# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>License and disclaimer</td>
<td>2</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.1 arabluatex is for LuaLaTeX</td>
<td>4</td>
</tr>
<tr>
<td>2 The basics of arabluatex</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Activating arabluatex</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Options</td>
<td>5</td>
</tr>
<tr>
<td>2.2.1 Classic contrasted with modern typesetting of Arabic</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Typing Arabic</td>
<td>7</td>
</tr>
<tr>
<td>2.3.1 Local options</td>
<td>9</td>
</tr>
<tr>
<td>3 Standard ArabLaTeX input</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Consonants</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Additional characters</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Vowels</td>
<td>11</td>
</tr>
<tr>
<td>3.3.1 Long vowels</td>
<td>11</td>
</tr>
<tr>
<td>3.3.2 Short vowels</td>
<td>12</td>
</tr>
<tr>
<td>4 arabluatex in action</td>
<td>13</td>
</tr>
<tr>
<td>4.1 The vowels and diphthongs</td>
<td>13</td>
</tr>
<tr>
<td>Short vowels</td>
<td>13</td>
</tr>
<tr>
<td>4.2 Other orthographic signs</td>
<td>15</td>
</tr>
<tr>
<td>Ta’ marbūtah</td>
<td>15</td>
</tr>
<tr>
<td>Hamzah</td>
<td>15</td>
</tr>
<tr>
<td>Maddah</td>
<td>17</td>
</tr>
<tr>
<td>Šaddah</td>
<td>17</td>
</tr>
<tr>
<td>The definite article and the ‘alif</td>
<td>18</td>
</tr>
<tr>
<td>4.3 Special orthographies</td>
<td>21</td>
</tr>
<tr>
<td>The name of God</td>
<td>21</td>
</tr>
<tr>
<td>The conjunctive</td>
<td>21</td>
</tr>
<tr>
<td>4.4 Quoting</td>
<td>22</td>
</tr>
<tr>
<td>novoc</td>
<td>22</td>
</tr>
<tr>
<td>voc</td>
<td>23</td>
</tr>
<tr>
<td>fullvoc</td>
<td>23</td>
</tr>
<tr>
<td>4.4.1 Quoting the hamzah</td>
<td>23</td>
</tr>
<tr>
<td>4.5 The “pipe” character</td>
<td>24</td>
</tr>
<tr>
<td>4.6 Putting back on broken contextual analysis rules</td>
<td>24</td>
</tr>
<tr>
<td>4.7 Stretching characters: the lāṭwil</td>
<td>26</td>
</tr>
<tr>
<td>4.8 Digits</td>
<td>26</td>
</tr>
<tr>
<td>4.8.1 Numerical figures</td>
<td>26</td>
</tr>
<tr>
<td>4.8.2 The abjad</td>
<td>26</td>
</tr>
<tr>
<td>4.9 Additional characters</td>
<td>27</td>
</tr>
<tr>
<td>4.10 Arabic emphasis</td>
<td>28</td>
</tr>
<tr>
<td>5 Arabic poetry</td>
<td>28</td>
</tr>
<tr>
<td>Scaling and distortion of characters</td>
<td>31</td>
</tr>
<tr>
<td>Footnotes</td>
<td>31</td>
</tr>
<tr>
<td>Line numbering</td>
<td>31</td>
</tr>
<tr>
<td>5.1 Example</td>
<td>31</td>
</tr>
<tr>
<td>6 Special applications</td>
<td>32</td>
</tr>
<tr>
<td>Linguistics</td>
<td>32</td>
</tr>
<tr>
<td>Brackets</td>
<td>32</td>
</tr>
<tr>
<td>Additional Arabic marks</td>
<td>33</td>
</tr>
<tr>
<td>6.1 The Qur’ān</td>
<td>35</td>
</tr>
<tr>
<td>Caveat</td>
<td>35</td>
</tr>
<tr>
<td>7 Color</td>
<td>35</td>
</tr>
<tr>
<td>7.1 Tricks of the trade</td>
<td>37</td>
</tr>
<tr>
<td>Diacritics</td>
<td>37</td>
</tr>
<tr>
<td>Ta’ marbūtah and maddah</td>
<td>38</td>
</tr>
<tr>
<td>Šaddah</td>
<td>38</td>
</tr>
<tr>
<td>The definite article and the euphonic tašdīd</td>
<td>39</td>
</tr>
<tr>
<td>Hamzah</td>
<td>40</td>
</tr>
</tbody>
</table>
Abstract

This package provides for LuaLaTeX an ArabTeX-like interface to generate Arabic writing from an ASCII transliteration. It is particularly well-suited for complex documents such as technical documents or critical editions where a lot of left-to-right commands intertwine with Arabic writing. arabluatex is able to process any ArabTeX input notation. Its output can be set in the same modes of vocalization as ArabTeX, or in different roman transliterations. It further allows many typographical refinements. It will eventually interact with some other packages yet to come to produce from .tex source files, in addition to printed books, TEI xml compliant critical editions and/or lexicons that can be searched, analyzed and correlated in various ways.

License and disclaimer

License applicable to the software

arabluatex — Processing ArabTeX notation under LuaLaTeX.
Copyright © 2016–2019 Robert Alessi
Please send error reports and suggestions for improvements to Robert Alessi:

- email: mailto:alessi@roberalessi.net
- website: http://www.robertalessi.net/arabluatex
- development: http://git.robertalessi.net/arabluatex
- comments, feature requests, bug reports: https://gitlab.com/ralessi/arabluatex/issues

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

This release of arabluatex consists of the following source files:
- arabluatex.ins
- arabluatex.dtx
- arabluatex.lua
- arabluatex_voc.lua
- arabluatex_fullvoc.lua
- arabluatex_novoc.lua
- arabluatex_trans.lua
- arabluatex.el

License applicable to this document

Copyright © 2016–2019 Robert Alessi

This document is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

1 Introduction

In comparison to Prof. Lagally’s outstanding ArabTEX,\(^1\) ArabLuaTEX is at present nothing more than a modest piece of software. Hopefully—if I may say so—it will eventually provide all of its valuable qualities to the LuaLaTeX users.

arabtex dates back to 1992. As far as I know, it was then the first and only way to typeset Arabic texts with \TeX{} and \LaTeX{}. To achieve that, arabtex provided—and still does—an Arabic font in Naskh style and a macro package that defined its own input notation which was, as the author stated, “both machine, and human, readable, and suited for electronic transmission and e-mail communication”.\(^2\) Even if the same can

\(^1\)See http://ctan.org/pkg/arabtex

\(^2\)Lagally (2004, p. 2).
be said about Unicode, ArabTEX ASCII input notation still surpasses Unicode input, in my opinion, when it comes to typesetting complex documents, such as scientific documents or critical editions where footnotes and other kind of annotations can be particular abundant. It must also be said that most text editors have trouble in displaying Arabic script connected with preceding or following \LaTeX\ commands: it often happens that commands seem misplaced, not to mention punctuation marks, or opening or closing braces, brackets or parentheses that are unexpectedly displayed in the wrong direction. Of course, some text editors provide ways to get around such difficulties by inserting invisible Unicode characters, such as LEFT-TO-RIGHT or RIGHT-TO-LEFT MARKS (U+200E, U+200F), RTL/LTR “embed” characters (U+202B, U+202A) and RLO/LRO “bidi-override” characters (U+202E, U+202D). Nonetheless, it remains that inserting all the time these invisible characters in complex documents rapidly becomes confusing and cumbersome.

The great advantage of ArabTEX notation is that it is immune from all these difficulties, let alone its being clear and straightforward. One also must remember that computers are designed to process code. ArabTEX notation is a way of encoding Arabic language, just as \TeX\ “mathematics mode” is a way of processing code to display mathematics. As such, not only does it allow greater control over typographical features, but it also can be processed in several different ways: so without going into details, depending on one’s wishes, ArabTEX input can be full vocalized Arabic (scriptio plena), vocalized Arabic or non-vocalized Arabic (scriptio defectiva); it further can be transliterated into whichever romanization standard the user may choose.

But there may be more to be said on that point, as encoding Arabic also naturally encourages the coder to vocalize the texts—without compelling him to do so, of course. Accurate coding may even have other virtuous effects. For instance, hyphens may be used for tying particles or prefixes to words, or to mark inflectional endings, and so forth. In other words, accurate coding produces accurate texts that can stand to close grammatical scrutiny and to complex textual searches as well.

Having that in mind, I started arabluatex. With the help of Lua, it will eventually interact with some other packages yet to come to produce from .tex source files, in addition to printed books, TEI XML compliant critical editions and/or lexicons that can be searched, analyzed and correlated in various ways.

1.1 **arabluatex is for LuaLaTEX**

It goes without saying that arabluatex requires LuaLaTeX. \TeX\ and \LaTeX\ have arabtex, and XeLaTeX has arabxetex. Both of them are much more advanced than arabluatex, as they can process a number of different languages, whereas arabluatex can process only Arabic for the time being. More languages will be included in future releases of arabluatex.

---

3Gáspár Sina‘i’s Yudit probably has the best Unicode support. See [http://www.yudit.org](http://www.yudit.org).

4To date, both packages support Arabic, Maghribi, Urdu, Pashto, Sindhi, Kashmiri, Uighuric and Old Malay; in addition to these, arabtex also has a Hebrew mode, including Judeo-Arabic and Yiddish.
In comparison to \texttt{arabxetex}, \texttt{arabluatex} works in a very different way. The former relies on the \texttt{TECkit} engine which converts \texttt{ArabTPX} input on the fly into Unicode Arabic script, whereas the latter passes \texttt{ArabTPX} input on to a set of Lua functions. At first, \LaTeX{} commands are taken care of in different ways: some, as \texttt{\texttt{\textbf{}}}, and the like are expected to have Arabic text as arguments, while others, as \texttt{\texttt{\texttt{LR}}}, for “left-to-right text”, are not. Then, once what is Arabic is carefully separated form what is not, it is processed by other Lua functions which rely on different sets of correspondence tables to do the actual conversion in accordance with one’s wishes. Finally, Lua returns to \TeX{} the converted strings— which may in turn contain some other \texttt{ArabTPX} input yet to be processed—for further processing.

\section{The basics of \texttt{arabluatex}}

\subsection{Activating \texttt{arabluatex}}

\texttt{arabluatex} is loaded the usual way:
\begin{verbatim}
\usepackage{arabluatex}
\end{verbatim}

The only requirement of \texttt{arabluatex} is Lua\LaTeX{}; it will complain if the document is compiled with another engine. That aside, \texttt{arabluatex} does not load packages such as \texttt{polyglossia}. Although it can work with \texttt{polyglossia}, it does not require it.

\textbf{Font setup} \quad Any Arabic font can be defined to be used with \texttt{arabluatex}. For example, assuming that \texttt{fontspec} is loaded, this line may be inserted in the preamble, just above the line that loads \texttt{arabluatex}:
\begin{verbatim}
\newfontfamily\arabicfont\{⟨fontname⟩\}[Script=Arabic]
\end{verbatim}

where \texttt{⟨fontname⟩} is the standard name of the Arabic font to be used.

By default, if no Arabic font is selected, \texttt{arabluatex} will issue a warning message and attempt to load the Amiri font\footnote{\cite{hosny2017}} like so:—
\begin{verbatim}
\newfontfamily\arabicfont\{Amiri\}[Script=Arabic]
\end{verbatim}

\textbf{Rem.} \quad By default Amiri places the \textit{kasrah} in combination with the \textit{laṣīl} below the consonant, like so: \texttt{\texttt{١}}. That is correct, as at least in the oldest manuscripts \texttt{١} may stand for \texttt{١} as well as \texttt{١}. See Wright (1996, i. 14 C–D). The placement of the \textit{kasrah} above the consonant may be obtained by selecting the \texttt{ss05} feature of the Amiri font, like so:—\footnote{See the documentation of \texttt{amiri}, Hosny (2017, p. 6).}
\begin{verbatim}
\newfontfamily\arabicfont\{Amiri\}[Script=Arabic,RawFeature={+ss05}]
\end{verbatim}

Other Arabic fonts may behave differently.

\subsection{Options}

\texttt{arabluatex} may be loaded with five global options, the first four of which are mutually exclusive and may be over ridden at any point of the document (see below section 2.3.1 on page 9):
In this mode, which is the one selected by default, every short vowel written generates its corresponding diacritical mark: ẓammah (ُـ), fatḥah (َـ) and kasrah (ِـ). If a vowel is followed by ن، viz. ⟨uN, aN, iN⟩, then the corresponding tanwīn (ٌـ،ٌّـ،ٌْـ) is generated. Finally, ⟨u, a, i⟩ at the commencement of a word indicate a “connective ‘alif” (‘alifu 'l-wasli), but voc mode does not show the waṣlah above the ‘alif; instead, the accompanying vowel may be expressed at the beginning of a sentence (ُ ٍّ ٍّ ِّ ِّ).

In addition to what the voc mode does, fullvoc expresses the sukūn and the waṣlah.

None of the diacritics is showed in novoc mode, unless otherwise specified (see “quoting” technique below section 4.4 on page 22).

This mode transliterates the ArabTgX input into one of the accepted standards. At present, three standards are supported (see below section 8 on page 40 for more details):

- **dmg** Deutscher Morgenländische Gesellschaft, which is selected by default;
- **loc** Library of Congress;
- **arabica** Arabica.

More standards will be included in future releases of arabluatex.

This option acts as a named argument and does not need a value as it defaults to true if it is used. It enables arabluatex to produce a duplicate of the original .tex source file in which all ASCII strings are replaced with Unicode equivalents. See below section 12 on page 57 for more information.

### 2.2.1 Classic contrasted with modern typesetting of Arabic

By default, arabluatex typesets Arabic in a classic, traditional style the most prominent features of which are the following:

- ‘Classic’ maddah: when ‘alif and hamzah accompanied by a simple vowel or tanwīn is preceded by an ‘alif of prolongation (ِـ)، then a mere hamzah is written on the line, and a maddah is placed over the ‘alif, like so:—
  
  samā’a ℰNa، jā’ā bā yatasā’alūna

  (see on page 17 for further details).

- The euphonic tašdīd is generated (see on page 17).

- In fullvoc mode, the sukūn is expressed.

- In such words as ثُمَّما، ثُمَّما، ثُمَّما، Thūmā, and the like, the hamzah alone is not written over the letter yā’ with no diacritical points below as in ُّّْ، ُّّْ، ُّّْ، but over a horizontal stroke placed in the continuation of the preceding letter.

Note that in old mss. such forms as ُّّّ، ُّّّ، ُّّّ، are also found; see Wright (1896, i. 24 D).
Such refinements as ‘classic’ maddah may be discarded by the \SetArbEasy command, either globally in the preamble or locally at any point of the document. The difference between \SetArbEasy and its ‘starred’ version \SetArbEasy* is that the former keeps the sukūn that is generated by the fullvoc mode, while the latter further takes it away. Default ‘classic’ rules may be set back at any point of the document with the \SetArbDflt command. Assimilation rules laid on item (b) on page 18 may also be applied by the ‘starred’ version of this command \SetArbDflt* either in the preamble or at any point of the document.\footnote{For an example, see section 5.1 on page 31.} Examples follow:—

(a) \SetArbDflt:
   i. voc وَمَاتَ اسْتَسْقَآءَ قَيْلَ أَنْ يَمِّيْكَهُ فِي نَجَومِ السَّمَاءِ
   ii. fullvoc وَمَاتَ اسْتَسْقَآءَ قَيْلَ أَنْ يَمِّيْكَهُ فِي نَجَومِ السَّمَاءِ
   iii. trans wa-māta 'stisqā‘uun qabla 'an yutimma kilāba-hu fī nuğūm' ‘ṣ-samā‘

(b) \SetArbDflt*:
   i. voc وَمَاتَ اسْتَسْقَآءَ قَيْلَ أَنْ يَمِّيْكَهُ فِي نَجَومِ السَّمَاءِ
   ii. fullvoc وَمَاتَ اسْتَسْقَآءَ قَيْلَ أَنْ يَمِّيْكَهُ فِي نَجَومِ السَّمَاءِ
   iii. trans wa-māta 'stisqā‘uun qabla 'ay yutimma kilāba-hu fī nuğūm' ‘ṣ-samā‘

(c) \SetArbEasy:
   i. voc وَمَاتَ اسْتَسْقَآءَ قَيْلَ أَنْ يَمِّيْكَهُ فِي نَجَومِ السَّمَاءِ
   ii. fullvoc وَمَاتَ اسْتَسْقَآءَ قَيْلَ أَنْ يَمِّيْكَهُ فِي نَجَومِ السَّمَاءِ
   iii. trans wa-māta 'stisqā‘uun qabla 'an yutimma kilāba-hu fī nuğūm' ‘ṣ-samā‘

(d) \SetArbEasy*:
   i. voc وَمَاتَ اسْتَسْقَآءَ قَيْلَ أَنْ يَمِّيْكَهُ فِي نَجَومِ السَّمَاءِ
   ii. fullvoc وَمَاتَ اسْتَسْقَآءَ قَيْلَ أَنْ يَمِّيْكَهُ فِي نَجَومِ السَّمَاءِ
   iii. trans wa-māta 'stisqā‘uun qabla 'an yutimma kilāba-hu fī nuğūm' ‘ṣ-samā‘

Please note that this document is typeset with \SetArbDflt throughout.

2.3 Typing Arabic

Once arabluatex is loaded, a \arb{⟨Arabic text⟩} command is available for inserting
From Wright (1896, i. 1 A):— Arabic, like Hebrew and Syriac, is written and read from right to left. The letters of the alphabet (ِءآَجِهلاُفوُرُحْ،ُفوُرُحْيِّجَّهَّتلا،ُةَّيِئآَجِهلاُفوُرُحلَا، orِمَجعُملاُفوُرُحْ) are twenty-eight in number and are all consonants, though three of them are also used as vowels (see § 3).

Caveat  For some reason, left-to-right paragraphs that start with Arabic words lose their indentation. For the time being, this can be circumvented by appending the \indent command at the commencement of such paragraphs.

