Serṭo – a font for Syriac (Aramaic)

Johannes Heinecke

Version 1.3, 4th November 2021

Contents

1 Introduction 2

2 The alphabets 2
  2.1 Consonants ............................................. 3
  2.2 Vowels ....................................................... 5
  2.3 Transliteration and long vowels .......................... 6
  2.4 Punctuation and paragraph marks ..................... 7
  2.5 Unicode ..................................................... 7
  2.6 The encoding .............................................. 7

3 The preprocessor serto[2].py 8
  3.1 Using the preprocessor .............................. 9

4 The format of the *.font files 11

5 Two examples 12

6 Things still missing 14

7 Installation 14
  7.1 Using Metafont sources ............................. 14
  7.2 Using vector fonts .................................. 15
  7.3 Other Files ............................................. 15

8 License 16

9 Changelog 16
1 Introduction

Serṭo is a form of the alphabet used for Aramaic (a western semitic language) which has been spoken in the Near East since at least 1100 BC. More precisely, Serṭo is used for Syriac which is the variant of Aramaic spoken since the second century AD.¹

Syriac used other alphabets as well, notably Estrangelo, which is indirectly contained in this package (for use with XƎLÂTEX only, see below). 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, the Chaldean alphabet or Estrangelo. Since Syriac split up in two main dialects in the fifth century AD, two differing systems of vowel-marking were established: Whereas the western dialect (Edessean) used Greek letters as vowel symbols, the eastern dialects uses dots to indicate the vowels (Chaldean vowels).

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 a recent version of pdfLÂTEX is needed, which can handle the right-to-left typesetting. For older version of LÂTEX, the preprocessor must be used with the option -o (see section 3).

This package also includes an adapted version of a Chaldean font (thanks to Tony Khoshaba, who put this font to the Web).

In order to use the Estrangelo alphabet, you have to use XƎLÂTEX instead of pdfLÂTEX, and install the font Estrangelo Edessa. For copyright reason, this font cannot be included in this package². The support for Estrangelo is still experimental.

2 The alphabets

Every letter in Serṭo (and some letters in the Chaldean alphabet and Estrangelo) 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

---

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

²The font can be easily found on the Web, or downloaded at https://www.wfonts.com/font/estrangelo-edessa
the preprocessor described below (section 3). If you do not want to use the preprocessor, please refer to the encoding table in section 2.6.

# 2.1 Consonants

<table>
<thead>
<tr>
<th></th>
<th>Serṭo</th>
<th>Estr.</th>
<th>Chaldean</th>
<th>name</th>
<th>translit.</th>
<th>coding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>isolated</td>
<td>final</td>
<td>medial</td>
<td>initial</td>
<td>isolated</td>
<td>final</td>
</tr>
<tr>
<td>1</td>
<td>ܐ</td>
<td>ܐ ܐ</td>
<td>ܒ</td>
<td>ܒ ܒ</td>
<td>ܓ</td>
<td>ܓ ܓ</td>
</tr>
<tr>
<td>6</td>
<td>ܒ</td>
<td>ܒ ܒ</td>
<td>܉</td>
<td>܉ ܉</td>
<td>Ͼ</td>
<td>Ͼ Ͼ</td>
</tr>
<tr>
<td></td>
<td>ḫ</td>
<td>ḫ ḫ</td>
<td>ܘ</td>
<td>ܘ ܘ</td>
<td>ܓ</td>
<td>ܓ ܓ</td>
</tr>
<tr>
<td></td>
<td>ṭ</td>
<td>ṭ ṭ</td>
<td>ܟ</td>
<td>ܟ ܟ</td>
<td>ܟ</td>
<td>ܟ ܟ</td>
</tr>
<tr>
<td></td>
<td>Serṭo</td>
<td>Estr.</td>
<td>name</td>
<td>translit.</td>
<td>coding</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>-----------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>isolated final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>medial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isolated final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isolated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isolated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chaldean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>h</td>
<td></td>
<td></td>
<td></td>
<td>_k</td>
<td></td>
</tr>
<tr>
<td></td>
<td>č</td>
<td></td>
<td></td>
<td></td>
<td>ˇk</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lāmad</td>
<td>mīn</td>
<td>nūn</td>
<td>semkat</td>
<td>s</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pē</td>
<td>p</td>
<td>p</td>
<td>semkat</td>
<td>s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>f</td>
<td>f</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>qāp</td>
<td>q</td>
<td>q</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rīš</td>
<td>r</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>šīn</td>
<td>š</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>taw</td>
<td>t</td>
<td>t</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.2 Vowels

