# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</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 LuaL</td>
<td>4</td>
</tr>
<tr>
<td>2 The basics of arabluatex</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Activating arabluatex</td>
<td>4</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>5</td>
</tr>
<tr>
<td>2.3 Typing Arabic</td>
<td>7</td>
</tr>
<tr>
<td>2.3.1 Local options</td>
<td>8</td>
</tr>
<tr>
<td>3 Standard ArabTEX input</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Consonants</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Additional characters</td>
<td>9</td>
</tr>
<tr>
<td>3.3 Vowels</td>
<td>10</td>
</tr>
<tr>
<td>3.3.1 Long vowels</td>
<td>10</td>
</tr>
<tr>
<td>3.3.2 Short vowels</td>
<td>11</td>
</tr>
<tr>
<td>4 arabluatex in action</td>
<td>11</td>
</tr>
<tr>
<td>4.1 The vowels and diphthongs</td>
<td>11</td>
</tr>
<tr>
<td>4.2 Other orthographic signs</td>
<td>13</td>
</tr>
<tr>
<td>4.2.1 Other orthographic signs</td>
<td>13</td>
</tr>
<tr>
<td>4.2.2 Other orthographic signs</td>
<td>13</td>
</tr>
<tr>
<td>4.2.3 Other orthographic signs</td>
<td>13</td>
</tr>
<tr>
<td>4.3 Special orthographies</td>
<td>18</td>
</tr>
<tr>
<td>4.3.1 Special orthographies</td>
<td>18</td>
</tr>
<tr>
<td>4.4 Quoting</td>
<td>19</td>
</tr>
<tr>
<td>4.4.1 Quoting the hamzah</td>
<td>20</td>
</tr>
<tr>
<td>4.5 The ‘pipe’ character</td>
<td>21</td>
</tr>
<tr>
<td>4.6 Putting back on broken contextual analysis rules</td>
<td>21</td>
</tr>
<tr>
<td>4.7 Stretching characters: the šī‘</td>
<td>23</td>
</tr>
<tr>
<td>4.8 Digits</td>
<td>23</td>
</tr>
<tr>
<td>4.8.1 Numerical figures</td>
<td>23</td>
</tr>
<tr>
<td>4.8.2 The abjad</td>
<td>23</td>
</tr>
<tr>
<td>4.9 Additional characters</td>
<td>24</td>
</tr>
<tr>
<td>4.10 Arabic emphasis</td>
<td>24</td>
</tr>
<tr>
<td>5 Arabic poetry</td>
<td>25</td>
</tr>
<tr>
<td>5.1 Example</td>
<td>28</td>
</tr>
<tr>
<td>6 Special applications</td>
<td>30</td>
</tr>
<tr>
<td>6.1 The Qurʿān</td>
<td>32</td>
</tr>
<tr>
<td>7 Color</td>
<td>33</td>
</tr>
<tr>
<td>7.1 Tricks of the trade</td>
<td>34</td>
</tr>
<tr>
<td>7.2 Diacritics</td>
<td>35</td>
</tr>
<tr>
<td>7.3 dashes</td>
<td>36</td>
</tr>
<tr>
<td>7.4 The definite article and the euphonic tašdīd</td>
<td>36</td>
</tr>
<tr>
<td>8 Transliteration</td>
<td>37</td>
</tr>
<tr>
<td>8.1 Additional note on dng convention</td>
<td>39</td>
</tr>
<tr>
<td>8.2 Examples</td>
<td>41</td>
</tr>
<tr>
<td>9 Buckwalter input scheme</td>
<td>41</td>
</tr>
<tr>
<td>9.1 Additional note on dng convention</td>
<td>42</td>
</tr>
<tr>
<td>10 Unicode Arabic input</td>
<td>43</td>
</tr>
<tr>
<td>11 LATEX Commands in Arabic environments</td>
<td>44</td>
</tr>
<tr>
<td>11.1 New commands</td>
<td>46</td>
</tr>
<tr>
<td>11.2 Environments</td>
<td>47</td>
</tr>
<tr>
<td>11.2.1 Lists</td>
<td>47</td>
</tr>
<tr>
<td>12 Caveat</td>
<td>49</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. Furthermore, it can interact with the ekdosis package 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

OpenBSD arabluatex is licensed under the terms of the so-called OpenBSD license, as it is modelled after the ISC copyright, which is functionally equivalent to a two-term BSD copyright with language removed that is made unnecessary by the Berne convention.²

²More information about the OpenBSD policy to which arabluatex adheres: https://www.openbsd.org/policy.html.
MERCHANDABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Please send error reports and suggestions for improvements to Robert Alessi:
  – email: mailto:alessi@roberalessi.net
  – website: https://sr.ht/~ralessi/arabluatex/
  – development: http://git.robertalessi.net/arabluatex
  – comments, feature requests, bug reports: https://todo.sr.ht/~ralessi/arabluatex

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

1 Introduction

In comparison to Prof. Lagally’s outstanding ArabT\TeX,\(^2\) ArabLuaT\TeX 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 LuaLa\TeX\ users.

