent vowel sign may be viewed as an alternative spelling of the phonetic information normally represented by an isolated independent vowel form. However, these two representations should not be considered equivalent for the purposes of rendering. Higher-level text processes may choose to consider these alternative spellings equivalent in terms of information content; however, such an equivalence is not stipulated by this standard.

Virama. Devanagari and other Indian scripts employ a sign known as the *virama*, *halant*, or vowel omission sign. A *virama* sign (for example, U+094D DEVANAGARI SIGN VIRAMA) nominally serves to cancel (or kill) the inherent vowel of the consonant to which it is applied. The *virama* functions as a combining character, its shape varying from script to script. When a consonant has lost its inherent vowel by the application of *virama*, it is known as a *dead consonant*; in contrast, a *live consonant* is a consonant that retains its inherent vowel or is written with an explicit dependent vowel sign. In the Unicode Standard, a dead consonant is defined as a sequence consisting of a consonant letter followed by a *virama*. The default rendering for a dead consonant is to position the *virama* as a combining mark bound to the consonant letterform.

For example, if C_n denotes the nominal form of consonant C, and C_d denotes the dead consonant form, then a dead consonant is encoded as shown in Figure 6-11

Figure 6-11. Dead Consonants

$$TA_n + VIRAMA_n \rightarrow TA_d$$

त + ् → त्

Consonant Conjuncts. The Indian scripts are noted for a large number of consonant conjunct forms that serve as orthographic abbreviations (ligatures) of two or more adjacent

letterforms. This abbreviation takes place only in the context of a *consonant cluster.* An orthographic consonant cluster is defined as a sequence of characters which represent one or more dead consonants (denoted C_d) followed by a normal, *live* consonant letter (denoted C_l) or an independent vowel letter.

Under normal circumstances, a consonant cluster is depicted with a conjunct glyph if such a glyph is available in the current font(s). In the absence of a conjunct glyph, the one or more dead consonants that form part of the cluster are depicted using half-form glyphs; or, in the absence of half-form glyphs, the dead consonants are depicted using the nominal consonant forms combined with visible *virama* signs (see Figure 6-12).

Figure 6-12. Conjunct Formations

(1) $GA_d + GHA_l \rightarrow GA_h + GHA_n$

ग् + घ → ग्घ

(2) $KA_d + KA_l \rightarrow K.KA_n$

क् + क → क्क

(3) $KA_d + SSHA_l \rightarrow K.SSHA_n$

क् + ष → क्ष

(4) $RA_d + RI_n \rightarrow RI_n + RA_{sup}$

र् + ऋ → र्ऋ

Anumber of types of conjunct formations appear in these examples: (1) a half-form of *GA* in its combination with the full form of *GHA*; (2) a vertical conjunct *K.KA*; (3) a fully ligated conjunct *K.SSHA,* in which the components are no longer distinct; and (4) a rare conjunct formed with an independent vowel letter, in this case the vowel letter *RI* (also known as *VOCALIC R*). Note that in example (4), the dead consonant RA_d is depicted with the non-spacing combining mark RA_{sup} (*repha*).

A well-designed Indian script font may contain hundreds of conjunct glyphs, but they are not encoded as Unicode characters because they are the result of ligation of distinct letters. Indian script rendering software must be able to map appropriate combinations of characters in context to the appropriate conjunct glyphs in fonts. (See "Rendering of Devanagari Script" later in this section.)

When an independent vowel appears as the terminal element of a consonant cluster, as in example (4) in Figure 6-12, the independent vowel should not be depicted as a dependent vowel sign, but as an independent vowel letterform.

Explicit Virama. Normally a *virama* character serves to create dead consonants which are, in turn, combined with subsequent consonants in order to form conjuncts. This behavior usually results in a *virama* sign not being depicted visually. Occasionally, however, this default behavior is not desired when a dead consonant should be excluded from conjunct formation, in which case the *virama* sign is visibly rendered. In order to accomplish this, the Unicode Standard adopts the convention of placing the character U+200C ZERO WIDTH NON-JOINER immediately after the encoded dead consonant that is to be excluded from conjunct formation. In this case, the *virama* sign is always depicted as appropriate for the consonant to which it is attached.

For example, in Figure 6-13, the use of ZERO WIDTH NON-JOINER prevents the default formation of the conjunct form क्ष (K.SSHA$_n$).

Explicit Half-Consonants. When a dead consonant participates in forming a conjunct, the dead consonant form is often absorbed into the conjunct form, such that it is no longer distinctly visible. In other contexts, however, the dead consonant may remain visible as a *half-consonant form.* In general, a half-consonant form is distinguished from the nominal

Figure 6-13. Preventing Conjunct Forms

$KA_d + ZWNJ + SSHA_l \rightarrow KA_d + SSHA_n$

क् + [ZW NJ] + ष → क्ष

consonant form by the loss of its inherent vowel stem, a vertical stem appearing to the right-side of the consonant form. In other cases, the vertical stem remains but some part of its right-side geometry is missing.

In certain cases, it is desirable to prevent a dead consonant from full conjunct formation yet still not appear with an explicit *virama*. In these cases, the half-form of the consonant is used. In order to explicitly encode a half consonant form, the Unicode Standard adopts the convention of placing the character U+200D ZERO WIDTH JOINER immediately after the encoded dead consonant. The ZERO WIDTH JOINER denotes a non-visible letter that presents linking or cursive joining behavior on either side (that is, to the previous or following letter). Therefore, in the present context, the ZERO WIDTH JOINER may be considered to present a context to which a preceding dead consonant may join in order to create the half-form of the consonant.

For example, if C_h denotes the half-form glyph of consonant C, then a half consonant form is encoded as shown in Figure 6-14.

Figure 6-14. Half-Consonants

$KA_d + ZWJ + SSHA_l \rightarrow KA_h + SSHA_n$

क् + [ZW J] + ष → क्ष

➥ In the absence of the ZERO WIDTH JOINER, this sequence would normally produce the full conjunct form क्ष ($K.SSHA_n$).

This encoding of half consonant forms also applies in the absence of a base letterform; that is, this technique may also be used to encode independent half-forms, as shown in Figure 6-15.

Figure 6-15. Independent Half-Forms

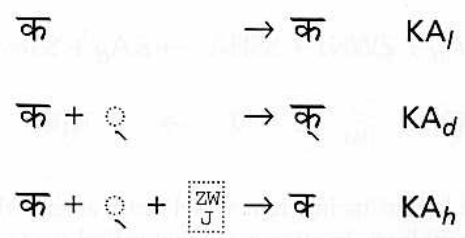
$GA_d + ZWJ \rightarrow GA_h$

ग् + [ZW J] → ग

Consonant Forms. In summary, each consonant may be encoded such that it denotes a live consonant, a dead consonant that may be absorbed into a conjunct, or the half-form of a dead consonant (see Figure 6-16).

Rules for Rendering. The following provides more formal and complete rules for minimal rendering of Devanagari as part of a plain text sequence. It describes the mapping between Unicode characters and the glyphs in a Devanagari font. It also describes the combining and ordering of those glyphs.

Figure 6-16. Consonant Forms



These rules provide minimal requirements for legibly rendering interchanged Devanagari text. As with any script, a more complex procedure can add rendering characteristics, depending on the font and application.

