ܣܶܪܛܳܐ *Serṭo* – a font for Syriac (Aramaic)

Johannes Heinecke

Version 0.3, 9th June 2003

Contents

1 Introduction

Serṭo is the form of the alphabet used for Syriac (Aramaic, a western semitic language).[1] It was mainly used in Syria since the 16th century. Syriac used other alphabets as well, notably *Esṭrangelo*, which is not contained in this package. Since Serṭo is as the Arabic alphabet a syllabic script, vowels are marked by diacritic marks above (or under) the consonantic letters. Modern forms of Aramaic still use either Serṭo or Esṭrangelo.

This package enables you to typeset words or paragraphs in Serṭo using a preprocessor which chooses the correct letter form depending on context. In order to typeset paragraphs the use of `elatex` is needed. In this case the preprocessor must be used with the option `-e` (see section 6).

2 The Alphabet

Every letter in Serṭo has several forms, depending on its position in the word: An initial, medial or final form. Since some letters do not connect to the following letter, there are isolated forms as well (i.e. a letter which is not connected to the right nor the left). The coding column in the following table refers to the preprocessor described below (section 6). If you do not want to use the preprocessor, please refer to the encoding table in section 2.5.

2.1 consonants

forms				name		coding
isolated	final	medial	initial			
ܐ	ܐ			ܐܳܠܰܦ	*Ōlaf*	'
ܒ	ܒ	ܒ	ܒ	ܒܶܝܬ݂	*Bēṯ*	`b`
ܓ	ܓ	ܓ	ܓ	ܓܳܡܰܠ	*Gōmal*	`g`

[1] See Ungnad 1932, Brockelmann 1960, Costaz 1986 or Nöldeke 1986f (English translation Nöldeke 2001) for further information on Syriac.

forms				name	coding
isolated	final	medial	initial		
ܕ	ܕ			ܕܳܠܰܬ *Dōlaṯ*	`d`
ܗ	ܗ			ܗܶܐ *Hē*	`h`
ܘ	ܘ			ܘܰܘ *Waw*	`w`
ܙ	ܙ			ܙܰܝܢ *Zain*	`z`
ܚ	ܚ	ܚ	ܚ	ܚܶܬ *Ḥēṯ*	`.h`
ܛ	ܛ	ܛ	ܛ	ܛܶܬ *Ṭēṯ*	`.t`
ܝ	ܝ	ܝ	ܝ	ܝܽܘܕ *Yōḏ*	`y`
ܟ	ܟ	ܟ	ܟ	ܟܳܦ *Kōf*	`k`
ܠ	ܠ	ܠ	ܠ	ܠܳܡܰܕ *Lōmaḏ*	`l`
ܡ	ܡ	ܡ	ܡ	ܡܺܝܡ *Mīm*	`m`
ܢ	ܢ	ܢ	ܢ	ܢܽܘܢ *Nūn*	`n`
ܣ	ܣ	ܣ	ܣ	ܣܶܡܟܰܬ *Semkaṯ*	`s`
ܥ	ܥ	ܥ	ܥ	ܥܶܐ $^{c}\bar{e}$	`` ` ``
ܦ	ܦ	ܦ	ܦ	ܦܶܐ *Pē*	`p`
ܨ	ܨ			ܨܳܕܶܐ *Ṣōḏē*	`.s`
ܩ	ܩ	ܩ	ܩ	ܩܳܦ *Qōf*	`q`
ܪ	ܪ	ܪ	ܪ	ܪܺܝܫ *Rīš*	`r`
ܪ̈	ܪ̈	ܪ̈	ܪ̈		`R`
ܫ	ܫ	ܫ	ܫ	ܫܺܝܢ *Šīn*	`^s`
ܬ	ܬ			ܬܰܘ *Taw*	`t`

2.2 Using the preprocessor

The preprocessor recognizes two types of commands. Within a single line you can put Syriac words between <S> and </S>: For example <S>ser.to'</S> becomes ܣܶܪܛܳܐ.

