



## Research Report on Pashto Font Development

Rafiq Ullah Kakar, Habib-ur-Rahman Ghaznawe  
*Afghan Computer Science Association (ACSA), Kabul, Afghanistan*  
[rafiqullah.kakar@acsa.org.af](mailto:rafiqullah.kakar@acsa.org.af), [habib.ghaznawe@acsa.org.af](mailto:habib.ghaznawe@acsa.org.af)

### Abstract

*This document is a research report on Pashto Font (PaykanRawan) Development conducted by Afghan Computer Science Association. This document gives a brief picture of the Font Development process using FontLab and Ms. Volt technology.*

### 1. Introduction

Fonts play special role for the development of rendering any computer software. Unfortunately, no fonts were developed from the scratch for Pashto so far. In most scenarios, Farsi or Arabic fonts have been modified to meet the basic needs of the end users. With the development of PaykanRawan special fonts developed from the scratch by ACSA Calligrapher, Mr. Paykan Rawan, meets all the grammar and other relevant needs in Pashto Language.

Background information on True Type and Open Type Fonts and how they are being supported in Linux operating system is provided here.

Font is collection of glyphs (basic element of the font) to show data characters (minimal unit of the written language) images visually. Each font has a structure made of:

- Character Set
- Code Page
- Font Code

### 2. Methods:

Font Character Set:

In computer and machine-based telecommunications terminology, a character is a unit of information that roughly corresponds to a grapheme, or symbol, in the written form of a natural language.

An example of a character is a letter, numeral, or punctuation mark. The concept also includes control characters, which do not correspond to natural language symbols but to other bits of information used to process texts of the language, such as instructions to printers or other devices that display such texts

Character properties are seen in the positions of characters i.e.:

A character baseline demonstrates an alignment on the line for writing.

- The way the character will be printed from its space dimension.
- The character position in its space.
- Each character has its character ID.

A character encoding consists of a code that pairs a set of characters (representations of graphemes or grapheme-like units, as might be found in an alphabet or syllabify for the communication of a natural language) with a set of something else, such as numbers or electrical pulses, in order to facilitate the storage and transmission of text in computers and through telecommunication networks.

Character Set is the string processing and the logic with which you manipulate text and it can also be defined as the way to store and transmit data. In brief encoding is mapping a code number to a character.

There are many types of encoding systems:

- DOS character sets, also known as IBM code pages: CP437, CP737, CP850, CP852, CP855, CP857, CP858, CP860, CP861, CP863, CP865, CP866, CP869
- Windows character sets: Windows-1250-1251-1252-1253-1254-1255-1256-1257 ...
- UTF-16/UCS-2, UTF-32/UCS-4, UTF-7, UTF-8, UTF-9 and UTF-18, UTF-EBCDIC...

	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0	◻	◻	◻	ذ	-	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
1	◻	◻	◻	ر	ف	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
2	◻	◻	◻	ز	ق	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
3	◻	◻	◻	س	ك	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
4	◻	◻	◻	و	ش	ل	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
5	◻	◻	◻	ص	م	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
6	◻	◻	◻	ض	ن	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
7	◻	◻	◻	ط	ه	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
8	◻	◻	◻	ظ	و	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
9	◻	◻	◻	ع	ي	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
A	◻	◻	◻	غ	ي	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
B	◻	◻	◻	ث	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
C	◻	◻	◻	ج	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
D	◻	◻	◻	ح	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
E	◻	◻	◻	خ	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
F	◻	◻	◻	د	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻

**Code page:**

Codepages are tables that map character codes (one byte long) to the Unicode indexes. Depending on the size of the page, these tables may have 256 or 65,536 records, one for each possible character code. Long codepages are called double-byte codepages and are primarily used to represent codes used in Chinese, Japanese, Korean languages.

Codepages are necessary because we need to somehow encode text written in different languages in the one-byte code space. So when we have a text file encoded according to some codepage, we use the codepage table to find which characters were used in this text.

Codepages are used not only to identify characters, but also to simplify text sorting, conversion of lowercase to uppercase characters, spell-checking and in many other applications where it is necessary to know which characters are used in the text.

For example Arabic script has code page 1256  
True Type Fonts

Apple Computer, Inc originally designed the True Type digital font format. It was a means of avoiding per-font royalty payments to the owners of other technologies, and a solution to some of the technical limitations of Adobe's Type 1 format. Originally code named "Bass" (as they were scalable fonts) and later "Royal", the TrueType format was designed to be efficient in storage and processing, and extensible. It was also built to allow the use of hinting approaches already in use in the font industry as well as the development of new hinting techniques, enabling the easy conversion of already existing fonts to the TrueType format. This degree of flexibility in TrueType's implementation of hinting makes it extremely powerful when designing characters for display on the screen. Microsoft had also been looking for an outline format to solve similar problems, and Apple agreed to license True Type to Microsoft.