\(\text{arabtex}\) dates back to 1992. As far as I know, it was then the first and only way to typeset
Arabic texts with T\TeX and \La\TeX. To achieve that, \(\text{arabtex}\) provided—and still does—an
Arabic font in \textit{Nas\ifmmode\textstyle\hbox{\vphantom{h}}\else\vphantom{t}\fi\hbox{\vphantom{h}}}\ 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”\(^3\). Even if the same can be said about Unicode,
ArabT\TeX 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 particulary abundant. It must also be said
that most text editors have trouble in displaying Arabic script connected with preceding
or following \La\TeX 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
\texttt{LEFT-TO-RIGHT} or \texttt{RIGHT-TO-LEFT MARKS (U+200E, U+200F)}, RTL/LTR “embed”
characters (\texttt{U+202B, U+202A}) and RLO/LRO “bidi-override” characters (\texttt{U+202E, U+202D})\(^4\).
Nonetheless, it remains that inserting all the time these invisible characters in complex
documents rapidly becomes confusing and cumbersome.

The great advantage of ArabT\TeX 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. ArabT\TeX notation is a way of encoding Arabic language,

\(^2\)\texttt{See http://ctan.org/pkg/arabtex}
\(^3\)Klaus Lagally, \textit{ArabT\TeX: Typesetting Arabic and Hebrew} (version 4.00) [User Manual Version 4.00] (Nov. 3,
\(^4\)Gáspár Sinai’s Yudit probably has the best Unicode support. \texttt{See http://www.yudit.org.}
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, Arab\TeX input can be full vocalized Arabic (\textit{scriptio plena}), vocalized Arabic or non-vocalized Arabic (\textit{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 \texttt{arabluatex}. With the help of Lua, it will eventually interact with some other packages yet to come to produce from \texttt{.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.

\section{arabluatex is for \LaTeXX}

It goes without saying that \texttt{arabluatex} requires \LaTeXX. \TeXX and \LaTeXX have \texttt{arabtex}, and \Xe\TeXX has \texttt{arabxetex}. Both of them are much more advanced than \texttt{arabluatex}, as they can process a number of different languages,\textsuperscript{5} whereas \texttt{arabluatex} can process only Arabic for the time being. More languages will be included in future releases of \texttt{arabluatex}.

In comparison to \texttt{arabxetex}, \texttt{arabluatex} works in a very different way. The former relies on the \texttt{TECkit} engine which converts Arab\TeX input on the fly into Unicode Arabic script, whereas the latter passes Arab\TeX input on to a set of Lua functions. At first, \LaTeXX commands are taken care of in different ways: some, as \texttt{\textbf\textit}, and the like are expected to have Arabic text as arguments, while others, as \texttt{\textbf \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 Arab\TeX 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 \LaTeXX; 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} 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}:

\texttt{\usepackage{arabluatex}}

\textsuperscript{5}To date, both packages support Arabic, Maghribi, Urdu, Pashto, Sindhi, Kashmiri, Uighuric and Old Malay; in addition to these, \texttt{arabtex} also has a Hebrew mode, including Judeo-Arabic and Yiddish.
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 like so:

\begin{verbatim}
\newfontfamily\arabicfont\{Amiri\}[Script=Arabic]
\end{verbatim}

Rem. \textit{a} By default Amiri places the \textit{kasrah} in combination with the \textit{tashdīd} below the consonant, like so:

\begin{verbatim}
\newfontfamily\arabicfont\{Amiri\}[Script=Arabic,RawFeature={+ss05}]
\end{verbatim}

Other Arabic fonts may behave differently.

Rem. \textit{b} \texttt{newfontfamily} can be used to have either Indian of Arabic numbers printed. See on page 23 for more information.

2.2 Options

\texttt{arabluatex} may be loaded with five global options, the first four of which are mutually exclusive and may be overriden at any point of the document (see below sect. 2.3.1 on page 8):

\begin{description}
\item[voc] In this mode, which is the one selected by default, every short vowel written generates its corresponding diacritical mark: \textit{dammah} (’.\textsuperscript{a}), \textit{fatḥah} (’.\textsuperscript{e}) and \textit{kasrah} (’.\textsuperscript{e}). If a vowel is followed by \textit{N}, viz. \texttt{(uN, aN, iN)}, then the corresponding \textit{tanwīn} (’.\textsuperscript{a}, ’.\textsuperscript{a} or ’.\textsuperscript{a}) is generated. Finally, \texttt{(u, a, i)} at the commencement of a word indicate a “connective ‘ālīf” (‘ālīf ‘-waṣli), but \texttt{voc} mode does not show the \textit{waṣlah} above the ‘ālīf; instead, the accompanying vowel may be expressed at the beginning of a sentence (’.\textsuperscript{a}i).\textsuperscript{6}

\item[fullvoc] In addition to what the \texttt{voc} mode does, \texttt{fullvoc} expresses the \textit{sukūn} and the \textit{waṣlah}.