<ST> and </ST> generate the enclosed part in Serṭo and generates a transliteration as well (<ST>foruwqo'</ST> becomes ܦܽܘܪܩܳܐ *'orūqō*), whereas <T> and </T> can be used for parts only need in transliterated form (<T>ser.to'</T> becomes *serṭō*). Since in transliteration a "neutral vowel" is needed, which does not appear in Serṭo, the code @ can be used: <ST>^s@mo`</ST> produces ܫܡܳܥ *šᵉmoʿ*

For multiple lines, start a block using <SERTO> in a line on its own. This block can be closed by </SERTO>. If you need transliterated Syriac, use </TRANS> and </TRANS>. The commands <SERTO>/</SERTO> and </TRANS>/</TRANS> do not work properly without the -e option of the preprocessor and elatex. If you add TEX-commands in these blocks, elatex is obligatory.

For the time being the preprocessor tries to set the hard sign ܩܘܫܝܐ *quššōyō* on top of a consonant if the consonant is doubled in the input:

<S>q.tl</S> yields ܩܛܠ but <S>q.t.tl</S> yields ܩܛ݁ܠ

To get the soft sign ܪܽܘܟܳܟ݂ܳܐ *rukkōḵō* a + must *follow* the letter:

<S>'ab+do'</S> yields ܐܰܒ݂ܕܳܐ

An Ōlaf is autmatically prefixed before an initial vowel:

<S>etqa.tel</S> and <S>'etqa.tel</S> both yield ܐܶܬܩܰܛܶܠ

Sometimes the letter *rīš* is written with two points. To achieve this, use R instead of r in the input:

<S>^sapiyRe'</S> yields ܫܰܦܺܝܪ̈ܶܐ

Silent consonants have a bar ܡܒܰܛܠܳܢܳܐ *mvaṭlōnō* under the line:

`<S>'an=tt</S>` yields ܐܢ݁ܬ

There is no automatic stretching yet, but the `--` can be used to insert a “manual stretch”:

`<S>napiyqto'</S>` becomes ܢܦܝܩܬܐ but `<S>na--piyq--to'</S>` is printed as ܢܦܝܩܬܐ

If you do not use the preprocessor, you can activate Serṭo by the command `\serto`. In this case you have to choose the correct letters yourself, and use the commands `\upperserto{`*vowelnumber*`}{`*letter*`}` or `\lowerserto{`*vowelnumber*`}{`*letter*`}` to set vowels.

In order to get bold letters, you can use `\sertob` with or without preprocessor (see section 6 for more information on the preprocessor.

2.3 Vowels

For the time being only the greek vowels. The Chaldean letters may follow one day. To have the vowel symbol written in inversed form under the consonant, user upper case input.

form	name	coding
◌ܰ	ܦܬܳܚܳܐ *pṯōḥō*	`a`
◌ܶ	ܪܒܳܨܳܐ *rvōṣō*	`e`
◌ܺ	ܚܒܳܨܳܐ *ḥvōṣō*	`i`
◌ܳ	ܙܩܳܦܳܐ *zqōfō*	`o`
◌ܽ	ܥܨܳܨܳܐ *ᶜṣōṣō*	`u`

form	coding
◌ܱ	`A`
◌ܷ	`E`
◌ܻ	`I`
◌ܴ	`O`
◌ܾ	`U`

To avoid that the simple preprocessor does not mess about with vowels in ligatures (notably *Lōmaḏ-ōlaf* and *Ōlaf-lōmaḏ* the vowel must be set after both consonants of the ligature:

<S>lo'</S> yields (incorrect) ܠܳܐ but <S>l'o</S> yields the vocalized ligature ܠܳܐ

<S>'AlOho'</S> yields (incorrect) ܐܰܠܳܗܳܐ but <S>'LaOho'</S> yields the vocalized ligature ܐܰܠܳܗܳܐ

