

# SCIENTIA BRUNEIANA

2009

## Brief Communications

Can Brunei Darussalam be Asia's Next Leading Location for Regional Treasury Centres?.....	<b>Rady Roswanddy Roslan and Petr Polak</b>	<b>3</b>
---	---	----------

## Research Articles

Font Development for Arwi: An Addition to Arabic Unicode Characters .....	<b>M.I. Seyed Mohamed Buhari</b>	<b>9</b>
Making Rational Decisions by Heuristic Semiquantitative Prognosis .....	<b>David J.H. Brown</b>	<b>35</b>
Option Pricing with Long Memory Interest Rate .....	<b>Abby Tan</b>	<b>49</b>
Emulation of Space Robotics Control by Implementing a Test-Bed System .....	<b>Md. Mahmud Hasan, Ten Seng Teik and Shafina Sultana</b>	<b>57</b>
Determination of Hydroxymethylfufural in Brunei Honey Samples Via the White Method <b>Kooh Chern Yuan, Adeline Sung Chien Yun, Franz L. Wimmer and Linda B.L. Lim</b>		<b>65</b>
Evaluation Study of Heavy Metals Status in Vegetable Cultivation Areas of Brunei Darussalam .....	<b>H.M. Thippeswamy, Hjh Suria Zanuddin, Pg Hjh Rosidah bte Pg Hj Metussin and Hjh Normah Suria Hayati bte Hj Md Jamil Al Sufri</b>	<b>75</b>

# FONT DEVELOPMENT FOR ARWI: AN ADDITION TO ARABIC UNICODE CHARACTERS<sup>1</sup>

*M.I. Seyed Mohamed Buhari*

Department of Computer Science, Faculty of Science, Universiti Brunei Darussalam, Brunei  
Darussalam Email: mibuhari@gmail.com, mibuhari@fos.ubd.edu.bn

---

**Abstract:** The Arwi (also called Arabu-Tamil or Arabic-Tamil) script was widely used by the Muslims of Tamil Nadu (in India), Malaysia, Thailand and Sri Lanka to write many religious, literary and poetry texts and for communication. The Arwi script represents the Tamil language (left-to-right script) using an Arabic style of script (right-to-left script). Arwi is written using Arabic script with additions of certain characters and diacritics. To the author's knowledge, no font or editor to type Arwi exists. In order to uplift the Arwi language into the Information Era, the author has studied and developed Arwi scripting tools using Unicode characters. A JavaScript-based HTML form and an Arwi font have been developed. Using the HTML form, users who know Tamil typing, can type in Tamil script. Others can use the keypad provided to type in Tamil and Arwi scripts and save them in HTML format or copy and paste the contents into any editor. Additionally, a keyboard layout has been provided which can be added to the language bar of the Windows or Linux operating systems and users can type using any editor software. This keyboard layout can be used with the existing Unicode characters on the Operating System or after installing the newly generated Arwi font. Various issues and concerns with regard to the rendering of Arwi Unicode characters in different browsers, different operating systems and different editors are discussed. The Arwi script needs to be incorporated into Unicode while catering for all the rendering issues raised.

---

## 1. Introduction

### 1.1. Tamil Script

The Tamil script, like Devanagari, Malayalam, Telugu, etc, is based on the Brahmi script. Tamil is written in a left-to-right pattern and has 65 characters and variants. To represent the Tamil script, HTML codes in the range 2944 to 3071 or their Unicode equivalents (U+0B80 to U+0BFF) are used. Tamil is spoken in various countries such as Tamil Nadu, Malaysia, Sri Lanka, Singapore, etc. Historically, the Tamil script dates from 100BC [1].

### 1.2. Arabic Script

Arabic Script is recorded to date back thousands of years and belongs to the Semitic language family. Similar to Persian and Urdu, the Arabic script is written right-to-left. The Arabic script consists of 28 characters and 6 vowels. The presence of different forms (initial, middle, final and isolated) for a character and diacritical marks (damma, fatha, kasrah) makes the Arabic script a complex one. Arabic is used as an official language among most countries in the Middle East and North Africa.

---

<sup>1</sup>This report is based on the paper entitled "Arwi: Case Study of Arabic, Syriac, and Diacritical Unicode Characters" presented at the 32nd Internationalization and Unicode Conference, September 8 – 10, 2008, San Jose, CA, USA.

### 1.3 Arwi Script

The Arwi (also called as Arabu-Tamil or Arabic-Tamil) script was widely used by the Muslims of Tamil Nadu (in India) and Sri Lanka [2] to write their religious texts as well as correspondence. The Arwi script [2] was developed as a result of cultural relations between Arabs and Tamil-speaking Muslims in Tamil Nadu. The Arwi script represents the Tamil language using an Arabic style of scripts. This is similar to writing the Malay language in English and the Jawi script. Actually, the Tamil language is written in the Tamil script, which is of left-to-right pattern. Arwi is written using the Arabic script, which is of right-to-left pattern, with an addition of certain diacritics and characters. Arwi uses 13 letters in addition to Arabic scripts [2]. This helped the Muslim community to learn to write Arabic text faster, which is the language of the Holy Quran.

This script did spread to various countries such as Sri Lanka, Malaysia, Thailand, etc. A variety of Islamic books on various topics such as Belief, Law, Sufism, Medicine, etc have been written using the Arwi script. Due to usage of the Arwi script, people have started using various Arabic words as part of their spoken Tamil language. Some commonly used words are Umma (instead of amma in Tamil), Kithab, Mowth, etc. Arwi is still used in certain Islamic schools (Madrashas) in Tamil Nadu. Many books like Adabumalai (About Morale and Discipline), Thakkasuruth (About Rules for Prayer), etc that provide Islamic rules in poetry form were also written in the Arwi script. A sample scanned page from the book titled "Simthu Sibyan" is given in Figure 1.

**Figure 1:** Sample page written in Arwi script, from “Simthu Sibyan” [سيمطُ الصَّيَّان]



To keep Arwi in touch with modern developments, there has arisen a need to have an Arwi font. To the author's knowledge, there is no font that caters for the needs of the Arwi script. Developing a font for this complex scripted language, which has starting, ending and middle forms for each character along with various diacritical marks, is time-consuming. It would be better to use the existing Unicode characters from different languages to help users type in this language.

## 2. Font Development

Font development can be done in two ways. The first approach is to draw the characters using graphics-oriented software such as Photoshop or other tools and then provide the appropriate glyphs and diacritics to be included with them. This process of writing one's own fonts might

be interesting to people who wish to type articles in their own font. But, the major drawback in using this approach is that it is tedious if the characters appear differently when they appear at different positions in the text (at the start or middle or end). It then takes a lot of work to develop the font. Another known drawback of this method is that you need to share your font with others so that they can view articles written using your font and also write articles using your font.

The second approach to font develop is to use Unicode characters. Development in the area of Unicode has been very popular lately because of its wide support by various operating systems, software and browsers and thus there is no need to install any new fonts.

Keeping the above two approaches in mind, the author has ventured into the development of a Unicode font for Arwi which can be used by any user to type. People who wish to install a new font and those who wish to use the existing Unicode characters are duly considered in the development of this font.

### **3. Related Works**

The authors of [2] refer to the importance of the Arwi script. They indicate that Arwi was taught even in Far Eastern countries like Indonesia, Thailand, Malaysia, Myanmar and Pakistan. Famous books by great scholars like Imaam Shaafi (Radiallahu Anhu - May Allah be pleased with him) and Imaam Abu Hanifa (May Allah be pleased with him) have been translated into Arwi. The authors also indicate that the decline of Arwi has caused a steady decline in the education of the women in the latter half of the 20th century. Characters mentioned as work in progress in Figure 2 are handled in our work. Figure 2 provides the current status of Arwi font as described in [2]. From Figure 2, it is obvious that characters from the Arabic (06XX) and Syriac (07XX) ranges of Unicode are considered.