The same remark applies to left-to-right list environments: when items start with Arabic words, the \arb command must be prefixed with \indent. The following example comes from Wright (1896, i. 213 C):—

\begin{enumerate}[label=\Roman*, start=16]
\item \indent \arb{fawA`ilu}.
\item \indent \arb{fA`aluN}; as \arb{_hAtamuN} \textit{a signet-ring}, ...
\end{enumerate}

Running paragraphs of Arabic text should rather be placed inside an \texttt{Arabic environment}:

\begin{arab}
\begin{quote}
\'at_{A}. sadIquW 1IL_A ju.\hA ya.\tlubu min-hu .hisAra-hu
li-yarkaba-hu ff safra\textsc{tu}N qa.\slRa\textsc{ti}N fa-q\textsc{al}a la-hu:
\end{quote}
\end{arab}
2.3.1 Local options

As seen above in section 2.2 on page 5, arabluatex may be loaded with four mutually exclusive global options: voc (which is the default option), fullvoc, novoc and trans. Whatever choice has been made globally, it may be overriden at any point of the document, as the \arb command may take any of the voc, fullvoc, novoc or trans modes as optional argument, like so:

- \arb[voc]{⟨Arabic text⟩};
- \arb[fullvoc]{⟨Arabic text⟩};
- \arb[novoc]{⟨Arabic text⟩};
- \arb[trans]{⟨Arabic text⟩}.

The same optional arguments may be passed to the environment arab: one may have \begin{arab}{⟨mode⟩}...\end{arab}, where ⟨mode⟩ may be any of voc, fullvoc, novoc or trans.

3 Standard ArabTEX Input

3.1 Consonants

Table 1 gives the ArabTEX equivalents for all of the Arabic consonants.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>ArabTEX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>٣</td>
<td>ُأَأِإ</td>
<td>\texttt{a} \texttt{A} \texttt{e}  \texttt{a} or \texttt{e}</td>
</tr>
<tr>
<td>٤</td>
<td>ʾu ʾa ʾi</td>
<td>\texttt{u} \texttt{a} \texttt{i}  \texttt{u}, ʾa, ʾi</td>
</tr>
</tbody>
</table>

Table 1: Standard ArabTEX (consonants)

\[9\] See below section 8 on page 40.

\[10\] See below, Rem. a. For ʾalif as a consonant, see Wright (1896, i. 16 D). The hamzah itself is encoded <> and may be followed by either ⟨u, a⟩ or ⟨i⟩. See below section 4.2 on page 15.
<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>ArabTPE notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ب</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>ت</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>ث</td>
<td>ṯ th</td>
<td>_t</td>
</tr>
<tr>
<td>ج</td>
<td>ǧ j</td>
<td>ǧ or j</td>
</tr>
<tr>
<td>ح</td>
<td>ḥ ḥ ḥ</td>
<td>.h</td>
</tr>
<tr>
<td>خ</td>
<td>ḫ kh</td>
<td>_h or x</td>
</tr>
<tr>
<td>د</td>
<td>d d d</td>
<td>.d</td>
</tr>
<tr>
<td>ذ</td>
<td>ḍ dh</td>
<td>.d</td>
</tr>
<tr>
<td>ر</td>
<td>r r r</td>
<td>.r</td>
</tr>
<tr>
<td>ز</td>
<td>z z z</td>
<td>.z</td>
</tr>
<tr>
<td>س</td>
<td>s sh s</td>
<td>^s</td>
</tr>
<tr>
<td>ص</td>
<td>ṣ ṣ ṣ</td>
<td>.s</td>
</tr>
<tr>
<td>ض</td>
<td>ḍ ḍ ḍ</td>
<td>.d</td>
</tr>
<tr>
<td>ط</td>
<td>ṭ ṭ ṭ</td>
<td>.t</td>
</tr>
<tr>
<td>ظ</td>
<td>ẓ ẓ ẓ</td>
<td>.z</td>
</tr>
<tr>
<td>ع</td>
<td>ʿ `</td>
<td>`</td>
</tr>
<tr>
<td>غ</td>
<td>ġ gh ǧ</td>
<td>.g</td>
</tr>
<tr>
<td>ف</td>
<td>f f f</td>
<td>f</td>
</tr>
<tr>
<td>ق</td>
<td>q q q</td>
<td>q</td>
</tr>
<tr>
<td>ك</td>
<td>k k k</td>
<td>k</td>
</tr>
<tr>
<td>ل</td>
<td>l l l</td>
<td>l</td>
</tr>
<tr>
<td>م</td>
<td>m m m</td>
<td>m</td>
</tr>
<tr>
<td>ن</td>
<td>n n n</td>
<td>n</td>
</tr>
<tr>
<td>ه</td>
<td>h h h</td>
<td>h</td>
</tr>
<tr>
<td>و</td>
<td>w w w</td>
<td>w</td>
</tr>
<tr>
<td>ي</td>
<td>y y y</td>
<td>y</td>
</tr>
<tr>
<td>ء</td>
<td>ah ah a</td>
<td>a T</td>
</tr>
</tbody>
</table>

Table 1: Standard ArabTPE (consonants)

**Rem. a.** Please note that in all cases of elision, the 'alif 'l-waṣli is expressed only by the vowel that accompanies the omitted hamzah: ⟨u, a, i⟩ as in wa-`nhazama َمَزَهْنٱَو wa-`nhazama. For more details on the definite article and the 'alif 'l-waṣli see section 4.2 on page 18.

That said, ⟨⟩ as a consonant is actually the spiritus lenis of the Greeks and is distinguished by the hamzah ⟨⟩ as it is shown in the above table. However, the bare 'alif may also be encoded as ⟨⟩ whether it be followed by a vowel or not, like so: wa- An َآنَو wa-. An َآنَو (where the dot symbolizes the absence of vowel), wa- An َآنَو wa-. An َآنَو wa-. An َآنَو.

11 For the letter ى with no diacritical points below, see Rem. b. below.
The letter َيَ، َاَهِتحَتنِمُةاَّنَثُملاُءآَيلَا may also be written without diacritical points as َيَ. When it is used as a consonant, it is encoded aY, where a recalls the fatḥah placed above the preceding letter in vocalized Arabic, like so: qaYٌء i ُءىَق qayʾٌء، َءىَش saYٌء i ُءىَش šayʾٌء، َءىَش saYaNٌء i ُءىَش šayʾٌء، َءىَش saY'aNٌء i َءىَش šayʾٌء.

The same result may be achieved by encoding this letter as .y, like so: qa.yٌء i ُءىَق qayʾٌء، َءىَش sa.yٌء i ُءىَش šayʾٌء، َءىَش sa.Y'aNٌء i َءىَش šayʾٌء.

### 3.2 Additional characters

Table 2 gives the ArabTEX equivalents for some additional Persian characters.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>ArabTEX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>پ</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>چ</td>
<td>ch</td>
<td>ḍ</td>
</tr>
<tr>
<td>ز</td>
<td>zh</td>
<td>ṣ</td>
</tr>
<tr>
<td>ژ</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>گ</td>
<td>g</td>
<td>g</td>
</tr>
<tr>
<td>ڭ</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>

Table 2: Standard ArabTEX (additional characters)

The alveolar consonants چ and ژ are processed as solar letters by arabluatex.

### 3.3 Vowels

#### 3.3.1 Long vowels

Table 3 gives the ArabTEX equivalents for the Arabic long vowels.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>ArabTEX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ا</td>
<td>ā</td>
<td>A</td>
</tr>
<tr>
<td>او</td>
<td>ā</td>
<td>U</td>
</tr>
<tr>
<td>ی</td>
<td>ī</td>
<td>I</td>
</tr>
</tbody>
</table>

Table 3: Standard ArabTEX (long vowels)

---

12 See below section 8 on page 40.
13 The characters that are listed in this table are not included in this standard. However, as arabica is based on dmg, the dmg equivalents have been used here.
14 This character is not found in Brockelmann et al. (1935, p. 2). It is taken from the DIN 31 635 (2011) standard.
15 See note 14.
16 See below section 8 on page 40.
17 For the letter َیَ with no diacritical points, see Rem. c. below.
3.3.2 Short vowels

Table 4 gives the ArabTeX equivalents for the Arabic short vowels.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>ArabTeX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>â</td>
<td>á</td>
<td>a</td>
</tr>
<tr>
<td>̄â</td>
<td>ā</td>
<td>a</td>
</tr>
<tr>
<td>̄u</td>
<td>ū</td>
<td>u</td>
</tr>
<tr>
<td>̄i</td>
<td>ī</td>
<td>i</td>
</tr>
</tbody>
</table>

Table 4: Standard ArabTeX (short vowels)

---

18 See below section 8 on page 40.
Whether Arabic texts be vocalized or not is essentially a matter of personal choice. So one may use voc mode and decide not to write vowels except at some particular places for disambiguation purposes, or use novoc mode, not write vowels—as novoc normally does not show them—except, again, where disambiguation is needed.\footnote{See below section 4.4 on page 22.}

However, it may be wise to always write the vowels, leaving to the various modes provided by arabluatex to take care of showing or not showing the vowels.

That said, there is no need to write the short vowels fatḥah, dammah or kasrāh except in the following cases:—

- at the commencement of a word, to indicate that a connective 'alif is needed, with the exception of the article (see below section 4.4 on page 22);
- when arabluatex needs to perform a contextual analysis to determine the carrier of the hamzah;
- in the various transliteration modes, as vowels are always expressed in romanized Arabic.

# arabluatex in action

## 4.1 The vowels and diphthongs

### Short vowels

As said above, they are written ⟨a, u, i⟩:

<table>
<thead>
<tr>
<th>Arabic</th>
<th>Romanization</th>
</tr>
</thead>
<tbody>
<tr>
<td>ḥalaqa (or xalaqa) خلْقَ</td>
<td>ḥalaqa, ṣamsun ʿšamsun, Karīmun ʿKarīmun.</td>
</tr>
<tr>
<td>bi-hi بِهِ</td>
<td>bi-hi, ʿaqitun ʿʾaqitun.</td>
</tr>
<tr>
<td>la-hu لاِهُ</td>
<td>la-hu, ḥuǧġatun ʿḥuǧġatun.</td>
</tr>
</tbody>
</table>

### Long vowels

They are written ⟨U, A, I⟩:

<table>
<thead>
<tr>
<th>Arabic</th>
<th>Romanization</th>
</tr>
</thead>
<tbody>
<tr>
<td>qalā  قَالَ</td>
<td>qalā, biʿa بِعَ, ṭūrun ʿṭūrun, ṭīnun ʿṭīnun, murūʿatun ʿmurūʿatun.</td>
</tr>
</tbody>
</table>

### ʿalif maqṣūrah

It is written ⟨_A⟩ or ⟨Y⟩:

<table>
<thead>
<tr>
<th>Arabic</th>
<th>Romanization</th>
</tr>
</thead>
<tbody>
<tr>
<td>al-fāṭā  آلَفَتا</td>
<td>al-fāṭā, al-maqhā ʿal-maqhā, ʾīlā ʾīlā.</td>
</tr>
</tbody>
</table>
ʾalif otiosum Said ʿalif ʾl-wiqāyat, “the guarding ʿalif”, after ʾ at the end of a word, both when preceded by dammah and by fathah is written ⟨UA⟩ or ⟨aW, aWA⟩:

na.ṣarUA اوُرَصَن naṣarū, katabUA اوُبَتَك katabū, ya.ɡzUA اوُزَي ṣūṣū, ramaW اوْمَر ramaḍ, banaw, banaWA اوُنَب.

ʾalif maḥḏūfah and defective ū, ī They are written ⟨_a, _i _u⟩:

al-l_ah-uُهّٰللَا al-lāh u,ʾil_ahuNٌهٰلِإ ʾilāh un, al-ra.hm_an-uُنٰمحَّرلَا ar-raḥmān u, l_akinنِكم lākin, h_ahunAاَنُهٰه hāhunā, Ḥunayn-u bn-u 'is.h_aq-aَقٰحسِإُنبُنُح Ḥunayn "bn" Ṣīqā, rabb_iرببī, al-ʿA.s_iِصٰعلَا al-ʿĀṣī.

Silent / Some words ending with ُة are usually written ُةوَـ or ُةوٰ instead of ُةاَـ: see Wright (1896, i. 12 A). arabluatex preserves that particular writing; the same applies to words ending in ُةيَـ for ُةاَـ. Long vowels ⟨U, I⟩ shall receive no sukūn after a ʾalif maḥḏūfah and are discarded in trans mode:

.ḥay_aUTuNٌةوٰيَح ḥayāt un, .sal_aUTuNٌةوٰلَص ṣalāt un, mi^sk_aUTuNٌةوٰكشِم miš-kāt un, tawr_aITuNٌةيٰروَت tawrāt un.

And so also: al-rib_aIT-uُةيٰبِّرلَا ar-ribāt u.

ʿAmr, and the silent To that name a silent ُة is added to distinguish it from ʿUmar: see Wright (1896, i. 12 C). In no way this affects the sound of the tanwīn, so it has to be discarded in trans mode:

`amruNUُةوٰرَمْرُع ʿAmr un, `amraNUُةوٰرَمْرُع ʿAmr an, `amriNUُةوٰرَمْرُع ʿAmr i

When the tanwīn falls away (Wright 1896, i. 249 B): `amr-uU bn-u mu.ḥammadiNُةوٰرَمْرُع ʿAmr "bn" Muḥammad un, mu.ḥammad-u bn-u `amr-iU bn-i _hAlidiNُةوٰرَمْرُع ʿAmr i bn' Muḥammad "bn" ʿAmr i bn' Ḥālid un.

And so also: al-rib_aUA اوٰرَبَیَا ar-ribā, ribaNU اوٰرَبَیَا riban.

tanwīn The marks of doubled short vowels, َـ, ًـ, ٍـ are written ⟨uN, aN, iN⟩ respectively. arabluatex deals with special cases, such as ًـ taking an َ after all consonants except ُ, and tanwīn preceding ِ as in ِهُدِي, which is written ⟨aN_A⟩ or ⟨aNY⟩:
mAluN māl
dAbaN bāb
madInaTaN madīnat
bintiN bint
maqhaN maqha
fataN fata

arabluatex is aware of special orthographies: ▼say'uN šayʾ
▼say'aN šayʾ
▼say'iN šayʾ.

In some cases, it may be useful to mark the root form of defective words so as to produce a more accurate transliteration of ending tanwīn. As seen above, tanwīn preceding ى is written ⟨aN⟩ or ⟨aNY⟩. Such forms as قاضي may likewise be written ⟨iNI⟩:

al-qA.dI al-qāḍī
qA.diyaN qāḍiy
qA.diNI qāḍi

4.2 Other orthographic signs
tā’ marbūṭah It is written ⟨T⟩:

madInaTuN madīnat
madInaTaN madīnat
madInaTiN madīnat

hamzah It is written ⟨'⟩, its carrier being determined by contextual analysis. In case one wishes to bypass this mechanism, he can use the “quoting” feature that is described below in section 4.4 on page 22.

Initial hamzah: 'asaduN ʾasad,
' u_htuN ʾuḥt,
'iqlIduN ʾiqlīd,
'anna ʾanna,
'inna ʾinna.

hamzah followed by the long vowel, is encoded ' _U: ' _UL_A ʾūlạ,
' _ULU ʾūlū,
' _ULA'ika ʾūlāʾika.

hamzah followed by the long vowel ي is encoded ' _I: ' _ImAnuN ʾīmā-

Middle hamzah: xa.ti'-Ina ḥāṭiʾina,
rU'usuN ṭu'us,
xa.tI'aTuN ṭuʿalun,
su'ila suʿila,
'as'ilatuN as'ilatun,
mas'alatuN mas'ālatun,
'as'alun yatasāʾalūna,
mU'aTuN tw'amun,
taw'amuN tw'amun,
murU'aTuN mūrālun,
ta'xIruN taʾḫīr,
ta'axxara taʾahhara,
jI'tu-ka jiṭu-ka,
qA'iluN qāʾilūn,
.hIna'i_din ḥīnaʾidin.

15

21 For another way of encoding the initial hamzah followed by a long vowel, see the tābī' 'l-hamzah' on the following page.
From Wright (1896, i. 14 B):— All consonants, whatsoever, not even ʾalif hemzatum excepted, admit of being doubled and take tašdīd. Hence we speak and write raʾʾās, saʾʾāl, naʾʾāǧ.

Final hamzah: xa.ta'uN, xa.ta'aN, xa.ta'iN, yaqra'u, taqra'Ina, taqra'Una, jA'a, ridA'uN, ridA'aN, jI'a, radi'in, su'UuN, daw'uN, qay'in, šifA'I, šay'uN, šay'aN, šay'iN, zim'aN, al-šay'-u, šay'Ua, ūmul, wa-'sir, fa-'ḏan, fa-ʾtazarat.

tahfīf ℒ-ḥamzat: if the hamzah has ǧazmah and is preceded by ʾalif hamzatum, it must be changed into the letter of prolongation that is homogeneous with the preceding vowel; hence: ʾa'mana, ʾūminu, ʾīmān, ʾīsir, ʾīḏan, ʾūmul, wa-'sir, fa-'ḏan, fa-ʾtazarat, baʿda 'ʾtilāf.
The strange spelling of miʾat nun: miʾatun, miʿatāni, miʾatayni, mūna, miʾātun, miʾatānu, miʿtānu. Of course, the ‘pipe’ character can be used to prevent this rule from being applied (see section 4.5 on page 24): miʾatun.

maddah At the beginning of a syllable, ʾalif with hamzah and fatḥah (א) followed by ʾalif ʾal-maddi (ʾalif of prolongation) or ʾalif with hamzah and ǧazmah (א) are both represented in writing ʾalif with maddah: א (see Wright 1896, i. 25 A–B).

Hence one should keep to this distinction and encode 'aʾkulu ʾākilun and 'Akilun respectively.

šaddah tašdīd is either necessary or euphonic.

The necessary tašdīd always follows a vowel, whether short or long (see Wright 1896, i. 15 A–B). It is encoded in writing the consonant that carries it twice:

'allāqa ʿallāqa, māddun māddun, 'ammara ʾāmmara, murrun murrun.

