South Asia Language Pedagogy and Technology, Vol. 1 (2008)
http://salpat.uchicago.edu

Digital Encoding of South Asian Languages: A Contemporary Guide to Unicode and Fonts

Sean Pue
South Asia Language Resource Center
University of Chicago

The World Wide Web and your computer can already display text in the scripts of most South Asian languages. What makes that possible is a system of digital encoding called Unicode. This article will explain how best to understand Unicode and its benefits by addressing a few primary questions. Following a general introduction, the article concludes with a discussion of specific issues involving Unicode and South Asian languages.

What is a Digital Text?

Any text that you see on a computer—on a webpage, in a word processor, and so on—is treated by the computer program as a series of numbers. For example, the word "Text" is a sequence consisting of the following numbers: 84, 101, 120, 116. The number a character refers to is called the character's *codepoint*. The codepoint of a character is usually represented using *hexadecimal* notation, which uses the numbers 0-9 plus the letters A-F.

What is Character Encoding?

Character Encoding refers to the underlying standard or code that governs which character will be represented as what number. An example is the Indian Script Code for Information Interchange (ISCII), which provides a character encoding for many South Asia languages. The most frequently used and versatile contemporary standard for character encoding is Unicode.

What is Unicode?

Unicode is an evolving industry standard for character encoding that maps the characters of nearly all languages and scripts as a specific number. This unique number provides a universal reference for storing, sorting, and manipulating texts.

Before Unicode, most character encodings could only deal with a character map of 256 entries. With Unicode, there is no limit to how many characters can be added to the standard. There are currently over 100,000 characters mapped in Unicode from over thirty writing systems. Unicode codepoints are usually referred to in their hexadecimal form, often preceded by "U+"

as in U+200D. Each codepoint also has a character name.

Unicode is the current cross-platform standard for character encoding. All contemporary operating systems (i.e. OS X, Windows, Linux) and most contemporary computer programs process text following this standard.

Unicode is only a standard for what number refers to what character. The actual display of characters involves a font.

What is a Font?

A computer font is a data file containing rules for how to display certain characters in a particular typeface. These characters are referred to by numbers.

There are currently several different font formats. At the present moment, the most versatile font format for rendering complex scripts is *OpenType*, developed originally by Microsoft and later by Adobe. It is supported by the three major operating systems (Windows, Linux, and Apple's OS X). Support for complex scripts using OpenType is not as complete in OS X as it is in Windows and Linux. Apple has developed its own font format, *Apple Advanced Typography (AAT)*, which is used exclusively in OS X. Apple earlier developed a popular font format called *TrueType*, which is unable to properly render complex scripts following the Unicode Standard.

> Technical Note: Many pre-Unicode fonts would simply encode all the possible character combinations of complex scripts as separate numbers in TrueType fonts. As a result, it is impossible to read text encoded in those formats in other fonts. Many fonts used for transliteration with diacritics also used their own system of encoding. Certain standards for these encodings have developed, such as CSX+ (Classical Sanskrit Extended). However, Unicode also provides encodings for diacritics (see below).

Because a font is essentially a data file, it is necessary for a computer program to be able to read that data and display it appropriately as pixels on the screen. Programs use a *text rendering engine* to do so. These engines vary based on operating systems and computer programs.

What is a Unicode Font?

A 'Unicode font' is a font that conforms to the Unicode standard. Like any other font, it provides rules for how to display characters, which are mapped as specific numbers. In a Unicode font, these numbers follow the Unicode standard. A Unicode font may contain the

rules for displaying numerous scripts, or maybe just one. For example, the Tahoma font contains rules for how to display the Latin alphabet, as well as the scripts used by Urdu and Thai. All *OpenType* and *AAT* fonts are Unicode fonts. TrueType fonts may follow the standard to some extent but are unable to properly render South Asian scripts using Unicode. Therefore, TrueType fonts should be avoided for South Asian languages.

Why Use Unicode?

If you write documents using Unicode, your text can be displayed in numerous Unicode fonts. If you send a document or put it up on the web and your readers do not have the same font you used when writing the text, they can still read it, provided they have some other Unicode font that can display those characters.

How Does One Type in Unicode?

In order to type in Unicode, one needs to install or enable an appropriate keyboard. The latest version of most current operating systems (Windows, OS X, Linux) already contain keyboard layouts for most South Asian scripts. Moreover, there are often different layouts available for the different scripts, such as a phonetic option or one closer to the layout used in the type writer of the native script. The South Asia Language Resource Center lists a number of keyboard options on its fonts website. Most operating systems also have an onscreen keyboard feature, which is helpful while learning to type.

How Does Unicode Handle South Asian Languages?

