Ref: Proceedings of INMIC2001, Organised by IEEE & Lahore University of Management Sciences, Lahore, December 28-30, 2001, pp: 223-228

Urdu Computing Standards: Urdu Zabta Takhti (UZT) 1.01

L2/02-004

Sarmad Hussain* and Muhammad Afzal**

Introduction

Software development in Urdu has been going on for more than three decades. However, until recently there were no industry standards for coding in Urdu, similar to ASCII standard for English. Therefore, all the individual or industry efforts were isolated and based on ad hoc mapping of Urdu characters on binary codes. Every developer was using arbitrary code pages (character set and its mapping onto binary code), making it incompatible with other vendors. In addition, due to the competition in the industry these code pages have been well-guarded trade-secrets. Thus, this absence of a standard has been a significant hurdle to the development and propagation of Urdu software.

Keeping in view the necessity and urgency of standardization needs, three years ago a national effort was initiated to get relevant people together and formulate a common standard for everybody to follow. This effort, detailed in another paper within this volume (Afzal and Hussain, 2001), resulted in the creation of Urdu Zabta Takhti (UZT) version 1.01. UZT 1.01 has been accepted by the Government of Pakistan (GoP) as the standard code page for Urdu. This paper discusses the contents and related specifications of Urdu Zabta Takhti 1.01 in detail. The discussion is divided into sub-sections on logical groups of characters that form this *takhti*, Urdu text file format and how sorting may be done in Urdu with UZT.

Contents of UZT

UZT 1.01 is a 256 bit code page. It has been divided into various logical sections, as described later. Figure 1 below shows the specification of UZT 1.01, as approved by the Government of Pakistan.

UZT 1.01 is divided into the following logical sections:

i. control characters (0 – 31, 127)
ii. punctuation and arithmetic symbols (32 – 47, 58 – 65)
iii. digits (48 – 57)
iv. Urdu aerab/diacritics (66 – 79, 123 – 126)
v. Urdu characters (80 – 122)
vi. reserved control space (128 – 159, 255)
vii. special symbols (160 – 176, 192 – 199)
viii. reserved expansion space (177 – 191, 200 – 207, 240 – 253)
ix. vendor area (208 – 239)
x. toggle character (254)