The package allows to typeset the greek vowels or Chaldean vowels symbols. To have the vowel symbol written in inversed form under the consonant, user upper case input.

<table>
<thead>
<tr>
<th>Greek</th>
<th>Chaldean</th>
<th>name</th>
<th>transliteration</th>
<th>coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>ἀ</td>
<td>ḯ</td>
<td>ptāhā</td>
<td>a</td>
<td>A</td>
</tr>
<tr>
<td>ἐ</td>
<td>ḯ</td>
<td>rbāṣā</td>
<td>e</td>
<td>E</td>
</tr>
<tr>
<td>ἰ</td>
<td>ṣ</td>
<td>ḫbāṣā</td>
<td>i</td>
<td>I</td>
</tr>
<tr>
<td>ṳ</td>
<td>ṣ</td>
<td>ḫqāp̄ā</td>
<td>ā</td>
<td>=A</td>
</tr>
<tr>
<td>ἰ</td>
<td>ṣ</td>
<td>ḫṣāšā</td>
<td>u</td>
<td>U</td>
</tr>
<tr>
<td>ἀ</td>
<td>ḯ</td>
<td>syāmē</td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>

Note: The ḫbāṣā and ḫṣāšā of theEastern or Chaldean vowels do in general occur together with the mater lectionis: ḫff t (or ḫff IL, ḫff OL)

The ḫṣāšā can be typeset directly.

Modern Aramaic dialects using the Chaldean alphabet have diacritic symbols (dots and tildes) which can be typeset directly.

3 The ḫdkt are not yet always processed. In general, the doubling of the consonant creates a ḫdkt quššāyā in the syriac text and does not change the transliteration. On the other hand, a consonant followed by + will receive a ḫdkt rūkkāhā and in the transliteration ḫdkt will appear as ḫdkt.

4 Modern Aramaic dialects using the Chaldean alphabet have diacritic symbols (dots and tildes) which can be typeset directly.
both consonants of the ligature:

\(<S>l=a'</S>\) yields (incorrect) \(\La\) lā but \(<S>l'=a'</S>\) yields the vocalized ligature \(\La\) lā

\(<S>'A1=Ah=a'</S>\) yields (incorrect) \(\La\) alahā but \(<S>'l=a=Ah=a'</S>\) yields the vocalized ligature \(\La\) alaahā

The Chaldean letters do not have this ligature. Instead, a tawālā ligature is provided: \(\La\) siprāyīta

The default vowels are the greek-based vowels. In order to get Chaldean vowels, it suffices to add : in front of the vowel in coding. Thus you can set the most famous Aramaic phrase in all Syriac alphabets in either vowel system:

\(<S>eliy eliy lm=an=a' s=ab=akt=aniy</S>\)

\(\La\) elī - elī lmānā sābāktānī

For Estrangelo the same vowel codings can be used

### 2.3 Transliteration and long vowels

As mention in section 3 below, the preprocessor can produce a transliteration as well. The transliteration can be defined in the table used by the preprocessor serto.font and assyr.font. The current definition uses the transcription as shown in the tables in sections 2.1 and 2.2, with the exception of long vowels. In words using a vowel symbol together with a mater lectionis, the transliteration shows the transcription instead, for instance \(\La\) transliterates as \(\La\) het̲ and not \(\La\) heyt̲ and \(\La\) qā̄yem: 
### 2.4 Punctuation and paragraph marks