The euphonic tašdīd always follows a vowelless consonant which is passed over in pronunciation and assimilated to a following consonant. It may be found (Wright 1896, i. 15 B–16 C):—

(a) With the solar letters ال، ض، ص، ش، س، ط، ز، د، ث، ت, after the article ال:—
Unlike arabtex and arabxetex, arabluatex never requires the solar letter to be written twice, as it automatically generates the euphonic taṣdīd above the letter that carries it, whether the article be written in the assimilated form or not, e.g. al-\(^{\ddagger}\)sams-u ُّتَّسَمُّلَا \(\text{aš-šams}^{\ddagger}\), or a\(^{\ddagger}\)-s-\(^{\ddagger}\)sams-u ُّتَّسَمُّلَا \(\text{aš-šams}^{\ddagger}\).

\begin{align*}
al-tamr-u ُّتَّمُّرلَا, & \quad \text{al-ra.\_hm\_an-u ُّتَّرَحَمُّنلَا}, \quad \text{al-} \\
zulm-u ُّتَّزُلمٍلَا, & \quad \text{al-lu.\_g\_a-t-u ُّتَّلُعَّجَنٍلَا}. \quad \text{al-lu\_g\_a-t-u ُّتَّلُعَّجَنٍلَا}
\end{align*}

(b) With the letters \(\text{ر, ل, م, و, ي}\) after \(\text{n}\) with ǧazmah, and also after the tanwīn:—

Note the absence of sukūn above the passed over \(\text{n}\) in the following examples, each of which is accompanied by a consistent transliteration:

\begin{align*}
\text{min rabbi-hi ِهّٰبَّرنِم}, & \quad \text{mir rabbi-hi, min layliN ٍلْيَّلنِم} \\
\text{mil layl\_in}, & \quad \text{an yaqtula َلُتْقَّينَأ} \quad \text{āy yaqtula}. \quad \text{āy yaqtula}
\end{align*}

With tanwīn: kit\_Abu\_N \text{mubInuN ٌنيِبُّمٌباَتِك} \text{kitāb} \text{um} \text{mubīn un}.

Rem. This particular feature must be put into operation by the \texttt{SetArbDflt*} command explicitly. See above section 2.2.1 on page 6 for further details. Other kinds of assimilations, including the various cases of ḵḍām, will be included in arabluatex gradually.

(c) With the letter \(\text{ت}\) after the dentals \(\text{ث, د, ذ, ض, ط, ظ}\) in certain parts of the verb: this kind of assimilation, e.g. ُّتَّثِبَل for ُّتَّثِبَل \(\text{labiṭtu}\), will be discarded here, as it is largely condemned by the grammarians (see Wright 1896, i. 16 B–C).

The definite article and the 'alif" 'l-waṣl i At the beginning of a sentence, \(\text{i}\) is never written, as ُّتَّلَخَشٍلَا \(\text{al-ḥamāl}\); instead, to indicate that the 'alif is a connective 'alif ('alif" 'l-waṣl'), the hamzah is omitted and only its accompanying vowel is expressed:

\begin{align*}
al-.\_hamd-u li-l-l_ah-i ُّتَّلَخَشٍلَا & \quad \text{al-ḥamāl" li-l-llāh‘}. \quad \text{li-l-llāh‘}
\end{align*}

As said above on page 6, fullvoc is the mode in which arabluatex expresses the sukūn and the waṣlah. arabluatex will take care of doing that automatically provided that the vowel which is to be absorbed by the final vowel of the preceding word be properly encoded, like so:—

(a) Definite article at the beginning of a sentence is encoded
\begin{align*}
al-, & \quad \text{or } <\text{solar letter}>-
\end{align*}

if one wishes to mark the assimilation—which is in no way required, as arabluatex will detect all cases of assimilation.

(b) Definite article inside sentences is encoded
\begin{align*}
'l-, & \quad \text{or } <\text{solar letter}>-.
\end{align*}
(c) In all remaining cases of elision, the 'ālifu 'l-waṣli is expressed by the vowel that accompanies the omitted hamzah: (u, a, i).

**Article:** bāb-u 'l-madrasaT-i ُةَسَرْمْلٱُباَب bāb u 'l-madrasat i, al-maqA laT-u 'l-'_Ul_A ىَلوُأْلٱُةَلاَقَمْلَا al-maqālat u 'l-'ara biyyaT-u ُةَّيِبَرَعْلٱُةَغُّللَا al-ru`bT-u 'l-ʿarabiyyat u 'l-ʿulạ̄ ُةِبْعُّرلٱِب bi-'l-qalam-i ِمَلَقْلٱِب bi-'l-qalam, bi-'l-ru`b-i ِبْعُّرلٱِب bi-'r-ruʿb i, fa-lammA raʾawu 'n-naǧm a, fī ṣināʿat i 'l-.tibb-i ِّبِّطلٱِةَعاَنِف fī ṣināʿat i ṣṭ-ṭibb i, 'ilạ 'l-intiqāḍ i ِضاَقِتْنٱِلٱىَلِإ ʾilạ 'l-intiqāḍ i, 'ti 'l-ibtidāʾ i ِءآَدِتْبٱِلٱيِف fi l-ibtidāʾ i, 'abU 'l-wazIr-i ِرُيِزَوْلٱ ʾabu 'l-wazīr i, fa-lammA raʾawu 'n-naǧm a.

**Particles:**

(a) li-: 'ālif 'l-waṣli is omitted in the article ُةَلْيَّلِل when it is preceded by the preposition ل ل: li-l-rajul-i ِلُجَّرلِل li-r-raǧul i. If the first letter of the noun be ل ل, then the ل of the article also falls away, but arabluatex is aware of that: li-l-laylat i ِةَلْيَّلِل li-l-‌laylat i.

(b) la-: the same applies to the affirmative particle ُةَقَحْلَل la-l-ḥaqq-u ُّقَحْلَل la-l-ḥaqq u.

(c) With the other particles, 'ālif 'l-waṣli is expressed: fi 'l-madIna T-i ِةَنيِدَمْلٱيِف fi l-madīnat i, wa-'l-rajul-u ُلُجَّرلٱَو wa-'r-raǧul u, bi-'l-qalam-i ِمَلَقْلٱِب bi-'l-qalam, bi-'l-ru`b-i ِبْعُّرلٱِب bi-'r-ruʿb i.

**Perfect active, imperative, nomen actionis:** qAla isma˚ qāla qala 'sma˚, qāla uqtul-i ُةَلْيَّلِل qāla 'qtul, huwa inbazama ُهُمْسٱِوَأ hūwa 'nhazama, wa-ustumila ُوُسُعْمِلَل 'wa-stuˈmila, qad-i in-sarafa ُقَدْ أَصَفَ āsāfa ُنْسارافا, al-iqtidAr-u ُرَدِتْقٱِلَا al-iqtidār, 'ilạ 'l-intiqāḍ, lawistaqbala ُلَبْقَتْسٱِوَل lawi 'staqbala.

**Other cases:** 'awi ismu-hu ُهُمْسٱِوَأ 'awi 'smu-hu, zayduN ibn-u ʾamriNU ُرْمَعُنْبٱٌدْيَز Zayd ʾamirN ibn 'Amr; 22 umar-u ibn-u 'l- ha.t.tAb-i ُنْبٱُرَمِع ʿUmar, 22 umar-u ibn-u 'l- ha.t.tAb-i ُنْبٱُرَمِع ʿUmar in 'l-ḫaṭṭāb; 22 imru-u 'l-qays-i ُبِلْقَيْسَ إِمْرُوُء إِلَّي Imruʾ in 'l-qays, la-aymun-u 'l-ah-i ُلْيَمْنُهِّلَبَل la-trām 'l-lāh ʾ, 23 ُرُمِيِّنَلَّ أَلْيَنَّ أَلْوَّهُنَّ 'l-lāh ʾ.

---

22 "Zayd is the son of 'Amr": the second noun is not in apposition to the first, but forms part of the predicate. Hence زَيْدُ ُنْبِي ُرْمَعُنْبٱٌدْيَز زَيْدُ ُنْبِي ُرْمَعُنْبٱٌدْيَز "Zayd, son of Zayd", زَيْدُ ُنْبِي ُرْمَعُنْبٱٌدْيَز زَيْدُ ُنْبِي ُرْمَعُنْبٱٌدْيَز "Zayd, son of Amr", زَيْدُ ُنْبِي ُرْمَعُنْبٱٌدْيَز زَيْدُ ُنْبِي ُرْمَعُنْبٱٌدْيَز "Zayd, son of Amr".

23 "Umar is the son of ʾal-Ḥaṭṭāb" (see note 22).
ʾalif ʾl-waṣl\(^1\) preceded by a long vowel The long vowel preceding the connective ʾalif is shortened in pronunciation (Wright 1896, i. 21 B–D). This does not appear in the Arabic script, but arabluatex takes it into account in some transliteration standards:—

\[
\begin{align*}
&\text{ff 'l-nAs-i, 'abU 'l-wazIr-i,} \\
&\text{ff 'l-ibtidA'-i, }_\text{dU 'l-i`lAl-i, }_\text{du 'l-ilāl, maqh_A 'l-'amIr-i,} \\
&\text{[\text{fi 'n-nāṣ, 'abU 'l-wazīr}, \\fi 'l-ibtidā', }_\text{dU 'l-i`lAl, }_\text{du 'l-ilāl, maqh_A 'l-'amīr}.}
\end{align*}
\]

ʾalif ʾl-waṣl\(^1\) preceded by a diphthong The diphthong is resolved into two simple vowels (Wright 1896, i. 21 D–22 A) viz. ay → āĭ and aw → āŭ. arabluatex detects the cases in which this rule applies:—

\[
\begin{align*}
&\text{ff 'aynay 'l-malik-i, ix'say 'l-qaw} \\
&\text{[\text{fi 'aynayi 'l-malik, }_\text{x'sayi 'l-qawm}, \text{mu.s.tafaw 'l-ah-i,} \\
&\text{muŷta-fawu 'l-lāh}.}
\end{align*}
\]

rāmāW 'l-.hijAraT-a, ra'aW 'l-najm-a \\
\[
\begin{align*}
&\text{[\text{ramawu 'l-ḥiǧārat, fa-lammā ra'awu 'n-naǧm}.}
\end{align*}
\]

ʾalif ʾl-waṣl\(^1\) preceded by a consonant with sukūn The vowel which the consonant takes is either its original vowel, or that which belongs to the connective ʾalif or the kasrah; in most of the cases (Wright 1896, i. 22 A–C), it is encoded explicitly, like so:—

\[
\begin{align*}
&\text{[\text{'antumu 'l-kA_dib-Una, ra'aytumu} \\
&\text{'rajul-a, mani 'l-ka_d_dAb-u} \\
&\text{mani 'l-kaḏḏāb, qatalati 'l-rUm-u,} \\
&\text{qatalati 'r-Rūm}.}
\end{align*}
\]

However, the Arabic script does not show the kasrah or the dammah which may be taken by the nouns having tanwīn although it is explicit in pronunciation and must appear in some transliteration standards. arabluatex takes care of that automatically:—

\[
\begin{align*}
&\text{mu.hammaduN 'l-nabā'ī, Muḥammad\text{"\(\text{m}\) 'n-nabī, saLAmuN ud_hulUA} \\
&\text{saLām\text{"\(\text{m}\) 'dhulā, qa.sidata-hu fi qatl-i 'uc{'a}bī \uc{m}\} \\
&\text{uslimiN 'llatī yaqUlu fī-hA 'Abī Muslim\text{"\(\text{m}\) 'llatī yaqūlu fī-hā.}}}
\end{align*}
\]

20
### 4.3 Special orthographies

**The name of God** The name of God, ﷽ tầng, is compounded of the article ﷽ , and ﷽ (noted ﷽ with the defective ʾalif) so that it becomes ﷽ ; then the hamzah is suppressed, its vowel being transferred to the ل before it, so that there remains ﷽ (I refer to Lane, Lexicon, I. 83 col. 1). Finally, the ل is made quiescent and incorporated into the other, hence the lašdid above it. As arabluatex never requires a solar letter to be written twice (see above, on page 17), the name of God is therefore encoded al-l_ah-u or 'l-l_ah-u:—

al-l_ah-u ﷽ al-lāh, ya|24 al-l_ah-u ﷽ yā al-lāh", 'a-fa|25 al-
l_ah-i la-ta.g' alanna 'a-fa-al-lāh' la-taq alanna, bi-'l-
l_ah-i bi-'l-lāh', wa-'l-l_ah-i wa-'l-lāh', bi-sm-i 'l-l_ah-i
bi-sm 'l-lāh', al-.hamd-u li-l-l_ah-i al-.hamd u li-l-lāh,
li-l-l_ah-i 'l-qA'il-u li-l-lāh li-l-lāh:—

**The conjunctive** Although it is compounded of the article ﷽ , the demonstrative letter ل and the demonstrative pronoun ذا, both masculine and feminine forms that are written defectively are encoded alla_dI and allatI respectively. Forms starting with the connective ʾalif are encoded 'lla_dI and 'llatI:—

'a_hAfu mina 'l-malik-i 'lla_dI ya.zimu 'l-nAs-a 'a_hAfu mina 'l-malik 'llaḏī yazimu 'n-nās", 'udtu 'l-
say_h-a 'lla_dI huwa marΙ.dun 'llaḏī huwa marid", ma 'anA bi-'lla_dI qA'iluN la-ka sayaN mā 'anā bi-'llaḏī qā ilun la-ka šay’un.

'ari-nA 'lla_dayni 'a.dallA-nA mina 'l-jinn-i wa-'l-'ins-i 'ari-na 'llaḏayni 'adallā-nā mina 'l-ğinni wa-'l-
ins'.

The other forms are encoded regularly as al-l or 'l-l:—

fa-'innA na_dkuru 'l-sawt-ayni 'l-la_dayni rawayN-humA 'an ja.h.zaT-a fa-'innā naḏkuru 'ṣ-sawt'ayni
'l-ladayni rawayN-humā 'an Ǧahzat".

---

24 Note the "pipe" character '|' here after ya and below after fa before footnote mark 25; it is needed by the dmg transliteration mode as in this mode any vowel at the commencement of a word preceded by a word that ends with a vowel, either short or long, is absorbed by this vowel viz. ʿalạ ʿṭ-ṭarīq. See section 4.5 on page 24 on the "pipe" and section 8 on page 40 on dmg mode.

25 See note 24.
And also: al-la_dAni، al-ladāni، al-la_dayni، al-latAni، al-ladānī، al-latayni، al-latāni، and so forth.

4.4 Quoting

It is here referred to “quoting” after the arabtex package. The “quoting” mechanism of arabluatex is designed to be very similar in effect to the one of arabtex.

To start with an example, suppose one types the following in novoc mode:

```plaintext
للهايلا
```

is it "ullima، or, he was taught the science of astronomy، or "ullm، he taught the science of astronomy؟ In order to disambiguate this clause، it may be sensible to put a dammah above the first غ، which is achieved by “quoting” the vowel u، like so: "ullima، or، with no other vowel than the required u: "ullm.

This is how the “quoting” mechanism works: metaphorically speaking، it acts as a toggle switch. If something، in a given mode، is supposed to be visible، “quoting” hides it؛ conversely، if it is supposed not to، it makes it visible.

As shown above، “quoting” means inserting one straight double quote (" scratch) before the letter that is to be acted upon. Its effects depend on the mode which is currently selected، either novoc، voc or fullvoc:

novoc In this mode، “quoting” essentially means make visible something that ought not to be so.

(a) Quoting a vowel، either short or long، makes the dammah، fatḥah or kasrah appear above the appropriate consonant:—

```plaintext
ullima 'ilm-a 'l-hay'aT-i
```

ya.gz "ullima ʿilm a 'l-hayʾat i،

(b) The same applies when “quoting” the tanwīn:—

```plaintext
wa-'innAsawfatudriku-na 'l-manAyAmuqadd"araT"an
```

wa-ʾinnā sawfa tudrikun- na 'l-manāyā muqaddarat

(c) If no vowel follows the straight double quote، then a sukūn is put above the preceding consonant:—

```plaintext
qAla isma" "gāla 'isma، jA'at" hindun
```

qāla isma "gāla 'isma، jA'at hindun

26 Note here the "pipe" character '|': as already stated on page 17، the sequence 'A usually encodes ʾalif with hamzah followed by ʾalif of prolongation، which is represented in writing ʾalif with maddah: \. The “pipe” character prevents this rule from being applied. See section 4.5 on page 24.

27 See Lagally (2004، p. 22)
(d) At the commencement of a word, the straight double quote is interpreted as 'ʾalif' 'l-waṣl':—

wa-"ust"u`mila, huwa "inhazama ho An`am wa-stu`mila, huwa "inhazama, al-"intiqA.d-u al-intiqād".

voc In accordance with the general rule, in this mode, “quoting” makes the vowels and the tanwīn disappear, should this feature be required for some reason:—
(a) Short and long vowels:—

q"Ala q"A'iluN qāla qā'il"un, ibn-u 'abI 'u.saybi'aT- 'a Ibn Abī Uṣaybi'at.

(b) tanwīn:—

madīnAT"aN madīnat"un, bAb"aN bāb"an, hud"aN_A hudā"un, "say" 'aN šay"un.

One may more usefully “quote” the initial vowels to write the waṣlah above the 'ʾalif or insert a straight double quote after a consonant not followed by a vowel to make the sukūn appear:—
(a) 'ʾalif' 'l-waṣl':—

fI "istiq.sA'-iN fi 'istiq.sā', wa-"istiq.sA'-uN wa-'stiqla"un, qAla "uhrub fa-lan tuqtala qāla 'hrub fa-lan tuqtala.

(b) sukūn:—

qAla "uqtul" fa-lan tuqtala qāla 'gtul fa-lan tuqtala ma JA'at mini imra'TIN ma gā'at mini 'mra'at'in, kam "qad" ma.dat" min" laylaTim kam qad maḍat mini laylat"un.

fullvoc In this mode, “quoting” can be used to take away any short vowel (or tanwīn, as seen above) or any sukūn:—

al-jamr-u 'l-.sayfiyy-u 'lla_dI kAna bi-q"rAn"nUn-a

4.4.1 Quoting the hamzah

As said above in section 4.2 on page 15, the hamzah is always written (‘), its carrier being determined by contextual analysis. “Quoting” that straight single quote character like so: (‘) allows to determine the carrier of the hamzah freely, without any consideration for the context. Table 5 gives the equivalents for all the possible carriers the hamzah may take.
Table 5: “Quoted” hamzah

As one can see from table 5, the carrier of the hamzah is inferred from the letter that precedes the straight double quote ("'). Of course, any “quoted” hamzah may take a short vowel, which is to be written after the ArabTEX equivalent for the hamzah itself, namely ⟨"⟩. For example, َؤ is encoded ⟨w'⟩, while ْؤ is encoded ⟨w'"⟩. In the latter example, the second straight double quote encodes the sukūn in voc mode in accordance with the rule laid above on pages 22–23.