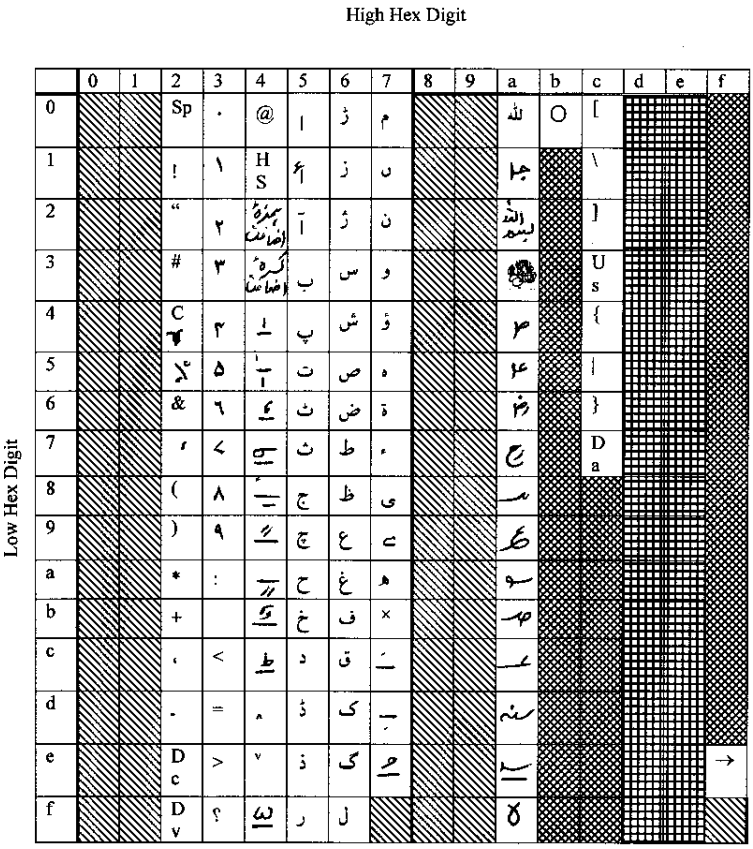

Figure 1: UZT 1.01

Though the driving force behind the contents and their arrangement in UZT were the Terms of References (TOR) (discussed in Afzal and Hussain 2001, this volume) agreed with the National Language Authority, care was taken to keep UZT similar to ASCII code (where possible). This is because people are familiar with the character

* Head, Center for Research in Urdu Language Processing, FAST National University of Computer and Emerging Science, B Block, Faisal Town, Lahore. Email: sarmad.hussain@nu.edu.pk.
** Professor, Fauji Foundation Institute of Management and Computer Science, Rawalpindi.

distributions in ASCII as it is a worldwide standard. In addition, owing to its universal acceptability, many hardware and software systems (especially the earlier ones, some of which are still deployed) conform very closely to ASCII standard. Incompatibility of UZT 1.01 with ASCII would mean incompatibility with these systems as well, which would not be a practical solution.

The team therefore had to propose a solution that had underlying conformance with ASCII along with the explicit conformance to TOR. This point of view is necessary to understand some of the logical groups presented here.

Control Characters

ASCII contains special characters that are not visible but act as control characters for various hardware and software operations. These characters are placed from decimal positions 0 – 31 and 127, and include null (0), new line (10), escape (27), delete (127), etc. As these characters are used universally and are hard coded in software and hardwired in the hardware, UZT has used the same slots for these characters to avoid conflicts. Complete list of these characters and their details are available in all introductory books on computer programming.

Punctuation and Arithmetic Symbols

The symbols used in ASCII are also mapped to their Urdu equivalents (same symbols in many cases) to provide the symbol coverage for Urdu and conform to existing ASCII. Table 1 below gives explanations for these symbols. Though the symbols could have been put in one continuous space, they were broken into two spaces (slots 32 – 47 and 58 – 65) to conform with ASCII. The slots 48 – 57 in ASCII have been used for digits and correspondingly for UZT 1.01 as well.

Table 1. Punctuation and Arithmetic Symbols in UZT

#	Symbol	Remarks
31	space	Break in Urdu connected words, not explicit space between words. Explicit space is achieved by hard space (65) (see Figure 2 for details)
33	!	Exclamation mark
34	“	Double apostrophe
35	#	Hash symbol
36	currency	Currency symbol ($ in ASCII) but may be replaced by Rs. or equivalent Urdu symbol in UZT
37	percent	Percent sign, may be inverted for Urdu
38	&	Ampersand sign
39	‘	Single apostrophe
40	(	Open parenthesis (close parenthesis in Urdu)
41	)	Close parenthesis (open parenthesis in Urdu)
42	*	Asterisk or multiplication symbol
43	+	Plus or addition symbol
44	comma	Comma, inverted for Urdu (different from English)
45	-	Minus sign
46	decimal	Decimal point (different for Urdu than English e.g. small hamza is sometimes used for “ishariya” in Urdu)
47	/	Division sign or slash
58	:	Colon
59	semi-colon	Semi-colon , inverted for Urdu (different from English)
60	<	Less-than symbol
61	=	Equal sign
62	>	Greater-than symbol
63	question mark	Question mark: inverted for Urdu (different from English)
64	@	“at” symbol now frequently used in email addresses
65	hard space	Actual space in Urdu (also see space (32) and Figure 2)

The same symbols have been included in UZT 1.01 as in ASCII, as these symbols are used in Urdu and English languages. Some symbols are logically similar however transformed into a different shape. Their logical names are given and physical realizations are left to the vendors. For example, the dollar symbol ‘$’ is used in ASCII which is replaced by currency in UZT 1.01. Vendor may choose to give relevant symbol for currency (e.g. rupee in Pakistan) as this symbol is not standardized. In addition percent, comma, decimal, semi-colon and question mark symbols get inverted in Urdu. Even though not relevant for Urdu, at slot 64 ‘@’ symbol has been included because of its universal usage in email addresses. These changes have been incorporated.