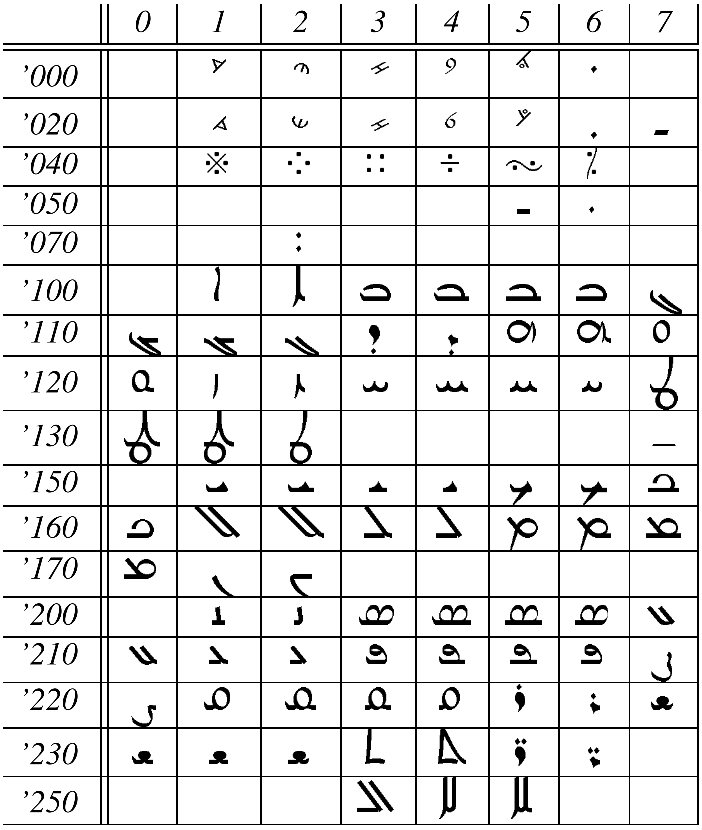
2.4 Punctuation and Paragraph marks

form	coding
·	`.`
⁚	`:`
※	`.X.`
⁘	`.:.`

form	coding
∷	`::`
÷	`:-`
·~·	`.~.`
⁒	`./.`

2.5 The encoding

The following table shows the internal encoding of the defined letters of Serṭo.



3 Two Examples

Some of the following can be found in `example.ptex` which comes with this package. If you do not have `elatex`, you cannot set texts in Serṭo but only single words.

The following input was used to generate the output below:

```
\documentclass[12pt,a4paper]{article}
usepackage{serto}

\begin{document}

<S>men qadiy^se' ho' tetqada^s</S>

<ST>kmo' dat+basb@suwn pagdo'</ST>

\end{document}
```

ܡܢ ܡܕܝܢܬܐ ܗܘܐ ܠܐܠܗܐ

ܟܡܐ ܕܬܒܣܒܣܘܢ ܦܓܕܐ *kmō daṯbasbesūn pagdō*

In order to set more than just words `elatex` (needed for the `TeX--XeT` extension) must be used. `elatex` properly sets text from the right to the left.

ܘܟܕ ܚܙܐܘ ܟܘܡܪ̈ܐ ܕܚܛܝܬܐ: ܡܕܡ ܕܥܒܕ ܢܨܝܚܐ: ܠܐܒܟܘܗܝ ܘܐܣܪܘܗܝ ܘܐܝܬܝܘܗܝ ܩܕܡ ܡܠܟܐ. ܘܐܫܬܥܝܘ ܠܡܠܟܐ ܡܕܡ ܕܥܒܕ ܠܐܠܗܝܗܘܢ. ܡܠܟܐ ܕܝܢ ܐܡܪ. ܠܐ ܗܘܐ ܡܫܬܘܕܝܘ ܐܫܬܘܕܝܬ ܠܝ ܕܬܕܒܚ ܠܐܦܘܠܘ: ܢܨܝܚܐ ܐܡܪ. ܠܝ ܐܠܗܐ ܐܝܠܝܢ ܕܡܐܬܥܝܢ. ܠܐ ܗܘܐ ܐܠܗܐ ܐܢܘܢ. ܩܪܒ ܐܢܘܢ ܠܝ ܠܗܪܟܐ. ܘܠܐ ܫܘܒܚܕܗܘܢ ܕܟܝܣܛܝܘܢܐ ܠܐ ܫܒܩ ܐ̱ܢܐ ܚܕ ܡܢܗܘܢ ܕܠܐ ܡܕܩܩ ܐ̱ܢܐ ܥܕܡܐ ܠܪܡܫܐ.