4.5 The “pipe” character (|)

In the terminology of ArabTEX, the “pipe” character ‘|’ is referred to as the “invisible consonant”. Hence, as already seen above in section 4.4.1 on the preceding page, its usage to encode the hamzah alone, with no carrier: ⟨|"⟩.

Aside from that usage, the “pipe” character is used to prevent almost any of the contextual analysis rules that are described above from being applied. Two examples have already been given to demonstrate how that particular mechanism works in note 24 on page 21 and in note 26 on page 22. One more example follows:—

bi-Qrānnūn|nūn-a“ in Crannon” (Thessaly, Greece).

As one can see, the “pipe” character between the two ⟨n⟩ prevents the necessary tašdīd rule (page 17) from being applied.

4.6 Putting back on broken contextual analysis rules

In complex documents such as critical editions where footnotes and other kind of annotations can be particularly abundant, the contextual analysis rules that are described above may be broken by LTEX commands. To take an example, consider the following:—

See below section 8 on page 40.
See more context on the previous page.
According to the rule stated on page 20, the diphthong in raʾaw must be resolved into two simple vowels before the ‘ālifu ‘l-waṣlī, as ُرَا أْلاَمِلَفْبِلْمَجَّنلٱاُوَأَر. The \arbnul{} command is provided so as to put back on contextual analysis rules in such situations. It takes as argument the word that must be brought back for any given rule to be applied as it ought to. Depending on the contexts that have to be restored, \arbnul{} may be found just after or before Arabic words.

In any case, no space must be left after or before the Arabic word that \arbnul{} is applied to.

The following shows how the Arabic should have been written in the preceding example and gives further illustrations of the same technique:—
4.7 Stretching characters: the taṭwil

A double hyphen (- -) stretches the ligature in which one letter is bound to another. Although it is always better to rely on automatic stretching, this technique can be used to a modest extent, especially to increase legibility of letters and diacritics which stand one above the other:

\textit{.hunayn-u bn-u 'is.h--a q-a a H}unayn\textsubscript{u} \textit{bn\textsubscript{u} Ishāq\textsubscript{u}}

4.8 Digits

4.8.1 Numerical figures

The Indian numbers, ar-raqam\textsuperscript{u} \textit{1-hindiyy\textsuperscript{u}}, are ten in number, and they are compounded in exactly the same way as our numerals:

\textit{1874 ١٨٧٤, 123-456,789 ١٢٣٤٥٦٧٨٩, fI sanaT-i 1024 ١٠٢٤ِتَنَسَيِف}

4.8.2 The abjad

The numbers may also be expressed with letters from right to left arranged in accordance with the order of the Hebrew and Aramaic alphabets (see Wright\textit{1896}, i. 28 B–C). The \textit{abğad} numbers are usually distinguished from the surrounding words by a stroke placed over them.

\textit{abğad} numbers are inserted with the \texttt{\abjad{number}} command in any of the \texttt{voc}, \texttt{fullvoc} and \texttt{novoc} modes, where \texttt{\{number\}} may be any number between 1 and 1999, like so:

\texttt{\abjad{45} kitAbu-hu fI '1-'AdAt-i 45 kitābu-hu fī 'l-ādāt\textsuperscript{i}.}

Rem. \textit{a}. As can be seen in the above given example, \texttt{arablutex} expresses the \textit{abğad} numbers in Roman numerals if it finds the \texttt{\abjad} command in any of the transliteration modes.

Rem. \textit{b}. \texttt{\abjad} may also be found outside Arabic environments. In that case, \texttt{arablutex} does not print the stroke as a distinctive mark over the number for it is not surrounded by other Arabic words. In case one nonetheless wishes to print the stroke, he can either use the \texttt{aemph} command that is described below in section \textit{4.10} on page \texttt{28} or insert the \textit{abğad} number in \texttt{\arb[novoc]{}}:

\texttt{26}
The \arb[trans]{'abjad} number for 1874 is \abjad{1874}. The 'abjad number for 1874 is دعضغ.

The \arb[trans]{'abjad} number for 1874 is \textit{\abjad{1874}}. The ‘abjad number for 1874 is دعضغ.

The \arb[trans]{'abjad} number for 1874 is \arb[novoc]{\abjad{1874}}. The ‘abjad number for 1874 is دعضغ.

\abjad{} may also be used to convert values of counters into ‘abjad numbers, like so:

1. The \arb[trans]{'abjad} number for the current page (\thepage) is \abjad{\thepage}.
2. \abjad{\thepage}.

The ‘abjad number for the current page (27) is زك.

This technique can be used to produce abjad-numbered lists as will be demonstrated on page 53.

4.9 Additional characters

In the manuscripts, the unpointed letters, al-ḥurūf ‘l-muhmalat, are sometimes further distinguished from the pointed by various contrivances, as explained in Wright (1896, i. 4 B–C). One may find these letters written in a smaller size below the line, or with a dot or another mark below. As representing all the possible contrivances leads to much complexity and also needs to be agreed among scholars, new ways of encoding them will be proposed and gradually included as arabluatex will mature.

For the time being, the following is included:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration\textsuperscript{30}</th>
<th>ArabTpX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ٗ</td>
<td>b</td>
<td>.b</td>
</tr>
<tr>
<td>ٙ</td>
<td>d</td>
<td>.d</td>
</tr>
<tr>
<td>ٝ</td>
<td>f</td>
<td>.f</td>
</tr>
<tr>
<td>ٞ</td>
<td>q</td>
<td>.q</td>
</tr>
<tr>
<td>ٙ</td>
<td>k</td>
<td>.k</td>
</tr>
<tr>
<td>٘</td>
<td>n</td>
<td>.n</td>
</tr>
</tbody>
</table>
| ٙ | ( | (( |}

Table 6: Additional Arabic codings

\textsuperscript{30}See below section 8 on page 40.
4.10 Arabic emphasis

As already seen in section 4.8.2 on page 26, the ṣabgad numbers are distinguished from the surrounding words by a stroke placed over them. This technique is used to distinguish further words that are proper names or book titles.

\texttt{\textbackslash emph{⟨Arabic text⟩}}

One may use the \texttt{\textbackslash emph{⟨Arabic text⟩}} command to use the same technique to emphasize words, like so:—

\texttt{\textbackslash abjad{45}: kit\textbackslash a\textbackslash b\textbackslash u-hu \textbackslash emph{fi \ 'l-\'A\textbackslash d\textbackslash a\textbackslash t-i}}

\texttt{45: kit\textbackslash a\textbackslash b\textbackslash u-hu fi \ 'l-\'A\textbackslash d\textbackslash a\textbackslash t-i}.

Rem. a. As the above example shows, \texttt{arabluatex} places the horizontal stroke under the emphasized words in any of the transliteration modes.

Rem. b. \texttt{\textbackslash emph*} is also provided should one wish to always have the horizontal stroke printed over the emphasized words, like so: \texttt{\textbackslash abjad{45}: kit\textbackslash a\textbackslash b\textbackslash u-hu \textbackslash emph*{fi \ 'l-\'A\textbackslash d\textbackslash a\textbackslash t-i}}

\texttt{45: kit\textbackslash a\textbackslash b\textbackslash u-hu fi \ 'l-\'A\textbackslash d\textbackslash a\textbackslash t-i}.

5 Arabic poetry

\texttt{arabluatex} provides a special environment for typesetting Arabic poetry. Every line in this environment must end with \textbackslash .

\texttt{\textbackslash ar\textbackslash a\textbackslash b\textbackslash v\textbackslash e\textbackslash r\textbackslash s\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash v\textbackslash e\textbackslash n\textbackslash t\textbackslash i\textbackslash o\textbackslash n\textbackslash g\textbackslash o\textbackslash o\textbackslash r\textbackslash o\textbackslash r\textbackslash f\textbackslash o\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbackslash e\textbackslash r\textbackslash e\textbackslash n\textbackslash t\textbacklash...}
If the name of the metre is expressed, it is printed after the lines and set flush left in voc, fullvoc and novoc modes or flush right in trans mode.

\SetHemistichDelim \SetHemistichDelim{delimiter}

This named argument does not need a value as it defaults to true if it is used. If so, a delimiter is printed between each of the hemistichs. By default, it is set to the 'star' character '*'. The \SetHemistichDelim{delimiter} command may be used at any point of the document to change this default setting.

utf utf=true|false

As the preceding one, this named argument does not need a value as it defaults to true if it is used. If so, Unicode Arabic input is expected in the arabverse environment instead of ASCII ArabTEX or Buckwalter input schemes. See section 10 on page 47 for more details.

color color=(color name)

The color in which lines of poetry are to be rendered.

\begin{arabverse}[mode=fullvoc, width=.3\linewidth]
\bayt(ṣadr){(tadwīr)} {(‘aḡuz)}\end{arabverse}

That two subsequent hemistichs should be connected with one another is technically named tadwīr. Should that happen, either the ṣadr or the ‘aḡuz or both of them, may be connected to one another by letters that are naturally bound to the following or the preceding ones over the tadwīr. The optional argument of the \bayt command is designed to deal with the various situations that may arise:—

(a) If the two hemistichs be connected with one another by a prominent horizontal flexible stroke, the taṭwīl should be used, like so: [--] (see section 4.7 on page 26). Of course, the ending word of the ṣadr and the word at the commencement of the ‘aḡuz must have the taṭwīl too so that the proper shapes of the letters be selected. Consider for example the following:—

\begin{arabverse}[mode=fullvoc, width=.3\linewidth]
\bayt(ṣadr){(tadwīr)} {(‘aḡuz)}\end{arabverse}

1 A 'starred' version \bayt* is also defined. arabluatex uses it internally when export is set to true to instruct some Lua functions that lines of poetry have already been processed. That aside, \bayt and \bayt* do the same, and only \bayt should be used.
As one can see, triple hyphens have been used. In the sadr, the first hyphen triggers the rules that are related to the definite article and the ‘alif ‘l-waṣl, while the following two select the figure of the letter lām connected with a following letter. In the ‘aǧuz, the last two hyphens select the letter yā’ connected with a preceding letter, while the first one is simply discarded in this mode, but still may appear as it should, if the trans mode be selected:—

\begin{arabverse}[mode=trans, width=.4\linewidth]
\bayt{lā 'arā man 'ahidtu fī-hā fa-ʾabkī 'l-\(\ldots\)}{---ya\(\ldots\)}
dalhān wa-mā yaruddu 'l-bukā'u
\end{arabverse}

(a) In some other cases, it may seem difficult, if not fairly impossible, to split a given word into two parts. This happens mostly because of the šaddah. Consider for example the following:—

\begin{arabverse}[mode=fullvoc, width=.25\linewidth, gutter=1cm]
\bayt{.gayra} annī qad 'asta\(\ldots\)\(\ldots\)}{---di\(\ldots\)}
lā ṣā́f bi-'l-tawiyyī 'l-\(\ldots\)}{---mūmum\(\ldots\)}
dsawiyyya\(\ldots\)}{---mūmum\(\ldots\)}
\end{arabverse}

(b) In some other cases, it may seem difficult, if not fairly impossible, to split a given word into two parts. This happens mostly because of the šaddah. Consider for example the following:—

\begin{arabverse}[mode=fullvoc, width=.25\linewidth, gutter=1cm]
\bayt{gayra} annī qad 'asta\(\ldots\)\(\ldots\)}{---di\(\ldots\)}
lā ṣā́f bi-'l-tawiyyī 'l-\(\ldots\)}{---mūmum\(\ldots\)}
dsawiyyya\(\ldots\)}{---mūmum\(\ldots\)}
\end{arabverse}

In the first line, the word "\(\ldots\)" should be split into "\(\ldots\)"\(\ldots\) as the first part of it belongs to the sadr and the second to the ‘aǧuz. One solution to avoid splitting this word in such a way is to write inside the tadwīr the part of it that belongs to either hemistich, without omitting to add a space after it. In the second line, the word "\(\ldots\)" should be split into "\(\ldots\)"\(\ldots\), so that the only way to avoid splitting it into two parts is to write it all inside the tadwīr. In that case, as the word is to be placed in the middle, it has been surrounded by spaces.

\footnote{See section 4.2 on page 18}
Scaling and distortion of characters  The \texttt{arabverse} environment and the \texttt{bayt} command are designed to typeset the verses in a two-column, fixed width layout. This may result in a somewhat distorted text. Should that happen, one may adapt the layout by modifying the values of the above described width and gutter named arguments until the visual aspect of the layout be satisfactory. It has to be noted that distortion and warping may be even more perceptible in Roman than in Arabic characters.

Footnotes  Footnotes are not set by default inside the \texttt{bayt} command, but there are two easy ways to have them printed.

If they are little in number, each footnote may be split into pairs of \texttt{footnote mark}\{\} (please mind the braces or “declare” \texttt{footnotemark} using \texttt{\MkArbBreak} to take it out of the Arabic environment\footnote{See section 11.1 on page 50.}) in the argument of the \texttt{bayt} command and \texttt{footnotetext} outside the \texttt{bayt} command.

If the footnotes are abundant in number, it is advised to load the \texttt{footnotehyper} package which \texttt{arabluatex} will then use to typeset any kind of footnote that is called from the arguments of the \texttt{bayt} command.\footnote{The \texttt{footnote} package can also be used for the same effect. However, it must be loaded \textit{after} \texttt{arabluatex}.}

Line numbering  Inside the \texttt{arabverse} environment, the \texttt{linenumbers} environment of the \texttt{lineno} package can be used to have the lines of succeeding verses numbered. Please refer to the documentation of this package for more information or to the example below for a basic implementation of this technique.

5.1 Example

Here follow the first lines of Imru’u ’l-Qaysi’s \textit{Mu’allaqah}. In this example, \texttt{\SetArbDflt*} has been selected so as to mark the ʾidgām that is fit to this declamatory poetry:—\footnote{Please note that for the time being only the assimilation rules that are laid on item (b) on page 18 are applied. See section 2.2.1 on page 6 for more information. None of the editions of the \textit{Mu’allaqāt} that I know of feature the ʾidgām in the Arabic text, although it is often strongly marked in declamation.}

\begin{verbatim}
\begin{arab}[fullvoc]
qAla imru’u ’l-\uc{q}aysi fI mu’allaqati-hi:
\end{arab}
\begin{arabverse}[mode=fullvoc, metre={((al-.darbu 'l-_tAnI mina 'l-_tawIli))}]
\SetArbDflt*
\begin{linenumbers*}
\bayt{qifA nabki min _dikr_A .habIbiN wa-manzili}{bi-saq.ti\
'1-\uc{l}-iU.A bayna \uc'{1-d}a_bulli fa-\uc{h}.lasmalil}\}
\bayt{fa-\uc{t}U.di ha fa-'l-\uc{m}liqrATi lam ya\uc{.h}awmali}\}
\bayt{\{limA nasa\uc{.g}at-hA min _ganUbI\uc{.n} wa-\uc{.m}sam\uc{'a}li\}\}
\bayt{\{tar_A ba\uc{.a}ra 'l-\uc{l}-arAmi fI \uc{.a}rAsAti-hA\{\uc{w}-\uc{q}I\uc{.A}ni-hA\}
\ka-\annA-hu .habbu fulfuli}\}
\end{linenumbers*}
\end{arabverse}
\end{verbatim}
6 Special applications

6.1 Linguistics

The same horizontal stroke as the tašdīd (see section 4.7 on page 26) may be encoded ⟨B⟩; ⟨BB⟩ will receive the tašdīd. This is useful to make linguistic annotations and comments on vowels:

Bu Ba Bi BuN BaN BiN ⋯ ⋯ u a i un an in, BBu BBa BBI ⋯ ⋯ u a i, B-–aN

6.2 Brackets

The various bracket symbols are useful in technical documents such as critical editions for indicating that some words or some letters must be added or removed. arabluatex will automatically fit those symbols to the direction of the text. For the time being, the following symbols are supported:
Parentheses, square and angle brackets may be input directly at the keyboard; however, words or letters that are to be read between braces must be passed as arguments to the \abrases command:

\begin{arab}
\abrases{wa-qAla} 'inna 'abI kAna mina 'l-muqAtilaTi
wa-kAna--<--t> 'ummA min 'u.zamA'i buyUt 'l-zamAzimaTi.
\end{arab}

Additional Arabic marks In addition to common letters, many symbols and ligatures are encoded in Arabic Unicode standard, such as honorifics consisting of complex ligatures, and annotation signs used in the Qurān or in classical poetry. \arbmark[⟨r(l)r⟩]{〈shorthand〉} can be used to insert such characters either in Unicode or in romanized Arabic environments. It takes as argument a shorthand defined beforehand in a default list which consists of the following at the time of writing:

<table>
<thead>
<tr>
<th>Codepoint</th>
<th>Shorthand</th>
<th>Glyph</th>
<th>Transliteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDFD</td>
<td>bismillah</td>
<td>بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ</td>
<td>bi-'smi 'Llāhi 'r-raḥmāni 'r-raḥīmi</td>
</tr>
<tr>
<td>FDF5</td>
<td>salam</td>
<td>صَلَّم</td>
<td>šallā 'Llāhu 'alay-hi wa-sallama</td>
</tr>
<tr>
<td>FDAF</td>
<td>slm</td>
<td>صلّ</td>
<td>šallā 'Llāhu 'alay-hi wa-sallama</td>
</tr>
<tr>
<td>FDFB</td>
<td>jalla</td>
<td>ﺡَالْ</td>
<td>jalla ḥa-lāla-hu</td>
</tr>
</tbody>
</table>

Table 7: Additional Arabic marks

The mark to be inserted is determined by contextual analysis, or by an optional argument, either rl to have the Arabic glyph printed, or lr to print the transliterated equivalent.

\newarbmark is also provided should one wish to define new marks in addition to the marks defined above. This command takes three arguments, like so:

\newarbmark{⟨shorthand⟩}{⟨RTL codepoint⟩}{⟨LTR rendition⟩}
As regards the right-to-left codepoint, it may be either typed in Unicode or selected as Unicode codepoint. To that end, the IATpX command \symbol{"XYZT"} or its plain TpX variant \char"XYZT\relax may be used, where XYZT are uppercase hex digits (0 to 9 or A to F).

It is also possible to use the so-called ‘~~~~ notation’ like so: ~~~~~xyzt, where x y z t are lowercase hex digits (0 to 9 or a to f).

As regards the third argument (left-to-right rendition), it may be either left empty or typed by means of \arb[trans]{⟨arabtex code⟩} so as to have it printed in romanized Arabic.

It must be noted that \newarbmak expects ArabTEx input scheme inside \arb[trans]{⟨arabtex code⟩} to the exclusion of buckwalter input scheme.