It is important to emphasize that in a font that is capable of rendering Devanagari, the set of glyphs is greater than the number of Devanagari Unicode characters.

Notation. In the next set of rules, the following notation applies:

C_n	Nominal glyph form of consonant C as it appears in the code charts.
C_l	A live consonant, depicted identically to C_n.
C_d	Glyph depicting the dead consonant form of consonant C.
C_h	Glyph depicting the half consonant form of consonant C.
L_n	Nominal glyph form of a conjunct ligature consisting of two or more component consonants. A conjunct ligature composed of two consonants X and Y is also denoted $X.Y_n$.
RA_{sup}	A non-spacing combining mark glyph form of the U+0930 DEVANAGARI LETTER RA positioned above or attached to the upper part of a base glyph form. This form is also known as *REPHA*.
RA_{sub}	A non-spacing combining mark glyph form of the U+0930 DEVANAGARI LETTER RA positioned below or attached to the lower part of a base glyph form.
V_{vs}	Glyph depicting the dependent vowel sign form of a vowel V.
$VIRAMA_n$	The nominal glyph form non-spacing combining mark depicting U+094D DEVANAGARI SIGN VIRAMA.

➥ A *virama* character is not always depicted; when it is depicted, it adopts this non-spacing mark form.

Dead Consonant Rule. The following rule logically precedes the application of any other rule in order to form a dead consonant. Once formed, a dead consonant may be subject to other rules described next.

R1 ***When a consonant C_n precedes a $VIRAMA_n$, it is considered to be a dead consonant C_d. A consonant C_n that does not precede $VIRAMA_n$ is considered to be a live consonant C_l.***

$$TA_n + VIRAMA_n \rightarrow TA_d$$

त + ् → त्

Consonant RA Rules. The character U+0930 DEVANAGARI LETTER RA takes one of a number of visual forms depending on its context in a consonant cluster. By default, this letter is depicted with its nominal glyph form (as shown in the code charts); however, in two contexts, it is depicted using a non-spacing glyph form that combines with a base letterform.

R2 ***If the dead consonant RA_d precedes either a consonant or an independent vowel, then it is replaced by the superscript non-spacing mark RA_{sup}, which is positioned so that it applies to the logically subsequent element in the memory representation.***

$$RA_d + KA_l \rightarrow KA_l + RA_{sup}$$ *Displayed Output*

र् + क → क + ◌ं → र्क

$$RA^1_d + RA^2_d \rightarrow RA^2_d + RA^1_{sup}$$

र् + र् → र् + ◌ं → र्र्

R3 ***If the superscript mark RA_{sup} is to be applied to a dead consonant and that dead consonant is combined with another consonant to form a conjunct ligature, then the mark is positioned so that it applies to the conjunct ligature form as a whole.***

$$RA_d + JA_d + NYA_l \rightarrow J.NYA_n + RA_{sup}$$ *Displayed Output*

र् + ज् + ञ → ज्ञ + ◌ं → र्ज्ञ

R4 ***If the superscript mark RA_{sup} is to be applied to a dead consonant that is subsequently replaced by its half-consonant form, then the mark is positioned so that it applies to the form that serves as the base of the consonant cluster.***

$$RA_d + GA_d + GHA_l \rightarrow GA_h + GHA_l + RA_{sup}$$ *Displayed Output*

र् + ग् + घ → ग + घ + ◌ं → ग्र्घ

R5 ***If the dead consonant RA_d precedes*** ZERO WIDTH JOINER, ***then the half consonant form RA_h, known as the eyelash-RA, is used instead of RA_{sup}. This form of RA is commonly used in writing certain languages such as Marathi.***

$$RA_d + ZWJ \rightarrow RA_h$$

र् + ZWJ → ⟌

R6 ***Except for the dead consonant RA_d, when a dead consonant C_d precedes the live consonant RA_l, then C_d is replaced with its nominal form C_n, and RA is replaced***

by the subscript non-spacing mark RA_{sub}, which is positioned so that it applies to C_n.

$$THA_d + RA_l \rightarrow THA_n + RA_{sub}$$ *Displayed Output*

ठ् + र → ठ + ◌् → ठ्र

R7 ***For certain consonants, the mark RA_{sub} may graphically combine with the consonant to form a conjunct ligature form. These combinations, such as the one shown here, are further addressed by the Ligature Rules described shortly.***

$$PHA_d + RA_l \rightarrow PHA_n + RA_{sub}$$ *Displayed Output*

फ् + र → फ + ◌् → फ्र

R8 ***If a dead consonant (other than RA_d) precedes RA_d, then the substitution of RA for RA_{sub} is performed as described above; however, the VIRAMA that formed RA_d remains so as to form a dead consonant conjunct form.***

$$TA_d + RA_d \rightarrow TA_n + RA_{sub} + VIRAMA_n \rightarrow T.RA_d$$

त् + र् → त + ◌् + ◌् → त्र्

A dead consonant conjunct form that contains an absorbed RA_d may subsequently combine to form a multipart conjunct form.

$$T.RA_d + YA_l \rightarrow T.R.YA_n$$

त्र् + य → त्र्य

Modifier Mark Rules. In addition to vowel signs, three other types of combining marks may be applied to a component of an orthographic syllable or to the syllable as a whole. These three types of marks are *nukta*, *bindus*, and *svaras*.

R9 ***The nukta sign, which modifies a consonant form, is placed immediately after the consonant in the memory representation and is attached to that consonant in rendering. If the consonant represents a dead consonant, then NUKTA should precede VIRAMA in the memory representation.***

$$KA_n + NUKTA_n + VIRAMA_n \rightarrow QA_d$$

क + ◌़ + ◌् → क़्

R10 ***The other modifying marks, bindus and svaras, apply to the orthographic syllable as a whole and should follow (in the memory representation) all other characters that constitute the syllable. In particular, the bindus should follow any vowel signs, and the svaras should come last. The relative placement of these marks is horizontal rather than vertical; the horizontal rendering order may vary according to typographic concerns.***

$$KA_n + AA_{vs} + CANDRABINDU_n$$

क + ◌ा + ◌ँ → काँ

Ligature Rules. Subsequent to the application of the rules just described, a set of rules governing ligature formation apply. The precise application of these rules depends on the availability of glyphs in the current font(s) being used to display the text.