\item[novoc] None of the diacritics is showed in \texttt{novoc} mode, unless otherwise specified (see “quoting” technique below sect. 4.4 on page 19).

\item[trans] This mode transliterates the ArabTEX input into one of the accepted standards. At present, three standards are supported (see below sect. 8 on page 37 for more details):

\begin{description}
\item[dmg/dmg+] Deutsche Morgenländische Gesellschaft \texttt{dmg} is selected by default;
\item[arabica] \texttt{Arabica}.
\end{description}

\item[export] \texttt{export=true|false} \texttt{New feature v.1.13} This option acts as a named argument and does not need a value as it defaults to \texttt{true} if it is used. It enables \texttt{arabluatex} to produce a duplicate of the original .\texttt{tex} source file in which all \texttt{ascn} strings are replaced with Unicode equivalents. See below sect. 12 on page 51 for more information.

\end{description}

\textsuperscript{6}Khaled Hosny, \textit{Amiri} (Dec. 13, 2017), \url{http://www.amirifont.org/}.
\textsuperscript{8}See the documentation of \texttt{amiri}, \textit{ibid.}, 6.
2.2.1 Classic contrasted with modern typesetting of Arabic

New feature v.1.2 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 (ا), then a mere hamzah is written on the line, and a maddah is placed over the 'alif, like so:

\[\text{samā'ūn} \rightarrow \text{samā'} \rightarrow \text{yatasā'ulūn} \]

(see on page 15 for further details).

- The euphonic tašdīd is generated (see on page 15).
- In fullvoc mode, the sukūn is expressed.
- In such words as اًءيَش and the like, the hamzah alone is not written over the letter yāʾ with no diacritical points below as in 

\[\text{yāš} \rightarrow \text{yā'as} \]

but over a horizontal stroke placed in the continuation of the preceding letter.

Please note that only few Arabic fonts provide such contrivances. In case this feature is not supported by some Arabic font, it is advisable to use \SetArbEasy.

\SetArbEasy

Such refinements as 'classic' maddah may be discarded by the \SetArbEasy command, \SetArbEasy* 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.

\SetArbDflt

Default 'classic' rules may be set back at any point of the document with the \SetArbDflt command. Assimilation rules laid on (b) on page 15 may also be applied by the 'starred' \SetArbDflt* version of this command \SetArbDflt* either in the preamble or at any point of the document. Examples follow:

(a) \SetArbDflt:

i. voc

وُمَّاتَ استسقَّةَ قَبْلَ أن يُثْمِكَهُ فِي نُجُوٍّ السَّمَآءِ

ii. fullvoc

وُمَّاتَ استسقَّةَ قَبْلَ أن يُثْمِكَهُ فِي نُجُوٍّ السَّمَآءِ

iii. trans wa-māta 'stisqāʾ an yutimma kitāba-hu fī nuğūm 's-samāʾ

(b) \SetArbDflt*:

i. voc

وُمَّاتَ استسقَّةَ قَبْلَ أن يُثْمِكَهُ فِي نُجُوٍّ السَّمَآءِ

ii. fullvoc

وُمَّاتَ استسقَّةَ قَبْلَ أن يُثْمِكَهُ فِي نُجُوٍّ السَّمَآءِ

iii. trans wa-māta 'stisqāʾ an yutimma kitāba-hu fī nuğūm 's-samāʾ

(c) \SetArbEasy:

i. voc

وُمَّاتَ استسقَّةَ قَبْلَ أن يُثْمِكَهُ فِي نُجُوٍّ السَّمَآءِ

ii. fullvoc

وُمَّاتَ استسقَّةَ قَبْلَ أن يُثْمِكَهُ فِي نُجُوٍّ السَّمَآءِ

iii. trans wa-māta 'stisqāʾ an yutimma kitāba-hu fī nuğūm 's-samāʾ

(d) \SetArbEasy*:

i. voc

وُمَّاتَ استسقَّةَ قَبْلَ أن يُثْمِكَهُ فِي نُجُوٍّ السَّمَآءِ

---

Note that in old mss. such forms as جَاءَ جاً are also found; see W. LL.D Wright, A Grammar of the Arabic Language, rev. W. Robertson Smith and M. J. de Goeje, with a foreword by Pierre Cachia, 2 vols. (3rd edn., Beirut: Librairie du Liban, 1896), t. 24 D.