<table>
<thead>
<tr>
<th>Form</th>
<th>Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>*</td>
<td>.X.</td>
</tr>
<tr>
<td>⋯</td>
<td>⋯</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Form</th>
<th>Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>::</td>
<td>::</td>
</tr>
<tr>
<td>÷</td>
<td>÷:</td>
</tr>
<tr>
<td>∼</td>
<td>∼:</td>
</tr>
<tr>
<td>⋯</td>
<td>⋯:</td>
</tr>
</tbody>
</table>

### 2.5 Unicode

This package has a limited Unicode support in that texts encoded in UTF8 can be directly typeset. In order to activate the UTF8 interpretation, either use UTF8 for your whole document by declaring `\usepackage[utf8]{inputenc}` in the preamble of your document, or just put `\usepackage[utf8]{inputenc}` somewhere at the beginning of your document.

### 2.6 The encoding

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

<table>
<thead>
<tr>
<th>Greek</th>
<th>Chaldean</th>
<th>Transliteration</th>
<th>Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ̌</td>
<td>μ̌</td>
<td>ǎ</td>
<td>=a'</td>
</tr>
<tr>
<td>ṃ̌</td>
<td></td>
<td>ě</td>
<td>ey</td>
</tr>
<tr>
<td>ṭ̌</td>
<td></td>
<td>ě</td>
<td>e'</td>
</tr>
<tr>
<td>ṹ̌</td>
<td></td>
<td>ǐ</td>
<td>iy</td>
</tr>
<tr>
<td>α̌</td>
<td></td>
<td>ǔ</td>
<td>ǔ</td>
</tr>
</tbody>
</table>
3 The preprocessor serto[2].py

Typesetting of texts is still not yet possible with standard \LaTeX, since the right-to-left typesetting (as for instance as in Arab\LaTeX) has not yet been implemented. For the time being I propose a preprocessor (written in PYTHON\(^5\)) and pdfl\LaTeX\(^6\).

I'm well aware that serto[2].py is not part of the most beautiful pieces of software code, on the contrary, it’s rather spaghetti code. Many things could have been in a more intelligent way, but it works, which is the most

\(^5\)Every version from 2.4 onwards up to version 2.7 should do for serto2.py; for PYTHON 3.\(^*\) use serto.py.

\(^6\)XƎL\LaTeX can handle UTF-8. However I have not yet adapted this package to XƎL\LaTeX.
important thing. If you find the time to improve it please share your changes with me!

3.1 Using the preprocessor

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

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

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

The option `-o` is necessary if you use an older version of `\LaTeX` which is not capable to typeset texts from the right to the left (`\TeX--\XeT` extension). The `-o`-option tells the preprocessor to inverse the letters on its own. In order to typeset whole paragraphs `pdf\LaTeX` is the better solution. Usually it comes with every modern `\TeX`-distribution. At least since Ubuntu 12.04 (texlive package), the standardly installed `pdf\LaTeX` behaves correctly.

The preprocessor recognizes two types of commands. Within a single line you can put Syriac words between `<S>` and `</S>`: For example `<S>ser.t=a'</S>` becomes ܒܨܢ ܒܨܢ.

`<ST>` and `</ST>` generate the enclosed part in Serṭo and generates a transliteration as well (`<ST>mdiyt=a'</ST>` becomes ܡܕܝܬ ܡܕܝܬ “city”), whereas `<T>` and `</T>` can be used for parts only need in transliterated form (`<T>ser.t=a'</T>` becomes ܡܪܛܐ). Since in transliteration a “neutral vowel” is needed, which does not appear in Serṭo, the code @ can be used: `<ST>`^s@m=a`</ST>` produces ܠܙ ܢܐ.

For multiple lines, start a block using `<SERTO>` in a line on its own. This block must be closed by a line containing `</SERTO>`. If you need transliterated Syriac, use `</TRANS>` and `</TRANS>`. The commands `<SERTO>`/`</SERTO>` and `</TRANS>`/`</TRANS>` do not work properly with the `-o` option of the preprocessor and and older `\LaTeX`. If you add `\TeX`-commands in these blocks, a right-to-left typesetting version of `\LaTeX` is obligatory.