R11 ***If a dead consonant immediately precedes another dead consonant or a live consonant, then the first dead consonant may join the subsequent element to form a two-part conjunct ligature form.***

$JA_d + NYA_l \rightarrow J.NYA_n$ $\quad$ $TTA_d + TTHA_l \rightarrow TT.TTHA_n$

ज् + ञ → ज्ञ $\quad$ ट् + ठ → ट्ठ

R12 ***A conjunct ligature form can itself behave as a dead consonant and enter into further, more complex ligatures.***

$$SA_d + TA_d + RA_n \rightarrow SA_d + T.R.A_n \rightarrow S.T.RA_n$$

स् + त् + र → स् + त्र → स्त्र

A conjunct ligature form can also produce a half-form.

$$K.SSHA_d + YA_l \rightarrow K.SSH_h + YA_n$$

क्ष् + य → क्ष्य

R13 ***If a nominal consonant or conjunct ligature form precedes RA_{sub} as a result of the application of rule R2, then the consonant or ligature form may join with RA_{sub} to form a multipart conjunct ligature (see rule R2 for more information).***

$KA_n + RA_{sub} \rightarrow K.RA_n$ $\quad$ $PHA_n + RA_{sub} \rightarrow PH.RA_n$

क + ◌् → क्र $\quad$ फ + ◌् → फ्र

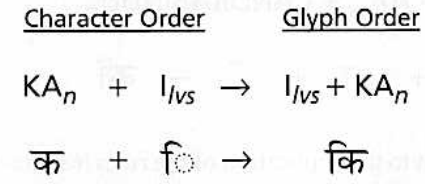
R14 ***In some cases, other combining marks will also combine with a base consonant, either attaching at a non-standard location or changing shape. In minimal rendering there are only two cases, RA_l with U_{vs} or UU_{vs}.***

$RA_l + U_{vs} \rightarrow RU_n$ $\quad$ $RA_l + UU_{vs} \rightarrow RUU_n$

र + ◌ु → रु $\quad$ र + ◌ू → रू

Memory Representation and Rendering Order. The order for storage of plain text in Devanagari and all other Indian scripts generally follows phonetic order; that is, a *CV* syllable with a dependent vowel is always encoded as a consonant letter *C* followed by a vowel sign *V* in the memory representation. This order is employed by the ISCII standard and corresponds with both the phonetic and keying order of textual data (see Figure 6-17).

Figure 6-17. Rendering Order



Since Devanagari and other Indian scripts have some dependent vowels that must be depicted to the left side of their consonant letter, the software that renders the Indian scripts must be able to reorder elements in mapping from the logical (character) store to the presentational (glyph) rendering. For example, if C_n denotes the nominal form of consonant C and V_{lvs} denotes a left-side dependent vowel sign form of vowel V, then a reordering of glyphs with respect to encoded characters occurs as just shown.

R15 ***When the dependent vowel I_{lvs} is used to override the inherent vowel of a syllable, it is always written to the extreme left of the orthographic syllable. If the orthographic syllable contains a consonant cluster, then this vowel is always depicted to the left of that cluster. For example:***

$$TA_d + RA_l + I_{lvs} \rightarrow T.RA_n + I_{lvs} \rightarrow I_{lvs} + T.RA_d$$

त् + र + ि → त्र + ि → त्रि

Sample Half Forms. Table 6-10 shows examples of half-consonant forms that are commonly used with the Devanagari script. These forms are glyphs, not characters. These forms may be encoded explicitly using ZERO WIDTH JOINER as shown; in normal conjunct formation, they may be used spontaneously to depict a dead consonant in combination with subsequent consonant forms.

Table 6-10. Sample Half-Forms

क	◌्	ZWJ	क्	न	◌्	ZWJ	न्
ख	◌्	ZWJ	ख्	प	◌्	ZWJ	प्
ग	◌्	ZWJ	ग्	फ	◌्	ZWJ	फ्
घ	◌्	ZWJ	घ्	ब	◌्	ZWJ	ब्
च	◌्	ZWJ	च्	भ	◌्	ZWJ	भ्
ज	◌्	ZWJ	ज्	म	◌्	ZWJ	म्
झ	◌्	ZWJ	झ्	य	◌्	ZWJ	य्
ञ	◌्	ZWJ	ञ्	ल	◌्	ZWJ	ल्
ण	◌्	ZWJ	ण्	व	◌्	ZWJ	व्
त	◌्	ZWJ	त्	श	◌्	ZWJ	श्
थ	◌्	ZWJ	थ्	ष	◌्	ZWJ	ष्
ध	◌्	ZWJ	ध्	स	◌्	ZWJ	स्

Sample Ligatures. Table 6-11 shows examples of conjunct ligature forms that are commonly used with the Devanagari script. These forms are glyphs, not characters. Not every writing system that employs this script uses all of these forms; in particular, many of these forms are used only in writing Sanskrit texts. Furthermore, individual fonts may provide fewer or more ligature forms than are depicted here.

Table 6-11. Sample Ligatures

क	◌्	क	क्क
क	◌्	त	क्त
क	◌्	र	क्र
क	◌्	ष	क्ष
ङ	◌्	क	ङ्क
ङ	◌्	ख	ङ्ख
ङ	◌्	ग	ङ्ग
ङ	◌्	घ	ङ्घ
ञ	◌्	ज	ञ्ज
ज	◌्	ञ	ज्ञ
द	◌्	घ	द्ध
द	◌्	द	द्द
द	◌्	ध	द्ध
द	◌्	ब	द्ब
द	◌्	भ	द्भ
द	◌्	म	द्म
द	◌्	य	द्य
द	◌्	व	द्व
ट	◌्	ट	ट्ट

ट	◌्	ठ	ट्ठ
ठ	◌्	ठ	ठ्ठ
ड	◌्	ग	ड्ग
ड	◌्	ड	ड्ड
ड	◌्	ढ	ड्ढ
त	◌्	त	त्त
त	◌्	र	त्र
न	◌्	न	न्न
फ	◌्	र	फ्र
श	◌्	र	श्र
ह	◌्	म	ह्म
ह	◌्	य	ह्य
ह	◌्	ल	ह्ल
ह	◌्	व	ह्व
ह		◌ृ	ह्र
र		◌ु	रु
र		◌ू	रू
स	◌्	त्र	स्त्र

Sample Half Ligature Forms. In addition to half form glyphs of individual consonants, half forms are also used to depict conjunct ligature forms. A sample of such forms is shown in Table 6-12. These forms are glyphs, not characters. These forms may be encoded explicitly using ZERO WIDTH JOINER as shown; in normal conjunct formation, they may be used spontaneously to depict a conjunct ligature in combination with subsequent consonant forms.

Combining Marks. Devanagari and other Indian scripts have a number of combining marks that could be considered diacritic. One class of these, known as *bindus*, is represented by U+0901 DEVANAGARI SIGN CANDRABINDU and U+0902 DEVANAGARI SIGN ANUSVARA. These indicate nasalization or final nasal closure of a syllable. U+093C DEVANAGARI SIGN NUKTA is a true diacritic. It is used to extend the basic set of consonant letters by

Table 6-12. Sample Half-Ligature Forms

क	◌्	ष	◌्	ZWJ	क्ष
ज	◌्	ञ	◌्	ZWJ	ज्ञ
त	◌्	त	◌्	ZWJ	त्त
त	◌्	र	◌्	ZWJ	त्र
श	◌्	र	◌्	ZWJ	श्र

modifying them (with a subscript dot in Devanagari) to create new letters. U+0951 → U+0954 are a set of combining marks used in transcription of Sanskrit texts.

Digits. Each Indian script has a distinct set of digits appropriate to that script. These may or may not be used in ordinary text in that script. European digits have displaced the Indian script forms in modern usage in many of the scripts. Some Indian scripts—notably Tamil—lack a distinct digit for zero.

Punctuation and Symbols. U+0964 DEVANAGARI DANDA is similar to full stop. Corresponding forms occur in many other Indian scripts. U+0965 DEVANAGARI DOUBLE DANDA marks the end of a verse in traditional texts.