Tschacher [3] refers to the use of Arwi as a mode of teaching Islam and writing poems within the Muslim community in Tamil Nadu. The preference for Arwi by Malaysians in their daily life is attested by Shuayb Alim [4]. The use of Arwi by Muslim communities in Sri Lanka is indicated in [5].

A detailed analysis of the Arwi language is provided in [6, 7]. Various reasons behind the decline of Arwi attested by Shuayb Alim [4] are discussed in [6]. One of the reasons cited is about the lack of printing facilities and the use of Urdu as the teaching medium in many Muslim schools in Tamil Nadu. Our work tries to rectify the problem of printing facilities in this language.

Nuhman [8] refers to a bill to make Sinhala the only official language of Sri Lanka. In the discussion on that bill, some Tamil speakers mentioned that Arwi was used as a writing script by Muslims using the Tamil language. Those speakers were arguing for the importance of making Tamil one of the official languages of Sri Lanka. Nuhman [8] refers to the issues of understanding Arwi scripts by people who understand Arabic and those who understand Tamil. People who understand Arabic can read Arwi but cannot understand Arwi. Those who know Tamil and not Arabic cannot read Arwi but if someone reads it for them they can understand Arwi. The author describes the use of the Arabic script for other languages (such

as Malayalam and Bengali. The author mentions that the problem of the one Tamil letter (L) being equivalent to various Arabic letters (ل, هـ, خ, ح, ق, ك) was solved using the Arwi script.

The author indicates that there were 200 published and around 2000 unpublished literary works written in Arwi. Nuhman [8] quotes the following passage:

“Arabic follows a consonantal system that is, it has distinct symbols or letters only for consonants while the vowels are optional and not

represented by separate letters but by a few diacritical marks without which Arabic texts can still be read and understood. Arabic has 28 consonants and 6 vowels and it is written usually from right to left. Tamil has 30 basic letters comprising 12 vowels and 18 consonants and follows essentially a syllabic system of writing as many other Indian languages, in which vowel sign occur only in the initial position of the words and when they occur after a consonants, the combination of the consonant and the vowel is represented by a syllabic letters. Tamil has 216 syllabic symbols apart from the basic symbols of vowels and consonants and it is written from left to right.”

Thus, two drastically different languages were combined to form a scripting language, Arwi, instead of developing a brand new language. Nuhman [8] concludes by saying that we can use any writing script to write any other language with some modifications, except for those languages like Chinese which use ideographs. There is an indication that Arwi was the writing script of the famous religious book Maghani (The Treasure), written by an influential South Indian scholar from Kayalpatinam, Mapillai Lebbe Alim @ Seyed Mohamed Ibnu Ahamed Lebbe (May Allah be pleased with him) (1816 – 1898).

Mohan [9] indicates that Arwi was used for writing many literary works.

**Figure 2.** Status of the Arwi font

<i>Tamil Equivalent</i>	<i>Arwi Letter</i>	<i>English Equivalent</i>	<i>Pronunciation</i>	<i>Unicode</i>
சச	چ	chā	‘chā’ in ‘chance’	0686
ل	د	dā	‘dā’ in ‘dawn’	068A
ل ل	ذ	tā	‘tā’ in ‘top’	068D
ر	ر	Ra	‘R’ (soft ر)	0694
ظ	ض	‘zha’	Unique to Arwi	06FB
پ	ب	pa	‘pa’ in ‘pause’	06A3
ண	ن	nā	Unique to Arwi	06B9
ஞ	ن	gnā	Unique to Arwi	0767
ஒ	و	o	‘o’ in ‘pot’	0657
ள	ص	l,ā	Strong l.	Work in Progress to Encode these 4 characters
ங	ع	‘nga’	‘ng’ in ‘bang’	
க	ك	gā	‘go’ in ‘gold’	
ம	م	e	‘e’ in ‘men’	

#### 4. Arwi and Its Features

In order to develop the font, we need to know the mapping of the Tamil script with its Arwi counterparts. Such a mapping is given in Table 1.

**Table 1** Equivalence of Tamil letters with Arwi alphabets

அ	ا	க	ك or ك dot below	ய	ي	ஃ	هـ
ஆ	ا	ங	غ	ர	ر	ஔ	و
இ	ا	ச	ج	ற	ر	ஐ	ي
ஈ	اي	ஜ	ج	ல	ل	ஓ	و
உ	ا	ஞ	ج or ج	ள	ض or ص with dot below	ஔ	و
ஊ	أو	ட	د	ழ	ض	ஓ	و
எ	ي or ي	ண	ن	வ	و	ௌ	و
ஏ	ي or ي	த	د or د	ஷ	ص	ஂ	و
ஐ	اي	ந	ن	ஸ	ش	ள	
ஓ	ا	ன	ن	ஹ	ح	ஃ	
ஔ	أو	ப	ب	ா	ا	து	د
ஔ		ம	م	ி	و	ச்ச	ج
				ீ	ي	ட்ட	د

Having seen the mapping, we consider various characters present in the Arwi script and try to find their Unicode equivalents. A few characters and diacritics needed by the Arwi script are not present in Unicode. They are:

- 1 ARABIC LETTER KAF (0643) character which has a dot below.
- 2 ARABIC LETTER AIM (0639) character with three dots below.
- 3 ARABIC INVERTED DAMMAH (064F) below.

Table 2 shows the mapping between the Arwi characters and Unicode. From this table, it is obvious that characters are incorporated from Arabic, Syriac and Combining Diacritical Marks.

**Table 2** Arwi letters with their respective Unicode characters

ا	0627	ص	0635	و	0648	ّ	064D	،	066C
ب	0628	ض	0636	ى	0649	َ	064E	َ	0670
ت	062A	ط	0637	ي	064A	ُ	064F	أ	0671
ث	062B	ظ	0638	ة	0629	ِ	0650	چ	0686
ج	062C	ع	0639	ء	0621	ّ	0651	پ	068A
ح	062D	غ	063A	ﻩ	06E9	ّ	0652	ر	0693
خ	062E	ف	0641	،	0328	َ	0653	ف	06A3
د	062F	ق	0642	؛	061B	ُ	0654	ن	06BA
ذ	0630	ك	0643	؟	061F	ِ	0655	غ	06A0
ر	0631	ل	0644	آ	0622	ِ	0656	ض	06FB
ز	0632	م	0645	-	0640	ُ	0657	ِ	0734
س	0633	ن	0646	ّ	064B	.	065C	ِ	0746
ش	0634	ه	0647	ّ	064C	,	066B		

Table 3 shows differences in the number system used in Arabic (present at U+0661 to U+0669) and Arwi (present at U+06F1 to U+06F9) writings. In some cases, the number 7 is represented similarly to the English letter "L".

**Table 3** Different number representations in the Unicode Arabic character range

٠	١	٢	٣	٤	٥	٦	٧	٨	٩
0660	0661	0662	0663	0664	0665	0666	0667	0668	0669
٠	١	٢	٣	٤	٥	٦	٧	٨	٩
06F0	06F1	06F2	06F3	06F4	06F5	06F6	06F7	06F8	06F9

## 5. Solutions Provided:

To cater for the needs of the Arwi community, two options are provided.

- First, we provide an option for those who wish to type a document in Arwi without installing any new font. An HTML-based typing tool has been provided.
- Second, we provide the keyboard layout for the Windows and Linux operating systems which could be set up using something similar to the Regional and Language Settings in the Control Panel. The keyboard layout has been made to cater for those who just use the existing Unicode characters and those who use the new Arwi font.