Finally a hard space (65) has been included and distinguished from normal space (32) because Urdu has two distinct space requirements in word processing (unlike a single space requirement for English). In English, the writing system is unconnected which entails that a single space can mark a word boundary. In Urdu, the script is inherently connected. However breaks between characters may occur within words or across words as illustrated in Figure 2. Within

words, some breaks are natural (e.g. after characters which do not connect at the end, like *alif*) and some may be intentionally inserted, still preserving the word, as shown in Figure 2, to create multiple ligatures (sequence of connected characters forming a word or sub-word). To distinguish word boundary from ligature boundary hard space and space are used respectively. Figure 2 illustrates that a single ligature can be broken into two ligatures with a Space, without breaking the word. Hard space breaks ligatures into different words.

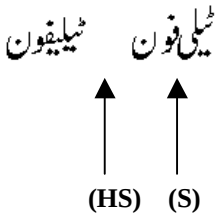

Figure 2. Hard Space (HS) between Characters at Word Boundary and Space (S) between Characters within Ligatures

Digits

Urdu numerals from zero through nine are included in slots 48 – 57 to keep its parallelism to ASCII. The shape conforms to Urdu numerals and is different from English. Urdu zero is written as a dot (slot 48) and should not be confused with English decimal (slot 46).

Urdu Diacritics (Aerab)

Urdu is very rich is diacritics (or aerab, though many are implicit in normal writing). However, word-processing for Urdu still requires these aerab to be represented in UZT 1.01. The aerab are included from slots 66 – 79 and slots 123 – 126. Again the aerab could have been included as a single continuous set of slots. However, they were broken into two logical groups to facilitate sorting. According to the Terms of Reference the aerab in slots 66 – 79 do not effect sorting order of words and aerab in slots 123 – 126 effect sorting order when used. The latter group is placed after the Urdu characters to get the sorting order required by Urdu. Sorting is discussed in detail in a subsequent section. Explanations of these diacritics are given in Table 2. Some of these symbols are not in common usage of Urdu.

Table 2. Diacritics (Aerab) in UZT

Code	Name	Remarks
66	Hamza-e-Izafat	Hamza used to connect words, e.g. in *Idara-e-Tehqiq*
67	Kasr-e-Izafat	Zer used to connect words, e.g. in "Bang-e-Dera"
68	Khari Zabar	
69	Khari Zer	
70	Ulta Pesh	
71	Leta Pesh	Not in normal usage
72	Leti Zer	Not in normal usage
73	Do Zabar	
74	Do Zer	
75	Do Pesh	
76	Chota Toay	Not in normal usage
77	Jazm	
78	Noon Ghunna	Not in normal usage; used as a diacritic to indicate nasalization
79	Shad	
123	Null diacritic	To be used where aerab are necessary to be typed but there are no aerab present; see section on sorting.
124	Zabar	
125	Zer	
126	Pesh	

Aerab will take a byte to represent; therefore when characters are written with aerab, two bytes are taken per character (and three bytes if the character includes do-chashmey hay, as explained below).

Urdu Characters

There is not an agreement on total number of characters in Urdu. Various researchers have indicated different number of Urdu characters (e.g. see Bokhari 1986, Hussain 1997, Kachru 1987, Khan 1997, and Masica 1993). Differences in various proposals are not in the scope of current paper. However, it may be pointed out here that the main disagreement in the total number of characters arises on the number of Urdu voiced and voiceless aspirates (e.g. /l^h/, /m^h/, /n^h/, /w^h/, etc.), i.e. the letters formed by characters and 'do-chashmey hay'. To solve the problem, National Language Authority was requested to provide the current list of characters in 'standard' Urdu and they were included in the Terms of References. The