Many modern languages written in the Devanagari script intersperse punctuation derived from the Latin script. Thus U+002C COMMA and U+002E FULL STOP are freely used in writing Hindi, and the *danda* is usually restricted to more traditional texts.

Encoding Structure. The Unicode Standard organizes the nine principal Indian scripts in blocks of 128 encoding points each. The first six columns in each script are isomorphic with the ISCII-1988 encoding, except that the last eleven positions (U+0955 → U+095F in Devanagari, for example), which are unassigned or undefined in ISCII-1988, are used in the Unicode encoding.

The seventh column in each of these scripts, along with the last eleven positions in the sixth column, represent additional character assignments in the Unicode Standard which are matched across all nine scripts. For example, positions U+xx66 → U+xx6F or U+xxE6 → U+xxEF code the Indic script digits for each script.

The eighth column for each script is reserved for script-specific additions that do not correspond from one Indian script to the next.

The character block for the Devanagari script is divided into the following specific ranges:

U+0901	→	U+0903	Various signs
U+0905	→	U+0914	Independent vowels
U+0915	→	U+0939	Consonants
U+093C	→	U+093D	Various signs
U+093E	→	U+094C	Dependent vowel signs
U+094D			Devanagari *virama*
U+0950	→	U+0954	Various signs
U+0958	→	U+095F	Additional consonants composed with Nukta
U+0960	→	U+0961	Additional independent vowels
U+0962	→	U+0963	Additional dependent vowel signs
U+0964	→	U+0965	Additional punctuation
U+0966	→	U+096F	Devanagari digits
U+0970			Devanagari-specific addition: abbreviation sign

Bengali: U+0980—U+09FF

The Bengali script is a North Indian script closely related to Devanagari. It is used to write the Bengali language primarily in West Bengal state (India) and in the nation of Bangladesh. It is also used to write Assamese in Assam (India) and a number of other minority languages (Daphla, Garo, Hallam, Khasi, Manipuri, Mizo, Naga, Munda, Rian, and Santali) in northeastern India.

Two-Part Vowel Signs. The Bengali script, along with a number of other Indian scripts, makes use of two-part vowel signs; these are vowels in which one half of the vowel is placed on each side of a consonant letter or cluster; for example: U+09CB BENGALI VOWEL SIGN O and U+09CC BENGALI VOWEL SIGN AU. The vowel signs are coded in each case in the position in the charts isomorphic with the corresponding vowel in Devanagari. Hence U+09CC BENGALI VOWEL SIGN AU is isomorphic with U+094C DEVANAGARI VOWEL SIGN AU. In order to provide compatibility with existing implementations of the scripts that use two-part vowel signs, the Unicode Standard explicitly encodes the right half part of these vowel signs; for example, U+09D7 BENGALI AU LENGTH MARK represents the right half part glyph component of U+09CC BENGALI VOWEL SIGN AU.

Special Characters. U+09F2 → U+09F9 are a series of Bengali additions for writing currency and fractions.

Rendering Behavior. For rendering of the Bengali script, see the rules for rendering in the Devanagari block description.

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

U+0980	→	U+09EF	Follow the Devanagari prototype
U+09F0	→	U+09F9	Bengali-specific additions

Gurmukhi: U+0A00—U+0A7F

The Gurmukhi script is a North Indian script historically derived from an older script called Lahnda. It is quite closely related to Devanagari structurally. Gurmukhi is used to write the Punjabi language in the Punjab in India.

Rendering Behavior. For rendering of the Gurmukhi script, see the rules for rendering in the Devanagari block description.

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

U+0A00	→	U+0A6F	Follow the Devanagari prototype
U+0A70	→	U+0A75	Gurmukhi-specific additions: letters, diacritics

Gujarati: U+0A80—U+0AFF

The Gujarati script is a North Indian script closely related to Devanagari. It is most obviously distinguished from Devanagari by not having a horizontal bar for its letterforms, a characteristic of the older Kaithi script to which Gujarati is related. The Gujarati script is used to write the Gujarati language of the Gujarat state in India.

Rendering Behavior. For rendering of the Gujarati script, see the rules for rendering in the Devanagari block description.

Encoding Structure. The block for the Gujarati script is divided into the following ranges:

U+0A80 → U+0AEF Follow the Devanagari prototype

Oriya: U+0B00—U+0B7F

The Oriya script is a North Indian script structurally similar to Devanagari, but with semi-circular lines at the top of most letters instead of the straight horizontal bars of Devanagari. The actual shapes of the letters, particularly for vowel signs, show similarities to Tamil. The Oriya script is used to write the Oriya language, of Orissa state, India, as well as minority languages such as Khondi and Santali.

Special Characters. U+0B57 ORIYA AU LENGTH MARK is provided as an encoding for the right side of the surroundrant vowel U+0B4C ORIYA VOWEL SIGN AU.

Rendering Behavior. For rendering of the Oriya script, see the rules for rendering in the Devanagari block description.

Encoding Structure. The block for the Oriya script is divided into the following ranges:

U+0B00	→	U+0B6F	Follow the Devanagari prototype
U+0B70			Oriya-specific addition

Tamil: U+0B80—U+0BFF

The Tamil script is a South Indian script. South Indian scripts are structurally related to the North Indian scripts, but they are used to write Dravidian languages of southern India and of Sri Lanka, which are genetically unrelated to the North Indian languages such as Hindi, Bengali, and Gujarati. The shapes of letters in the South Indian scripts are generally quite distinct from the shapes of letters in Devanagari and its related scripts. This is partly a result of the fact that the South Indian scripts were originally carved with needles on palm leaves, a technology that apparently favored rounded letter shapes rather than square, block-like shapes.

The Tamil script is used to write the Tamil language of Tamil Nadu state in India as well as minority languages such as Badaga. Tamil is also used in Sri Lanka, Singapore, and parts of Malaysia. The Tamil script has fewer consonants than the other Indian scripts. It also lacks conjunct consonant forms. Instead of conjunct consonant forms, the *virama* (U+0BCD) is normally fully depicted in Tamil text.

Naming Conventions for Mid Vowels. The Unicode character encoding for Tamil uses a distinct set of naming conventions for mid vowels in the South Indian (Dravidian) scripts. These conventions are illustrated by U+0B8E TAMIL LETTER E and U+0B8F TAMIL LETTER EE, to be contrasted with the isomorphic positions in Devanagari: U+090E DEVANAGARI LETTER SHORT E and U+090F DEVANAGARI LETTER E. The Dravidian languages have a regular length distinction in the mid vowels which is not reflected in normal Devanagari. U+090E DEVANAGARI LETTER SHORT E is an addition to Devanagari to enable transcription of the Dravidian short vowel forms. The naming conventions are chosen to best reflect the actual nature of the vowels in question in the Dravidian scripts, as well as in Devanagari and the other North Indian scripts.

Special Characters. U+0BD7 TAMIL AU LENGTH MARK is provided as an encoding for the right side of the surroundrant (or two-part) vowel U+0BCC TAMIL VOWEL SIGN AU.

Rendering of Tamil Script. The South Indic scripts function in much the same way as Devanagari, with the additional feature of two-part vowels. As in the Devanagari example, the words "TAMIL LETTER" and "TAMIL VOWEL SIGN" will be omitted where this does not cause ambiguity.