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

`<S>`q.t1`</S>` yields ܩ ܩ but `/<S>`q.t.t1`</S>` yields ܩ ܩ.

---

7 Using `<SERTO>` or `<TRANS>` cannot work correctly with the `-o` option of the preprocessor `serto.py`. Use `<CHALDEAN>` ... `<CHALDEAN>` and `<ESTRANGELO>` ... `<ESTRANGELO>` for the Chaldean or Estrangelo alphabets.
In cases where you need a `quššāyā` without wanting to double the consonant, a `*` can be used after the letter to typeset a dot above a letter: 

`<S>h*=anon</S>` produces `ḥānon` and `<S>s1=amk+on</S>` results in `šlāmhon`.

To avoid a `quššāyā` (when you need to adjacent identical consonants, either use a vowel on the first, use the stretching symbol:

`<S>maml'e</S>` yields `maml-`e
`<S>m^ml'</S>` yields `mm-l`-`

To get the soft sign `rūkkāh̲ā` a must follow the letter: `<S>'ab+d=a'</S>` yields `ṭāb-d`

An `ālaf` is automatically prefixed before an initial vowel:

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

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

`<S>'sapiyRe'</S>` yields `šp-i-yè`

Silent consonants have a bar `mb̲aṭlānā` (linea occultans) under the line which is produced by `=` just before the consonant (attention `=a`, however, yields `l`:

`<S>'an=tt</S>` yields `yī`.

In order to have the `linea occultans` on top of the letter, use `==`:

`<S>h==wiyt</S>` yields `dūšū`

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

`<S>napiyqt=a'</S>` becomes `n-p-i-y-q-t-a` but `<S>na--piyq--t=a'</S>` is printed as `n-p-i-y-q-t-a`.
This works also for the Chaldean letters \(<C>n:ap:iyqt:=a'</C> becomes بَضَبَعُن but \(<C>n:a--p:iyq--t:=a'</C> is printed as بَضَبَعُن.

If you do not use the preprocessor, you can activate Serto 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. Please see the encoding table in section 2.6 for the correct vowel numbers.

In order to get bold letters, you can use \sertob with or without preprocessor (see section 3 for more information on the preprocessor.

4 The format of the *.font files

These files are necessary to tell the preprocessor where (in the font) a certain letter is found, and whether it has different forms. The format is straightforward, with, however, a few idiosyncrasies. In general there are two sections, the first (starting with a line \#FONT) indicates which letter has which form in which position and a second (\#TRANS) to define the transcription.

The first part consists of lines like the following
\begin{verbatim}
b beth 66+124 66 66 66+124 1
\end{verbatim}
which reads, coding \(b\) is for the letter \(beth\), its isolated form is character 66 followed by character 124, its initial and medial form is character 66, its final form is character 66 followed by character 124 and the next letter (if any) must take its medial form.

However, if one of the four last columns has a value of \(-1\), no form is provided in the font. Values from 0 to 15 are reserved for accents/vowels above the line, values from 16 to 31 are reserved for vowels under the line.

The lines
\begin{verbatim}
- blank 32 32 32 32 0
Q shadda 6 6 6 6 2
-- stretch 45 45 45 45 1
\end{verbatim}

must not be deleted.

The coding for digits (starting with 0 up to 9 in the first column cannot have the + in the position definition.

The transcription definition defines for each coding symbol (defined in the
font-section) a valid TeX-string to be used if transcription is needed, e.g.: .t \d{t}

i.e. the coding .t will be representend by \ in transcribed portions.

5 Two examples

Some of the following can be found in example.ptex which comes with this package.