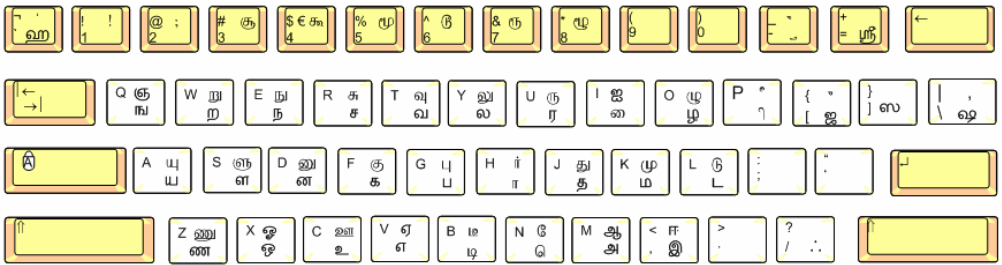
At first, to enable typing in the Arabic or Arwi fonts in the Windows operating system, the "Install files for complex script and right-to-left languages (including Thai)" option must be enabled on the user's PC. This is done using the Regional and Language Settings in the

Control Panel. In Linux variants, BD (Bidirectional) or Multilingual Support is enabled by default. For further information, refer to [10].

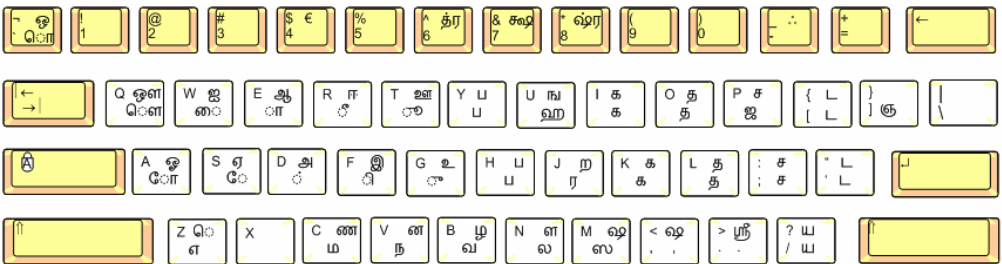
### 5.1. HTML Based Arwi Typing Tool

The Unicode characters present in the range U+0600 to U+06FF (Arabic), U+0700 to U+074F (Syriac) and U+0300 to U+036F (Combining Diacritical Marks) are used. The use of Arabic Presentation Forms A (U+FE70 to U+FEFF) and B (U+FB50 to U+FDFF) was considered but not implemented. This is because Arabic Presentation Forms are not included by many Arabic or Unicode fonts and these forms were included only to provide compatibility with preexisting standards and legacy implementations. A HTML page which has the options shown in Figure 5 is provided. Options are provided so that the user can type if he/she knows to type in Tamil or Arwi. If a user does not know how to type in Tamil or Arwi, he could use the virtual keypad provided in the software. Due to the presence of various keymaps (as shown in Figures 3 and 4) for Tamil typing, we provide two common options: One is for Unicode fonts like Latha and the other for fonts like Bamini or Sarukesi. Those who have learnt Tamil typing using a typewriter find it difficult to move on to Unicode-based Tamil Fonts. There exists certain software that could convert text from one font to another.

**Figure 3** Tamil keypad for Unicode font (Latha)



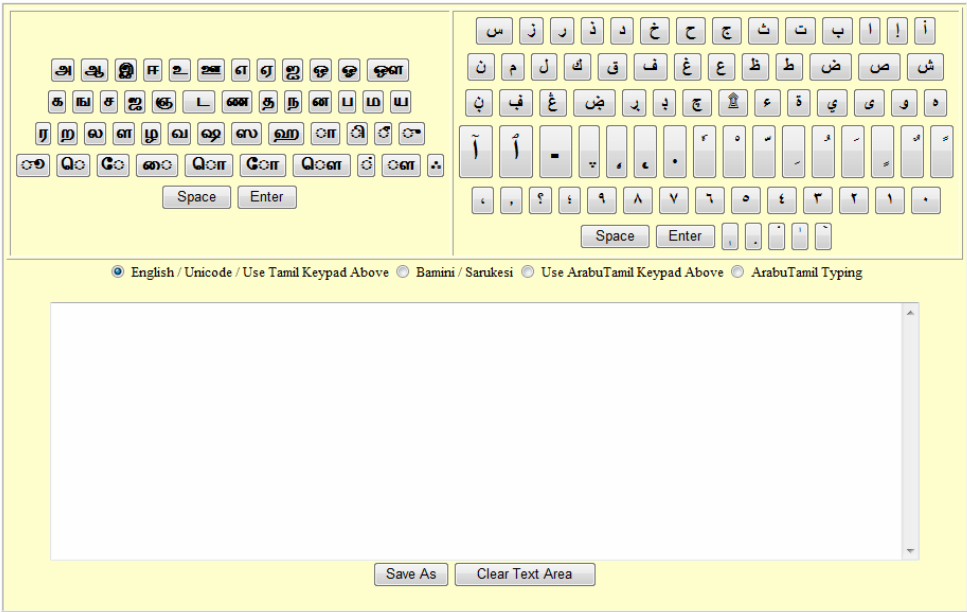
**Figure 4.**Tamil keypad for Bamini or Sarukesi fonts



This software is made using JavaScript and thus does not require any server side support. You can obtain the whole code and run it on any machine. It runs on both the Windows and Linux operating systems. This software provides options for users to mix both Tamil and Arwi scripts even though that is not the normal method of writing Arwi script. When the user types in Arwi, the character alignment becomes right-to-left



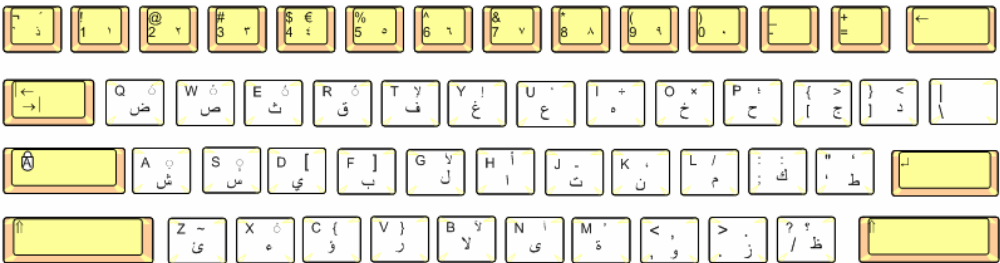
**Figure 5** HTML-based Arwi typing tool



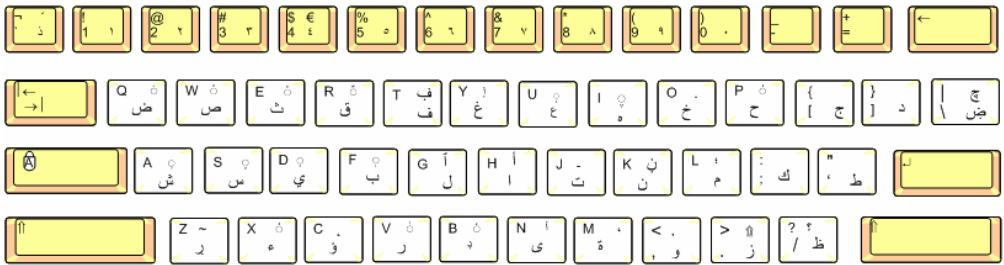
**5.2. Using a Keymap Facility to Type Arwi Using Various Editors**

If the user wishes to type in Arwi using various editors, he/she needs the Arwi keymap to be installed. It would be good to have an entry in the language bar, which any user can use to start typing in Arwi. We have designed the keypad for Arwi to be similar to that of Arabic, so as to make it easier for those who already knew Arabic typing. A tradeoff exists on whether to match the Arwi keyboard layout to that of the Tamil keyboard because people who speak Tamil might not know Arabic typing. In doing so, we have tried our best to make the keypad as close as possible to that of the Arabic keypad. But it is a known fact that people who use the Arwi script know the Arabic script. Also, most Tamil-speaking people know English typing rather than Tamil typing. Figures 6 and 7 provide pictorial views of the Arabic and Arwi keyboard layouts respectively.