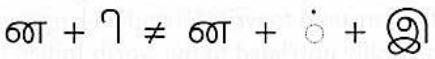
It is important to emphasize that in a font that is capable of rendering Tamil, the set of glyphs is greater than the number of Tamil characters.

Table 6-13 is a summary of the Tamil letters.

Table 6-13. Tamil Letter Summary

க KA	ங NGA	ச CA	ஜ JA	ஞ NYA	ட TTA	ண NNA	த TA	ந NA	ன NNNA	ப PA	
ம MA	ய YA	ர RA	ற RRA	ல LA	ள LLA	ழ LLLA	வ VA	ஷ SSA	ஸ SA	ஹ HA	
அ A	ஆ AA	இ I	ஈ II	உ U	ஊ UU	எ E	ஏ EE	ஐ AI	ஒ O	ஓ OO	ஔ AU
◌ A	◌ா AA	◌ி I	◌ீ II	◌ு U	◌ூ UU	◌ெ E	◌ே EE	◌ை AI	◌ொ O	◌ோ OO	◌ௌ AU
◌் VIRAMA	◌ௗ AU LENGTH										

Independent versus Dependent Vowels. As with Devanagari, the dependent vowel signs are not equivalent to a sequence of *virama* + independent vowel. For example:

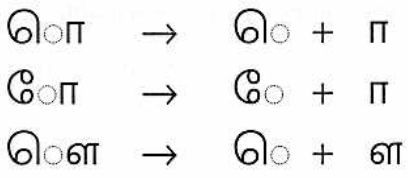


As in the case of Devanagari, a consonant cluster is any sequence of one or more consonants separated by viramas, possibly terminated with a *virama.*

Two-Part Vowels. Certain Indic vowels consist of two discontiguous elements. As in other cases of discontiguous elements, there are two sequences of Unicode values that can be used to express equivalent spellings. This is similar to the case of letters such as "â", which can either be spelled with "a" followed by non-spacing "^", or spelled with a single Unicode character "â".

Note that the ௗ in the third example is *not* U+0BB3 TAMIL LETTER LLA; it is rather U+0BD7 TAMIL AU LENGTH MARK.

If the precomposed forms are used in the memory representation instead of the separate characters, then a similar transformation occurs in the rendering process. The precomposed form on the left is transformed into the two separate forms equivalent to those on the right, which are then subject to vowel reordering, as below. Thus in rendering:



Vowel Reordering. As shown in Table 6-14, the following vowels are always reordered in front of the previous consonant cluster, similar to the rendering behavior of the DEVANAGARI VOWEL SIGN I.

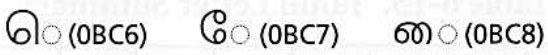


Table 6-14. Vowel Reordering

Memory Representation			Display
க		→	
க		→	
க		→	

The same effect occurs with the results of vowel splitting (see Table 6-15).

Table 6-15. Vowel Splitting and Reordering

Memory Representation				Display
க	◌ொ		→	கொ
க	◌ெ	ா	→	கா
க	◌ோ		→	கோ
க	◌ே	ா	→	கா
க	◌ௌ		→	கௌ
க	◌ெ	ள	→	கௌ

In both cases, the ordering of the elements is *unambiguous*: the consonant (cluster) occurs *first* in the memory representation. The vowel ஔ also has two discontinuous parts and can also be composed using the AU LENGTH MARK.

Ligatures. The following examples illustrate the range of ligatures available in Tamil. These changes take place after vowel reordering and vowel splitting. Unlike Devanagari, there are very few conjunct consonants; most ligatures are located between a vowel and a neighboring consonant.

1. Conjunct consonants.

க + ◌் + ஷ → க்ஷ

As with Devanagari, vowel reordering occurs around conjunct consonants. For example:

க + ◌் + ஷ + ◌ொ + ா → கொக்ஷா

2. The vowel ா optionally ligates with ண, ன, or ற on its left:

ண + ா → ணா

ன + ா → னா

ற + ா → றா

Since this process takes place after reordering and splitting, the following ligatures may also occur:

Separate Vowels	*Precomposed Vowels*
ண + ◌ெ + ா → ணெ	ண + ◌ொ → ணொ
ண + ◌ே + ா → ணே	ண + ◌ோ → ணோ
ன + ◌ெ + ா → னெ	ன + ◌ொ → னொ
ன + ◌ே + ா → னே	ன + ◌ோ → னோ
ற + ◌ெ + ா → றெ	ற + ◌ொ → றொ
ற + ◌ே + ா → றே	ற + ◌ோ → றோ

3. The vowel signs ◌ி and ◌ீ form ligatures with ட on their left.

ட + ◌ி → டி

ட + ◌ீ → டீ

These vowels often change shape or position slightly in order to link up with the appropriate shape of the consonant on their left:

ல + ◌ி → லி

ல + ◌ீ → லீ

4. The vowel signs ◌ெ and ◌ை typically change form or ligate (see Table 6-16).

Table 6-16. Ligating Vowel Signs

x	*x*+◌ெ	*x*+◌ை
க	கு	கூ
ங	ங்	ஙூ
ச	சு	சூ
ஞ	ஞு	ஞூ
ட	டு	டூ
ண	ணு	ணூ
த	து	தூ
ந	நு	நூ
ன	னு	னூ

x	*x*+◌ெ	*x*+◌ை
ப	பு	பூ
ம	மு	மூ
ய	யு	யூ
ர	ரு	ரூ
ற	று	றூ
ல	லு	லூ
ள	ளு	ளூ
ழ	ழு	ழூ
வ	வு	வூ

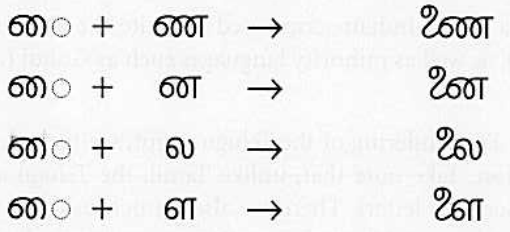
To the right of ஐ, ஃத, ஸ, ஹ, or க்ஷ, these forms have a spacing form (see Table 6-18).

Figure 6-18. Spacing Forms of Vowels

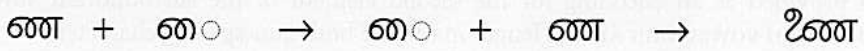
ஐ + ◌ெ → ஐ◌ெ

ஐ + ◌ை → ஐ◌ை

5. The vowel sign ◌ொ changes to ெ◌ to the left of ணா, னா, ல, or ளா.

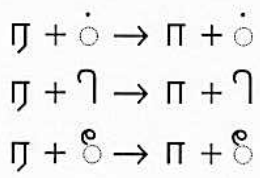


Remember that this change takes place after the vowel reordering; in the first example, the vowel ◌ொ follows ணா in the memory representation. After vowel reordering, it is on the left of ணா, and thus changes form. The complete process is



6. The consonant ர changes shape to ற.

This occurs when the ற form of ர U+0BB0 TAMIL LETTER RA would not be confused with the nominal form ற of U+0BBE TAMIL VOWEL SIGN AA (for example, when ர is combined with ◌ி, ◌ீ, or ◌ூ).



