Serto – a font for Syriac (Aramaic)

Johannes Heinecke

Version 1.0, 30th March 2013

Contents

1 Introduction 2

2 The alphabets 2
   2.1 Consonants .............................................. 2
   2.2 Vowels ................................................. 4
   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.py 8
   3.1 Using the preprocessor ................................... 8

4 The format of the *.font files 11

5 Two examples 11

6 Things still missing 13

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

8 License 15

9 Changelog 15
1 Introduction

Serto 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, Serto is used for Syriac which is the variant of Aramaic spoken since the second century AD.\footnote{See Ungnad 1932, Brockelmann 1960, Costaz 1986 or Nöldeke 1986f (English translation Nöldeke 2001) for further information on Syriac.}

Syriac used other alphabets as well, notably Estrangelo, which is not (yet) contained in this package. Since Serto 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 Serto, 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 (these are the only ones provided by this package for the time being), the eastern dialects uses dots to indicate the vowels (Chaldean vowels).

This package enables you to typeset words or paragraphs in Serto using a pre-processor which chooses the correct letter form depending on context. In order to typeset paragraphs the use of a recent version of \texttt{pdflatex} is needed, which can handle the right-to-left typesetting. For older version of \LaTeX, the preprocessor must be used with the option \texttt{-o} (see section 3).

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

2 The alphabets

Every letter in Serto (and some letters in the Chaldean alphabet) 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 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>serfo forms</th>
<th>isolated</th>
<th>final</th>
<th>medial</th>
<th>initial</th>
<th>Chaldean</th>
<th>name</th>
<th>translit.</th>
<th>coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ālaṭ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bēṭ</td>
<td></td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gāmal</td>
<td></td>
<td>g</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dālaṭ</td>
<td></td>
<td>d</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>hē</td>
<td></td>
<td>h</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>waw</td>
<td></td>
<td>w</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>zayn</td>
<td></td>
<td>z</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>hēṭ</td>
<td></td>
<td>.h</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>tēṭ</td>
<td></td>
<td>.t</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>yāḍ</td>
<td></td>
<td>y</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>kāp</td>
<td></td>
<td>k</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ḫāʾ</td>
<td></td>
<td>ḫ</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lāmad</td>
<td></td>
<td>l</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mūn</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>nūn</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>semkaṭ</td>
<td></td>
<td>s</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.

\textbf{ATTENTION:} I had to change the coding of \textit{zqāpā} (\_\_) from o to =a in order to accommodate the new letter Ā (coding o). In texts encoded up to version 0.4 o has to be replaced by =a. Sorry for any inconveniences.

\footnote{\textit{beγadkeφat} are not yet always processed. In general, the doubling of the consonant creates a \textit{quššāyā} in the syriac text and does not change the transliteration. On the other hand, a consonant followed by + will receive a \textit{rakkaḥa} and in the transliteration \textit{bgdkft} will appear as \textit{bgdlft}.}

\footnote{Modern Aramaic dialects using the Chaldean alphabet have diacritic symbols (dots and tildes) which can be typeset directly.}
The Chaldean name transliteration coding

<table>
<thead>
<tr>
<th>Greek</th>
<th>Chaldean</th>
<th>name</th>
<th>transliteration</th>
<th>coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>( )</td>
<td>( )</td>
<td>( pt̄hā )</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>( )</td>
<td>( )</td>
<td>( rh̄šā )</td>
<td>e</td>
<td>e</td>
</tr>
<tr>
<td>( )</td>
<td>( )</td>
<td>( h̄šā )</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>( )</td>
<td>( )</td>
<td>( zqāpā )</td>
<td>ā</td>
<td>=a</td>
</tr>
<tr>
<td>( )</td>
<td>( )</td>
<td>( ṧšā )</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>( )</td>
<td>( )</td>
<td>sÿāmē</td>
<td>o</td>
<td></td>
</tr>
</tbody>
</table>

Note: The \( h̄šā \) and \( ṧšā \) of the Eastern or Chaldean vowels do in general occur together with a mater lectionis: \( {\text{LH}}\) (or \( {\text{LH}}\)).

The \( sÿāmē \) is processed as a vowel sign, even if it is not so from a linguistic point of view. Its coding \(<\text{S}>P</\text{S}>\) is chose because of its plural signification. If you do not want it over a letter, put it over a word stretch: The Aramaic at the beginning of this paragraph has been typset in the following: \(<\text{S}>sy=a--Pme’</\text{S}>\).