For an example, see sect. 5.1 on page 28.
2.3 Typing Arabic

Once arabluatex is loaded, a \arb{⟨Arabic text⟩} command is available for inserting Arabic text in paragraphs, like so:—

From Wright:— Arabic, like Hebrew and Syriac, is written and read from right to left. The letters of the alphabet (ِﺀآَجِهلﺍُفوُرُح, يِّجَّتلﺍُفوُرُح, آَجِهلﺍُفوُرُحلَﺍ, ِةَّي,*مَجعُملﺍ) are twenty-eight in number and are all consonants, though three of them are also used as vowels (see § 3).

The following example comes from Wright:—

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

Running paragraphs of Arabic text should rather be placed inside an Arabic environment like so:—

| \begin{arab} |
| ... |
| \end{arab} |

\*Wright, see n. 7, i. 1 A.

\*\*Wright, see n. 7, i. 213 C.
2.3.1 Local options

As seen above in sect. 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:

voc – \arb[voc]{⟨Arabic text⟩};
fullvoc – \arb[fullvoc]{⟨Arabic text⟩};
novoc – \arb[novoc]{⟨Arabic text⟩};
trans – \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 ArabT\TeX{} input

3.1 Consonants

Table 1 gives the ArabT\TeX{} equivalents for all of the Arabic consonants.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration(^{12})</th>
<th>ArabT\TeX{} notation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dmg+ loc arabica</td>
<td></td>
</tr>
<tr>
<td>ل</td>
<td>ل</td>
<td>ل</td>
</tr>
<tr>
<td>ل</td>
<td>ل</td>
<td>ل</td>
</tr>
<tr>
<td>د</td>
<td>د</td>
<td>د</td>
</tr>
<tr>
<td>ت</td>
<td>ت</td>
<td>ت</td>
</tr>
<tr>
<td>ت</td>
<td>ت</td>
<td>ت</td>
</tr>
</tbody>
</table>

Table 1: Standard ArabT\TeX{} (consonants)

\(^{12}\) See below sect. 8 on page 37.

\(^{13}\) See below, Rem. a. For \textit{alif} as a consonant, see Wright, \textit{A Grammar of the Arabic Language}, i. 16 D. The \textit{hamzah} itself is encoded \texttt{'} and may be followed by either \langle u, a \rangle or \langle i \rangle. See below sect. 4.2 on page 13.
<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>ArabTeX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ﻓ</td>
<td>ǧ</td>
<td>ǧ</td>
</tr>
<tr>
<td>ﻰ</td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>ﻱ</td>
<td>kh</td>
<td>h</td>
</tr>
<tr>
<td>ﻲ</td>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>ﻳ</td>
<td>dh</td>
<td>d</td>
</tr>
<tr>
<td>ﻴ</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>ﻵ</td>
<td>z</td>
<td>z</td>
</tr>
<tr>
<td>ﻶ</td>
<td>s</td>
<td>s</td>
</tr>
<tr>
<td>ﻷ</td>
<td>š</td>
<td>š</td>
</tr>
<tr>
<td>ﻸ</td>
<td>ś</td>
<td>ś</td>
</tr>
<tr>
<td>ﻹ</td>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>ﻺ</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>ﻼ</td>
<td>z</td>
<td>z</td>
</tr>
<tr>
<td>﻽</td>
<td>g</td>
<td>g</td>
</tr>
<tr>
<td>﻾</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>﻿</td>
<td>q</td>
<td>q</td>
</tr>
<tr>
<td>ﻷ</td>
<td>k</td>
<td>k</td>
</tr>
<tr>
<td>ﻸ</td>
<td>l</td>
<td>l</td>
</tr>
<tr>
<td>ﻹ</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>ﻺ</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>ﻻ</td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>﻽</td>
<td>w</td>
<td>w</td>
</tr>
<tr>
<td>ﻼ</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>﻽</td>
<td>ah</td>
<td>ah</td>
</tr>
</tbody>
</table>

Table 1: Standard ArabTeX (consonants)

Rem. a Please note that in all cases of elision, the ‘alif ‘u-wasi is expressed only by the vowel that accompanies the omitted hamzah: ⟨u, a, i⟩ as in wa-inhazama and wa-nhazama. For more details on the definite article and the ‘alif ‘u-wasi see sect. 4.2 on page 16.

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 .A whether it be followed by a vowel or not, like so: wa-.An, wa-.n (where the dot symbolizes the absence of vowel), wa-.An, wa-.n (where the dot symbolizes the absence of vowel), wa-.An, wa-.n (where the dot symbolizes the absence of vowel).

Rem. b The letter ي with two points below, اَهِتحَتنِمُﺓاَّنَثُملﺍُﺀآَيلَﺍ, may also be written without diacritical points as ى. 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: qaY'uNٌﺀىَق qayʾun, "saY'uNٌﺀىَش šayʾun, "saY'aNٌﺀىَش šayʾan.

The same result may be achieved by encoding this letter as .y, like so: qa.y'uNٌﺀىَق qayʾun, "sa.y'uNٌﺀىَش šayʾun, "sa.y'aNٌﺀىَش šayʾan.