The example below provides an implementation of this technique. It may be observed that \arbcolor is used so as to have the marks printed in red:

\begin{verbatim}
\SetArbDflt*
\newarbmak{sly}\{\arbcolor[red]{~~~~06d6}\}{}
\newarbmak{jim}\{\arbcolor[red]{~~~~06da}\}{}
\begin{arab}
sU_rTu 'l-nisA'i, 19:
\end{arab}
\end{center}
\begin{arab}[fullvoc]
\arbmark{bismillah}
\end{arab}
\end{verbatim}
6.1 The Qurʾān

This sub-part is destined to become a part of its own, as fine typesetting of Qurʾānic text is planned in the versions of arabluatex to come in the medium-term. New functions and new Arabic modes will be available as arabluatex will mature.

For the time being, \ayah{⟨3-digit number⟩} is provided so as to typeset the number of the āyah that it is referred to inside the dedicated mark – Unicode U+06DD: 

\ayah(123) (123).

An example follows:—

Caveat For some reason, most of the Arabic fonts do not show the number properly: some are only able to display at most two digits, while others display the digits outside the ‘end of āyah’ sign, let alone those that print the digits stacked. To the knowledge of the writer, this should be reported to the developers of those fonts.

7 Color

arabluatex is able to render in color either words, parts of words or diacritics. As
the techniques implemented in this section may lead to some complexity, the reader should first become well acquainted with the following points:

(a) The “pipe” character (|, section 4.5 on page 24);
(b) ‘Quoting’ technique (section 4.4 on page 22), and more specifically ‘quoting the hamzah’ (on page 23);
(c) Putting back on broken contextual analysis rules (section 4.6 on page 24);
(d) Arabic marks (section 6 on page 33).

\arbcolor{color}{(Arabic text)}

As this example shows, \arbcolor has been used to render headings in red with the same encoding both in vocalized and in romanized Arabic. The same technique also applies to syllabes inside words. arabluatex takes care of selecting the appropriate shape of the letters while coloring them:—

\begin{arab}
\arbcolor[red]{al-bābu 'l-ḥāmisu} fī ṭabaqāti 'l-ʾaṭibbāʾi 'llaḏīna kānū munḏu zamāni Ǧālīnūsa wa-qarīb
\end{arab}

As this example shows, \arbcolor has been used to render headings in red with the same encoding both in vocalized and in romanized Arabic. The same technique also applies to syllabes inside words. arabluatex takes care of selecting the appropriate shape of the letters while coloring them:—

\begin{arab}
\{'voc'\ mode:
'i^stara\arbcolor[red]{niN}tu-hu bi-ṯama\arbcolor[blue]{^ga}ba-ka
\end{arab}

36Regarding the colors themselves and the way new colors can be defined in addition to those that are already available, please refer to the xcolor package.
### 7.1 Tricks of the trade

#### Diacritics

Depending on the mode selected, either `voc`, `novoc` or `fullvoc`, coloring the diacritics requires more attention for the insertion of `\arbcolor` may prevent contextual analysis from being applied.

Furthermore, depending on the surrounding letters, the standard encoding of short vowels (`u, a, i`) may result either in diacritics or in a connective `ʾalif` with the `wašlah` or its accompanying vowel. As for the `sukūn`, it is generated by contextual analysis. Thus applying colors to bare diacritics requires them to have specific encodings.

Table 8 gives the ArabTEX equivalents for the diacritics to be printed inside or just after `\arbcolor`.

<table>
<thead>
<tr>
<th>Diacritic</th>
<th>Transliteration</th>
<th>ArabTEX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>'</td>
<td>a a a</td>
<td>.a</td>
</tr>
<tr>
<td>'</td>
<td>u u u</td>
<td>.u</td>
</tr>
<tr>
<td>'</td>
<td>i i i</td>
<td>.i</td>
</tr>
<tr>
<td>'</td>
<td>o</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: ArabTEX diacritics for `\arbcolor`

The following examples show how the letters, or the diacritics above or under them or both the letters and the diacritics can be rendered in different colors:—

‘voc’ mode:

\begin{verbatim}
 i*'staraytu-hu bi-_taman\arbcolor[red]{niN} '{a`\arbcolor[blue]{^ga}ba-ka \textit{ištaraytu-hu bi-ṭaman} in 'a้งaba-ka.
\end{verbatim}

\begin{verbatim}
 i*'staraytu-hu bi-_tama\arbcolor[red]{niN} '{a`\arbcolor[blue]{^ga}ba-ka \textit{ištaraytu-hu bi-ṭaman} in 'a้งaba-ka.
\end{verbatim}

\begin{verbatim}
 i*'staraytu-hu bi-_tama\arbcolor[red]{niN} \arbcolor[blue]{ba-ka} \textit{ištaraytu-hu bi-ṭaman} in 'a้งaba-ka.
\end{verbatim}

---

37 See below section 8 on page 40.
As can be seen, fullvoc required the letters y, n and ^g before \arbcolor to be ‘quoted’. Otherwise, unwanted sukūns would have been generated because of the absence of a vowel after those consonants.

\textbf{tanwin} \arbnull must be used with futhatān (ۛ) so as to put back on contextual analysis rules:

\begin{verbatim}
mu`allim\arbcolor\{\arbnul{l}m\}aN, \istisqA\arbcolor\{\arbnul{l}A'\}aN, ^say\arbcolor\{\arbnul{l}aY'\}aN, ^gAmi`a\arbcolor\{\arbnul{l}T\}aN.
\end{verbatim}

\textbf{Rem.} Note that in the last example (\textit{gāmiʿat} \arbnull), the ‘pipe’ character has been inserted before \arbcolor. Otherwise, the \textit{dmg} mode of the transliteration rules would have interpreted the tāʾ marbūṭah as final (e.g. h instead of the expected t).\footnote{See also on page 44 “Discarding the iʿrāb” for more information.}

The \textit{tanwin} preceding ی conveys even more intricate business to the rendering with the utmost accuracy in both romanized and non-romanized modes. First, a new Arabic mark needs to be defined. It should print ی in Arabic script and not a thing in transliteration. It is to be appended after \arbcolor, like so:

\begin{verbatim}
1 \newarbmark(Y){0649}()
2 \arb[\trans]{hud\arbcolor{red}(\arbnul{\_})aN}\arbmark(Y)
3 \arb[\trans]{hud\arbcolor{red}(\arbnul{\_})aN}\arbmark(Y)
\end{verbatim}

\textit{huda}
waṣlah and maddah  Both can be generated with the help of \arbnull:—

wa-\arbcolor{red}{\arbnull{wa}}\isqA'uN
fi "al".i-\arbcolor{red}{\arbnull{\'l-}i}btidA'i.
\arpred{a}\arbnull{k}}\isqA'\uN
\arpred{a}\arbnull{k}kilun.

The Unicode codepoint of the maddah is 0653, while bare ʾalif is 0627. So:—

\newarbmark{alifmaddahred}{\arbcolor{red}{\arbnull{a}}}{kulu}.
\newarbmark{alifmaddahred}{\arbcolor{red}{\arbnull{k}}}{kulu}.

Rem. In the preceding example, any consonant could have been passed as argument to the \arbnull command.

šaddah  In the following example, it is assumed that the šaddah above the letter ل in لَنوُمِّلَعُمْلَا, al-muʿallimūna, is to be rendered in red. Thus the Arabic mark must generate the šaddah alone—of which the Unicode codepoint is 0651—in Arabic script and the letter ‘l’ in transliteration:—

The definite article and the euphonic tašdid  The intricate business of rendering in color the initial ʾalif al-waṣl of the definite article followed by a solar consonant must be unraveled.

From the examples provided above, in fI ʾl-nAsi fi ʾn-nāsi, the initial ʾalif” ʾl-waṣl can be rendered in red like so: \arbnull{al-}a. Then, the following two letters, namely 1-n, must print the string lām + nūn + šaddah in Arabic, and exactly n-n in transliteration. Thus an Arabic mark is needed:—

39 To the knowledge of the writer, the waṣlah alone is not part of the Arabic Unicode block.
The ‘quoting’ technique provides an easy way to determine the carrier of the hamzah, as shown in table 5 on page 24—:

\begin{verbatim}
yatasA\arbn\{a\}\arbcolor{red}{|''}.alUna yatasą'a-lāna, \"say\arbcolor{red}{|''}\{a\N\} alUna \"say\ arb\{a\N\}in, \arbcolor{red}{|''}.iN \"say\arb\{a\}N, \arbcolor{red}{|''}.as\arbcolor{red}{y''}.ilaTuN  ❇️ as Ɨal\"n.
\end{verbatim}

8 Transliteration

It may be more appropriate to speak of “romanization” than “transliteration” of Arabic. As seen above in section 2.2 on pages 5–9, the “transliteration mode” may be selected globally or locally.

This mode transliterates the ArabTEX input into one of the accepted standards. As said above on page 6, three standards are supported at present:

\textbf{dmg Deutsche Morgenländische Gesellschaft}, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.\textsuperscript{40} dmg transliteration convention is selected by default;

\textbf{loc Library of Congress}: this standard is part of a large set of standards for romanization of non-roman scripts adopted by the American Library Association and the Library of Congress;\textsuperscript{41}

\textbf{arabica Journal of Arabic and Islamic Studies/Revue d'études arabes et islamiques}: this standard is most widely used by scholars in the field of Arabic studies.\textsuperscript{42}

More standards will be included in future releases of arabluatex.

\textbf{\SetTranslitConvention Convention} The transliteration mode, which is set to dmg by default, may be changed at any point of the document by the \SetTranslitConvention{⟨mode⟩} command, where ⟨mode⟩ may be either dmg, loc or arabica. This command is also accepted in the preamble should one wish to set the transliteration mode globally, e.g.:—

\textsuperscript{40}See Brockelmann et al. (1935).

\textsuperscript{41}See \url{http://www.loc.gov/catdir/cpso/roman.html} for the source document concerning Arabic language.

\textsuperscript{42}See \url{http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf}.
\SetTranslitStyle \textbf{Style} \hspace{2pt} Any transliterated Arabic text is printed in italics by default. This also can be changed either globally in the preamble or locally at any point of the document by the \SetTranslitStyle{⟨style⟩} command, where ⟨style⟩ may be any font shape selection command, e.g. \texttt{\upshape}, \texttt{\itshape}, \texttt{\slshape}, and so forth.

\SetTranslitFont \textbf{Font} \hspace{2pt} \SetTranslitFont{⟨font selection command⟩} allows any specific font to be selected for rendering transliterated text with the font-selecting commands of the fontspec or luaoTfload package. Of course, this font must have been defined properly. To take one example, here is how the \textit{Gentium Plus} font can be used for rendering transliterated text:—

\newfontfamily\translitfont{Gentium Plus}[Ligatures=TeX]
\SetTranslitFont{\translitfont}

\uc \textbf{Proper names} \hspace{2pt} Proper names or book titles that must have their first letters uppercased may be passed as arguments to the \texttt{\uc{⟨word⟩}} command. \texttt{\uc} is a clever command, for it will give the definite article \textit{al-} in lower case in all positions. Moreover, if the initial letter, apart from the article, cannot be uppercased, viz. ’ or ‘, the letter next to it will be uppercased:—

\uc{.hunayn-u} \text{bn-u} \uc{.i\_s.h_aq-a} \text{Hunayn” bn”}
\uc{Tshāq”,} \uc{.u\_tm_an-u} \text{Utmān”,} \text{.daraba} \uc{.zayd-u} \text{bn-u}
\uc{.h_alidinN} \uc{.sa\_d-a} \text{bn-a} \uc{.awf-i} \text{bn-i} \uc{.abd-i}
\uc{.’l-1\_ah-i} \text{daraba Zayd” bn”}
\uc{Hālid”} \text{Sa’d”} \text{bn’a} \text{‘Auf}” \text{bn’} \text{‘Abd}” \text{Llāh’}.

However, \texttt{\uc} must be used cautiously in some very particular cases, for the closing brace of its argument may prevent a rule from being applied. To take an example, as seen above on page 20, the transliteration of مُحَمَّد َالْنَبِي must be Muhammad\textsuperscript{um}‘ \textit{n-nabī}, as nouns having the \\textit{tanwi\textsuperscript{n}} take a kasrah in pronunciation before \textit{alifu ’l-waṣli}. In that case, encoding مُحَمَّد like so: \texttt{\uc{mu.hammaduN}} is wrong, because the closing brace would prevent \texttt{arabluatex} from detecting the sequence ⟨\texttt{-uN}⟩ immediately followed by ⟨’l⟩. Fortunately, this can be circumvented in a straightforward way by inserting only part of the noun in the argument of \texttt{\uc} vz. up to the first letter that is to be uppercased, like so: \texttt{\uc{m}u.hammaduN}.

\textbf{Hyphenation} \hspace{2pt} In case transliterated Arabic words break the \TeX hyphenation algorithm, one may use the \texttt{\-} command to insert discretionary hyphens. This command will be discarded in all of the Arabic modes of \texttt{arabluatex}, but will be processed by any of the transliteration modes:—
\texttt{\textbackslash uc{'abU \uc{bakraN} \uc{mu\-\ham\-madu} bnu \uc{za\-ka \-}\riy\-yA\-a} \uc{'l\-rAziiyyu}} ‘Abū Bakr Muḥammad b. Zakariyyā’ ar-Rāziyyu.

\texttt{\textbackslash arb[trans]{\uc{'abUzaydiN.hunaynubnu'is.h_aqa'l-`ibAdiyyu}} ʾAbū Zayd Ḥunayn ibn ʾIsḥāq al-ʿIbādiyyu.

\texttt{\textbackslash prname} \texttt{\textbackslash \prname} Proper names outside Arabic environments  \texttt{\prname} Transliterated proper names inserted in paragraphs of English text should be printed in the same typeface as the surrounding text. \texttt{\prname{\textbackslash (Arabic proper name)}} is provided to that effect:43—

From Wright (1896, i. 23 C)— If the name following ʾān be that of the mother or the grandfather, the ʾ is retained; as \texttt{\arb[fullvoc]{`Is\_A ibnu maryama}}, “Jesus the son of Mary”; \texttt{\arb[fullvoc]{`ammār ibnu man.sUrN}}, “Ammār the (grand)son of Mansūr”.

The following example shows how \texttt{\prname} can be used in conjunction with the \texttt{nameauth} package to have Arabic proper names printed first in full then in partial forms:44—

\begin{verbatim}
1 \begin{nameauth}
2 \texttt{\textless\ Hunayn & \prname{'abU zayd} & \prname(.hunayn), \prname{i}bn
3 ʾis.h.aq al-\textasciitilde{}ibAdiyy & > %
4 \prname{\textbackslash (Arabic proper name)}} & \texttt{Razi & \textless\ \prname{'abU bakr mu.hammad ibn zakariyyA'} &}
5 \prname{\textbackslash (Arabic proper name)}} & > %
6 \end{nameauth}
7
8 On first occurrence, proper names are printed as \texttt{\textless\ Hunayn, \textless\ Razi.}
9 Then as \texttt{\textless\ Hunayn, \textless\ Razi.}
\end{verbatim}

On first occurrence, proper names are printed as ʾAbū Zayd Ḥunayn ibn ʾIshāq al-ʿIbādī, ʾAbū Bakr Muḥammad ibn Zakariyyā’ ar-Rāzi. Then as Ḥunayn, ar-Rāzi.

\texttt{\prname*} \texttt{\textbackslash prname} Rem. \texttt{arabluatex} also provides \texttt{\prname} which only renders in upright roman style already

---

43Just as \texttt{\uc} \texttt{\prname} is also able to process proper names consisting of several subsequent words.

44See the documentation of \texttt{nameauth} for more details: \url{https://ctan.org/pkg/nameauth}
transliterated proper names without applying any further processing. It is mostly used internally and applied to proper names exported in Unicode to an external selected file.\textsuperscript{45}

\section{Additional note on dmg convention}

According to Brockelmann et al. (1935, p. 6), Arabic ʿrāb may be rendered into dmg in three different ways:
(a) In full: ʿAmrun;
(b) As superscript text: ʿAmrun;
(c) Discarded: Amr.

\verb|\arbup| By default, arabluatex applies rule (b). Once delimited by a set of Lua functions, ʿrāb is passed as an argument on to a \verb|\arbup| command which is set to \verb|\textsuperscript|.

\verb|\NoArbUp| \verb|\ArbUpDflt|\verb|\SetArbUp| Finally, \verb|\SetArbUp|{\textit{formatting directives}} can be used to customize the way ʿrāb is displayed. To take one example, here is how Arabic ʿrāb may be rendered as subscript text:—

\begin{verbatim}
1 \SetArbUp{\textsubscript{#1}}
2 Arabic |dmg| transliteration for \arb{raʾaytu ḍAmīʾan
3 muḥaddamaTaN miʾdanatu-hu}: \arb[trans]{raʾaytu
4 ṭAmīʾan muḥaddamaTaN miʾdanatu-hu.}

\end{verbatim}

Arabic dmg transliteration for ʿrābaytu ḍAmīʾan muḥaddamaTaN miʾdanatu-hu.

As shown in the above example, \#1 is the token that is replaced with the actual tanwīn in the formatting directives of the \verb|\SetArbUp| command.

\section*{ṭrāb boundaries} Every declinable noun (muʿrab) may be declined either with or without tanwīn, viz. munṣarif\textsuperscript{"an} or ḡayr\textsuperscript{"a} munṣarif\textsuperscript{"a}. The former is automatically parsed by arabluatex, whereas the latter has to be delimited with an hyphen, like so:—

munṣarif: muʿallim\textsuperscript{uN} muʿallim\textsuperscript{uN}, kAʿin\textsuperscript{uN} kāʾin\textsuperscript{uN}, kAʿin\textsuperscript{aTuN}
        kāʾin\textsuperscript{uN}, \uc{`amraNU} ʿAmr\textsuperscript{uN}, fata\textsuperscript{uN}, qA. diNI
        qāḍ\textsuperscript{uN}.

ḡayr munṣarif: al-muʿallim-u ḍalāl al-muʿallim\textsuperscript{uN}, kitAb-Ani ḍālib\textsuperscript{aNi},
               raʾsaʾ-ANi raʾsaʾ-ANi, sAriq-Una sāriq\textsuperscript{uN}, qA. d-Una ṣād\textsuperscript{uN},
               al-.zulm-Atu ṣād zulm\textsuperscript{aNi}.