**Figure 6** Arabic keyboard layout



**Figure 7** Arwi keyboard layout



In order to provide that option in Windows operating systems, we have used the Keyboard Layout Manager Software [11]. Using the Keyboard Layout Manager software, we provide the user with the Keyboard Layout file, which is named ArabuTamil.klm2000. To install the ArabuTamil keypad on any Windows machine, first insrall and open the Keyboard Layout Manager Software, then click New under Keyboards and in the layout type "ArabuTamil" and select any language that you are not using or planning to use. We have used Arabic (Yemen) for testing purposes because there is no Arwi layout up till now. Then click Create, and once the Option is created select the ArabuTamil option and click Edit. Click on Import and select the ArabuTamil.klm2000 file. Then click Open followed by OK twice. Finally, Confirm changes. Then, in the language bar, the user will find the necessary ArabuTamil (as Arabic (Yemen)) option present. Now the user can type ArabuTamil in any editor software by selecting the ArabuTamil option present in the language bar. This process is shown in Figure 8. The process of developing a keymap for a Unix-based operating System is discussed in Section 7.

## 6. Problems Faced in Arwi Unicode Development

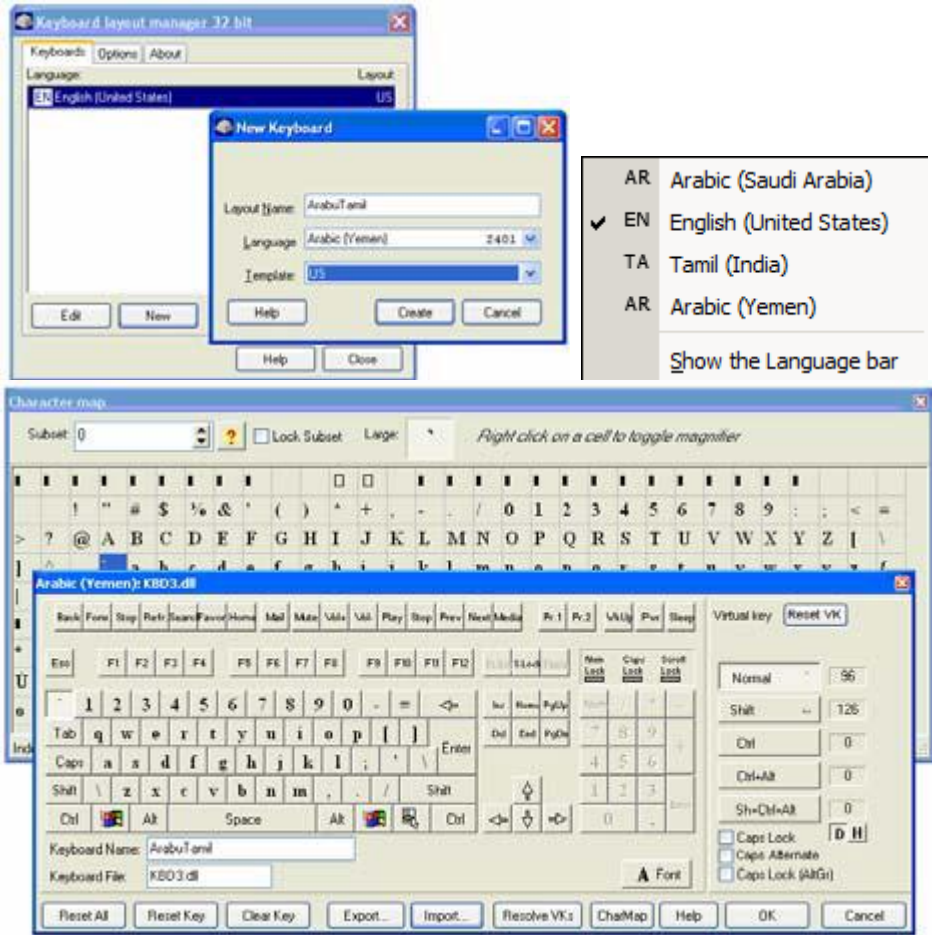
As stated earlier, Arwi uses Arabic characters which have lots of character representations. It needs to use certain scripts from Syriac and many others from Arabic and Arabic Supplements. As per the Unicode Standard 4.0, chapter 8, Syriac Qushshaya (U+0741 – dot above) and Syriac Rukkakha (U+0742 – dot below) can be used with only specific Syriac letters. Combining Syriac with the Arabic and Arabic supplements cause problems in aligning the character along with the diacritics. The diacritics meant for the previously typed character might stand alone without joining the previous and next characters. Combining characters or diacritics from different ranges like Syriac and Arabic in our case can cause many rendering problems.

Generally, only one character range should be used to type in a language. Certain rendering engines have their rules based on fixed code points. If we use different ranges, the presence of different code points makes the rendering ineffective. But Combining Diacritical Marks (U+0300 to U+036F) can be used with different character ranges. As a rendering issue, if one existing diacritic in Arabic is copied into another code point it will be found that the positioning of the new diacritic is not the same as that of the original diacritic, even though both of them are just copies of one another.

A similar rendering problem has been widely noted on Firefox and Opera browsers when people use Unicode characters from languages such as Tamil. This is one of the reasons why when developing a website in Tamil using Unicode the page is best viewed using Internet Explorer. A sample website [<http://zeyarath.blogspot.com>] being rendered differently in different browsers is shown Figure 9. It should be noted that Internet Explorer 7 and Mozilla

Firefox 3 render properly. Mozilla Firefox 2 does not render the text properly. Mozilla Firefox 3 uses the Uniscribe rendering tool.

**Figure 8** Arwi keyboard setup for Windows clients



**Figure 9** A Tamil Website rendered on different browsers

**Friday, April 25, 2008**

**மரணத்திற்கப்பாலும் ஹயாத்துண்டா?**

தத்துவத்தையுடையவர்கள் மரணித்து மண்ணோடு  
மண்ணாகிப் போய் விடுவதில்லை! உடல்களும்  
நசிப்பதில்லை, ஜீவியத்தில் இருந்தது போலவே  
கபுரிலும் சடலம் கோர்வை குலையாமலிருக்கும்.

Internet Explorer 7

Friday, April 25, 2008

**மரணத்திற்கப்பாலும் ஹயாத்துண்டா?**

தந்தைவந்தையுடையவர்கள் மரணித்தோ மண்ணோடு  
விட்டுவதில்லை! உடல்களும் தனிப்பதில்லை, துவ  
பேரலவே கபூரிலும் சடலம் தோர்வன கொலையாய்  
ஆனபியர்களுடைய உடல்களந்த தனிப்பதன் இறன்

Mozilla Firefox 2.0.0.16

Friday, April 25, 2008

**மரணத்திற்கப்பாலும் ஹயாத்துண்டா?**

தத்துவத்தையுடையவர்கள் மரணித்து மண்ணோடு  
மண்ணாகிப் போய் விடுவதில்லை! உடல்களும்  
தனிப்பதில்லை, துவியத்தில் இருந்து போலவே கபூரிலும்  
சடலம் தோர்வை குலையாமலிருக்கும்.

Mozilla Firefox 3.0.1

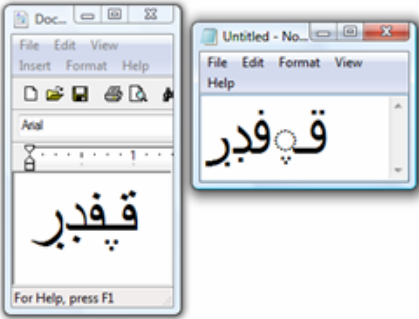

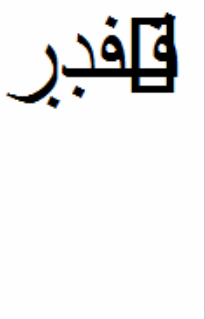
### 6.1. Unicode Rendering in WordPad and Notepad

Using the keyboard layout provided, the user can type directly into Notepad or WordPad or Microsoft Word. The user could type using the HTML provided and copy and paste the contents onto the Notepad or WordPad or Microsoft Word or any other software. As can be seen from Figure 10 below, the rendering of Unicode characters is done better in WordPad than in Notepad or Microsoft Word.