characters included in UZT 1.01 are all those characters listed in TOR. However, UZT 1.01 has provision to expand these characters (with time) and also allows for (possibly) all characters to form aspirated versions with 'do-chashmey hay'. Urdu characters fill slots 80 – 121, therefore there are 42 characters included. 'Do-chashmey hay' is included at the end of the list in slot 122 (instead of its traditional position adjacent to 'gol hay' in slot 117) for two reasons. First, though written separately, 'do-chashmey hay' is not a character of Urdu but forms characters when combined with other characters (e.g. /b^h/, /p^h/, etc.) and therefore needs to be distinguished from characters. Second, the sorting sequence, which requires all words starting with /b/ to be before all words starting with /b^h/ (and similarly with other unaspirated-aspirated consonantal pairs), comes out naturally if 'do-chashmey hay' is placed at the end of the character list.

A few other notable additions are 'alif-hamza' and 'wao-hamza' in positions 81 and 116 respectively. These were also included as characters as prescribed by National Language Authority (NLA) because they take aerab (zer, zabar, pesh etc.) like other characters. In addition, according to NLA specifications, 'noon-ghunna' was placed before 'noon'.

Reserved Control Space

ASCII is a seven-bit standard, going from 0-127. Earlier systems, some of which are possibly still deployed, still conform to this standard. Therefore, if one byte code is sent to these systems, they truncate the most significant bit and convert the byte code to seven-bit code. This would be especially dangerous if slots 128 – 159 are used, because truncating the most significant bit would map these slots onto the control characters resulting in unpredictable behavior. Therefore, these slots have not been used. In addition, slot not 255 is also not used, as in such a case it would map onto its seven-bit equivalent 'delete' in slot 127. These slots may be used in the future, when eight-bit ASCII is universally acceptable.

For the same reasons of truncation, all the Urdu characters, aerab and digits have been included in lower 128 slots. This ensures that even if a system conforms to a seven-bit code, it may still effectively use UZT to store information in Urdu.

Special Symbols

Urdu writing system is very rich in symbols. These symbols come from a variety of sources including religion, poetry and calligraphy and are used in Urdu word processing. Slots 160 – 176 include these symbols, with room for further expansion after it. Slots 192 – 199 include more general symbols also in ASCII, including square brackets, curly brackets, under-score and dash. More room is also left after these symbols for future expansion.

Reserved Expansion Space

Slots 177 – 191, 200 – 207 and 240 – 253 have been reserved for future expansion. The committee devising Urdu standards will fill these slots in the future. The slots have been left vacant in anticipation of the future needs of Urdu.

Vendor Area

The code page covers the general and widely used Urdu language characters, aerab and symbols. However, there may be special needs that may arise for formatting or for including other more specialized characters not commonly used in Urdu. Slots 208 – 239 have been reserved for this purpose. Vendors may use these slots to define their own symbols for specific applications or formatting (though see Urdu Text File Format section for some limitations).

Toggle Character

Finally, a character in slot 254 has been reserved to toggle between various code pages. Currently toggle character followed by character zero (slot 48) would mean start of UZT 1.01 and toggle character followed by one (slot 49) would mean ASCII. This will be relevant for the *filename*.**utx** (Urdu Text File) format discussed below. Other standard code pages and their codes will be defined by the Urdu Standards Committee in due course of time. No toggle would default to UZT 1.01.

Urdu Text File Format (*filename*.utx)

In addition to defining the contents of UZT 1.01, a file format for standard Urdu text files has also been defined. This simple standard has been devised to enable exchange of files across different Urdu applications. This is a text-based format that stores the UZT 1.01 eight-bit codes in key-press order. It may store all the characters in UZT 1.01 *except those in the vendor area*, as the vendor-defined characters are not portable across vendors. Vendors are encouraged to define their own file formats to store data which contains vendor-specific codes. For example, using Microsoft Word one may use vendor specific *filename*.doc extension to store vendor specific formatting etc. but may also ignore vendor specific information and store information as plain *filename*.txt, which is readable by other software as well.