As a standard, Unicode is more concerned with characters and scripts than with languages. The characters used by different South Asian languages are therefore organized by script. Most often, characters in the same script have numbers that are next to each other in a *character block*. Currently (in Unicode 5.1) there are twelve "Indic script" character blocks in Unicode. They are: Bengali, Devanagari, Gujarati, Gurmukhi, Kannada, Limbu, Malayalam, Oriya, Sinhala, Syloti Nagri, Tamil, and Telugu. The characters used by right-to-left languages like Urdu, Pashto, and Sindhi are found on the Arabic character code chart. Tibetan has its own code chart.

How Does Unicode Handle the Special Issues Common to Left-to-Right South Asian Scripts?

The twelve "Indic scripts" and Tibetan all function in a similar way in Unicode, as the writing

systems themselves are similar. All of these scripts are segmented in a such a way that one or more consonants are written as a cluster followed by a specific vowel sign, and consonants in a cluster will change their shape—either assuming a half-form or a combined form—before a following vowel. Unicode is concerned with the underlying letter and not with the shapes they assume in a word. There are additional control characters, discussed below, that are inserted between consonants to indicate whether or not they are combined together. As a result, each of these scripts are assigned only up to 128 separate codepoints even though there are often many more separate visual forms for these letters. In this way, Unicode differs from many proprietary fonts and other standards, such as the Indian Script Code for Information Interchange (ISCII).

For left-to-right South Asian scripts, Unicode contains codepoints for independent vowels (used when a vowel is not preceded by a consonant), consonants, and dependent vowel signs (used when vowels follow consonants), as well as digits and other "various signs." The common punctuation marks are found in the Devanagari range.

There are a few ways to indicate that consonants should be combined, and they all involve inserting certain characters in between the consonants. The first of these characters is actually a sign in the left-to-right scripts referred to as a *virām*, *halant*, or *hasant*. All of the left-to-right South Asian scripts in Unicode contain this character. For your reference, here are the Unicode codepoints for this character in the different South Asian scripts:

Codepoint	Character Name
U+094D	DEVANAGARI SIGN VIRAMA
U+09CD	BENGALI SIGN VIRAMA
U+0A4D	GURMUKHI SIGN VIRAMA
U+0ACD	GUJARATI SIGN VIRAMA
U+0B4D	ORIYA SIGN VIRAMA
U+0BCD	TAMIL SIGN VIRAMA
U+0C4D	TELUGU SIGN VIRAMA
U+0CCD	KANNADA SIGN VIRAMA

U+0D4D	MALAYALAM SIGN VIRAMA
U+0F84	TIBETAN MARK HALANTA
U+A806	SYLOTI NAGRI SIGN HASANTA

When a VIRAMA Unicode character is added in between consonants, the consonants will be displayed in their conjoined form. For example, the typical way to write the Hindi Word *kyā* is:

U+0915	क	DEVANGARI LETTER KA
U+094D	◌्	DEVANAGARI SIGN VIRAMA
U+092F	य	DEVANAGARI LETTER YA
U+093E	◌ा	DEVANAGARI VOWEL SIGN AA
Typical Result:	क्या	

This typical result shows the DEVANAGARI LETTER KA in its combined form with DEVANAGARI LETTER YA. What is actually displayed when the VIRAMA is added depends on the text rendering engine and the font. In Hindi, the word *kyā* can also be written using the regular form of DEVANAGARI LETTER KA with the VIRAMA sign below it.

There are two special control characters that can be added after the VIRAMA that can request that the virama be written separately or that the combined form be displayed. These characters are called the ZERO WIDTH NON-JOINER (U+200C) [ZWNJ] and the ZERO WIDTH JOINER (U+200D) [ZWJ]. Neither of these characters has any shape of its own—it has "zero width"—but rather governs whether the VIRAMA will be displayed (using the ZWNJ) or the consonants will be joined (using the ZWJ) and the virama not displayed:

क + ◌् + ZWNJ + य + ◌ा = क्‌या

क + ◌् + ZWJ + य + ◌ा = क्या

In summary, the important aspect of the way Unicode deals with left-to-right South Asian scripts is that the conjunct forms are created by adding a VIRAMA character between the consonants. For additional information, see Unicode's Indic Scripts and Languages FAQ and Chapter 9 of the Unicode Standard, South and Southeast Asian Scripts.

How Does Unicode Handle the Special Issues Common to Right-to-Left South Asian Scripts?

Right-to-left South Asian languages use scripts derived from the script used for Arabic. The special characters used by South Asian languages have unique codepoints in Unicode. Unicode does not distinguish between variants of right-to-left scripts, such as *nastaliq* or *naskh.* It is only concerned with the underlying letters and signs, not with the shape in which they combine together. Therefore, there is no need to worry about initial, medial, or final forms of letters when entering text.

> Technical Note: There are a few "Arabic Presentation" code charts in Unicode that do contain separate initial, medial, and final forms of Arabic letters. These are left over for compatibility with an earlier Arabic-language character encoding and are not used for South Asian languages.