The authors of [12] have described rendering problems for Unicode characters in various Indian languages. They have specifically addressed the problems with Tamil Unicode characters. The authors of [12] also mention that characters coupled with characters such as the Zero Width Joiner (ZWJ), Zero Width Non Joiner (ZWNJ) etc., can cause serious headaches to the text processing applications. At the same time, the presence of zero-width glyphs is very important for Indian language fonts.

A Zero Width Non Joiner (ZWNJ) is permitted by Wordpad but not by Notepad. Thus, Wordpad 6.0 renders Arwi better than Notepad 6.0, Microsoft Word 2003 and OpenOffice.org Writer 2.1.

**Figure 10** Rendering Arwi Unicode in Notepad or Wordpad or Microsoft Word

Words copied from the browser to WordPad and Notepad	From WordPad to Microsoft Word	From Browser to Microsoft Word
		

## 6.2. Problems with different Browsers on different operating Systems:

We have tested our HTML page on various operating systems such as Windows XP, Vista and SuSe Linux 10.2.

### 6.2.1. Issues on Windows Vista

Both HTML-based Arwi typing and Keyboard Layout based Arwi typing approaches work fine in Internet Explorer (Version: 7.0.6000.16386) and Firefox (Version: 2.0.0.13).

### 6.2.2. Issues on Windows XP

Internet Explorer (Version: 6.0.2900.2180.xpsp\_sp2\_rtm.040803-2158) does not display a few characters such as those with Unicode numbers 0656 (Arabic Subscript Alef), 0657 (Arabic Inverted Damma), 065C (Arabic Vowel Sign Dot Below), 0328 (Combining Ogonek, part of Combinatorial Diacritical Marks) properly. Even after upgrading the Internet Explorer to 7.0.5730.13 version, same problems persist. The Arwi virtual keypad for the Internet Explorer is shown in Figure 11.

**Figure 11** Arwi virtual keypad on Internet Explorer (Windows XP)



To verify whether this is a problem with Internet Explorer or with the operating system itself, we tested our HTML-based Arwi Typing page on Firefox 2.0.0.13 on Windows XP. In Firefox also, certain Unicode characters like 0656, 0657 and 0328 did not appear properly. Syriac Unicode characters like 0746 (Syriac three dots below) and 0734 (Syriac Zqapha below) did not join with the previous character and appeared separately. General diacritical marks which belong to Arabic Script (Like Fathah, Damma, etc) did not appear on the display in the virtual keypad provided with the HTML script, but when clicked they worked fine. The size of them seemed to be very small when viewed, but we could insert an image of them instead by using the following code:

```
document.write('<TD>');
document.write('<INPUT type="button" style="font-size: 30; font-weight:bold"
name="\u0746" value="\u0746 "
onclick=AppendCharacter("\u0640\u200d\u070f\u0746")>');
document.write('</TD>');
```

It would be better to replace all the characters on the virtual keypad with appropriate images. In the code above, the problem of the Syriac characters not joining with the previous

and next Arwi Script was handled using Unicode character 0640 (hyphen), 200D (Zero Width Joiner) and 070F (Syriac Abbreviation Mark). This works fine on Firefox and Internet Explorer browsers on Microsoft Vista. We did download the Arial Font from the Internet (Arial32.exe) and used it with Windows XP. After doing this, Unicode character 0328 did work fine but did not appear properly in the display.

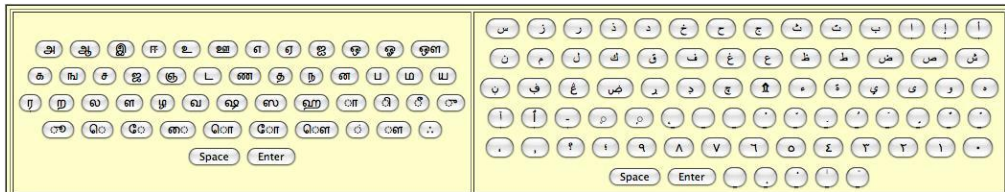
In Mozilla Firefox 3, the Arabic HTML appears better than expected for the U+0657 character. Also, Syrian characters did not require the joining sequence we have used in the code (\u0640\u200d\u070f) to join properly.

**Figure 12** Arwi virtual keypad on Mozilla Firefox 3 (Windows XP)



In Safari Version 3.1.2 (525.21) on Windows XP, all the characters seem to work fine both on the virtual keypad display and when used.

**Figure 13** Tamil and Arwi virtual keypad on Safari 3.1.2 (Windows XP)



### 6.2.3. Issues on SuSe 10.2 with Firefox 2.0.0.6:

In the Tamil virtual keypad part of our HTML-based Arwi typing software, characters 〰 and 〰 did not appear in the view but worked properly when we use them. With regards to the Arwi virtual keypad, all general diacritical marks worked fine but did not appear in the view. Characters with Unicode numbers 0657 and 0328 appeared as separate characters and did not join with the previous characters. A few other Unicode characters like 06E9, 065C, 0734 and 0746 did not appear properly. The issues discussed here are illustrated in Figure 14.



[illegible]

**Figure 15** Arwi virtual keypad on Firefox 3 (SuSe 10.2)



After finding that few characters do not appear properly on the Windows XP and SuSe Linux operating systems, we checked for the presence of the Unicode character in Windows XP. This is done using Charmap with Advanced view. We selected the Unicode subrange in the “Group by” option and selected Combining Diacritical Marks and Arabic to verify the presence of the Unicode characters. We could conclude that few characters were not present in Windows XP and thus could not be displayed. It is to be noted that Arial Unicode MS is not being frequently updated. Overall, characters or diacritics from different language ranges in Unicode should not be mixed. A comparison about the fonts and glyphs present in Windows XP and Windows Vista is presented in [13].

Font	XP File Size	Vista File Size	XP Glyphs	Vista Glyphs
Arial	359KB	749KB	1680	3381
Arial Black	115KB	117KB	669	674
Arial Bold	344KB	728KB	1680	3381
Arial Bold Italic	222KB	539KB	966	2516
Arial Italic	203KB	534KB	966	2516

**Table 4b** Comparing font support in Windows Vista and Windows XP

Criteria	Windows XP SP2	Windows Vista
Size	91.4 MB	290+ MB
Number of Glyphs	218,725	712,000+ Glyphs Almost 500,000 new glyphs
Number of Fonts	133	191 Fonts installed by default 68 New fonts, 10 Removed

## 7. Enabling and Setting Up the New Keymap in the Linux Operating System

### 7.1. Verification of Unicode Support in the Linux Operating System

In order to verify whether Unicode is enabled on the machine, the locale command is used. The user can use **locale -a** to find out what languages are supported by the machine. To change the locale settings for your account, open the ~/.profile file and add the line:

```
export LANG=en_US.UTF-8
```

If the user wishes to make the default font an Arabic font, the user can change the LANG option to something like ar\_SA.UTF-8, which stands for Unicode of Arabic (Saudi Arabia). Setting the LANG option to ar\_SA.UTF-8 seems to change the Firefox browser menus into the Arabic language in SuSe 10.2. In order to make the change in profile take effect, you need to login again. This can be done by pressing Ctrl+Alt+BackSpace.