\textsuperscript{45}See below section 12 on page 57 for more details.
Rem. a. As the tanwīn is passed over in pronunciation when it is followed by the letters ر، ل، م، و، ي (see item (b) on page 18), it may be desirable to further distinguish it by putting it above the line, but not to do the same for ġayr munṣarif terminations. This can be achieved by simply omitting the hyphen before any ġayr munṣarif termination:—

\[
\text{kāna} \cdot \text{ganiyyān} l_{\text{a}} \text{akinna-hu labisa} \cdot \text{ğubbat} \cdot \text{mumazzaq} \cdot \text{aydu-hā}.
\]

Rem. b. Although the hyphen before the tanwīn is optional as arbluatex always parses nouns with such termination, it may also be used to mark better the inflectional endings:—

\[
\text{mana`} \cdot \text{l-nās} \cdot \text{a-kaffat} \cdot \text{min mu}_h \cdot \text{tabati-hi} \cdot \text{a-had-uñ bi-sayyidi-nā}.
\]

Discarding the Ţrab As said above (item (e) on the previous page), the ţrab may be discarded in some cases, as in transliterated proper names or book titles. arbluatex is able to render words ending with tāʾ marbūṭah in different ways, depending on their function:—

(a) Nouns followed by an adjective in apposition: madiNaT kabIraT madiñah kabīrah, al-madiñah al-kabīrah.

(b) Nouns followed by another noun in the genitive (contract state): .hikmaT al-l_ah ḥikmat Allāh, fi.d.daT al-darāhim fiḍḍat ad-darāhim.

Uncertain short vowels In some printed books, it may happen that more than one short vowel be placed on a consonant in cases where the vocalization is uncertain or ambiguous, like so: َلَُِعَف. In transliteration, the uncertain vowels go between slashes and are separated by commas: faʿ/u,a,i/la.

8.2 Examples

Here follows in transliteration the story of Ḟuḥā and his donkey (ُهُراَمِحَواَحُج). See the code on page 8:—

\(\text{\textquoteleft dmg\textquoteright standard: atā ṣadīq} \cdot \text{ilā Ḟuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fi saʃraɪ} \cdot \text{gaʃrā} \cdot \text{fa-gāla la-hu: \textquoteleft sawfa uʾida-hu ilay-ka fi l-masāʾi wa-udfa }\cdot \text{la-ka uyṯrat} .\)

\(\text{\textquoteleft loc\textquoteright standard: atā ṣadīqun ilā Ḟuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fi saʃraɪ} \cdot \text{gaʃrā} \cdot \text{fa-gāla la-hu: \textquoteleft sawfa uʾida-hu ilay-ka fi l-masāʾi wa-udfa }\cdot \text{la-ka uyṯrat} .\)
Even though arabluatex is primarily designed to process the ArabTeX notation, it can also process the Buckwalter input scheme to a large extent. The Buckwalter scheme is actually processed in two steps, as it is first converted into ArabTeX. Then, once this is accomplished, the ArabTeX scheme is processed through the above described functions. In this way, the Buckwalter input scheme can make the most of the arabluatex special features that are presented in section 2.2 on page 5.

The input scheme, which is set to arabtex by default, may be changed at any point of the document by the \SetInputScheme{⟨scheme⟩} command, where ⟨scheme⟩ may be either arabtex or buckwalter. This command is also accepted in the preamble should one wish to set the input scheme globally, like so:

\begin{verbatim}
1 \usepackage{arabluatex}
2 \SetInputScheme{buckwalter}
\end{verbatim}

‘base’, ‘xml’ and ‘safe’ schemes arabluatex can use any of the so-called Buckwalter ‘base’, ‘xml’ or ‘safe’ schemes as they are described in Habash (2010, pp. 25–26). However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode ٱ and ؤ, must be replaced with square brackets viz. [ and ] respectively.

It is therefore recommended to use the Buckwalter ‘safe’ scheme.

Table 9 gives the Buckwalter equivalents that are currently used by arabluatex. The additional characters that are defined in table 6 on page 27 are also available.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>Buckwalter notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>\</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>A</td>
</tr>
<tr>
<td>loc arabica</td>
<td>base/xml</td>
<td>safe</td>
</tr>
</tbody>
</table>

Table 9: Buckwalter scheme

---

46 See [http://www.qamus.org/transliteration.htm](http://www.qamus.org/transliteration.htm)
47 I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in arabluatex.
48 See section 8 on page 40.
<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>Buckwalter notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>بـ</td>
<td>b b b</td>
<td>b b</td>
</tr>
<tr>
<td>رـ</td>
<td>t t t</td>
<td>t t</td>
</tr>
<tr>
<td>ٍ</td>
<td>ṭ th ṭ</td>
<td>v v</td>
</tr>
<tr>
<td>١</td>
<td>ḏ ḏ ḏ</td>
<td>j j</td>
</tr>
<tr>
<td>ه</td>
<td>h h h</td>
<td>H H</td>
</tr>
<tr>
<td>١</td>
<td>ḥ ḥ ḥ</td>
<td>x x</td>
</tr>
<tr>
<td>د</td>
<td>d d d</td>
<td>d d</td>
</tr>
<tr>
<td>ذ</td>
<td>dh dh dh</td>
<td>* V</td>
</tr>
<tr>
<td>ر</td>
<td>r r r</td>
<td>r r</td>
</tr>
<tr>
<td>ز</td>
<td>z z z</td>
<td>z z</td>
</tr>
<tr>
<td>س</td>
<td>s s s</td>
<td>s s</td>
</tr>
<tr>
<td>š</td>
<td>sh š</td>
<td>$ c</td>
</tr>
<tr>
<td>س</td>
<td>s s</td>
<td>s S</td>
</tr>
<tr>
<td>د</td>
<td>d d d</td>
<td>D D</td>
</tr>
<tr>
<td>ِ</td>
<td>ḫ ḫ ḫ</td>
<td>T T</td>
</tr>
<tr>
<td>ز</td>
<td>ḥ ḥ ḥ</td>
<td>E E</td>
</tr>
<tr>
<td>َ</td>
<td>ǧ ǧ ǧ</td>
<td>g g</td>
</tr>
<tr>
<td>q</td>
<td>q q q</td>
<td>q q</td>
</tr>
<tr>
<td>k</td>
<td>k k k</td>
<td>k k</td>
</tr>
<tr>
<td>l</td>
<td>l l l</td>
<td>l l</td>
</tr>
<tr>
<td>m</td>
<td>m m m</td>
<td>m m</td>
</tr>
<tr>
<td>n</td>
<td>n n n</td>
<td>n n</td>
</tr>
<tr>
<td>h</td>
<td>h h h</td>
<td>h h</td>
</tr>
<tr>
<td>w</td>
<td>w w w</td>
<td>w w</td>
</tr>
<tr>
<td>y</td>
<td>y y y</td>
<td>y y</td>
</tr>
<tr>
<td>ā</td>
<td>ā ā ā</td>
<td>Y Y</td>
</tr>
<tr>
<td>َ</td>
<td>ah ah a</td>
<td>p p</td>
</tr>
</tbody>
</table>

Table 9: Buckwalter scheme
Table 9: Buckwalter scheme

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>Buckwalter notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>an</td>
<td>an</td>
<td>F</td>
</tr>
<tr>
<td>un</td>
<td>un</td>
<td>N</td>
</tr>
<tr>
<td>in</td>
<td>in</td>
<td>K</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>o</td>
</tr>
<tr>
<td>ā</td>
<td>ā</td>
<td>ā</td>
</tr>
<tr>
<td>(tawālī)</td>
<td>—</td>
<td>e</td>
</tr>
</tbody>
</table>

Transliteration The Buckwalter notation can also be transliterated into any accepted romanization standard of Arabic. See above section 8 on page 40 for more information. However, it should be pointed out again that only accurate coding produces accurate transliteration. It is therefore at the very least highly advisable to use the hyphen for tying the definite article and the inseparable particles (viz. prepositions, adverbs and conjunctions) to words, like so:

\[
\text{Al-EaAlamu} \text{ al-ālam}^{\text{u}}, \text{Al-camsu} \text{ aš-šams}^{\text{u}}, \text{bi-SinaAEapi}
\]

\[
\text{Al-T~ib~i} \text{ bi-ṣināʿat} \text{ i } \text{ṭ-ṭibb}^{\text{i}}.
\]

\[
\text{wa-Al-l~ehi} \text{ wa-l-lāh}^{\text{i}}, \text{Al-Hamdu li-l~ehi} \text{ al-ḥamd}^{\text{u}} \text{ li-l[lāh}^{\text{i}}.
\]

Similarly, it is not advisable to use | and { ('base' and 'xml' schemes) or M and L ('safe' scheme) to encode the 'alif' 'l-mamdūdat' and the 'alif' 'l-waṣl' for such signs are supposed to be generated by arabluatex internal functions. Besides, as they do not per se convey any morphological information on what they are derived from, they cannot be transliterated accurately. To take one example, \text{<ilY Al-LntiqADi} gives \text{ālīn} \text{a-ḍā} as expected, but only \text{<iY Al-intiqADi} can be transliterated as \text{'ilā 'l-intiqādī} with the correct vowel (i) in place of the 'alif' 'l-waṣl'.

10 Unicode Arabic input

As said above in section 9 on page 45 about the Buckwalter input scheme, even though arabluatex is primarily designed to process the ArabTeX notation, it also accepts Unicode Arabic input. It should be noted that arabluatex does in no way interfere with Unicode Arabic input: none of the voc, fullvoc, novoc or trans options will have any effect on plain Unicode Arabic for the time being.

That said, there are two ways of inserting Unicode Arabic:
11 \LaTeX{} Commands in Arabic environments

**General principle** \LaTeX{} commands are accepted in Arabic environments. The general principle which applies is that any single-argument command with up to two optional arguments—that is: `\command\[⟨opt1⟩\][⟨opt2⟩]{⟨arg⟩}`—such as `\emph\{⟨text⟩\}`, `\textbf\{⟨text⟩\}` and the like, is assumed to have Arabic text in its mandatory argument:

```
\abjad\{45\} kitAbu-hu \emph{fI 'l-\uc{`AdAt-i}}
```

The same applies to footnotes:

```
\begin{arab}
\text{اَنَّ أَبِي كَانَ مِنَ الْمَعَاطِيَةِ، وَكَانَ أَمِي مِنْ عَظِيمَةٍ بَوْعُ الْرَّماَضَةِ.}
\text{المَعَاطِيَةُ: الْمَنْفَعَةُ.}
\text{الْرَّماَضَةُ: طَالِبَةُ مِنَ الْفِرْسِ.}
\end{arab}
```

Some commands, however, do not expect running text in their arguments, or one may wish to insert English text e.g. in footnotes or in marginal notes. \texttt{arabluatex} provides a set of commands to handle such cases.

---

49 This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.
50 `\rframebox` has been adapted from `\framebox` for insertions of right-to-left text.
\LR \LR\{\langle arg\rangle\} is designed to typeset its argument from left to right. It may be used in an Arabic environment, either \arb{\langle Arabic text\rangle} or \begin{arab} \langle Arabic text\rangle \end{arab}, for short insertions of left-to-right text, or to insert any \LaTeX\ command that would otherwise be rejected by arablatex, such as commands the argument of which is expected to be a dimension or a unit of measurement.

\RL \RL\{\langle arg\rangle\} does the same as \LR\{\langle arg\rangle\}, but typesets its argument from right to left. Even in an Arabic environment, this command may be useful.

\LR\footnote{\langle text\rangle} and \RL\footnote{\langle text\rangle} typeset left-to-right and right-to-left footnotes respectively in Arabic environments. Unlike \footnote{\langle text\rangle}, the arguments of both \LR\footnote{\langle text\rangle} and \RL\footnote{\langle text\rangle} are not expected to be Arabic text. For example, \LR\footnote{\langle text\rangle} can be used to insert English footnotes in running Arabic text:

\begin{verbatim}
1 \begin{arab}[fullvoc]
2 \uc{z}ayd-uN\arbnull\begin{arab}ibnu\end{arab}\LRfootnote{\
3 \enquote{\arb{\langle trans\rangle}{\uc{z}ayd} is the son of \arb{\langle trans\rangle}{\uc{a}mr}}: the second noun is not in ap\n4 \pred\textsuperscript{\LR\arabic{footnote}}\arbnull{\uc{a}mr-iNU\arbnull{ibnu-u\uc{a}mr\end{arab}}
7 \end{arab}
\end{verbatim}

\textquote{Zayd is the son of Amr}: the second noun is not in apposition to the first, but forms part of the predicate...

When footnotes are typeset from right to left, it may happen that the numbers of the footnotes that are at the bottom of the page be typeset in the wrong direction. For example, instead of an expected number 18, one may get 81. arablatex is not responsible for that, but should it happen, it may be necessary to redefine in the preamble the \LaTeX\ macro \thefootnote like so:

\begin{verbatim}
\renewcommand*{\thefootnote}{\textsuperscript{\LR\arabic{footnote}}}
\end{verbatim}

Another solution is to put in the preamble, below the line that loads arablatex, the \FixArbFtnmk command. However, for more control over the layout of footnotes marks, it is advisable to use the \textit{scrextend} package.\footnote{See \url{http://ctan.org/pkg/koma-script}; read the documentation of KOMA-script for details about the \deffootnotemark and \deffootnote commands.}

\LRmarginpar The \LRmarginpar\{\langle left\rangle\}\{\langle right\rangle\} command does for marginal notes the same as \LR\footnote does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that \marginpar also works in Arabic environments, but it acts as any other single-argument command inserted in Arabic environments. The general principle laid on the previous page applies.

\setRL and \setLR can be used to change the direction of paragraphs, either form left to right or from right to left. As an example, an easy way to typeset a right-to-left sectional title follows:

\begin{verbatim}
\setRL
\end{verbatim}
11.1 New commands

In some particular cases, it may be useful to define new commands to be inserted in Arabic environments. From the general principle laid on page 48, it follows that any command that is found inside an Arabic environment is assumed to have Arabic text in its argument which arabtex will process as such before passing it on to the command itself for any further processing. As a result of this feature, such a command as:

```
\newcommand{\fvarabic}[1]{\arb[fullvoc]{#1}}
```

will work as expected, but will always output non-vocalized Arabic if it is inserted in a novoc Arabic environment because its argument will have been processed by the novoc rules before the command \fvarabic itself can see it.

The \texttt{\MkArbBreak\{\textit{csv list of commands}\}} command can be used in the preamble to give any command—either new or already existing—the precedence over arabtex inside Arabic environments. It takes as argument a comma-separated list of commands each of which must be stripped of its leading character \, like so:—

```
\MkArbBreak{onecmd, anothercmd, yetanothercmd, ...}
```

For example, here follows a way to define a new command \texttt{\fvred} to distinguish words with a different color and always print them in fully vocalized Arabic:—

```
\MkArbBreak{fvred}
\newcommand{\fvred}[1]{\arbcolor[red]{\arb[fullvoc]{#1}}}
\begin{arab}[voc]
_\textit{tumma} “\textit{intalaqa} _\textit{dU} ‘l-qarn-\textit{ayni} ‘il_\textit{A} ‘ummaT-iN ‘u_\textit{hr}_\textit{A} f\textit{i}
\fvred{((ma.tli}‘l_\textit{=sams-i})) wa-lA bin\textit{A} a la-hum
\textit{yu’amminu-hum mina ‘l} ‘\textit{sams-i}.
\end{arab}
```
It must be noted that the arguments, either optional or mandatory, of commands declared with \MkArbBreak are not to be processed by arabluatex. Therefore, as in the previous example, any of their argument to be rendered in Arabic must be inserted again in \arb. These commands themselves may have up to two optional and/or mandatory arguments followed by one optional argument, like so:—

(a) \command (no argument, lowermost combination)
(b) \command[{opt1}] (one optional argument)
(c) \command{⟨arg1⟩} (one mandatory argument)
(d) \command[{opt1}]{⟨arg1⟩} (one optional and one mandatory argument)
(e) [...]  
(f) \command[{opt1}]{⟨opt2⟩}{⟨arg1⟩}{⟨arg2⟩}
(g) \command[{opt1}]{⟨opt2⟩}{⟨arg1⟩}{⟨arg2⟩}{⟨opt3⟩} (uppermost combination)

As said above, \MkArbBreak prevents arabluatex from processing the arguments of ‘declared’ commands as Arabic text. This technique proves sufficient in most cases. However, a ‘starred’ version of this command—\MkArbBreak*—is also provided. It goes a step further, as it directs arabluatex to close the current Arabic environment before any of the ‘declared’ commands, then resume it just after.

It must be noted that \MkArbBreak* must be used with the utmost care and should never be used if \MkArbBreak gives satisfaction. At any rate, the latter must always be tested before the former.

11.2 Environments

Environments such as \begin{quote} ... \end{quote} may be nested inside the arab environment. Up to one optional argument may be passed to each nested environment, like so:—

In the following example, the quoting package is used:—

1 \setquotestyle(arabic)
2 \begin{arab}
3 \begin{quote}
4 kÀna \uc(‘abU) \uc(‘l-hu_dayli) ‘ahd_A ‘il_A \uc(muwaysiN)
5 dajÀjatu-hu ‘llatI ‘ahdA-hA dÜna mÀ kÀna
6 yuttaxa_du li-\uc(muwaysiN). wa-l_akinna-hu bi-karami-hi
7 wa-bi-.husni xuluqi-hi ‘a.zhara ‘l-ta’aJJuba min simani-hA
8 wa-.tIbI la.hmi-hA. wa-kÀna \uc(‘abU) \uc(‘l-hu_dayli)}
9 yu’rafu bi-’l-‘imÂkI ‘l-‘sadldi. fa-qÀla: \endquote \begin{quote}
10 ra’ayta yA \uc(‘abA) \uc(‘imÀna) tilka ‘l-dajÀjatu-hu? qÀla:
Lists environments are also accepted inside the arab environment. One may either use any of the three standard list environments, viz. itemize, enumerate and description or use packages that provide additional refinements such as paralist or enumitem.

