L2/02-011R

Comments on L2/02-006: Towards Unicode Standard for Urdu

Jonathan Kew, SIL International
12 January 2002
updated: 15 January 2002

The document L2/02-006 discusses the "standard character set for Urdu" (UZT), recently approved in Pakistan, and its relation to Unicode. In particular, it gives Unicode mappings for the majority of the character codes in UZT, and lists 25 that "do not have a representation in Unicode". I would like to suggest several amendments to these lists.

First, a couple of the mappings listed are incorrect:

UZT code 0x78 (#121 in Table 1). The Urdu character concerned is "chota yeh"; this should be mapped to U+06CC ARABIC LETTER FARSI YEH, not to U+0649 ARABIC LETTER ALEF MAKSURA. The difference is important when linked to the following letter; linked ALEF MAKSURA does not acquire dots below, whereas YEH does. (Incidentally, L2/02-006 seems to consistently use Unicode 1.x names, which differ from the current Unicode names for several of the characters concerned here.)

UZT code 0xA4 (#165). It is incorrect to map this to U+FDF9. The UZT code represents a combining mark that is written above a word, while U+FDF9 is a base character. UZT code 0xA4 corresponds to the ARABIC SIGN SALLALLAHOU ALAYHE WASALLAM proposed in L2/01-425, along with other combining marks for Arabic-script honorifics.

Second, I believe that a number of the characters proposed for inclusion in Unicode do not merit encoding, or can be unified with characters already present. (Others correspond to characters already proposed in L2/01-425; obviously, in these cases I support their addition to the Standard.)

Following the item numbers in Table 2:

#1: UZT 0x2E ARABIC-URDU DECIMAL SIGN. It might be appropriate to treat this as a language-specific glyph variant of U+066B ARABIC DECIMAL SEPARATOR. The UTC has, I believe, already stated that the Urdu digit forms are considered glyph variants of the extended Arabic-Indic (or Farsi) digits; compare the representative glyph for U+06F4 in TUS 3.0 with that shown in L2/02-006, Table 1, item #53. Perhaps the form of the decimal separator should be handled in the same way.

#2: UZT 0x3A ARABIC-URDU COLON SIGN. This can be encoded as COLON followed by HYPHEN.

#3: UZT 0x41 ARABIC-URDU HARD SPACE. From the discussion of UZT in L2/02-004, it appears that the function of this code is to break the cursive connection between adjacent letters, without introducing whitespace. This sounds like U+200C ZERO WIDTH NON-JOINER.

#4: UZT 0x42 ARABIC-URDU HAMZA E IZAFAT. It is unclear to me why the existing hamzas in Unicode (U+0621 if it is to stand alone on the line, or U+0654 if it is to appear with another character as a base) would not serve here.

#5: UZT 0x43 ARABIC-URDU KASRA E IZAFAT. Again, I would be interested to know the justification for separating this from normal KASRA (U+0650).

#6: UZT 0x45 ARABIC-URDU ALEF BELOW. This has been proposed (as ARABIC SUBSCRIPT ALEF) in L2/01-425; thus, I support the addition of this character to Unicode.

#7: UZT 0x46 ARABIC-URDU PESH ABOVE. This is simply the Urdu name for U+064F ARABIC DAMMA. However, I believe this is an error in the document, as L2/02-004 shows an inverted

PESH at this code position, which would correspond to the ARABIC TURNED DAMMA proposed in L2/01-425.

#8: UZT 0x47 ARABIC-URDU SPECIAL INVERTED PESH. Compare the ARABIC TURNED DAMMA (see above). I am not aware whether adequate justification exists for two variants of inverted/turned damma/pesh, as suggested by UZT; unfortunately, the documents provided do not offer this background information.

#9: UZT 0x48 ARABIC-URDU ZARE BELOW. I don't know anything about this character.

#10: UZT 0x4C ARABIC-URDU SMALL TAH. I believe this occurs only as an integral part of existing letters (the Urdu retroflex series); it functions just like the nuktas (dots) of other Arabic letters, to distinguish letters with the same basic stroke shape.

Unicode (in common with virtually all other encodings for Arabic) works in terms of the commonly-understood letters of the alphabet, rather than on the basis of letter construction (with a code for each basic letter form, and separate codes for the various dot patterns that may be added). Therefore, there are no codes for Arabic dot patterns as combining marks; it would be inconsistent, then, to create such a code for the SMALL TAH.

Moreover, this would break normalization; DDAL, for example, would need to be canonically equivalent to DAL + SMALL TAH.

It is possible that languages other than Urdu may wish to use other letters including the SMALL TAH element; for example, one can imagine HAH WITH SMALL TAH, SEEN WITH SMALL TAH, etc. If such forms can be documented (I believe some of them probably can, and hope to eventually collect such materials), the correct response will be to encode these letters in their own right, not to construct them with a SMALL TAH code added to dotless letters.