There appears to be a certain amount of ambiguity in the Arabic code chart because a number of different characters look the same. However, the mapping of South Asian right-to-left languages is now more or less standardized in practice. One reason is the exclusion from popular fonts and keyboards of certain codepoints deemed inappropriate for South Asian languages as they are more properly part of Arabic. So while the letter ARABIC LETTER FARSI YEH (U+06CC) looks identical to ARABIC LETTER ALEF MAKSURA (U+0649), ALEF MAKSURA is often not included in current fonts. While ARABIC LETTER YEH (U+064A) looks the same as ARABIC LETTER FARSI YEH in initial and medial positions, it is best avoided, because many fonts for South Asian languages do not include it. Similarly, ARABIC LETTER HEH DOACHASHMEE (U+006BE) should be used in place of ARABIC LETTER HEH (U+0647), even though they look identical. For the regular *heh*, the preferred codepoint is ARABIC LETTER HEH GOAL (U+06C1).

Another issue is the letter *hamza*. The Unicode codepoint used for the "seated" hamza is ARABIC LETTER YEH WITH HAMZA ABOVE (U+0626). ARABIC LETTER HAMZA (U+0621) is used in the standalone, unconnected position. There is a third codepoint, ARABIC HAMZA ABOVE (U+0654), which renders a hamza above a letter, such as BAREE YEH, VAO, and so on. Though Unicode contains codepoints such as YEH BARREE WITH HAMZA ABOVE, HEH GOAL WITH HAMZA ABOVE, and so on, it is preferable to add the HAMZA ABOVE, as many current fonts do not render these combinations properly:

Hamza:	"Seated"	"Standalone"	"Above"
Character:	ئ	ء	ٔ
Codepoint:	U+0626	U+0621	U+0654
Example:	بَہِے	سماء	بَہاؤ

There is a separate set of numbers (U+06F0 to U+06F9) for the encoding of the form of numerical digits used by South Asian languages (as distinct from the form used by Arabic).

The ZERO WIDTH NON-JOINER (U+200C) can be used to break the combination of letters without adding an intermediate space.

> Historical Note: The Center for Research in Urdu Language Processing developed the first *nastaleeq* OpenType font named Nafees Nastaleeq. This font, as well as the CRULP Urdu Phonetic Keyboard (modeled after the commercial Inpage Urdu software), has helped to standardize the encoding of Urdu on the internet by not including certain letters, such as the ARABIC LETTER HEH or ARABIC LETTER YEH (as opposed to FARSI YEH), and certain combined forms. This font cannot properly render *aerabs* (diacritics) between combining letters. A new *nastaleeq* font named Pak Nastaleeq is currently being developed by the Center of Excellence for Urdu Informatics at the National Language Authority (Pakistan) and is available in beta format.

How Do Diacritics/Transliteration Work in Unicode?

Transliteration is the transformation of characters from one script into another. South Asian languages are frequently transliterated into Roman characters with *diacritics* (small signs added above or below letters) and other special characters. Transliteration can be used to make text in another language readable to people who do not know its script, and it can also provide more information than is available in the native script. For example, the proper pronunciation of vowels are not marked in certain scripts. Transliteration can provide that information. Additionally, transliteration provides a means of showing the component parts of compound words or phrases.

Unicode provides codepoints for all the diacritics needed for transliteration of South Asian languages. Not all fonts are able to display every character, however, so particular fonts are necessary to render diacritical marks.

Unicode has two ways of rendering diacritical marks. Some diacritical characters are precomposed, meaning certain codepoints are assigned to letters with diacritical marks. For example, there is a number assigned to LATIN SMALL LETTER T WITH DOT BELOW. Unicode also contains combining diacritical marks. For example, there is a character named COMBINING DOT BELOW. Entering this character after LATIN SMALL LETTER T will render a small letter t with a dot below. Every standard diacritical mark is available in a combining form in Unicode. Certain combinations are only possible using combining forms. For example, there is no unique code point for a small letter t with a combining diaeresis (two dots) below. Combining diacritical marks are also used to add multiple diacritics to a letter. Combining diacritical marks also can be added to precomposed letters with diacritics marks. Therefore, to compose an l with a macron (small line) above and a ring (small circle) below, one can enter either LATIN SMALL LETTER L WITH LINE BELOW (1E3A) followed by COMBINING RING BELOW (0325), or LATIN SMALL LETTER L (006C) followed by COMBINING MACRON BELOW (0331) followed by COMBINING RING BELOW (0325).

What are the Best Practices, and What Does the Future Hold?

The best practice for composing digital texts in South Asian languages is to follow the Unicode standard and to use a Unicode font, either of the OpenType or AAT variety. This practice will ensure that your documents will be easily accessible for years to come. Because Unicode is an industry standard, infinitely expandable, and used in all contemporary operating systems (Windows, OS X, Linux), as well as the World Wide Web, Unicode is the best character encoding standard. It is here to stay. Font technology, on the other hand, will continue to evolve as new display engines and fonts are continually developed. By composing texts according the Unicode standard, your texts will be able to easily take advantage of evolutions in font technology that will only improve the display of your documents.