3.2 Additional characters

New feature v1.8.5 Table 2 on the next page gives the ArabTeX equivalents for some additional Persian characters.

¹⁴For the letter ﻢ with no diacritical points below, see Rem. b below.
Table 2: Standard ArabTEX (additional characters)

Rem. 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>ﯼ</td>
<td>ā</td>
<td>ā</td>
</tr>
<tr>
<td>ﯼ</td>
<td>ā</td>
<td>ā</td>
</tr>
<tr>
<td>ﯼ</td>
<td>ī</td>
<td>ī</td>
</tr>
<tr>
<td>ﯼ</td>
<td>ā</td>
<td>ā</td>
</tr>
<tr>
<td>ﯼ</td>
<td>ā</td>
<td>ā</td>
</tr>
<tr>
<td>ﯼ</td>
<td>ī</td>
<td>ī</td>
</tr>
</tbody>
</table>

Table 3: Standard ArabTEX (long vowels)

Rem. a The long vowels ą, ą, ī, otherwise called ḥurūf 'l-madd', the letters of prolongation, involve the placing of the short vowels ą, Į, ī before the letters ą, Į, ī respectively. arabluatex does that automatically in case any from voc, fullvoc or trans modes is selected e.g. ﯼ qāla, ﯼ qīla, ﯼ yaqūlu.

15See below sect. 8 on page 37.
16The 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.
18See 17.
19See below sect. 8 on page 37.
20For the letter Ī with no diacritical points, see Rem. c. below.
21= al-ʾalif 'l-maqṣūrat".
Defective writings, such as ـٰ، al-ʾalif، and 'l-maḥḏūfat، or defective writings of ʿā and ʿi are encoded _a _u and _i respectively, e.g. _d_alika َكِلٰﺫ، al-mal_a'ikaT-u ‘l-ra.hm_an-u، hu_dayfaT-u bn-u ‘l-yamAn_i for Ḥuḏayfat u bn ṭ-Yamānī، etc.

The letter ي with two points below، may also be written without diacritical points as ى. When it is used as a long vowel، it is encoded iY، where ي recalls the kasrah placed below the preceding letter in vocalized Arabic، like so: ٖناَمَيْلٱُنْبُةَفْيَذُح for Ḥuḏayfat u bn u ‘l-Yamānī، etc.

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>َـ</td>
<td>a a a</td>
<td>a</td>
</tr>
<tr>
<td>ُـ</td>
<td>u u u</td>
<td>u</td>
</tr>
<tr>
<td>ِـ</td>
<td>i i i</td>
<td>i</td>
</tr>
<tr>
<td>ًـ</td>
<td>an an an</td>
<td>aN</td>
</tr>
<tr>
<td>ٌـ</td>
<td>un un un</td>
<td>uN</td>
</tr>
<tr>
<td>ٍـ</td>
<td>in in in</td>
<td>iN</td>
</tr>
</tbody>
</table>

Table 4: Standard ArabTEX (short vowels)

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. 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، ḍammah or kasrah except in the following cases:—
- at the commencement of a word، to indicate that a connective ʾalif is needed، with the exception of the article (see below sect. 4.4 on page 19);
- 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.

4 arabluatex in action

4.1 The vowels and diphthongs

Short vowels As said above، they are written ⟨a، u، i⟩:
Long vowels They are written ⟨U, A, I⟩:

qāla ḍāla, bīʿa ḍība, ṭūr ṭīr, ṭīn ṭīn, murūʿa ṭūrā.

ʾalif maqṣūrah It is written ⟨_A⟩ or ⟨Y⟩:


ʾalif otiosum Said ʾl-fāt “the guarding ʾalif”, after ʾwāqiyat, both when preceded by ʾdammah and by ṣubah is written (UA) or (AW, WA):

naṣarū ṭūṣarū, katabū ṭūṣabū, yaḡzū ṭūṣagū, ramaW ṭūṣaw, banaww, banaw.

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

al-lāh ṭūlāh, ʾilāh ṭūlāh.

Silent Some words ending with ʾā, ʾī are usually written ʾā, ʾī instead of ʾā, ʾī: see Wright. ʾArablatex preserves that particular writing; the same applies to words ending in ʾā, ʾī for ʾā, ʾī. Long vowels ⟨U, I⟩ shall receive no sukūn after a ʾalif maḥḏūfah and are discarded in trans mode:

ḥayāt ṭūḥāt, ṣalāt ṭūḥāt, miškāt ṭūḥāt, tawrāt ṭūḥāt.

And so also: al-rib ʾl-rūb “ับاراً”.