### 7.2. Setting Up the Keymap for Vim in the Linux Operating System

The necessary keymaps for vim are present in /usr/share/vim/current/keymap folder. For each language, you can find the keymap for both Unicode and non-Unicode characters. The Unicode keymap for the Arabic language is arabic\_utf-8.vim. In order to write the arwi keymap, we can copy the arabic\_utf-8.vim file into arwi\_utf-8.vim. The contents of this file indicate the link between the characters of the keyboard with the hexadecimal and decimal representation of the Unicode characters. You can provide a comment to indicate what each character means. Part of the Arwi keymap is given the Table 5. In Table 5, the character 'q' (Lowercase) is mapped to Arabic Dad, which is represented in hexadecimal and decimal as 0x0636 and 1590 respectively.

**Table 5** A portion of the keymap file in Linux

let b:keymap_name = "arwi"			
loadkeymap			
q	<char-0x0636>	" (1590)	- DAD
w	<char-0x0635>	" (1589)	- SAD
e	<char-0x062b>	" (1579)	- THEH

Once the keymap is ready, you can type in the vim using the statement **set keymap=arwi**, after pressing **Esc+:**. With the Arabic keymap in vim, certain characters like ‘, لا, ة, ل, لا, ئ, did not appear properly while typing using vim.



7.3. Setting up the Input Locals in the Linux Operating System

If you want to enable the language bar in the task bar of the desktop, you need to do the following:

- 1. Click Personal Settings (Configure Desktop).
- 2. Select Regional & Accessibility.
- 3. Select Keyboard Layout.
- 4. Select the language that you want to present on the language bar from the Available Layout and click Add >>.

Using these steps, you can click on the language bar and type in different languages. But, this will work only for those languages for which the keymap is already available. For this work, we wish to design a new Keymap for the Arwi script. In order to design a new keymap, we need to go to the /usr/share/X11/xkb or /etc/X11/xkb or /usr/X11R6/lib/X11/xkb folders. Different Linux variants have different folders for the xkb. We then need to add an entry under the **! layout** section of the base.lst file, which is present in /usr/share/X11/xkb/rules folder. This entry is shown in the Table 6.

**Table 6** A portion of the base.lst keymap file in Linux

! layout	
us	U.S. English
ad	Andorra
af	Afghanistan
ara	Arabic
arwi	Arwi (Arabu-Tamil)
al	Albania

As seen in Table 6, an entry was made for the Arwi font. This line can be added at any spot in the layout section because it will just appear in alphabetical order in the Available Layouts section of the Keyboard Layout. The next step is to add the entry for the Arwi script in the base.xml file which is present in the /usr/share/X11/xkb/rules folder. A partial copy of the layout section of the Arabic script can be taken and modified as shown in Table 7.

Then we need to make the actual keymap available in the /usr/share/X11/xkb/symbols folder. After the reserved keyword "key" we use a representation to indicate each row of the keyboard. AE stands for the row with numbers 1, 2, 3,... . AE01 indicates the first key, which is the number 1 key. AE02 indicates the second key, which is the number 2 key and so on. AD stands for the row starting with QWERT. AD01 represents the key "q". AC stands for the row starting with ASDF. AC01 represents the key "a". AB stands for the row starting with ZXCV. AB01 represents the key "z". In the keymap, the lowercase representation and uppercase representation are separated by a comma. A part of the Arwi keymap is shown in Table 8. The way by which we assign Unicode characters to keys is also shown in Table 8.

Whenever we make a change to the keymap file (present in the /usr/share/X11/xkb/symbols folder), we need to remove the keymap from the Keyboard layout and then add again from the Available Layout and click Apply, as shown in Figure 16. Only then will these changes come into effect.

Instead of making changes to the keymap permanent, if we want to temporarily test the keymap (for example, the arwi keymap), we could use the following command:

setxkbmap -layout arwi

**Table 7** A portion of the base.lst keymap file in Linux

```
<layout>
  <configItem>
    <name>arwi</name>
    <shortDescription>Arwi</shortDescription>
    <shortDescription xml:lang="es">Ara</shortDescription>
    <description>Arwi</description>
    <description xml:lang="af">Arabies</description>
  </configItem>
  <variantList>
    <variant>
      <configItem>
        <name>azerty</name>
        <description>azerty</description>
        <description xml:lang="en_GB">azerty</description>
      </configItem>
    </variant>
  </variantList>
</layout>
```

**Table 8** A portion of the Arwi keymap file in Linux

key <TLDE> {	[	Arabic_thal,	Arabic_shadda	]	};
key <AE01> {	[	1,	exclam	]	};
key <AE02> {	[	2,	at	]	};
key <AC07> {	[	0x1000698,	0x1000699	]	};

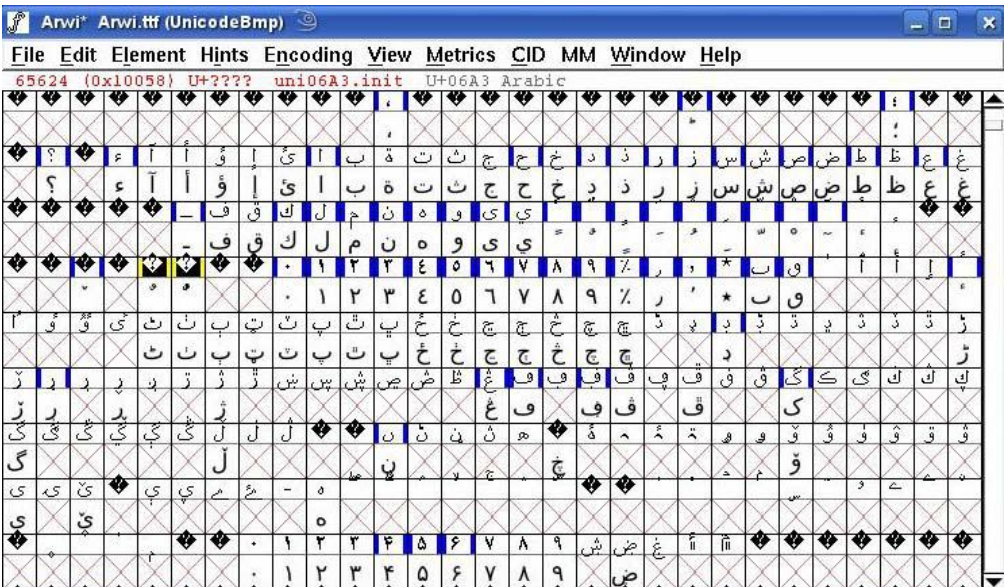
**8. Design and Development of the Arwi Font**

When developing the font, authors made a point of developing truetype font which could be used by both the Windows and Linux operating systems. Making the font a Unicode font was also considered. Development of the font was done using the Fontforge software on SuSe 10.2. The Open Source font “DejaVuSans.ttf” font present in the /usr/share/fonts/truetype folder was selected as the base font. Installation of the Fontforge software on a SuSe machine was straightforward with rpm (rpm -ivf fontforge-i386.rpm). In Fontforge, when we opened the DejaVuSans font, we could see that each character is shown as two cells (Figure 17). The cell on the top indicates the character and the bottom cell indicates the drawing or representation of the character. Certain characters are noted with an X mark in the bottom cell, indicating that they cannot be changed. The name of the font can be changed using Font Info under the Element menu in the Fontforge software. To generate the font, use the Generate Fonts option under the File menu and then select TTF type. It was noted that we needed to close Fontforge before the font was available for typing in any editor software.

Figure 16 Keyboard layout in Linux



Figure 17 Arwi Font development using Fontforge software



A few characters have been added to the DejaVuSans font. These characters along with their Unicode representations are shown in Table 3.