To avoid that the simple preprocessor does not mess about with vowels in ligatures (notably \( \text{Lāma}d-\text{ālaf} \) and \( \text{Ālaf-}l\text{āma}d \) the vowel must be set after both consonants of the ligature:

\(<\text{S}>l=a’</\text{S}>\ yields (incorrect) \( ā \) \( ā \) but \(<\text{S}>l’=a</\text{S}>\ yields the vocalized ligature \( {\text{L}}lā \)

\(<\text{S}>’\text{Al}=\text{Ah}=a’</\text{S}>\ yields (incorrect) \( \text{Lala}hā \) but \(<\text{S}>’\text{la}=\text{Ah}=a’</\text{S}>\ yields the vocalized ligature \( {\text{L}}’\text{ala}hā \)

The Chaldean letters do not have this ligature. Instead, a \( \text{taωalā} \) ligature is provided: \( \text{LH} \) \( \text{siprāyūta} \)
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:

\[
\text{<S>eliy eliy lm=an=a’ s=ab=akt=aniy</S>}
\]

\[
\text{<S>el:iy :el:iy lm:=an:=a’ s:=ab:=akt:=an:iy</S>}
\]

\[
\text{<C>eliy eliy lm=an=a’ s=ab=akt=aniy</C>}
\]

\[
\text{<C>el:iy :el:iy lm:=an:=a’ s:=ab:=akt:=an:iy</C>}
\]

\[
\text{<C>:el:iy :el:iy lm:=an:=a’ s:=ab:=akt:=an:iy</C>}
\]

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 \(\tilde{\text{I}}\jmath\kappa\tilde{\text{V}}\) transliterates as \(\text{h. }\tilde{\text{e}}\text{t}\tilde{\text{b}}\) and not \(\text{h. }\text{eyt}\) and \(\text{uB}\tilde{\text{k}}\tilde{\text{k}}\tilde{\text{q}}\text{ay}\text{em}\):

<table>
<thead>
<tr>
<th>Greek</th>
<th>Chaldean</th>
<th>transliteration</th>
<th>coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>(\text{\alpha})</td>
<td>(\tilde{\text{a}})</td>
<td>=a’</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>(\text{\epsilon})</td>
<td>(\tilde{\text{e}})</td>
<td>ey</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>(\text{\epsilon})</td>
<td>(\tilde{\text{e}})</td>
<td>e’</td>
</tr>
<tr>
<td>(\iota)</td>
<td>(\text{\iota})</td>
<td>(\tilde{\text{i}})</td>
<td>iy</td>
</tr>
<tr>
<td>(\omicron)</td>
<td>(\text{\omicron})</td>
<td>(\tilde{\text{u}})</td>
<td>uw</td>
</tr>
</tbody>
</table>
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>
</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 Serto and the Chaldean variant.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;0n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;1n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;2n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;3n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;4n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;5n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;6n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;7n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;8n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;9n&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>&quot;An&quot;</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>
3 The preprocessor serto.py

Typesetting of texts is still not possible with standard \LaTeX, since the right-to-left typesetting (as for instance as in Arab\TeX) has not yet been implemented. For the time being I propose a preprocessor (written in PYTHON\footnote{Every version from 2.4 onwards up to version 2.7 should do; Pytho... operating systems (http://www.python.org).}) and pdflatex.

I’m well aware that serto.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 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 \LaTeXed as usually. Please make sure to have the \texttt{\usepackage{serto}} included in your preamble.

The option \texttt{-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--XeTextension). The \texttt{-o} -option tells the preprocessor to inverse the letters on its own. In order to
typeset whole paragraphs \texttt{pdflatex} is the better solution. Usually it comes with every modern \TeX-distribution. At least on Ubuntu 12.04 (texlive package), the standardly installed \texttt{pdflatex} behaves correctly.

The preprocessor recognizes two types of commands. Within a single line you can put Syriac words between \texttt{	extless S\textgreater} and \texttt{\textless /S\textgreater}: For example \texttt{\textless S\textgreater}ser.t=a’\texttt{\textless /S\textgreater} becomes ـتً. ـتً

\texttt{\textless ST\textgreater} and \texttt{\textless /ST\textgreater} generate the enclosed part in Serto and generates a transliteration as well (\texttt{\textless ST\textgreater}mdiyt=a’\texttt{\textless /ST\textgreater} becomes ـتً ـتً “city”), whereas \texttt{\textless T\textgreater} and \texttt{\textless /T\textgreater} can be used for parts only need in transliterated form (\texttt{\textless T\textgreater}ser.t=a’\texttt{\textless /T\textgreater} becomes ـتً). Since in transliteration a “neutral vowel” is needed, which does not appear in Serṭǫ, the code \texttt{@} can be used: \texttt{\textless ST\textgreater}s@m=a’\texttt{\textless /ST\textgreater} produces ـتً ـتً ـتً ـتً.