The preceding text was set typeset with the following input:

```
\documentclass[12pt,a4paper]{article}
\usepackage{serto}
\begin{document}

<SERTO>

wkod .hzaw kuwmd'e da.h.tIytO' : medem
da`bad na.siy.ho': lAbkuwh=y=
wa'sruwh=y= wa'ytIywh=y= qdom malko'.
we'^stA`iyw lmalko' me----dem da`bad
{\sertob l'AlOhayhwn}. molko' deyn 'emar. l'o hwo'
me^stAwdoyow 'e^stEwdiyt lIy dtEdba.h
l'Apoluw: na.siy.ho' 'emar. liy 'laOhe'
'ayleyn dma't`eyn. l'o hwo' 'laOhe'
'Enuwn. qareb 'Enuwn lIy lhorko'. \sertob wl'O
^suwbhodhuwn dakiys.tyone' l'o ^sobeq
'=no' .had menhuwn dl'o mdaqeq '=no'
`admo' lram^so'.
```

```
</SERTO>

\end{document}
```

The file was preprocessed using `serto.py -e` (the option `-e` indicates the preprocessor, to generate a LaTeX-file with the necessary commands to enable `TeX--XeT`.

4 Things still missing

The todo-list is long. I try to add features etc. as soon as possible. Please tell me items you would like to have, but which are not yet on this list. Any volunteers are welcome!

- proper typsetting of texts (without preprocessor and `elatex`, maybe in the ArabTeXpackage)
- proper treatment of *matrēs lectionis* (long vowels)
- treatment of the ܣܝ̈ܡܐ *seyōmē*
- proper treatment of the silent bar ܡܒܛܠܢܐ *mvaṭlōnō*
- proper treatment of *beġaḏkefaṯ* with hard sign ܩܘܫܝܐ *quššōyō* and soft sign ܪܘܟܟܐ *rukkōḫō*
- automatic transcription mode
- interpunction
- proper dealing with ligatures
- Chaldean vowels
- unicode input
- ܐܣܛܪܢܓܠܐ and Chaldean glyphs
- numbers
- support for typical encodings and for unicode (UTF-8) either via the preprocessor or via the `inputenc.sty` package.

- creating Type 1 fonts from the METAFONT-sources
- support for existing Syriac fonts

5 Installation

Put the `*.mf` files into a subdirectory `serto` of your metafont branch in your `texmf`-directory. For example using teTEX under Linux, you should put them into `/usr/local/share/texmf/fonts/source/serto/`.

The Stylefile etc. `*.sty`, `*.fd` go into a directory for stylefiles, e.g. `/usr/local/share/texmf/tex/latex/serto/`.

The preprocessor `serto.py` and the encoding file `syriac.font` somewhere where it can be found (e.g `/usr/local/bin`). Possibly you have to adjust the first line of the preprocessor `#!/usr/bin/python` if your python interpreter is somewhere else. In order to find the encoding file `syriac.font` you should specify its absolute path in the programme.
(e.g. `FONTFILE="/usr/local/lib/syriac.font"`).

6 The preprocessor serto.py

Since this is an extremely early version of the font, typesetting of texts is not yet possible with standard LaTeX, since the right-to-left typesetting (as for instance as in ArabTEX) has not yet been implemented. For the time being I propose a preprocessor (written in PYTHON[2]) and `elatex`.

The preprocessor is called with the LaTeX-file as argument:

```
serto.py [-e] ppfilename.tex > filename.tex
```

The resulting LaTeX-file can be LaTeXed as usually. Please make sure to have the `\usepackage{serto}` included in your preamble.

The option `-e` is necessary if you use `elatex`. Since `elatex` is capable to typeset texts from the right to the lext, the `-e` -option tells the preprocessor not to inverse the letters on its own. In order to typeset whole paragraphs `elatex` is the better solution. Usually it comes with every modern TEX-distribution.

[2] Every version from 1.5.2 onwards should do. PYTHON is an interpreted object oriented programming language available for many operating systems (`http://www.python.org`).

References

Brockelmann, Carl: 1960. *Syrische Grammatik mit Paradigmen, Literatur, Chrestomathie und Glossar*. Leipzig: VEB Enzyklopädie, 8th edn.

Costaz, Louis: 1986. *Grammaire Syriaque*. Beyrouth: Dar el-Machreq.

Nöldeke, Theodor: 1986f. *Kurzgefaßte syrische Grammatik. (Neubearbeitung von Schall, Anton)*. Darmstadt: Wissenschaftliche Buchgesellschaft.

Nöldeke, Theodor: 2001. *Compendious Syriac Grammar. Translated by James A. Crichton*. Winona Lake, IN: Eisenbrauns.

Ungnad, Arthur: 1932. *Syrische Grammatik mit Übungsbuch*. München: Beck, 2nd edn.