ʾAmr, and the silent To that name a silent ʾ is added to distinguish it from ʾUmar: see Wright. In no way this affects the sound of the tanwīn, so it has to be discarded in trans mode:

24Wright, see n. 7, i. 12 A.
25Ibid., i. 12 C.
When the tanwīn falls away (Wright, *A Grammar of the Arabic Language*, i. 249 B): `
`Amr-u bN u mu.hammadNU `Amr-
u bN MuḥammadNU, mu.hammad-u bN-u `amr-i bN-i HālidNU `Amr-
i bN HālidNU.

And so also: al-rib_aUA ar-ribā, ribaNU rib an.

**tanwīn** The marks of doubled short vowels, `u, `a, `i are written ⟨uN, aN, iN⟩ respectively. arabluatex deals with special cases, such as ⟨a⟩ taking an ` after all consonants except ⟨t⟩, and tanwīn preceding ⟨i⟩ as in ⟨aN_A⟩, which is written ⟨aNY⟩. Such forms as ⟨IIN⟩ may likewise be written ⟨iIN⟩:

al-qA.dI yِضاَقلَﺍ al-qāḍī, qA.diyNU qāḍiy, qA.dIN qāḍi.

**4.2 Other orthographic signs**

**tāʾ marbūṭah** It is written ⟨T⟩:

madInaNU māl, bAbaNU bāb, madInaNU madīnat, bintIN bint maqhaNU maqha, fataNU fataq.

arabluatex is aware of special orthographies: ^say'uNU šayʾ, ^say'aNU šayʾ, ^say'iNU 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 ⟨i⟩ is written ⟨aNY⟩. Such forms as ⟨qāḍi⟩ may likewise be written ⟨iIN⟩:—

al-qA.dIN al-qāḍī, qA.diyNU qāḍiy, qA.dIN qāḍi.

**hamzah** It is written ⟨ʼ⟩, 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 sect. 4.4 on page 19.

**Initial hamzah:** 'asadNU, 'u_htuNU 'ublNU 'iqlīdNU, 'anna, 'inna, 'ānā, 'ānā, 'ānā, 'ānā, 'ānā, 'ānā.

hamzah followed by the long vowel ⟨u⟩ is encoded ⟨_U⟩, 'U_A, 'U, 'U, 'U, 'U, 'U. hamzah followed by the long vowel ⟨i⟩ is encoded ⟨_I⟩, 'I).

---

26 For another way of encoding the initial hamzah followed by a long vowel, see the lāhfiy `l-hamzat on the next page.
From Wright: 27—All consonants, whatsoever, not even ‘alif hêmazat excepted, admit of being doubled and take tašdīd. Hence we speak and write ra’ašun, sa’AluN, sa’ašun, na’Ajun. &; ra’asun.

Middle hamzah: xa.ta’aN, xa.ta’iN, xa.ta’un, xa.ta’ina, xa.ta’ina. 

Final hamzah: x.a.ta’iN, x.a.ta’un, x.a.ta’iN, x.a.ta’un.

27 Wright, see n. 7, i. 14 B.
The strange spelling of \( mi^\text{aat} \): \( m\text{i'aTuN} \), \( m\text{i'atAni} \), \( m\text{i'atayni} \), \( m\text{i'aNa} \), \( m\text{i'at} \). Of course, the ‘pipe’ character can be used to prevent this rule from being applied (see sect. 4.5 on page 21): \( m\text{i'a|TuN} \).

\textit{maddah} At the beginning of a syllable, \( '\text{alif} \) with hamzah and fatḥah (\( \text{ﺃ} \)) followed by \( '\text{alifu} \) \( '\text{l-maddi} \) (\( '\text{alif} \) of prolongation) or \( '\text{alif} \) with hamzah and \( \text{ǧazmah} \) (\( \text{ﺃ} \)) are both represented in \( '\text{alif} \) with maddah: \( \text{ﺁ} \) (see Wright, \textit{A Grammar of the Arabic Language}, i. 25 A–B).

Hence one should keep to this distinction and encode ‘\( \text{a'kulu} \)’ \( '\text{ākulu} \) and ‘\( \text{AkiluN} \)’ \( '\text{ākil} \)’ respectively.

\textit{šaddah} \( \text{tašdīd} \) is either necessary or euphonic.

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

\( \text{'allaqa} \) \( \text{mAdduN} \) \( m\text{ādd} \), \( '\text{ammar} \) \( m\text{ūr} \).

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

(a) With the \textit{solar} letters \( \text{ن} \), \( \text{l} \), \( \text{ض} \), \( \text{ط} \), \( \text{ص} \), \( \text{s} \), \( \text{ش} \), \( \text{س} \), \( \text{ز} \), \( \text{ذ} \), \( \text{ث} \), \( \text{ث} \), after the article \( \text{إ} \):

Unlike \textit{arabtex} and \textit{arabxetex}, \textit{arabluatex} never requires the solar letter to be written twice, as it automatically generates the euphonic \( \text{tašdīd} \) above the letter that carries it, whether the article be written in the assimilated form or not, e.g. \( \text{al-'sams-u} \) \( '\text{aš-sams} \), or \( \text{a'-s-'sams-u} \) \( '\text{aš-sams} \).