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

U+0B80	→	U+0BEF	Follows the Devanagari prototype
U+0BF0	→	U+0BF2	Tamil-specific additions

Telugu: U+0C00—U+0C7F

The Telugu script is a South Indian script used to write the Telugu language of Andhra Pradesh state in India, as well as minority languages such as Gondi (Adilabad and Koi dialects) and Lambadi.

Rendering Behavior. For rendering of the Telugu script, see the rules for rendering in the Tamil block description. Take note that, unlike Tamil, the Telugu script writes conjunct consonants with subscripted letters. There are also numerous consonant letters with contextual shape changes when used in conjuncts. Some vowel signs also change their shape in specified combinations.

Special Characters. U+0C55 TELUGU LENGTH MARK is provided as an encoding for the second element of the vowel U+0C47 TELUGU VOWEL SIGN EE. U+0C56 TELUGU AI LENGTH MARK is provided as an encoding for the second element of the surroundrant vowel U+0C48 TELUGU VOWEL SIGN AI. The length marks are both non-spacing characters.

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

U+0C00 → U+0C6F Follow the Devanagari prototype

Kannada: U+0C80—U+0CFF

The Kannada script is a South Indian script used to write the Kannada (or Kanarese) language of Karnataka state, as well as minority languages such as Tulu.

The Kannada script is very closely related to the Telugu script both with regard to the shapes of the letters and in the way conjunct consonants behave.

Special Characters. U+0CD5 KANNADA LENGTH MARK is provided as an encoding for the right side of the two-part vowel U+0CC7 KANNADA VOWEL SIGN EE. U+0CD6 KANNADA AI LENGTH MARK is provided as an encoding for the right side of the two-part vowel U+0CC8 KANNADA VOWEL SIGN AI. The Kannada two-part vowels actually consist of a non-spacing element above the consonant letter and one or more spacing letters to the right of the consonant letter.

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

U+0C80 → U+0CEF Follow the Devanagari prototype

Malayalam: U+0D00—U+0D7F

The Malayalam script is a South Indian script used to write the Malayalam language of Kerala state.

The shapes of Malayalam letters closely resemble those of Tamil. However, Malayalam has a very full and complex set of conjunct consonant forms.

Special Characters. U+0D57 MALAYALAM AU LENGTH MARK is provided as an encoding for the right side of the two-part vowel U+0D4C MALAYALAM VOWEL SIGN AU.

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

U+0D00 → U+0D6F Follow the Devanagari prototype

Thai: U+0E00—U+0E7F

The Thai script is used to write Thai and other Southeast Asian languages, such as Kuy, Lavna, and Pali. It is a member of the Indic family of scripts descended from Brahmi. Thai modifies the original Brahmi letter shapes and extends the number of letters to accommodate features unique to the Thai language, including tone marks derived from superscript digits.

Standards. Thai layout in the Unicode Standard is based on the Thai Industrial Standard 620-2529.

General Principles of the Thai Script. In common with the Indic scripts, each Thai letter is a consonant possessing an inherent vowel sound. Thai letters further feature inherent tones. The inherent vowel and tone can be varied by means of modifiers attached to the base consonant letter. These modifier letters consist of combining vowel signs, combining tone marks, and independent vowel letters. The combining signs and marks follow the modified consonant in the memory representation. However, the independent vowels are treated as other independent letters and precede or follow the consonant depending on their visual position. This encoding for Thai differs from the other Indic scripts since the latter always place vowels *after* the consonant that precedes them phonetically. The difference is necessitated by the encoding practice commonly employed with Thai character data as represented by the Thai Industrial Standards.

Thai Punctuation. Common Thai punctuation includes U+0E46 THAI CHARACTER MAIYAMOK to mark repetition of preceding letters and U+0E5A THAI CHARACTER ANGKHANKHU for ellipsis. U+0E5B THAI CHARACTER KHOMUT marks the beginning of religious texts. Thai also uses punctuation marks such as U+002E FULL STOP and U+002C COMMA, encoded in the ASCII and Latin1 blocks.

Thai Transcription of Pali and Sanskrit. The Thai script is frequently used to write Pali and Sanskrit. When so used, consonant clusters are represented by the explicit use of U+0E3A THAI CHARACTER PHINTHU (*virama*) to mark the removal of the inherent vowel. There is no conjoining behavior, unlike other Indic scripts. U+0E4D THAI CHARACTER NIKHAHIT is the Pali *nigghahita* and Sanskrit *anusvara*. U+0E30 THAI CHARACTER SARA A is the Sanskrit visarga. U+0E24 THAI CHARACTER RU and U+0E26 THAI CHARACTER LU are vocalic /r/ and /l/, with U+0E45 THAI CHARACTER LAKKHANGYAO used to indicate their lengthening.

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

U+0E01	→	U+0E2E	Consonant letters
U+0E2F			Punctuation
U+0E30	→	U+0E3A	Vowel signs
U+0E3F			Currency symbol (Baht)
U+0E40	→	U+0E44	Vowel signs
U+0E45	→	U+0E46	Punctuation
U+0E47			Vowel sign
U+0E48	→	U+0E4B	Tone marks (diacritics)
U+0E4C	→	U+0E4D	Vowel signs
U+0E4E	→	U+0E4F	Miscellaneous signs
U+0E50	→	U+0E59	Thai digits
U+0E5A	→	U+0E5B	Miscellaneous signs

Lao: U+0E80—U+0EFF

The Lao language and script are closely related to Thai. The Unicode Standard encodes the Lao script in the same relative order as Thai.

There are a few additional letters in Lao that have no match in Thai. These are

U+0EBB LAO VOWEL SIGH MAI KON

U+0EBC LAO SEMIVOWEL SIGN LO

U+0EBD LAO SEMIVOWEL SIGN NYO

The preceding two semivowel signs are the last remnants of the system of subscript medials, which in Burmese also includes original "rw." In Burmese and Khmer there is a full set of subscript consonant forms used for conjuncts. Thai no longer uses any of these; Lao has just the two.

There are also two ligatures in the Unicode character encoding for Lao: U+0EDC LAO HO NO and U+0EDD LAO HO MO. These correspond to sequences of [h] plus [n] or [h] plus [m] without ligating. Their function in Lao is to provide versions of the [n] and [m] consonants with a different inherent tonal implication.

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

U+0E80	→	U+0ED9	Follows the Thai prototype (see exceptions just noted)
U+0EDC	→	U+0EDD	Lao digraphs

Tibetan: U+0F00—U+0FBF

The Tibetan script is used for writing Tibetan and related languages, such as Ladakhi and Lahuli, spoken in the Himalayan region, including Tibet, Bhutan, India, and Nepal. The Tibetan script is a member of the Indic family of scripts descended from Brahmi. The original Brahmi letter shapes can still be clearly discerned in Tibetan, but Tibetan removes the Brahmi voiced aspirates and adds letters for Tibetan sounds not found in Brahmi.

General Principles of the Tibetan Script. As in all Indic scripts, each Tibetan letter is a consonant containing an inherent vowel sound. Tibetan letters each also contain an inherent tone related to the voicing or non-voicing of the original Brahmi letters; this is not marked in the script. As in other Indic scripts, the inherent vowels are modified by means of floating, non-spacing characters attached to the base letter. Removal of the inherent vowel is not always marked in native Tibetan words and must be determined from context.