For multiple lines, start a block using \texttt{\textless SERTO\textgreater} in a line on its own. This block can be closed by \texttt{\textless /SERTO\textgreater}. If you need transliterated Syriac, use \texttt{\textless TRANS\textgreater} and \texttt{\textless /TRANS\textgreater}. The commands \texttt{\textless SERTO\textgreater}/\texttt{\textless SERTO\textgreater} and \texttt{\textless TRANS\textgreater}/\texttt{\textless TRANS\textgreater} do not work properly with the \texttt{-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 ـتً ـتً quṭṣṭṭī on top of a consonant if the consonant is doubled in the input:

\texttt{\textless S\textgreater}q.t.l\texttt{\textless /S\textgreater} yields ـتً ـتً but \texttt{\textless S\textgreater}q.t.l\texttt{\textless /S\textgreater} yields ـتً ـتً

In cases where you need a ـتً quṭṣṭṭī without wanting to double the consonant, a \texttt{*} can be used after the letter to typeset a dot above a letter: \texttt{\textless S\textgreater}h*=anon\texttt{\textless /S\textgreater} produces ـتً ـتً hānon and \texttt{\textless S\textgreater}s@l=amk+on\texttt{\textless /S\textgreater} results in ـتً ـتً slāmḥon.

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

\texttt{\textless S\textgreater}maml’e\texttt{\textless /S\textgreater} yields ـتً ـتً mambē
\texttt{\textless S\textgreater}mˆml’e\texttt{\textless /S\textgreater} yields ـتً ـتً mambē
\texttt{\textless C\textgreater}maml’e\texttt{\textless /C\textgreater} yields ـتً ـتً mambē
\texttt{\textless C\textgreater}mˆml’e\texttt{\textless /C\textgreater} yields ـتً ـتً mambē

\footnote{Using \texttt{SERTO} or \texttt{TRANS} implies the using of the \texttt{-e} option of the preprocessor \texttt{serto.py}. If you do not use the \texttt{-e} option the resulting \LaTeX-file is likely to be syntactically incorrect.}
To get the soft sign ۝ rākkāḥā a must follow the letter: <S>’ab+d=a’</S> yields ܐܒ(Activity)

An ālaf is automatically prefixed before an initial vowel:

<S>etqa.tel</S> and <S>’etqa.tel</S> both yield ܐ&

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

<S>ˆsapiyRe’</S> yields ܐ&$

Silent consonants have a bar ܒ& in the output which is produced by = just before the consonant (attention =a, however, yields ܐ:

<S>’an=tt</S> yields ܐܒ(Activity)

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

<S>h==wiyt</S> yields ܕܥ(Activity)

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

<S>napiyqt=a’</S> becomes ܐܥ(Activity) but <S>na--piyq--t=a’</S> is printed as ܐܥ(Activity)

This works also for the Chaldean letters <C>n:ap:iyqt:=a’</C> becomes ܒܥ(Activity) but <C>n:a--p:iyq--t:=a’</C> is printed as ܒܥ(Activity)

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

\[ \text{b beth 66+124 66 66 66+124 1} \]

which reads, coding \text{b} is for the letter \text{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 \text{-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

\[ \text{~ blank 32 32 32 32 0} \]
\[ \text{Q shadda 6 6 6 6 2} \]
\[ \text{-- stretch 45 45 45 45 1} \]

must not be deleted.

The coding for digits (starting with 0 up to 9 in the first column cannot have the \text{+} 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 \text{.t} will be represented by \text{f} 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:
<S>men qadiyˆse' ho' tetqadaˆs</S>

<ST>kmo’ dat+basb@suwn pagdo’</ST>

"من قدیم هو تطقداس"

"كمًا دابسبًسًن پدو"
Replacing <SERTO> by <CHALDEAN> results in this:

The file was preprocessed using serto.py.

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 ٧صِ٧لَ٧نَ٧ا mbaṭlānā
- proper treatment of beğadkepāṭ with hard sign ٧فِ٧سُ٧ا quśšāyā and soft sign ٧فِ٧سُ٧ا rākkāhā
- automatic transcription mode
- interpunction
- proper dealing with ligatures
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 \texttt{pfb}, \texttt{tfm} and \texttt{afm} files for the fonts), the needed styles and the preprocessor, but not the METAFONT sources:

```bash
sudo dpkg -i serto-1.0.deb
```

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

7.1 Using Metafont sources

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

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

8 License

This Material is subject to the LaTeX Project Public License 1.3 (http://ctan.org/license/lppl1.3).

9 Changelog

• Version 1.0
  – adding a character for the linea occultans above the letter
  – SERTOFONTDIR 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 adjustments 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