(b) With the letters \( \text{ر} \), \( \text{ل} \), \( \text{م} \), \( \text{و} \), \( \text{ي} \), \( \text{م} \), \( \text{ل} \), \( \text{ر} \), after the \textit{tanwīn}:

Unlike \textit{arabtex} and \textit{arabxetex}, \textit{arabluatex} never requires the solar letter to be written twice, as it automatically generates the euphonic \( \text{tašdīd} \) above the letter that carries it, whether the article be written in the assimilated form or not, e.g. \( \text{al-'sams-u} \) \( '\text{aš-sams} \), or \( \text{a'-s-'sams-u} \) \( '\text{aš-sams} \).
Note the absence of sukūn above the passed over ن in the following examples, each of which is accompanied by a consistent transliteration:

\textit{min} rabbi-hi ِهِّبَّﺭنِم, \textit{mir} rabbi-hi, \textit{min} layliN ٍلْيَّلنِم \textit{mil} layl\textsuperscript{m}, 'an yaqtula أَلْتُقَّين \textit{ʾay} yaqtula.

With \textit{tanwīn}: kitÅbÅb mubänN ٌنيِبُّمٌﺏاَتِك kitāb\textsuperscript{um} um mubīn\textsuperscript{um}.

\textbf{Rem.} This particular feature must be put into operation by the \texttt{\SetArbDflt*} command explicitly. See above sect. 2.2.1 on page 5 for further details. Other kinds of assimilations, including the various cases of \textit{ʾidāgām}, will be included in arablatex gradually.

\begin{itemize}
\item[(c)] With the letter ت after the dentals ث, د, ذ, ض, ط, ظ in certain parts of the verb: this kind of assimilation, e.g. لُتبُت\textsuperscript{a} for لُتبُت\textsuperscript{a} labītu, will be discarded here, as it is largely condemned by the grammarians (see Wright, \textit{A Grammar of the Arabic Language}; i. 16 B–C).
\end{itemize}

The definite article and the \textit{ʾalif} \textit{l}-wașli At the beginning of a sentence, \texttt{t} is never written, as لُتْقَّين لُتْقَّين\textsuperscript{a} for لُتْقَّين \textit{ʾay} yaqtula. The hamzah is omitted and only its accompanying vowel is expressed:

\texttt{al-} hamd-u li-l-ah-i لُتْقَّين لُتْقَّين\textsuperscript{a} al-ḥamd\textsuperscript{u} li-l-lāh\textsuperscript{u}.

As said above on on page 5, \texttt{fullvoc} is the mode in which arablatex expresses the sukūn and the waṣlah. arablatex 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:

\begin{itemize}
\item[(a)] Definite article at the beginning of a sentence is encoded \\
\texttt{al-}, or \texttt{a<solar letter>}-
\end{itemize}

\begin{itemize}
\item[(b)] Definite article inside sentences is encoded \\
\texttt{1-} or \texttt{'<solar letter>'-}.
\end{itemize}

\begin{itemize}
\item[(c)] In all remaining cases of elision, the \textit{ʾalifu} \textit{l}-wașli is expressed by the vowel that accompanies the omitted hamzah: (א, א, י).
\end{itemize}

\begin{itemize}
\item Article: bāb-u 1-madrasat-i بَابُ المَدْرَسَةُ, al-maqālaT-u 1-\texttt{_UL}_A \textit{al-maqāla} \textit{l}-ulā, al-lu.qaT-u 1-\textit{arabiyyaT-u} \textit{al-luqātan} \textit{l}-\textit{arabiyyat}" in صناعة أطبّب 1-tibb\textsuperscript{a} fi ʾil\textit{A} 1-intiqā\textit{a}.d-i 1-ibtid\textit{A}'-i 1-ibtidā\textit{A}' in ʾil\textit{A} \textit{ʿabu} 1-wazīr-i fa-lammA raʾaw 1-\textit{najm}-a 1-najm\textsuperscript{a} fa-lamm\textsuperscript{a} raʾawu ʾna-jm\textsuperscript{a}.
\end{itemize}

\begin{itemize}
\item Particles:—
\item[(a)] \textit{li-}: \textit{ʾalif} \textit{l}-wașli is omitted in the article ُل when it is preceded by the preposition ل ُل-l-rajul-i لُلْرَجُلِ \textit{l}-r-rajul\textsuperscript{a}.
\end{itemize}

If the first letter of the noun be ل, then the ل of the article also falls away, but arablatex is aware of that: ل-l-laylaT-i لُلْلَيْلَةِ li-l-laylat\textsuperscript{a}. 

16
(b) ila-: the same applies to the affirmative particle ل: la-l-haqq-u لُحَق لَّا-l-haqq".