Consonant clusters are rendered in Tibetan as conjuncts formed by stacking letters along a vertical axis. Because of the prevalence of this practice and to simplify other operations, the Tibetan script contains two encodings of each consonant, used in the following manner: conjuncts are represented in the text stream by placing one or more of the subjoined letter forms following one of the nominal forms; vowel signs come after the clusters thus formed.

Tibetan Punctuation. Common Tibetan punctuation includes U+0F0D *shey* to mark phrases. *Shey* is doubled (U+0F0E) to mark full stops. U+0F11 *rinchenpungshey* is a decorative variant. U+0F0B *intersyllabic tsek* is a syllable delimiter. There are no interword or interphrase spaces in Tibetan. Line breaks normally occur at word boundaries (which are always marked with U+0F0C *tsek*). When a multisyllable word must be broken at the end of a line, three *tseks* are used to indicate the continuation, in a manner similar to western hyphenation. U+0F08 *drulshey* is sometimes used at the beginning (and less frequently the end) of a text.

The character U+0F04 *goyik* is an honorific flourish, double and triple forms of which are used at the beginnings of texts. It normally joins with one or two more occurrences of the same character to form ligatures and is almost never used alone; it is often followed by *shey* in a decorative form.

The *Wheel of Dharma*, which occurs sometimes in Tibetan texts, is encoded in the Miscellaneous Symbols block at U+2638.

The two characters U+0F3C *left ang khang* and U+0F3D *right ang khang* are paired punctuation, typically used together forming a roof over one or more digits, in which case kerning or special ligatures may be required for proper rendering; they may also be used much as a single closing parenthesis is used in forming lists. The marks U+0F3E *yue chu* and U+0F3F *ye chu* are paired signs used to combine with digits; special glyphs or compositional metrics are required for their use.

Tibetan Half-Numbers. The *half-number* forms (U+0F2A → U+0F33) are peculiar to Tibetan, though other scripts (for example, Bengali) have similar fractional concepts. The value of each half-number is 0.5 less than the number within which they appear.

Tibetan Transcription of Sanskrit. Tibetan is also used to write Sanskrit. The Sanskrit retroflex letters are retained. The voiced aspirates are represented by conjuncts formed of consonants placed above the letter U+0F67 *ha*; the relevant aspirates for Sanskrit are encoded within this block at U+0F93 *subjoined gha*, U+0F9D *subjoined ddha*, U+0FA2 *subjoined dha*, U+0FA7 *subjoined bha*, and U+0FAC *subjoined dzha*. The conjunct *kshr* is at U+0FB9. U+0F7F *namchey* is the *visarga* (see the Devanagari block description), and U+0F7E *ngaro* is the *anusvara*.

To maintain consistency in transliterated texts and for ease in transmission and searching, it is recommended that implementations of Sanskrit in the Tibetan script use the precomposed forms of aspirated letters (and *kshr*) whenever possible, rather than implementing these as completely decomposed stacks. However, implementations must ensure that decomposed stacks and precomposed forms are treated interpreted equivalently (see *Section 3.6, Decomposition*).

When the Tibetan script is used to write Sanskrit, consonants are frequently stacked in ways that do not occur in native Tibetan words; this usually indicates deletion of one or more vowel sounds. The stacking behavior is usually indicated by a number of subjoined consonants following a nominal consonant without any break between them. In some rare cases, head or subjoined forms need to be displayed in a manner conflicting with normal, contextual rendering of the script. In those cases, the *virama* (U+0F84) may be inserted between consonants to signal the other, less common, behavior. The letters *wa* (U+0F5D), *ra* (U+0F62), and *ya* (U+0F61) typically change shape when subscripted via subjoining; the nominal changed forms are shown in the chart for Tibetan. (Unchanged forms are possible through the mechanism of *virama* insertion.)

U+0F09 *enumeration* is a list enumerator used at the start of administrative letters in Bhutan, as is U+0F0A *petition honorific*. U+0F3A TIBETAN MARK GUG RTAGS GYON and U+0F3B TIBETAN MARK GUG RTAGS GYAS are paired punctuation marks (brackets). The sign U+0F39 *trang se* is a non-spacing mark corresponding to the flagged ends on characters such as U+0F59 *tsa*; it is sometimes used to form new letters for sounds that do not occur in the traditional script, such as *fa* and *va*.

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

U+0F00	→	U+0F1F,	Tibetan syllables, punctuation, and symbols
U+0F34	→	U+0F3F	
U+0F20	→	U+0F29	Tibetan digits
U+0F2A	→	U+0F33	Tibetan half-numbers
U+0F40	→	U+0F69	Tibetan nominal letterforms
U+0F71	→	U+0F8B	Tibetan combining vowels and other combining marks
U+0F90	→	U+0FB9	Tibetan subjoined letterforms

Georgian: U+10A0—U+10FF

The Georgian script is used primarily for writing the Georgian language. Upper- and lowercase pairs exist primarily in archaic forms of the script.

Archaic Script Form. The modern Georgian script is a lowercase style called *mkhedruli* (soldier's). It originated as the secular derivative of a form called *khutsuri* (ecclesiastical) that had uppercase and lowercase pairs. Although no longer used in most modern texts, the *khutsuri* style is still used for liturgical purposes; the Unicode Standard encodes the uppercase form of *khutsuri* as well as the lowercase letters of modern Georgian.

Georgian Paragraph Separator. The Georgian paragraph separator has a distinct representation, so it has been separately encoded at U+10FB. It is intended to be used in conjunction with U+2029 PARAGRAPH SEPARATOR, rather than as a replacement for it. See the discussion of the *paragraph separator* in the General Punctuation block.

Other Punctuation. For the Georgian full stop, use U+0589 ARMENIAN FULL STOP.

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

U+10A0	→	U+10C5	Historic (uppercase) letters
U+10D0	→	U+10F0	Modern letters
U+10F1	→	U+10F6	Archaic (lowercase) letters
U+10FB			Punctuation

Hangul Jamo: U+1100—U+11FF

Korean Hangul may be considered to be a syllabic script. As opposed to many other syllabic scripts, the syllables are formed from a set of alphabetic components in a regular fashion. These alphabetic components are called jamo.

The Unicode Standard contains both the complete set of precomposed modern Hangul syllable blocks, and the set of conjoining Hangul jamo in this block. This set of conjoining Hangul jamo can be used to encode all modern and ancient syllable blocks. For a description of conjoining jamo behavior and precomposed Hangul Syllables, see *Section 3.10, Combining Jamo Behavior,* and the Hangul Syllables character block description (U+AC00 → U+D7A3).

The Hangul jamo are divided into three classes: *choseong* (leading consonants, or syllable-initial characters), *jungseong* (vowels, or syllable-peak characters), and *jongseong* (trailing consonants, or syllable-final characters). In the following discussion, these can be abbreviated by *L* (leading consonant), *V* (vowel), and *T* (trailing consonant).

For use in composition, there are two invisible filler characters that act as placeholders for *choseong* or *jungseong*: U+115F HANGUL CHOSEONG FILLER and U+1160 HANGUL JUNGSEONG FILLER.