**Table 3** Arwi characters added to the DejaVuSans font

ء	0621	ّ	0653	ء	066C	ف	06A3
ا	06E9	ُ	0654	ا	0670	ن	06B9
ا	0328	ِ	0655	آ	0671	ع	06A0
آ	0622	ِ	0656	چ	0686	ض	06FB
-	0640	ُ	0657	د	068A	ِ	0734
ّ	0651	.	065C	ر	0694	ِ	0746
ّ	0652	,	066B				

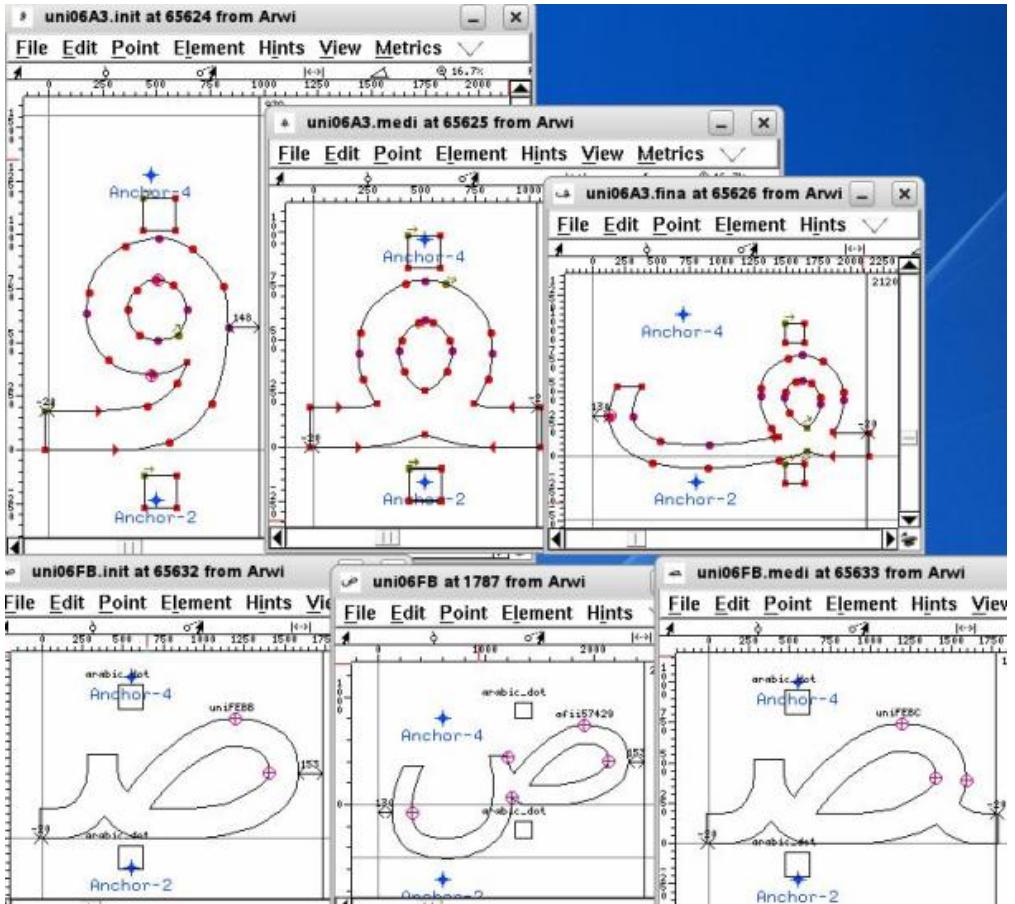
When using the font, the user needs to know which key on the keyboard is mapped to which character so that he/she can type. Once the font is ready, we can install the keyboard layout on the Windows machine (Section 5.2) or Linux machine (Section 7) and type using any editor software such as Microsoft Word or OpenOffice. Users can make HTML pages using this font by including the tag <font face="arwi">.

### 8.1. Issues with the Fontforge software

The font won't work if we just add a new glyph to a character for which there is no entry in the box. Add the entry in the box and then add the glyph (middle, initial and final). In order to add a new character, we must use the available Unicode code point. It should be noted that all the glyphs for a single character using the same Unicode code point of the base character. Let's say that we add a character at Uni06A1. Then, we need to provide the necessary glyph (media, init or fina), called as substitutions, as shown in Figure 18. We need to link the substitution glyph with the original base character. This is done using Element → Glyph Info → Substitutions. Select the appropriate substitution like 'init' or 'medi' or 'fina' and link to the newly created glyph.

- 'medi': Medial forms in Arabic Lookup 8 subtable
- 'fina': Terminal forms in Arabic Lookup 9 subtable
- 'init': Initial forms in Arabic Lookup 10 subtable

**Figure 18** Glyphs for two different characters



To add new glyphs to the given font, we need to add slots and proceed to enter the glyph and then link this glyph to the base character. This is done using the procedure described below:

1. Indicate the number of glyphs you want to add.
  - a. Encoding → Add Encoding Slots
2. Select each one of the newly added slots and do the following:
  - a. Element → Glyph Info:
    - i. For a Glyph
      1. Unicode Name: uni06A1.init
      2. Unicode Value: -1
    - ii. For a base character
      1. Unicode Name: uni06A1
      2. Unicode Value: U+06a1
  - b. Then click 'Set From Name' (in Glyph Info) and click 'OK'.
  - c. For each and every glyph created, we need to click File → Generate Fonts → Save as True type. It should be noted that the user's rights are to be considered when saving the fonts in the respective folders.

3. Make sure that the glyph is added to the substitutions option in the main or base Unicode font.
4. After doing the above, if you wish to see the impact on OpenOffice.org Writer, you need to close OpenOffice.org Writer and re-open it. Also, make sure the keymap entry is removed and added again (if there needs to be a change in the keymap).

At the base character, we need to make sure that the OT Glyph class type under Element → Glyph Info → Unicode is set to Base Glyph. From the glyph, we can see which base character it is linked to using: Element → Show Dependent → Substitutions. To view the glyph, use the following sample procedure: View → Goto → Uni06A1.medi.

In order to have the same lookup information for two characters, we can use Edit → Copy Lookup Data on the original and then paste into the character we want. Sometimes a link from an existing character to another existing character needs to be removed. For example, initially 06BA was linked to afii57446 and this link was removed using Edit → Unlink Reference. 06BA.medi and 06BA.init were modified to finish the task.

## 8.2. Testing of the Arwi font and the issues faced

The Arwi font was tested using SuSe 10.2, Ubuntu 7.04 and Ubuntu 8.04. The Arwi font did not appear properly in OpenOffice.org 2.0.4 build 2.0.4.7 and Opcion Font Viewer 1.1.1. Similar problems persisted in Ubuntu 7.04. Problems persisted when new diacritics were added. When tested with Ubuntu 8.04, the Arwi font worked as expected in OpenOffice.org 2.4.0, GTK+ Editor, gedit, kate and Opcion Font Viewer, but not in QT Editor. These differences were due to the different rendering engines used. Different OpenType capable text rendering engines exist: OpenOffice.org and SIL's XeTeX use IBM's International Components for Unicode (ICU), QT4 has its own engine based on HarfBuzz, GTK+ uses Pango which is using 21 HarfBuzz internally, the New TeX engine LuaTeX has its own OpenType engine, Microsoft has its Uniscribe engine, Adobe has a hybrid Apple Advanced Typography (AAT)-OpenType engine used on Mac OS X. Overall, one rendering engine is shared between GTK/QT4; another in OpenOffice.org; Microsoft has its own.

A simple test was done by copying the existing diacritical character (damma) to a different position. The results of this testing are shown in Figure 19. It should be noted that certain rendering engines work on specific locations for considering a character as a diacritic and so if a character is added to a new location, rendering won't work correctly.