A special point to note is that Urdu is written from right to left but its number system is written from left to right. The format assumes that all characters are written from left to right. Therefore the front-end formatting will have to take care of reversing the digit directions (this file format will store the numbers in the sequence they are typed).

Sorting

Sorting is a complex issue in Urdu because it is achieved through the characters *and* aerab, but the aerab are normally not written in Urdu. A native speaker of Urdu knows which aerab are implicitly present and therefore can sort these words. However a computer would need the aerab explicitly defined to do aerab-specific sorting. According to the TOR, the sequence in Figure 3 is used for sorting.

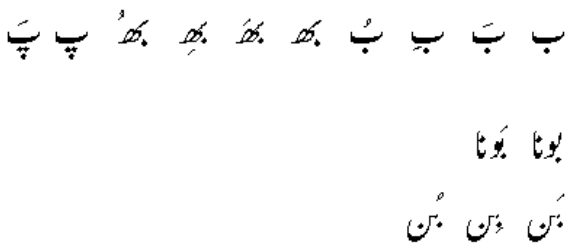

Figure 3. Sorting Sequence of Urdu Letters with Aerab and Do-Chashmey Hay

For each character, the words with that character without aerab (null aerab) come first, then the words with that character with 'zabar' then 'zaer' then 'pesh'. The first row of Figure 3 shows this sequence. In addition, this row also indicates that the letters without do-chashmey hay precede the same letter with do-chashmey hay. The second row shows two words [bona] ("to sow") and [b na] ("dwarf") with exactly the same character sequence but the first word with no aerab and the second word with a 'zabar'. In sorting, the first one would precede the second. Similarly, third row shows three words with exactly the same character sequence and also shows their sorting sequence (first the word with 'zabar' then with 'zaer' then with 'pesh').

These examples indicate that there are actually two levels of sorting in Urdu. First sorting is at character level. Second sorting is done within a character (or same sequence of characters) depending on aerab after the first level of sort has been done (whether aerab are written or implicit, the sorting order does not change). To enable correct sorting of words using computers, both levels of sort must be effectively implemented. It was not possible to achieve both levels of sorting directly through code page (even if separate codes are assigned for letters with different aerab, as proposed by one of the intermediate code pages, see Afzal and Hussain, 2001, this volume). However the characters in the code page have been organized such that first level sorting without aerab is achieved just based on the UZT 1.01 code of each character (including character with do-chashmey hay, even though they will take two bytes instead of one to store).

Level two sort must be achieved through the software. Therefore, to perform correct sorting, the sorting algorithm will require the aerab to be ignored in the first pass and sorting will be done on character codes. Then second pass will insert aerab to resolve conflicts of words with same initial character sequences.

Future Directions

Standardization effort for Urdu computing has just started. Standardizing the code page (UZT 1.01) was the first and basic step identified at the *First Seminar on the Standardization of Urdu Keyboard Layout and Internal Character Representation*, however there is much more work which needs to be accomplished. This section briefly highlights some of the areas that need to be addressed.

UZT 1.01

One of the first measures to be taken is to test the validity and usefulness of UZT 1.01 for word processing, internet and other applications. This would require various vendors to adapt their existing systems to UZT 1.01 and to test its robustness. Feedback will be required from the vendors to mature UZT 1.01 into a standard which is optimal for system development.

Inclusion of UZT within Unicode

To provide multilingual computing environment, software systems are slowly adopting the two-bit Unicode standards for character representation. For future usage, Urdu characters and symbols must also be included in the Unicode. As Urdu writing system is derived from Arabic and Persian, many characters are already represented in existing Unicode standard. However there are twenty-five characters and symbols in UZT 1.01, which are still missing from Unicode and need to be included (Zia 1999).