#11: UZT 0x4D ARABIC-URDU SAKOON. This is simply the usual form of U+0652 ARABIC SUKUN used with Urdu, particularly when written in the Nastaliq style of calligraphy.

#12: UZT 0x4E ARABIC-URDU REVERSE SAKOON. Not really a sukun at all, but a mark used to indicate that NOON represents nasalization rather than a consonantal /n/. Proposed in L2/01-425 with the name ARABIC NASALIZATION MARK.

#13: UZT 0x7B ARABIC-URDU NO-DICRITIC*[sic]* SIGN. I cannot see any justification for positively encoding the absence of a diacritic. The fact that no vowel diacritic is present is perfectly adequate. (Consider the idea of encoding unaccented French vowels by adding an invisible “no accent” code after the normal vowel characters.)

The documents describing the development and design of the UZT encoding make it apparent that this code was created to aid the process of collation. The intent is that this code acts as a “placeholder” for the potential vowel that is not in fact present, purely in order to provide something for a collation routine to compare to other cases where a vowel mark is present. Such a code is unnecessary; all that is needed is a suitable collation process (already needed for many other reasons in Unicode).

#14: UZT 0xA2 ARABIC-URDU LIGATURE BISMILLAH. This “character” as shown in the UZT documents is clearly not a character in any normal sense; the charts show that it is a rendering of a complete sentence of Arabic. (Not even just the word “bismillah”.) It is true that this sentence is a common, well-known “formula” used as an opening in many contexts, and often written in a somewhat ornate style. Nevertheless, it remains clearly a fully spelled-out sentence, not a single orthographic unit.

I would suggest that this sentence should be encoded just like any other Arabic text—by spelling it out, letter by letter. If implementers wish to provide a convenient shortcut to generate this text, that is of course perfectly reasonable. A “smart font” implementation might even provide a special ornate rendering of it. But if this is encoded as a Unicode “character”, the door should be open to any word or phrase that is commonly used in some stylized form. (*Coca-Cola*?)

#15: UZT 0xA5 ARABIC-URDU LIGATURE ALAYHE AS SALAM. Proposed in L2/01-425 with the name ARABIC SIGN ALAYHE ASSALAM. Note in the case of this and other similar characters: it is misleading to describe this as a “ligature”. It is true that its form originates from a simplified rendering of the Arabic text, but it is now a sign in its own right, functioning as a combining mark that appears above a base character. It is thus quite different from the “ligatures” encoded in the FDFx column, which I believe are regrettable (along with the rest of the Arabic presentation forms).

#16: UZT 0xA6 ARABIC-URDU LIGATURE RADIALLAH. Proposed in L2/01-425 as ARABIC SIGN RADI ALLAHU ANHU.

#17: UZT 0xA7 ARABIC-URDU LIGATURE REHMATULLAH. Proposed in L2/01-425 as ARABIC SIGN RAHMATULLAH ALAYHE.

#18: UZT 0xA8 ARABIC-URDU TAKHALLUS SIGN. Proposed in L2/01-425 as ARABIC SIGN NOM DE PLUME.

#19: UZT 0xA9 ARABIC-URDU MISRA SIGN. Although I have seen this sign, I am not sufficiently familiar with its use to know whether it should be regarded as simply a swash form of U+0639 ARABIC LETTER AIN, or deserves separate encoding. Further information would be helpful.

#20: UZT 0xAA ARABIC-URDU FOOTNOTE SIGN. Proposed in L2/01-425 as ARABIC FOOTNOTE MARKER.

#21: UZT 0xAB ARABIC-URDU SAFAH SIGN. No information on usage is offered. I am guessing that this would combine with numbers, similarly to the YEAR and NUMBER signs, and would mean “page…”. This may merit encoding, although it would be helpful to see examples from published materials to support its case.

#22: UZT 0xAC ARABIC-URDU NUMBER SIGN. Proposed in L2/01-425 as ARABIC NUMBER SIGN.

#23: UZT 0xAD ARABIC-URDU SANAH SIGN. Proposed in L2/01-425 as ARABIC YEAR SIGN.

#24: UZT 0xAE ARABIC-URDU LONG MADD. I don't know anything about this. Is it merely a stylistic variant of MADDAH, or is it a distinct character that requires separate encoding? How is it used? Published examples would help clarify this.

#25: UZT 0xB0 ARABIC-URDU END OF SECTION SIGN. This is a simple symbol; surely there are possible equivalents in Unicode already? For example, U+25CB WHITE CIRCLE.

In summary, of the 25 characters “proposed for inclusion”, I would suggest that:

- 1, 2, 3, 4, 5, 11, and 25 are probably adequately represented in Unicode already;

- 6, 7, 12, 15, 16, 17, 18, 20, 22, and 23 are among those proposed in L2/01-425, and will (I hope) be accepted in due course;

- 10, 13, and 14 are inappropriate for encoding;

- further information is required regarding 8, 9, 19, 21, and 24 to support their case for inclusion.