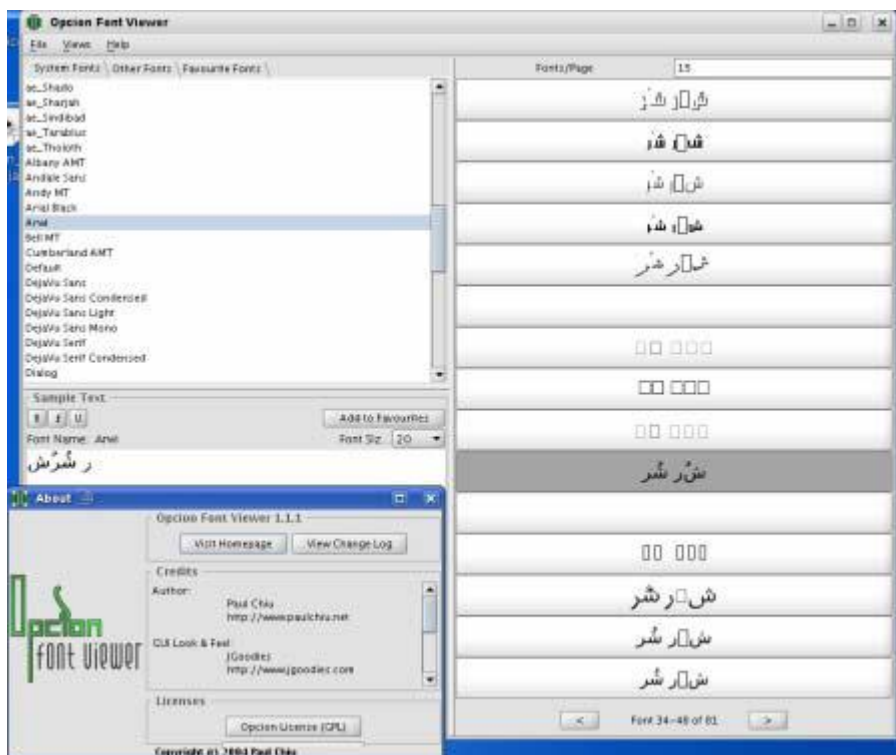
**Figure 19** Arwi font testing on different tools



PDF by XeTeX and ConTeXt

Specimen Font Previewer





Option Font Viewer (Ubuntu 7.04)



Ubuntu 8.04

As concluding remarks, we need to make sure that the initial, final and middle glyphs of the characters 0686, 068A, 068D, 0693, 0694, 06A3, 06B9, 06BA, 06A0, and 06FB are

present. The presence of diacritical marks 0653, 0656, 0657 and 0670 should also be verified. In addition, a proposal for the following is to be submitted to the Unicode Consortium:

1. ARABIC LETTER KAF (0643) Character which has a dot below.
2. ARABIC LETTER AIM (0639) Character with three dots below.
3. ARABIC INVERTED DAMMAH (064F) below.

A ligature joining the representation of the diacritic Maddah with the diacritic Fatha is also present in the Arwi script.

Certain characters in Arwi script have different possible representations. These could be handled by the fonts themselves. They are listed as follows:

1. The Arwi representation for the number 7 similar to the English character “L” might be supported by a font by providing multiple glyphs for U+06F7.
2. The equivalent of ARABIC LETTER NOON WITH TWO DOTS BELOW (0767).
3. The Unicode character for ARABIC LETTER SAD (0635) WITH A DOT BELOW

## 9. Conclusion and Future Work

The need for the development of the Arwi font has been the focus of this report. The Arwi language was used among Muslims in many countries to write many religious and literary works. Lack of adequate typing and printing facilities has caused the sharp decline in the use of the Arwi script. The authors have studied the development of this script and have designed an HTML based client-side script, Arwi font and a language bar oriented Arwi typing option. From the analysis, it is obvious that the HTML-based typing tool has many shortcomings with regards to operating systems other than Windows Vista. A proposal for the inclusion of Arwi characters is being prepared for submission to the Unicode consortium. As possible future work, collation of the Arwi script needs to be researched.

## 10. References

- [1] How to read and write Tamil characters:  
[Available at <http://www.xs4all.nl/~wjsn/tamil.htm>]  
and History of Writing  
[Available at <http://www.xs4all.nl/~wjsn/tekst/taalschriften.htm#QXQ>]
- [2] Arwi language – Wikipedia  
[Available at [http://en.wikipedia.org/wiki/Arwi\\_language](http://en.wikipedia.org/wiki/Arwi_language)]
- [3] Tschacher, T. 2004. How to die before dying? Sharia and Sufism in a 19th century Arabic Tamil Poem. Panel 38, 18th European Conference on Modern South Asian Studies, Lund, Sweden, 6 – 9 July 2004.  
[Available at <http://www.sasnet.lu.se/panelabstracts/38.html>]
- [4] Shuayb Alim. 1993. Arabic, Arwi and Persian in Sarandib and Tamil Nadu. Madras.
- [5] Sri Lanka – Ethnic Groups:  
[Available at <http://countrystudies.us/sri-lanka/38.htm>]



- [6] Arwi  
[Available at <http://www.armu.com/armu/works/archives/12dec1998/amc1.html>]
- [7] Tschacher, T. 2004. Arwi (Arabic-Tamil) – An Introduction [Available at <http://web.archive.org/web/20040822180630/www.fas.nus.edu.sg/journal/kolam/vols/kolam5&6/1AOldLit/Arwi.htm>]
- [8] Nuhman, M.A. 2007. Sri Lankan Muslims: Ethnic Identity within Cultural Diversity. International Centre For Ethnic Studies, Colombo, Sri Lanka.
- [9] Mohan, V. 1983. Muslims of Sri Lanka. Aalekh Publishers, Jaipur, India.
- [10] Help: Multilingual support (indic) – Wikipedia  
[Available at [http://en.wikipedia.org/wiki/Help:Multilingual\\_support\\_\(Indic\)](http://en.wikipedia.org/wiki/Help:Multilingual_support_(Indic))]
- [11] Keyboard Layout Manager  
[Available at <http://www.klm32.com/>]
- [12] Unicode rendering problems for Indic scripts.  
[Available at [http://acharya.iitm.ac.in/multi\\_sys/unicode/render/ren\\_07.php](http://acharya.iitm.ac.in/multi_sys/unicode/render/ren_07.php)]
- [13] Proceedings of 32nd Internationalization and Unicode Conference, San Jose, USA, 8 -10, September 2008.

## Appendix A: Basic Definitions

Reference: [13]

**Glyphs:** A “glyph” is screen unit of text. It is a picture of what users think of as a character; A “grapheme” is a single visual unit of text.

**Characters:** A “character” is a single logical unit of text. A “character set” is a set of characters, called a “repertoire”. A “code point” is a number assigned to a character in a character set. A “coded character set” is a character set in which each character has a code point.

**Bytes:** A “character encoding” maps a sequence of code points (“characters”) to a sequence of code units (such as bytes). A “code unit” is a single logical unit of storage.

**Ligature:** Ligatures are very common. Essentially a ligature is a single glyph that represents more than one underlying character. A number of ligatures vary between fonts of a single script. Here are few examples:

- The default position of U+0651 ARABIC SHADDA as a glyph, is above the base character, whereas for U+064D ARABIC KASRATAN is placed below the base character, as a glyph. However, certain computer fonts follow an approach that originated in metal typesetting and combine the kasratan with shadda in a ligature above the text.
- If we join ARABIC LETTER LAM (0644) and ARABIC LETTER ALEF (0627), we

get “Lam-Alef”.

- Joining of characters to form the words “Allah” or “Sallallahu Alaihiwassalam”.

## **Appendix B: Process for Submitting a Character Proposal to the Unicode Consortium**

Reference: [www.unicode.org](http://www.unicode.org)

1. First, check whether the character or diacritic you want to propose is something that already exists or something which is in the pipeline.

a. Check the chart: <http://www.unicode.org/charts/>

b. Pipeline: <http://www.unicode.org/alloc/Pipeline.html>

2. If not there, write the proposal, according to:

a. <http://std.dkuug.dk/JTC1/SC2/WG2/docs/summaryform.html>

3. Only the base character is proposed. But the forms need to be shown in the proposal. These glyphs have to be scanned from a printed text. Internet sources like Wikipedia are not accepted as reference. Also, we need to include the proposed character in a font and provide a document file with a screen shot or a pdf file with the font embedded.

4. Note: Character size, italic and bold forms of a character are handled by the encoding algorithm present in the application software.