To take a first example, should one wish to typeset a list of manuscripts, the description environment can be used like so:—

\begin{description}
  \item[b] max.tU.tu 'l-maktabaTi 'l-ahliyyaTi bi-\arbiy{bArIs} 2860
  \item[s] max.tU.tu 'l-maktabaTi 'l-ahliyyaTi bi-\arbiy{bArIs} 2859
  \item[m] max.tU.tu majlisi \arbiy{sUrAY malY} .tahr\'Ana 521.
\end{description}
As a second example, the contents of a treatise may be typeset with the standard list environments, like so:

\begin{itemize}
  \item \textbf{al-fanu} 'l-'awwalu fI 'l-.tibbi
  \item wa-maw.dU'Ati-hi mina 'l-'umUri 'l-.tabl'iyyaTl wa-ya`stamilu
  \item `al_A sittaTi ta`AllimiN
  \item [wa-huwa fa.slAni]
  \item [wa-ya^stamilu]
  \item [al_A sittaTi ta`]
  \item \textbf{al-fa.sl} [wa-huwa fa.slAni]
\end{itemize}

As a third example, abjad-numbered lists can be typeset in conjunction with the \texttt{enumitem} package,\footnote{See the documentation of \texttt{enumitem} for more details: \url{https://ctan.org/pkg/enumitem}} like so:

\begin{itemize}
  \item \textbf{fa`}ala
  \item fa`ala
  \item fA`ala
  \item 'af`ala
\end{itemize}
From Wright (1896, i. 29 B–C):— The derived forms of the triliteral verb are usually reckoned fifteen in number, but the learner may pass over the last four, because (with the exception of the twelfth) they are of very rare occurrence.

Caveat  The various French definition files of the babel package viz. acadian, canadien, francais, frenchb or french all redefine the list environments, which breaks the standard definition file that is used by arabluatex. Therefore, babel-french must be loaded with the StandardLists=true option, like so:—

This option will prevent babel-french from interfering with the layout of the document. Then the paralist or enumitem packages can be used to make the lists ‘compact’ as babel-french do.

11.3 csquotes

The recommended way of inserting quotation marks in running Arabic text is to use csquotes. With the help of the \DeclareQuoteStyle command, one can define an Arabic style, like so:—
\usepackage{csquotes}
\DeclareQuoteStyle{arabic}{\textquotedblright}{\textquotedblleft}{\textquoteright}{\textquoteleft}

Then, use this newly defined style with \setquotestyle, like so:

\setquotestyle{arabic}
\begin{arab}
fa-qAla la-hu ju.hA: \enquote{.garIb-uN \textquoteleft amru-ka \textquoteleft a-tu.saddiqu \textquoteleft l-.himAr-a wa-tuka_d_diba-nI?}
\end{arab}
\setquotestyle{english}

Rem. Do not forget to set back the quoting style to its initial state once the Arabic environment is closed. See the last line in the code above.

11.4 Two-argument special commands

textcolor The two-argument command \textcolor{⟨color⟩}{⟨Arabic text⟩} is supported inside \begin{arab} ... \end{arab}. One simple example follows:\textcolor{red}{—}

\begin{arab}
\textcolor{red}{\uc{m}uha_d_dabu \uc{m}u.hammadiN \uc{m}u.dIliyyiN \textquoteleft l-
\textquoteleft a.limAmu \textquoteleft l-.sadru \textquoteleft l-kabIru \textquoteleft l-.Alimu \textquoteleft l-fA.dilu \uc{m}u.wa-yu\textquoteleft rafu bi-\textquoteleft l-d}a.hwari.
\end{arab}

\textcolor{red}{arbcolor} provides its own \arbcolor command which is able to render syllabes or diacritics in colors. See section 7 on page 35.

reledmac  The two-argument command \edtext{(lemma)}{(commands)} is supported inside \begin{arab} ... \end{arab}.

11.5 quran

arabluatex is compatible with the quran package so that both can be used in conjunction with one another for typesetting the Qurʾān. As quran draws the text of the Qurʾān from a Unicode encoded database, its commands have to be passed as arguments to the \txarab command for short insertions in left-to-right paragraphs, or inserted inside the txarab environment for typesetting running paragraphs of Qurʾānic text (see above section 10 on page 47 for more details). Please note that arabluatex takes care of formatting the Arabic: therefore, it is recommended to load the quran package with the nopar option, after arabluatex itself has been loaded, like so:

\begin{verbatim}
\usepackage{arabluatex}
\usepackage[nopar]{quran}
\end{verbatim}

As an example, the following code will typeset the sūrat al-Fātiḥah:

\begin{verbatim}
\begin{txarab}
\quransurah[1]
\end{txarab}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
\begin{permalink}
12 Exporting Unicode Arabic to an external file

arabluatex is able to produce a duplicate of the original .tex source file in which all arabtex or buckwalter strings will have been replaced with Unicode equivalents, either in Arabic script or in any accepted standard of transliteration. Exporting ASCII strings to Unicode while preserving the exact selected global or local options is a fairly complex operation which may requireLuaTEX to be run several times as will be explained below.

12.1 Commands and environments

**export**  
First, arabluatex must be loaded with the **export** global option enabled, like so:—

```
\% preamble
\usepackage[export]{arabluatex}
\% or:
\usepackage[export=true]{arabluatex}
```

Once that is done, compiling the current file will produce a new empty external .tex file with the same preamble as the original file.

**\SetArbOutSuffix**  
By default, _out is appended as a suffix to the external file name. Any other suffix may be set with the command \SetArbOutSuffix{⟨suffix⟩}.

**arabexport**  
Exporting running paragraphs  
Then, the arabexport environment is provided to actually exporting running paragraphs with or without Arabic environments to the external selected file, like so:—

```
\begin{arabexport}
<Running paragraphs of either Arabic or non-Arabic text>
\end{arabexport}
```

arabluatex converts to Unicode and writes to the external file what is found inside Arabic environments. As to non-Arabic text, it is appended untouched to this file, which is formatted as follows:—

(a) Unicode Arabic text, either in Arabic script or in transliteration, is inserted as argument of \txtarb or \txttrans accordingly.

(b) Additionally, Arabic paragraphs may receive \arbpardir, which arabluatex uses to determine the direction of Arabic paragraphs to be set by default, or either \setRL or \setLR depending on what may have been set locally.

(c) Proper names are inserted as arguments of \prname.

---

55 See above on page 6 for more information.
56 See above section 10 on page 47.
57 \txttrans is used internally by several Lua functions to format transliterated Arabic. Therefore, it is not documented.
58 See above on page 49.
59 See above on page 42.
Appending words or commands to the external file only \ArbOutFile[(newline)]\ArbOutFile*[⟨argument⟩] silently exports its argument to the external file. It may take the string newline as an optional argument, in which case a carriage return is appended to the contents of the argument. \ArbOutFile*[⟨newline⟩]{⟨argument⟩} does the same as \ArbOutFile, but also inserts its argument into the current .tex source file.

Exporting Arabic poetry Lines of Arabic poetry are exported as described above on page 29 when the export option that is specific to the arabverse environment is set to true. As a result of this particular feature, arabverse environments must be left outside \begin{arabexport} ... \end{arabexport}.

Please note that inside arabverse environments \bayt is replaced with \bayt*.  

12.2 Nested Arabic environments

The exporting mechanism described above converts only the outermost level of nested Arabic environments. This may be sufficient in some cases, but if nested Arabic environments be found in the original .tex source file, then the Unicode converted file must be opened and compiled in turn, and so on until the innermost Arabic environment be converted and exported. In such cases, arabluatex issues a warning, so that authors do not have to check the entire file that just has been exported:—

1 Package arabluatex Warning: There are still 'arabtex' strings
2 to be converted. Please open ⟨jobname⟩⟨suffix⟩.tex and compile
3 it one more time.

Where ⟨jobname⟩ is the name of the original .tex source file, and ⟨suffix⟩ the suffix appended to the file that is to be opened and compiled again.

12.3 Further processing of Unicode converted files

Unicode files can be further processed by document converters such as John McFarlane’s pandoc. To take here one simple example, here is how file_out.tex can be converted from LuaLaTeX into Open Document format (.odt):—

1 pandoc file_out.tex -s -o file_out.odt

However, specific commands such as \txarb, \txtrans or \prname*, which are not known to pandoc, must be redefined explicitly in the preamble to prevent the converter from gobbling their arguments, like so:—

1 % preamble:
2 % note that 'export' has been removed
3 \usepackage{arabluatex}
4 \renewcommand{\txarb}[1]{\#1}
5 \renewcommand{\txtrans}[1]{\emph{\#1}}
6 \renewcommand{\arbup}[1]{\textsuperscript{\#1}}

See above note 31 on page 29 for more information.
See http://pandoc.org/
% now that \prname{} has been replaced with \prname*{} it should
% be safe to say:
\renewcommand{\prname}{[2] (#2)}
% &c

13 Future work

A short, uncommented, list of what is planned in the versions of \texttt{arabluatex} to come follows:
(a) Short-term:
i. TEI xml support: \texttt{arabluatex} will interoperate with TEI xml through new global and local options that will output Arabic in a TEI xml compliant file in addition to the usual PDF output: see on page 4.
(b) Medium-term:
i. More languages: the list of supported languages will eventually be the same as \texttt{arabtex}: see note 4 on page 4.
ii. Formulate propositions for extending the ArabTEX notation and the transliteration tables. Include them in \texttt{arabluatex}. See section 4.9 on page 27.

14 Implementation

The most important part of \texttt{arabluatex} relies on Lua functions and tables. Read the .lua files that accompany \texttt{arabluatex} for more information.
\begin{verbatim}
\RequirePackage{ifluatex}
\ifluatex
\PackageError{arabluatex}{lualatex needed}{%
Package `arabluatex' needs LuaTeX.\MessageBreak
So you should use `lualatex' to process your document.\MessageBreak
See documentation of `arabluatex' for further information.}%
\expandafter\expandafter\expandafter\csname endinput\endcsname
\else
\fi
\end{verbatim}

Declare the global options, and define them:
\begin{verbatim}
\RequirePackage{xkeyval}
\DeclareOptionX{voc}{\def\al@mode{voc}}
\DeclareOptionX{fullvoc}{\def\al@mode{fullvoc}}
\DeclareOptionX{novoc}{\def\al@mode{novoc}}
\DeclareOptionX{trans}{\def\al@mode{trans}}
\define@boolkey{arabluatex.sty}@pkg@{export}[true]{%
\if@pkg@export%
\AtBeginDocument{\luadirect{arabluatex.openstream()}%\MkArbBreak{@al@ob,@al@cb,@al@cb@sp}}
\else
\fi
\end{verbatim}
Packages that are required by \texttt{arabluatex}:

\begin{itemize}
\item \texttt{RequirePackage{xcolor}}
\item \texttt{requirepackage{luacolor}}
\item \texttt{requirepackage{etoolbox}}
\item \texttt{requirepackage{arabluatex-patch}}
\item \texttt{requirepackage{fontspec}}
\item \texttt{requirepackage{luacode}}
\item \texttt{requirepackage{xparse}}
\item \texttt{requirepackage{adjustbox}}
\item \texttt{requirepackage{xstring}}
\item \texttt{PassOptionsToPackage{normalem}{ulem}}
\item \texttt{requirepackage{ulem}}
\end{itemize}

The following boolean will be set to \texttt{true} in RL mode:

\begin{itemize}
\item \texttt{providebool{al\@rlmode}}
\end{itemize}

Here begins the real work: load \texttt{arabluatex.lua}:

\begin{verbatim}
\luadirect{dofile(kpse.find_file("arabluatex.lua"))}
\end{verbatim}

Font setup. If no Arabic font is selected, issue a warning message and attempt to load the Amiri font which is included in TEXlive:

\begin{verbatim}
\AtBeginDocument{\ifdefined\arabicfont\relax\else
\PackageWarning{arabluatex}{\string\arabicfont\ is not defined.^^J
I will try to load Amiri}%
\newfontfamily\arabicfont[Script=Arabic]{Amiri}\fi}\
\end{verbatim}

This neutralizes what may be defined by other packages:

\begin{verbatim}
\AtBeginDocument{\ifdef{\LR}\
{\RenewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\rmfamily#1\egroup}}
{\NewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\rmfamily#1\egroup}}}
\end{verbatim}

This command typesets its argument from left to right. As \LR may be already defined, we need to redefine for it to suit our purpose:

\begin{verbatim}
\AtBeginDocument{\ifdef{\LR}\
{\NewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\rmfamily#1\egroup}}
{\NewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\rmfamily#1\egroup}}}
\end{verbatim}

This one typesets its argument from right to left. Same remark as above regarding the need of redefinition.
The \texttt{\textbackslash MkArbBreak\{\textit{csv list of commands}\}} command can be used to give any command—either new or already existing—the precedence over \texttt{arabluatex} inside Arabic environments. It is actually coded in Lua.

\texttt{\textbackslash MkArbBreak*} goes a step further as it directs \texttt{arabluatex} to close the current Arabic environment before processing any ‘declared’ command then resume it just after.

\texttt{\textbackslash aemph} Arabic emphasis. Needs to be redefined as well. The function is actually coded in Lua.

The ‘starred’ version of this command always puts the stroke over its argument. As of v1.16 \texttt{arabluatex} uses \texttt{ulem} to render the strokes, thus allowing line breaks and manual hyphenation for transliterated Arabic.

\texttt{\textbackslash arbcolor} \texttt{[\textit{color}]}\{\textit{Arabic text}\} takes the Arabic text to be colored as argument.

\texttt{\textbackslash SetInputScheme} \texttt{arabluatex} is designed for processing Arab{	extsc{Tg}}X input notation. \texttt{\textbackslash SetInputScheme} may be used in the preamble or at any point of the document should the user wish to use a different notation such as the ‘Buckwalter scheme’. 
\SetArbEasy By default, \arabluatex applies complex rules to generate euphonic tašdīd, ʾalif mamdūdah and sukūn depending on the modes which are selected, either voc, fullvoc or trans. Such refinements can be discarded with \SetArbEasy, either globally in the preamble or at any point of the document. Note that \SetArbEasy keeps the sukūn that is generated, while the starred version \SetArbEasy* takes it away. Default complex rules can be set back at any point of the document with \SetArbDflt.

\SetArbDflt* As of v1.6, \arabluatex does not applies any more the assimilation rules that are laid on item (b) on page 18; a new starred version \SetArbDflt* is now available to the user should he wish to apply them.

\SetTranslitFont By default, the font that is used for transliterated text is the main font of the document. Any other font may also be selected with the font-selecting commands of the fontspec package.

\SetTranslitStyle By default any transliterated Arabic text is printed in italics. This can be changed either globally in the preamble or at any point of the document:

\SetTranslitConvention \SetTranslitConvention{⟨convention⟩} can be used to change the transliteration convention, which is dmg by default:

\arbup By default, \textsuperscript{⟨code⟩} is set to \textsuperscript{⟨code⟩}. This is how the tanwīn that takes place at the end of a word should be displayed in dmg mode. \NoArbUp may be used either in the preamble or at any point of the document in case one wishes to have the tanwīn on the line. The default rule can be set back with \ArbUpDflt at any point of the document. Finally \SetArbUp can be used to customize the way tanwīn is displayed; this command takes the formatting directives as argument, like so: \SetArbUp{⟨code⟩}.

\NewDocumentCommand{\al@arbup@dflt}{m}{\textsuperscript{#1}}%
\NewDocumentCommand{\al@arbup}{m}{\al@arbup@dflt{#1}}
\NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
\NewDocumentCommand{\NoArbUp}{}{}
\NewDocumentCommand{\ArbUpDflt}{}{}
\NewDocumentCommand{\SetArbUp}{}{}

62
\uc Proper Arabic names or book titles should be passed to the \uc command so that they have their first letters uppercased. \uc is actually coded in Lua.

\uc \uc can be used safely in all of the modes that are provided by \texttt{arabluatex} as any of the voc, fullvoc and novoc modes discard it on top of any other functions to be run. \uc does the same as \uc except that it is never discarded. For that reason, \uc should never be used outside the trans mode. \texttt{arabluatex} uses \uc internally so as to prevent \uc from being discarded in case words that are to be transliterated are inserted into Arabic commands or environments where transliteration is not required. Therefore, it is not documented.

\let\uc\uc

\prname \prname is to be used outside Arabic environments for proper names. It takes as argument one or more Arabic words, each of which will be rendered in upright roman style with its first letter uppercased.

\prname* Unlike \prname, \prname* does not take \texttt{arabtex} or \texttt{buckwalter} input as argument, but already Unicode converted names and renders them in upright roman style.

\txarb \txarb sets the direction to right-to-left and selects the Arabic font. It is used internally by several Lua functions, but available to the user should he wish to insert utf8 Arabic text in his document.

\txtrans \txtrans is used internally by several Lua functions to insert transliterated Arabic text. Therefore, it is not documented.

\txarab The \txarab environment does for paragraphs the same as \txarb does for short insertions of utf8 Arabic text.
\NewDocumentCommand{\arb}{O{\al@mode} +m}%
{\edef\@tempa{#1}%
  \ifx\@tempa\al@mode@voc%
    \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
    \luadirect{tex.sprint(arabluatex.processvoc(\luastringN{#2},
                                 \luastringO{\al@arb@rules},
                                 \luastringO{\al@input@scheme}))}\egroup%
  \else%
    \ifx\@tempa\al@mode@fullvoc%
      \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
      \luadirect{tex.sprint(arabluatex.processfullvoc(\luastringN{#2},
                                      \luastringO{\al@arb@rules},
                                      \luastringO{\al@input@scheme}))}\egroup%
    \else%
      \ifx\@tempa\al@mode@novoc%
        \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
        \luadirect{tex.sprint(arabluatex.processnovoc(\luastringN{#2},
                                                  \luastringO{\al@arb@rules},
                                                  \luastringO{\al@input@scheme}))}\egroup%
      \else%
        \ifx\@tempa\al@mode@trans%
          \bgroup\textdir TLT\al@trans@font\al@trans@style%
          \luadirect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
                                                        \luastringO{\al@trans@convention},
                                                        \luastringO{\al@arb@rules},
                                                        \luastringO{\al@input@scheme}))}\egroup%
        \else%
          \fi
        \fi
      \fi
  \fi}
\arbmark
\arbmark[⟨rl|lr⟩]{⟨shorthand⟩} takes one argument from a list of defined elements.
The mark to be inserted is determined by contextual analysis or by an optional argument, either rl or lr. This command is coded in Lua.
\NewDocumentCommand{\arbmark}{O{} m}{%
  \bgroup%
  \SetInputScheme{arabtex}%
  \luadirect{tex.sprint(arabluatex.processarbmarks(\luastringN{#2},
                                      \luastringN{#1}))}%
  \egroup}
\newarbmark
\newarbmark lets the user define additional Arabic marks. As \arbmark, this command is coded in Lua. It takes three arguments: the abbreviated form to be used as argument of \arbmark, the rendition in Arabic script and the rendition in romanized Arabic.
\NewDocumentCommand{\newarbmark}{m m m}{%
  \luadirect{arabluatex.newarbmark(\luastringN{#1},
                                      \luastringN{#2},
                                      \luastringN{#3})}}
arab
The arab environment does for paragraphs the same as \arb does for short insertions of Arabic text.
\NewDocumentEnvironment{arab}{O{\al@mode} +b}{%\par\edef\@tempa{#1}%
  \ifx\@tempa\al@mode@voc%
The `arabverse` environment may receive different options: `mode`, `width`, `gutter`, `metre`, `color`, `utf`, `delim` and `export`; all of them are defined here just before the `arabverse` environment.