Collation. The unit of collation in Korean text is normally the Hangul syllable block. Because of the arrangement of the conjoining jamo, their sequences may be collated with a binary comparison. For example, in comparing (a) *LVTLV* against (b) *LVLV*, the first syllable block (*LVT*) should be compared against the second (*LV*). Supposing the first two characters are identical—since all trailing consonants have binary values greater than all leading consonants—the *T* would compare as greater than the second *L* in (b). This produces the correct ordering between the strings. The positions of the fillers in the code charts were also chosen with this in mind.

- As with any coded characters, collation cannot depend simply on a binary comparison. Odd sequences such as superfluous fillers will produce an incorrect sort, as will cases where a non-jamo character follows a sequence (such as comparing *LVT* against *LVx*, where *x* is a Unicode character above U+11FF, such as U+3000 IDEOGRAPHIC SPACE).

If mixtures of precomposed syllable blocks and jamo are collated, the easiest approach is to decompose the precomposed syllable blocks into conjoining jamo before comparing.

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

U+1100	→	U+1159	Choseong (leading consonants)
U+115F			CHOSEONG FILLER (leading filler)
U+1160			JUNGSEONG FILLER (vowel filler)
U+1161	→	U+11A2	Jungseong (vowels)
U+11A8	→	U+11F9	Jongseong (trailing consonants)

Latin Extended Additional: U+1E00—U+1EFF

The characters in this block constitute a number of precomposed combinations of Latin letters with one or more general diacritical marks. These characters were added to the Unicode Standard as a result of the process of merging the Unicode Standard with the developing ISO 10646 standard. Each of the characters contained in this block may be alternatively representated with a base letter followed by one or more general diacritical mark characters found in the Combining Diacritical Marks block. A canonical form for such alternative representations is specified in *Chapter 3, Conformance.*

Vietnamese Vowel Plus Tone Mark Combinations. A portion of this block (U+1EA0 → U+1EF9) comprises vowel letters of the modern Vietnamese alphabet (quốc ngữ) combined with a diacritic mark which denotes the phonemic tone that applies to the syllable. In the modern Vietnamese alphabet, there are 12 vowel letters and five tone marks

Figure 6-19. Vietnamese Letters and Tone Marks

a ă â e ê i o ô ơ u ư y

◌́ ◌̀ ◌̉ ◌̃ ◌̣

Some implementations of Vietnamese systems prefer storing the combination of vowel letter and tone mark as a singly encoded element; other implementations prefer storing the vowel letter and tone mark separately. The former implementations will use characters defined in this block along with combination forms defined in the Latin-1 Supplement and Latin Extended-A character blocks; the latter implementations will use the basic vowel letters in the Basic Latin, Latin-1 Supplement, and Latin Extended-A blocks along with characters from the Combining Diacritical Marks block. For these latter implementations, the characters U+0300 COMBINING GRAVE, U+0309 COMBINING HOOK ABOVE, U+0303 COMBINING TILDE, U+0301 COMBINING ACUTE, and U+0323 COMBINING DOT BELOW should be given preference in representing the Vietnamese tone marks.

Encoding Structure. The Latin Extended Additional character block is divided into the following ranges:

U+1E00	→	U+1E9B	Additional Latin letter with diacritic combinations
U+1EA0	→	U+1EF9	Vietnamese vowel plus tone mark combinations

Greek Extended: U+1F00—U+1FFF

The characters in this block constitute a number of precomposed combinations of Greek letters with one or more general diacritical marks; in addition, a number of spacing forms of Greek diacritical marks are provided here. These characters were added to the Unicode Standard as a result of the process of merging the Unicode Standard with the developing ISO 10646 standard. In particular, these characters facilitate the representation of Polytonic Greek texts in a compatible manner with existing implementations of Polytonic Greek.

Each of the characters contained in this block may be alternatively representated with a base letter from the Greek block followed by one or more general diacritical mark characters found in the Combining Diacritical Marks block. A canonical form for such alternative representations is specified in *Chapter 3, Conformance.*

Spacing Diacritics. Sixteen additional spacing diacritic marks are provided in this character block for use in the representation of Polytonic Greek texts. Each of these has an alternative representation for use with systems that support non-spacing marks. The Unicode Standard considers the non-spacing alternative forms to be the canonical Unicode representation of the information represented by the spacing forms. The non-spacing alternatives appear in Table 6-17.

Table 6-17. Greek Spacing and Non-Spacing Pairs

Spacing Form	Non-Spacing Form
1FBD GREEK KORONIS	0313 COMBINING COMMA ABOVE
037A GREEK YPOGEGRAMMENI	0345 COMBINING YPOGEGRAMMENI
1FBF GREEK PSILI	0313 COMBINING COMMA ABOVE
1FC0 GREEK PERISPOMENI	0342 COMBINING GREEK PERISPOMENI
1FC1 GREEK DIALYTIKA AND PERISPOMENI	0308 COMBINING DIAERESIS + 0342 COMBINING GREEK PERISPOMENI
1FCD GREEK PSILI AND VARIA	0313 COMBINING COMMA ABOVE + 0300 COMBINING GRAVE ACCENT
1FCE GREEK PSILI AND OXIA	0313 COMBINING COMMA ABOVE + 0301 COMBINING ACUTE ACCENT
1FCF GREEK PSILI AND PERISPOMENI	0313 COMBINING COMMA ABOVE + 0342 COMBINING GREEK PERISPOMENI
1FDD GREEK DASIA AND VARIA	0314 COMBINING REVERSED COMMA ABOVE + 0300 COMBINING GRAVE ACCENT
1FDE GREEK DASIA AND OXIA	0314 COMBINING REVERSED COMMA ABOVE + 0301 COMBINING ACUTE ACCENT
1FDF GREEK DASIA AND PERISPOMENI	0314 COMBINING REVERSED COMMA ABOVE + 0342 COMBINING GREEK PERISPOMENI
1FED GREEK DIALYTIKA AND VARIA	0308 COMBINING DIAERESIS + 0300 COMBINING GRAVE ACCENT
1FEE GREEK DIALYTIKA AND OXIA	0308 COMBINING DIAERESIS + 0301 COMBINING ACUTE ACCENT
1FEF GREEK VARIA	0300 COMBINING GRAVE ACCENT
1FFD GREEK OXIA	0301 COMBINING ACUTE ACCENT
1FFE GREEK DASIA	0314 COMBINING REVERSED COMMA ABOVE

Canonicalization of Spacing Forms. When canonicalizing the spacing forms, the spacing status of the implied usage must be taken into account. Unless information is present to the contrary, these spacing forms would be translated to U+0020 SPACE followed by the non-spacing form equivalents shown in Table 6-17.

In archaic forms of Greek, U+0345 COMBINING GREEK YPOGEGRAMMENI and the precomposed forms that contain it have special case mappings. (See the Greek character block description for more information.)

Encoding Structure. The Greek Extended character block is divided into the following ranges:

U+1F00	→	U+1FBC,	Additional Greek letter with diacritic combinations
U+1FC2	→	U+1FCC,	
U+1FD0	→	U+1FDB,	
U+1FE0	→	U+1FEC,	
U+1FF2	→	U+1FFC	
U+1FBD	→	U+1FC1,	Additional Greek spacing diacritics
U+1FCD	→	U+1FCF,	
U+1FDD	→	U+1FDF,	
U+1FED	→	U+1FEF,	
U+1FFD	→	U+1FFE	