**Open Type Fonts**

The OpenType font format, jointly developed by Microsoft and Adobe, allows us to combine the best features of the TrueType and Type 1 font formats. For the end user there is little difference between OpenType and TrueType fonts: both are stored in a single font file, both are Unicode-encoded and Windows and Mac OS directly support both.

The real difference is the support of font features that is provided by the OpenType font format. Font features, which can be script and language dependent, can define behaviors of the font, which allow better typographic layout, and precise support of complex scripts.

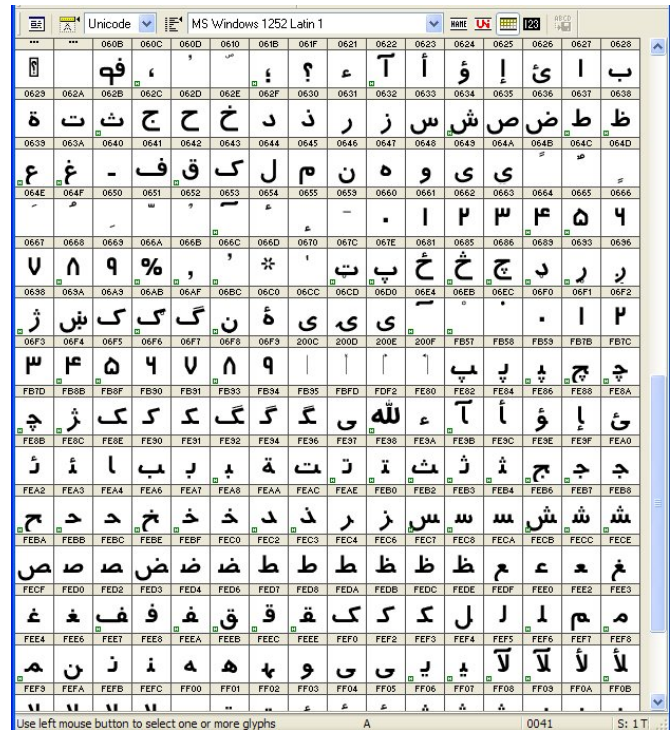
For detailed OpenType specification and other documents please refer to this page:

<http://www.microsoft.com/typography/otspec/default.htm>

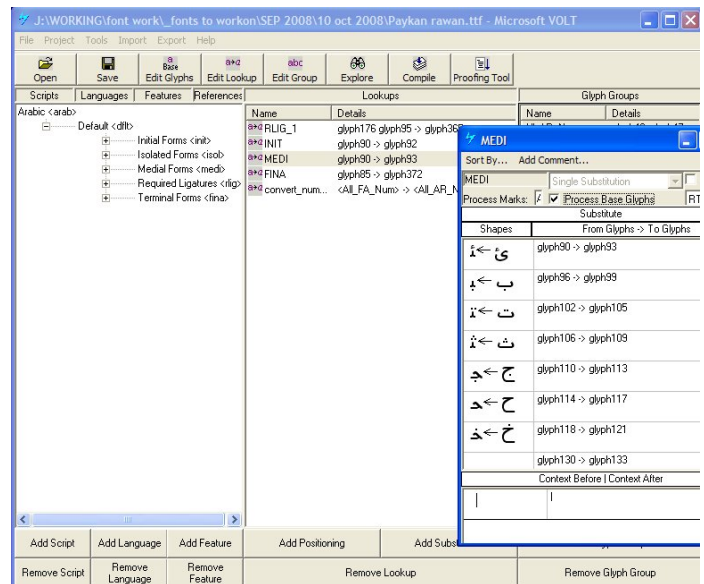
A good document covering OpenType glyph processing in Windows is:

<http://www.microsoft.com/typography/developers/opentype/default.htm>

**Font Lab**



**MS Volt**



MS.Volt image for Glyph substitution

**3. Development Tools**

The Development tools used were FontLab and Ms. Volt Technology.

#### 4. Results

The PaykanRawan font was successfully made and installed on MS. Windows XP. This font has the feature of obliging the typist to use the correct Yeh's used in Pashto writing language. Since there are 2 Yeh's that should not be written in the middle of a word.

The font was acceptable on the computer screen when the clear font option was enabled. In addition to this, better hinting is needed, for enhanced visibility on the screen.

#### 5. Conclusion

The font needs to be better hinted, in order to make it look better on the computer screen.

#### 6. References

1. [www.panl10n.net](http://www.panl10n.net)
2. [http://partners.adobe.com/public/developer/opentype/index\\_glyph2.html](http://partners.adobe.com/public/developer/opentype/index_glyph2.html)
3. <http://groups.msn.com/MicrosoftVOLTusercommunity/homepage.msnw>
4. <http://www.microsoft.com/typography/developers/volt/default.htm>
5. [http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&item\\_id=VOLT\\_Tutorial](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=VOLT_Tutorial)
6. <http://www.true-type-typography.com/>
7. <http://www.crupl.org/software/localization.htm>