Then follows the environment itself:

$$\text{arabverse}$$
Each verse consists of two hemistichs; therefore the \bayt command takes two arguments, the first receives the ṣadr and the second the ʿaǧuz. That two subsequent hemistichs should be connected with one another is technically named tādwiř. In some of these cases, the hemistichs may be connected by a prominent horizontal flexible stroke which is drawn by the \alverse@stroke command.

\SetHemistichDelim A hemistich delimiter also may be defined. By default, it is set to the ‘star’ character: *. The \SetHemistichDelim{⟨delimiter⟩} command can be used at any point of the document to change this default setting.
\abjad \abjad\{\textit{number}\} expresses its argument in Arabic letters in accordance with the ʾabǧad arrangement of the alphabet. \textit{number} must be between 1 and 1999. It is now coded in Lua so that polyglossia is no longer needed. See arabluatex.lua for more information.

\ayah \ayah\{\textit{number}\} prints up to 3-digit numbers inside ‘end of Ayah’ sign (U+06DD) or inside parentheses depending on the mode which is selected.

\arbnul The \arbnul command does nothing by itself. It is processed only if it is found in Arabic context so as to put back on contextual analysis in case it has been broken by other commands.
\NewDocumentCommand{\arbnull}{m}{\relax}
\abrases{⟨Arabic text⟩} puts its argument between braces. This macro is written in Lua and is dependent on the current value of tex.textdir.
\NewDocumentCommand{\abrases}{+m}{\luadirect{tex.sprint(arabluatex.abraces(\luastringN{#1}))}}
\LRmarginpar is supposed to be inserted in an Arabic environment. It typsets his argument in a marginal note from left to right.
\DeclareDocumentCommand{\LRmarginpar}{o m}{\IfNoValueTF{#1}{\marginpar{\textdir TLT #2}}{\marginpar\[	extdir TLT #1\]\[	extdir TLT #2\]}}
\LRfootnote and \RLfootnote are supposed to be used in Arabic environments for insertions of non Arabic text. \LRfootnote typesets its argument left-to-right... while \RLfootnote typesets its argument left-to-right.
\DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pardir TLT\textdir TLT\footnote{#1}\egroup}
\DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pardir TRT\textdir TRT\footnote{#1}\egroup}
\FixArbFtnmk In the preamble, just below \usepackage{arabluatex}, \FixArbFtnmk may be of some help in case the footnote numbers at the bottom of the page are printed in the wrong direction. This quick fix uses and loads \texttt{scrextend} if it is not already loaded.
\NewDocumentCommand{\FixArbFtnmk}{}{\@ifpackageloaded{scrextend}{\AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark} .\enskip}}}\RequirePackage{scrextend}\AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark} .\enskip}}}}

Exporting Unicode Arabic to external file
\SetArbOutSuffix By default, \_out is the suffix to be appended to the external file in which arabluatex exports Unicode in place of arabtex or buckwalter strings. Any other suffix may be set with \SetArbOutSuffix{⟨suffix⟩}.
\NewDocumentCommand{\SetArbOutSuffix}{m}{\luadirect{arabluatex.utffilesuffix(\luastringN{#1})}}
\ArbOutFile \ArbOutFile{⟨newline⟩}{⟨string⟩} silently exports ⟨string⟩ to the external selected file. It may take \texttt{newline} as an optional argument in which case a carriage return is appended to \texttt{string}.
\ArbOutFile* \ArbOutFile*{⟨newline⟩}{⟨string⟩} does the same as \ArbOutFile but also inserts ⟨string⟩ in the current .tex source file.
The \texttt{arabexport} environment processes and prints its argument unchanged to the current .pdf file. Additionally, if \texttt{arabluatex} is loaded with the \texttt{export} option, this argument is exported to the external selected .tex file with Unicode in place of the original arubtetx or buckwalter strings.

The \texttt{arab@v@export} environment does for \texttt{arabverse} the same as \texttt{arabexport}. It is used internally by \texttt{arabverse}.

\texttt{arbpardir} is automatically inserted by \texttt{arabluatex} at the beginning of Arabic paragraphs converted to Unicode so that they are printed in the right direction.

\textbf{Errors and Warnings}

- \newcommand{\al@warning}[1]{\PackageWarning{arabluatex}{#1}}
- \newcommand{\al@error}[2]{\PackageError{arabluatex}{#1}{#2}}
- \newcommand{\al@wrong@nesting}{\al@error{(RL/LR)\footnote{space is not allowed}}{Get rid of the surrounding \texttt{\string\\RL} or \texttt{\string\\LR} command.}}
- \newcommand{\al@wrong@mark}{\al@warning{}}
That is it. Say goodbye before leaving.

**Patches**

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{arabluatex-patch} [%
[2016/11/14 v1.0 patches for arabluatex]

I have put in a separate \texttt{.sty} file external lines of code that I had to patch for a good reason. I hate doing this, and hopefully, most of these lines will disappear as soon as they are not required anymore.

The following is taken from \texttt{latex.ltx}. I had to make this patch for I could not find a way to process the list environments in right-to-left mode. The \LaTeX{} primitives \texttt{\bodydir} and \texttt{\pagedir} will eventually allow us to get rid of this:

\begin{verbatim}
def\list#1#2{\ifnum \@listdepth >5\relax\@toodeep\else\global\advance\@listdepth\@ne\fi\rightmargin\z@\listparindent\z@\itemindent\z@\csname @list\romannumeral\the\@listdepth\endcsname\def\@itemlabel{#1}\let\makelabel\@mklab\@nmbrlistfalse\ifhmode\unskip\unskip \par\fi\@trivlist\parskip\parsep\parindent\listparindent\advance\linewidth -\rightmargin\advance\linewidth -\leftmargin\patch begins:\ifbool{al@rlmode}{\advance\@totalleftmargin \rightmargin}\else\advance\@totalleftmargin \leftmargin\patch ends.\parshape \@ne \@totalleftmargin \linewidth\ignorespaces\def\@item[\#1]{}\if\@noparitem\@donoparitem\else\if\@inlabel\indent \par\fi\ifhmode\unskip\unskip \par\fi\unskip}
\end{verbatim}
\fi
\if@newlist
  \ifnobreak
    \@nbitem
  \else
    \addpenalty\@beginparpenalty
    \addvspace@topsep
    \addvspace{-\parskip}\
  \fi
  \else
    \addpenalty\@itempenalty
    \addvspace\itemsep
  \fi
  \global\@inlabeltrue
\fi
\everypar{
\@minipagefalse
\global\@newlistfalse
\if@inlabel
  \global\@inlabelfalse
  \setbox\z@lastbox
  \ifvoid\z@
    \kern-\itemindent
  \fi
}\
\box\@labels
\penalty\z@
\fi
\if@nobreak
  \@nobreakfalse
  \clubpenalty \@M
\else
  \clubpenalty \@clubpenalty
  \everypar{}\
\fi
\if@noitemarg
  \@noitemargfalse
  \if@nmbrlist
    \refstepcounter\@listctr
  \fi
\fi
\if@nobreak
  \@nobreakfalse
  \clubpenalty \@M
\else
  \clubpenalty \@clubpenalty
  \everypar{}\
\fi
\if@noitemarg
  \@noitemargfalse
  \if@nmbrlist
    \refstepcounter\@listctr
  \fi
\fi
\patch begins:
  \ifbool{al@rlmode}{\sRLbox\@tempboxa{\makelabel{#1}}}{%
  \sbox\@tempboxa{\makelabel{#1}}}%
  \ifbool{al@rlmode}{\global\setbox\@labels\hbox dir TRT}{%
    \global\setbox\@labels\hbox}
\patch ends.
\unless\hbox\@labels
  \hskip \itemindent
  \hskip -\labelwidth
This is adapted from Vafa Khalighi’s bidi package. Thanks to him.

References


DIN 31 635 (July 2011). Information and Documentation - Romanization of the Arabic Alphabet for Arabic, Ottoman-Turkish, Persian, Kurdish, Urdu and Pushto. URL: http://www.din.de.


Change History

v1.0.

General: Initial release  ............ 1

v1.0.1.

General: Minor update of the documentation  ............ 1
v1.1.
\abjad: New and more flexible \abjad command. ............. 67

v1.2.
\SetArbEasy: New
\SetArbEasy/\SetArbDflt for ‘modern’ or ‘classic’ Arabic styles. ............... 62

v1.3.
\arbup: ʾiʿrāb is now written as superscript text in dsc mode by default. ............... 62

v1.4.
\SetInputScheme:
\SetInputScheme can be used to process other input schemes such as ‘Buckwalter’ ............... 61
\SetTranslitFont: For selecting a specific font for transliterated texts ......................... 62

v1.4.3.
\abraces: New \abraces command which expresses its argument between braces. ............... 68

v1.4.4.
\SetArbEasy*: this starred version discards the sukūn in addition to what is already discarded by \SetArbEasy. ............... 62

v1.5.
General: Compatibility with the quran package ............... 56
Environments may be nested inside the arab environment ... 51
\txarab: New \txarab environment for typesetting running paragraphs in Unicode Arabic . 63

v1.6.
\arabverse: New environment \arabverse for typesetting Arabic poetry ............... 65
\bayt: New macro \bayt for typesetting each verse inside the \arabverse environment .... 66
\SetArbDflt*: This starred version applies the assimilation rules in addition to what \SetArbDflt already does. ............... 62
\SetHemistichDelim: New
\SetHemistichDelim command for changing the default delimiter between hemistichs ............. 66

v1.7.
\arbnull: New \arbnull command for putting back on any contextual analysis rule broken by other commands. ............... 67

v1.8.
General: \arabica transliteration standard is now supported ... 40

v1.8.5.
General: Six additional Persian characters are now available ... 11

v1.9.
\\MkArbBreak: New \MkArbBreak command for inserting user-defined macros in Arabic environments ............... 61
\\aemph*: Starred version which always puts the stroke over its argument ............... 61

v1.10.
General: \uc supersedes \cap .... 42
\prname: New command for typesetting Arabic proper names in transliteration ............... 63

v1.11.
\arbmark: New command for inserting additional marks in Arabic environments ............... 64
\\newarbmark: Allows defining additional sets of Arabic marks . 64

v1.12.
General: \abjad can now process \LaTeX counters .......... 27
\arbcolor: Standard color command for Arabic environments ............... 61
\\MkArbBreak*: ‘starred’ version which closes Arabic environments before processing declared commands. ............... 61

v1.13.
\arabexport: Processes and print its argument in the current file and exports it in full Unicode in the external selected .tex file. ... 69
arabverse: New options color and export to arabverse environment. .......... 65
\arbmark: New optional argument: either rl or lr ................. 64
\ArbOutFile: Silently exports its argument in the selected external file. ........ 68
\arbpardir: Sets the direction of Arabic paragraphs once they are converted to Unicode. ...... 69
prname*: Renders proper names already converted to Unicode in upright roman style ........ 63
\SetArbOutSuffix: Sets a suffix to be appended to the filename of the external Unicode file. .... 68
v1.15.
\ayah: Prints End of Ayah sign ... 67
v1.16.
\aemph: Now uses ulem ............ 61

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols
\@oline ........... 58, 60 \@tempa ...... 116, 117, \@arbup ............ 95, 96, 97, 98, 100
\@M .............. 393 \@tempboxa ....... 404, \@arbup@dflt 94, 95, 97
\@beginparpenalty . 369 \@bays ... 163, 169, 240, 241 \@bays@width ....
\@clubpenalty ..... 395 \@toodeep ........ 334 \@error .... 321, 322
\@donoparitem ...... 357 \@topsep ...... 370 \@gutter@width ....
\@listctr ...... 388, 406, 407, 408 \@toodeepboxa .... 417, \@hemistich@delim 326, 327, 328
\@labels ........ 388, 406, 407, 408 \@trivlist ...... 346 \@input@scheme ...
\@listctr ...... 401 \@advance ...... 177, 179,
\@listdepth 333, 336, 341 \@aspenalty ... 179, 180, 233, 253
\@minipagetrue ... 380 \@aspace .... 181, 235, 237,
\@mklab .......... 343 \@trivlist .... 243, 245, 249, 251
\@nbitem ........ 367 \@item ........ 226, 227, 235
\@nbitemfalse ... 342 \@itemoffset .......... 40 \@input@scheme ..
\@labels ........ 388, 406, 407, 408 \@item .... 76, 77, 120,
\@listctr ...... 401 \@item .... 125, 130, 137,
\@listdepth 333, 336, 341 \@item ...... 155, 161, 167, 174
\@minipagetrue ... 380 \@mode .......... 122, 127, 132,
\@mklab .......... 343 \@mode .... 150, 151, 157,
\@nbitem ........ 367 \@mode@dflt 94, 95, 97
\@one .......... 336, 353 \@mode ........ 163, 169, 240, 241
\@newlistfalse ... 381 \@mode ...... \@bayt@width ....
\@nbitemfalse ... 344 \@mode@fullvoc ...
\@nobreakfalse ... 392 \@mode ........ 23, 122, 157
\@nointeritemfalse 399 \@mode ........ 24, 127, 163
\@nointeritemtrue 374 \@mode ........ 25, 132, 169,
\@line ........... 58, 59 \@mode ........ 25, 132, 169,
\abjad .......... 26, 256 \@mode ........ 82, 85, 86, 120,
\abvules .. 26, 256 \@mode ........ 125, 130, 136,
\abraces ....... 33, 273 \@mode ........ 155, 161, 167, 173
\addspace .... 370, 371, 375 \@mode ........ 211, 242, 248, 319
\addvspace ... 370, 371, 375 \@mode ........ 211, 242, 248, 319
\advance ...... 336, \@mode ........ 23, 122, 157
 \@mode ........ 349, 350, 351, 352
\@mode ........ 24, 127, 163
\@mode ........ 24, 127, 163
\@mode ........ 25, 132, 169,
<table>
<thead>
<tr>
<th>Package/Command</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\RequirePackage</td>
<td>1, 9, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 287</td>
</tr>
<tr>
<td>\rightmargin</td>
<td>338, 349, 351</td>
</tr>
<tr>
<td>\RL</td>
<td>49, 50, 324, 325</td>
</tr>
<tr>
<td>\RLfootnote</td>
<td>49, 279</td>
</tr>
<tr>
<td>\romannumeral</td>
<td>341</td>
</tr>
<tr>
<td>\savenotes</td>
<td>231</td>
</tr>
<tr>
<td>\setLR</td>
<td>45, 49, 206, 209, 211, 319</td>
</tr>
<tr>
<td>\setRL</td>
<td>43, 49, 206, 207, 209, 211, 319</td>
</tr>
<tr>
<td>\SetArbDflt</td>
<td>7, 78</td>
</tr>
<tr>
<td>\SetArbDflt*</td>
<td>7, 78</td>
</tr>
<tr>
<td>\SetArbEasy</td>
<td>7, 78</td>
</tr>
<tr>
<td>\SetArbEasy*</td>
<td>7, 78</td>
</tr>
<tr>
<td>\SetArbOutSuffix</td>
<td>57, 290</td>
</tr>
<tr>
<td>\SetArbUp</td>
<td>43, 94</td>
</tr>
<tr>
<td>\Setbox</td>
<td>384, 406, 407, 419</td>
</tr>
<tr>
<td>\SetHemistichDelim</td>
<td>29, 224, 227</td>
</tr>
<tr>
<td>\SetInputScheme</td>
<td>45, 76, 142</td>
</tr>
<tr>
<td>\setkeys</td>
<td>192, 204, 216, 219, 308</td>
</tr>
<tr>
<td>\setlength</td>
<td>178, 179, 180, 181</td>
</tr>
<tr>
<td>\setLR*</td>
<td>7, 78</td>
</tr>
<tr>
<td>\setRL*</td>
<td>7, 78</td>
</tr>
<tr>
<td>\settranslitconvention</td>
<td>40, 91</td>
</tr>
<tr>
<td>\SetTranslitFont</td>
<td>41, 87</td>
</tr>
<tr>
<td>\SetTranslitStyle</td>
<td>41, 89, 105</td>
</tr>
<tr>
<td>\space</td>
<td>323</td>
</tr>
<tr>
<td>\spevnotes</td>
<td>254</td>
</tr>
<tr>
<td>\sRLbox</td>
<td>404, 419</td>
</tr>
<tr>
<td>\SetArbDflt</td>
<td>7, 78</td>
</tr>
<tr>
<td>\SetArbDflt*</td>
<td>7, 78</td>
</tr>
<tr>
<td>\SetArbEasy</td>
<td>7, 78</td>
</tr>
<tr>
<td>\SetArbEasy*</td>
<td>7, 78</td>
</tr>
<tr>
<td>\SetArbOutSuffix</td>
<td>57, 290</td>
</tr>
<tr>
<td>\SetArbUp</td>
<td>43, 94</td>
</tr>
<tr>
<td>\Setbox</td>
<td>384, 406, 407, 419</td>
</tr>
<tr>
<td>\SetHemistichDelim</td>
<td>29, 224, 227</td>
</tr>
<tr>
<td>\SetInputScheme</td>
<td>45, 76, 142</td>
</tr>
<tr>
<td>\setkeys</td>
<td>192, 204, 216, 219, 308</td>
</tr>
<tr>
<td>\setlength</td>
<td>178, 179, 180, 181</td>
</tr>
</tbody>
</table>

**Notes:**
- \textcolor: \textcolor \textdir: \textdir \textsuperscript: \textsuperscript \textwidth: \textwidth \thefootnotemark: \thefootnotemark \txarab: \txarab \txtrans: \txtrans \U: \U \uc: \uc \voc: \voc \wd: \wd \xcolor: \xcolor \xleaders: \xleaders \z@: \z@