The following input was used to generate the output below:

```latex
\documentclass[12pt,a4paper]{article}
\usepackage{serto}
\begin{document}
\<S>men qadiy\se' ho' tetqada\se's</S>
\<ST>kmo' dat+basb@suwn pagdo'</ST>
\end{document}
```

Ů´L ´ŤŻ ˆŻ A ¨M B ˆŹl˜L ´Ť z ˆx
A¨LI ´Ő yP ˝ĚEĚ ´FŻ´K B ¨wp kmā datb̲s̲u̲n̲ pagdā

A long example in Serto:

The preceding text was set typeset with the following input:

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

12
Replacing <SERTO> by <CHALDEAN> and \sertob by \assyrb results in this:

Using <ESTRANGELO> (and deleting remaining \sertob and \assyrb) results in:

The file was preprocessed using serto3.py and then typeset with XƎL A TEX.
6 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, maybe in the ArabTEX package)
- proper treatment of matrēs lectionis (long vowels)
- proper treatment of the silent bar mbâtlânā
- proper treatment of beṯadkep̄at with hard sign quššāyā and soft sign rūkkāhā
- automatic transcription mode
- interpunction
- proper dealing with ligatures
- estrangelā
- numbers the inputenc.sty package.
- support for existing Syriac fonts

7 Installation

The easiest way to install the fonts and the preprocessor is by installing the debian package (this includes only the pfb, tfm and afm files for the fonts), the needed styles and the preprocessor, but not the METAFONT sources:

    sudo dpkg -i serto-1.3.deb

If you are not on a Debian or Ubuntu plateform, you need to install manually from the .tgz file:

7.1 Using Metafont sources

Put the *.mf files into a subdirectory serto of your metafont branch in your texmf-directory. For example using the texlive distribution under Linux, you should put them into /usr/local/share/texmf/fonts/source/serto/. Do
not forget to call **texhash** in order to make the tex software find the newly installed fonts.

### 7.2 Using vector fonts

Using vector fonts depends a little from the TeX-installation used, the following is tested for Ubuntu 12.04 and 10.04, it will probably work on Debian platforms as well, or other platforms using the teTeX installation.

- copy `syriac.map` to `/usr/local/share/texmf/fonts/map/dvips/config/`
- copy `*.afm` to `/usr/local/share/texmf/fonts/afm/syriac/serto/`
- copy `*.pfb` to `/usr/local/share/texmf/fonts/type1/syriac/`
- add Map `syriac.map` to `/etc/texmf/updmap.d/10local.cfg`
- run `sudo update-updmap`
- run `sudo updmap-sys`

### 7.3 Other Files

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 `serto.font` somewhere where it can be found (e.g `/usr/local/bin`). They must reside in the same directory unless you specify in the environment variable `SERTOFONTDIR` the directory containing `serto.font` and `assyr.font`. Possibly you have to adjust the first line of the preprocessor `#!/usr/bin/python` if your python interpreter is somewhere else.

In order to typeset using Estrangelo, an external (Unicode encoded) font has to be downloaded. The examples in this document use Estrangelo Edessa designed by Paul Nelson and George Kiraz and copyrighted by the Syriac Computing Institute. For copyright reasons this font is not included in this package. Once you have downloaded the `.ttf` file, install it with your system fonts (in both Linux (Ubuntu) and Windows, clicking on the file name usually opens a font viewer application which allows the (local) installation. Note, that Estrangelo only works with XeTeX and its `fontspec`-package.

If you use a different font, adapt `estrangelo.sty` accordingly.
8 License
This Material is subject to the LaTeX Project Public License 1.3 (http://ctan.org/license/lppl1.3).

9 Changelog

• Version 1.3
  – documentation updated
  – error in loading *.font-files corrected
  – serto.py (for PYTHON3), no more support for PYTHON2 any more.

• Version 1.1
  – encoding script for PYTHON3: serto3.py
  – experimental Estrangelo support using a .ttf (needs to be downloaded)

• Version 1.0
  – adding a character for the linea occultans above the letter
  – SERTOFONDDIR environment variable to specify the directory of *.font files
  – some UTF8 support

• Version 0.7
  – Chaldean vowels
  – Integration of the Chaldean font provided by Tony Khoshaba
  – Major adjustements to the serto.py preprocessor

• Version 0.2, 0.3 and 0.4
  – can’t remember, didn’t keep track of changelog those days...

• Version 0.1
  – Initial version
References