Project Meltho: Syriac Wordprocessing under Windows (Report I)

George Anton Kiraz
Computer Laboratory
University of Cambridge (St John's College)
Email: George.Kiraz@cl.cam.ac.uk
URL: http://www.cl.cam.ac.uk/users/gk105

July 30, 1996

Abstract

This paper forms the first report of Project Meltho, from the Syriac Computing Institute. The project aims at implementing a Syriac word processing system for Microsoft Word for Windows.

1 Introduction

Until the late 1980s only large publication houses were able to produce good-quality Syriac typesetting, though at substantial cost. As a result, authors had to resort to handwriting, while others – who were expected to produce their work in camera-ready-copy form – either had to resort to transliteration or to endure the high costs of typesetting. The development of Kiraz's Syriac font package (Kiraz, 7 92) for Multi-Lingual Scholar (Gamma Productions Inc.) – which runs under DOS – provided computer users with the means to produce good quality Syriac typesetting.

Today, however, with Windows being ubiquitous, there is a need for a new Syriac word processing system which runs under Microsoft Word. This is the aim of Project Meltho. When completed, Meltho will be available as a shareware product.

2 Current Windows Products

There are currently a number of programs which allow the user to input Syriac texts under Windows.

- Gamma UniVerse (by Gamma Production Inc.). This program is intended to be the Windows version of Multi-Lingual Scholar. The Syriac fonts are based on Kiraz's fonts for the latter. Although, UniVerse allows the user to enter Syriac texts freely, it does not provide for any of the word processing features one would expect from a word processor such as footnotes, columns, etc.

- Gamma UniType (by Gamma Productions Inc.). The Syriac fonts here are also based on Kiraz's fonts for Multi-Lingual Scholar. UniType is a localisation software which allows the user to enter Syriac texts (one sentence at a time) to any Windows application. This is a very useful tool and allows the user, for example, to prepare presentations, banners, leaflets etc. However, because one cannot enter more than one sentence at a time, editing texts can be very cumbersome (UniType was never meant to be a word processor).

- The Syriac Writer (by Esho Marcus and Sargon Hasso). This package provides an add-on to Al-Kaatib International, a bi-lingual word processor from Eastern Language Systems. Al-Kaatib provides many of the word processing features one would expect from a word processor. It is not clear, however, if there will be further developments to the program.

Additionally, there have been attempts to provide Syriac support for Arabic Windows, simply by providing Syriac fonts which map Syriac letters to their Arabic counterparts (see below).

3 Problem Statement

One of the main obstacles in Syriac word processing is that of bidirectionality which can easily be avoided using Arabic Windows. This, however, does not imply that implementing Syriac for Arabic Windows is a straightforward task.

Arabic and Syriac, though share common typesetting characteristics, are very different in many respects. The following is a list of the most important differences.

1. Contextual Analysis. Syriac *he, ṣadhe* and *taw* do not connect to the following character, unlike their counterparts in Arabic. This causes the following character to take the shape of middle or final form rather than initial or stand alone, respectively.

2. Character Hight. Arabic Windows makes use of 'high diacritics' for tall characters, and 'low diacritics' for short ones. The hight of Syriac and Arabic letters is not compatible, e.g. Arabic *ta* is short, while Syriac *taw* is tall. This causes diacritics to appear *on* consonants, rather than *above* them.

3. Overstrike. Arabic diacritics do not correspond to Syriac diacritics.

4. Ligatures. There is no correspondence between Syriac and Arabic ligatures.

The above problems can be resolved simply by designing a Syriac font in the following manner:

1. Contextual Analysis. Placing Syriac *he, ṣadhe* and *taw* to Arabic letters which do not connect to the following character such as *Hamzated Aliph, ta marbūṭa* etc. The disadvantage is that letters will be assigned to obscure keys in the keyboard layout.

2. Character Hight. Making all diacritics vertically hight. This, however, will result in bad positioning of vowels.

3. Overstrike. Limiting the number of Syriac oversrikes to the maximum number of oversrikes in Arabic fonts.

4. Ligatures. There are two solutions here: (1) Designing combinations of Syriac letters to mimic Arabic ligatures (e.g. *beth* + *mim* which is not a genuine Syriac ligature). In this case, the font will be full of such ligatures and there will be no room left for real Syriac ligatures. (2) Turning ligatures off. In this case, the user will not be able to make use of Arabic ligatures.

This crud solution does not provide Microsoft Word with any mean to distinguish between Syriac and Arabic texts. Any text-related operation (e.g. search, replace, spell checking, etc.) will cause corruption to the text.

4 New Approach

Meltho aims to provide a 'clean' solution to the problem at hand, where Syriac and Arabic texts remain always distinguished. Meltho makes use of the Arabic features which apply to Syriac. Syriac specific features are dealt with independently using `SyrWin`, a library of functions for handling Syriac texts.

4.1 The SyrWin Library

The SyrWin library provides for all the functions related to Syriac texts. Program developers can use SyrWin to add Syriac support to their Windows applications. The library contains definitions of letters and diacritics, their characteristics and any other information about them. When loaded, SyrWin reads a number of configuration files and builds tables to map symbols to their location in fonts, keyboard tables, sorting tables, etc.

The C code for using SyrWin takes the following form:

```
/* load SyrWin */
...

/* initialise SyrWin */
SW_Initialise();

/* use SyrWin functions */
...

/* shut down SyrWin */
SW_ShutDown();

/* unload SyrWin */
```

A complete reference to SyrWin will be provided upon completion of the library.

4.2 The Main of Meltho

The main core of Meltho is written in C and is compiled into a dynamic link library. When loaded, it runs behind the scene acting as a filter between the user and Microsoft Word. Meltho looks at all events caused by the user's input. If an event is related to Syriac, Meltho handles it accordingly and sends the necessary commands to Microsoft Word. Otherwise, Meltho passes the user's input untouched to Microsoft Word.

4.3 Features

The first version of Meltho will support the following features:

- Fonts. Estrangelo, Serto and East Syriac. Users will be able to use third party fonts.
- Keyboard. Various keyboard layouts will be provided. Additionally, users can produce their own layouts.

- Vowels. High vowels will be placed on tall consonants and low vowels on short one.
- Ligatures. Ligatures will apply automatically, but can be turned off by the user.
- Garshūnī. Macros will be provided to convert Arabic texts between Syriac and Arabic scripts.
- Transliteration. Macros will be provided to allow the automatic transliteration of Syriac texts.
- Find/Replace. This will be handled as to keep the Syriac and Arabic texts distinguished. Other text-related features might be added in the first release.

5 Conclusion

We are now at the preliminary stages of this project. The basic functionality of `SyrWin` has been designed and implemented. The same applies to the interface between Meltho and Microsoft Word.

In the future, we are also looking at providing Syriac support for Microsoft Access.

Acknowledgements

The Syriac Computing Institute thanks the supporters of Project Meltho: Mor Gabriel Monastery, Turkey; Microsoft Inc., Middle East Product Development, USA; Dr. Sebastian P. Brock, UK.

References

Kiraz, G. (1987-92). *Alaph Beth Font Kit: Aramaic, Coptic, Hieroglyphic, North Semitic, Phoenician, Sabaean, Syriac and Ugaritic Fonts.* Alaph Beth Computer Systems.