However, even when Urdu character and symbols are incorporated in Unicode, UZT 1.01 will be needed to complement it, because Unicode is an unordered set of characters (at least from the perspective of Urdu). To define the sorting

sequence, UZT 1.01 will still have to be consulted. In addition, for memory intensive monolingual Urdu applications, UZT 1.01 gives more optimal single byte storage (versus double byte storage for Unicode).

Standardization of Urdu Fonts

Urdu has a rich tradition of calligraphy with many distinct schools of writing. Among the more salient are the Naskh and Nastalique writing traditions acquired through Arabic and Persian (Majeed 1989). Within these schools there is a lot of variation. Work needs to be done to identify and define these fonts for Urdu. In addition, some fonts need to be developed and made available as shareware for use across vendors. These fonts need to be developed for character and ligature based systems.

Standardization of Urdu Keyboard

Keyboards for Urdu typewriters have existed since 1911 and there are currently more than thirty variations of these keyboards (Aziz 1987). All these keyboards have been devised for typewriters. Effort also needs to be made to look at the variations in the keyboards suggested to-date and standardize the keyboard layout for computers, incorporating additional symbols and characters, if applicable.

Conclusion

There have been many factors for slow growth of Urdu software. One of the contributing factors has been the lack of standards for Urdu computing. This paper has explained the details of UZT 1.01, one of the recently devised standards for Urdu computing.

Government of Pakistan has formally ratified UZT 1.01 as the standard code page for Urdu. This code page has been based on the Terms of References devised by the collaboration of experts who have experience with Urdu computing across Pakistan and relevant government departments including Muqtadara Qaumi Zaban and Pakistan Computer Bureau. The salient features of UZT 1.01 include its sole and complete coverage of Urdu (in regular usage) including Urdu Characters, Urdu aerab, punctuation marks, digits, special symbols and implicit conformance with ASCII where possible. In addition, UZT 1.01 also includes specific vendor area and further expansion slots. UZT 1.01 standard also gives automatic level-one sorting and defines standard Urdu text file format for data exchange across various software systems developed by different vendors.

However, just standardizing Urdu code page is not sufficient. Standardization efforts need to be expanded to represent Urdu completely in Unicode. In addition more work needs to be done in the areas of fonts and keyboard. Application of these standards to Urdu computing including desktop publishing, internet and other software systems need to be further explored. As more Urdu applications are developed these standards will be tested and must be revised to achieve optimal solutions.

Acknowledgements

The authors acknowledge the voluntary efforts of all the contributors, who have been involved in this standardization process, since 1998. Without their contributions, standardization would not have been possible. A partial list of contributors is given in Appendix B of another paper in this volume (Afzal and Hussain, 2001).

References

Afzal, M. and Hussain, S. 2001. "Urdu Computing Standards: Evolution of Urdu Zabta Takhti 1.01," in Proceedings of INMIC 2001, Lahore.

Aziz, T. 1987. *Urdu Type Machine ke Kaleedi Takhtay.* Muqtadara Qaumi Zaban, Islamabad.

Bokhari, S. 1985. *Phonolgy of Urdu Language.* Royal Book Company, Karachi.

Hussain, S. 1997. *Phonetic Correlates of Lexical Stress in Urdu.* Unpublished Ph.D. dissertation, Northwestern University, IL, USA.

Kachru, Y. 1987. "Hindi-Urdu," in *The Major Languages of South Asia, The Middle East and Africa*. Comrie, B (eds.). Routledge, London, UK.

Khan, M. 1997. *Urdu Ka Sauti Nizam.* Muqtadara Qaumi Zaban, Islamabad.

Majeed, S. (ed.) 1989. *Urdu Rasm-ul-Khat*. Muqtadara Qaumi Zaban, Islamabad.

Masica, C. 1991. *The Indo-Aryan Languages.* Cambridge University Press, Cambridge, UK.

Zia, K. 1999. "Towards Unicode Standard for Urdu," in *Proceedings of 4th Symposium on Multilingual Information Processing (MLIT4)*, Yangon, Myanmar (CICC Japan).