(c) With the other particles, ʾalif ʾl-waṣl is expressed: ʾa ʾl-madinat-i فيَلْمَدَّيْنَةَ، wa-ʾl-rajul-u وَأَرْجلُ، wa-ʾr-rajul"، bi-ʾl-qalam-i ويَلْقَّلُ، bi-ʾl-qalam، bi-ʾl-ruʾb-i بَلْقَّلُ، bi-ʾr-ruʾb

Perfect active, imperative, nomen actionis: qAla isma ْعَمْسٱَلاَق qāla ʿsma، qAla uqtul ْلُتْقٱَلاَق qāla ʿqtul، huwa inhazama َمَزَهْنٱَوُه huwa ʿnhazama، wa-ustuʿmila َلِمْعُتْسٱَو wa-ʿstuʿmila، qadi in.sarafa َفَرَصْنٱِدَق qadi ʿnṣarafa، al-iqtidAr-u ُﺭﺍَدِتْقٱِلَﺍ al-iqtidār، ʾil_Aʾl-intiqA.d-i ِﺽاَقِتْنٱِلٱىَل ʾilạl-intiqāḍ، law istaqbala َلَبْقَتْسٱِوَل lawi ʿstaqbala.

Other cases: ʾawi ismu-hu أُهُمْسٱِو ʾawi ʿsmu-hu، ZayduN ibn-u ʿamriNU ٌدْيَﺯ وٍرْمَعُنْبٱ Zayd، ʿUmar-u ibn-u ʿl-Ḫaṭṭāb ʿUmar، Imruʾ u ʾl-Qays-i ُرْمِﺍ ُﺅِسْيَقْلٱ Imruʾ، la-aymun-u ʾl-lāh-i ِهّٰللٱُنُمْيٱَل la-ʾymun u ʾl-lāh

ʾalif ʾl-waṣl preceded by a long vowel The long vowel preceding the connective ʾalif is shortened in pronunciation (Wright, A Grammar of the Arabic Language, i. 21 B–D). This does not appear in the Arabic script, but arabluatex takes it into account in some transliteration standards:

fi ʾl-nAs-i فيَنْيَعِف 'fi ʾl-nās)، ʿabU ʾl-wazIr-i ʾبُو أَوْزِيْرِ عُبَّا ʿabu ʾl-wazīr，fI ʾl-ibtidA'-i ِﺀﺁَدِتْبٱِلٱيِف 'fi ʾl-ibtidāʾ، ḍu ʾl-ʾaṭlāl، maqh_A ْمَقُحَّلُ ʾلَ-اَمِيرِ maqhā ʾl-amīr

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

fi ʾaynay ʾl-malik-i ِكِلَمْلٱِيَنْيَعيِف ʾiḫšayi ʿl-qawm، mu.s.tafaw ʿl-ʾaṭlāl، fa-lammā raʾawu ʾl-naḡm

ʾalif ʾl-waṣl 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 (ibid., i. 22 A–C), it is encoded explicitly, like so:

28 “Zayd is the son of ʿAmr"; the second noun is not in apposition to the first, but forms part of the predicate. Hence زِيَدُ ٌبُنُ عَمْرُ وَ زِيَدُ ٌبُنُ عَمْرُ، “Zayd, son of ʿAmr”.

29 "ʿUmar is the son of al-Ḥaṭṭāb" (see n. 28).
However, the Arabic script does not show the kasrah or the damaah which may be taken by the nouns having tanwin although it is explicit in pronunciation and must appear in some transliteration standards. arabluatex takes care of that automatically:—

mu.ammaduN'l-nabI
Muḥammad
uni
'un-nabī
, salAmuNud_hulUA
Salām
unu
'u'dḫulū,
qa.sIdata-hu fī qatl-i ʿAbī Muslim
i
'llatī yaqūlu fī-hā.

4.3 Special orthographies

The name of God  The name of God, الله, is compounded of the article ال, and ال (noted معهد) so that it becomes لا لله; then the hamzah is suppressed, its vowel being transferred to the ل before it, so that there remains لله (I refer to Edward William Lane, An Arabic-English lexicon, 8 vols. [London – Edinburgh: Williams and Norgate, 1863–1893] [henceforth Lane, Lexicon], I. 83 col. 1). Finally, the first ل is made quiescent and incorporated into the other, hence the laṣdid above it. As arabluatex never requires a solar letter to be written twice (see above, on page 15), the name of God is therefore encoded لله-u or للا-ح-u:—

al-l_ah-u
al-lāh
u
ya|30
al-l_ah-i
la-ta.g`alanna
ʾa-fa-al-lāh
i
la-tağʿalanna,
bi-'l-l_ah-i
bi-'l-
laṭāh,
wa-'l-l_ah-i
wa-'l-
laṭāh,
bi-sm-i 'l-l_ah-i
bi-smi ℓ-лаḥ, al-
-hamd-u li-l-l_ah-i ℓ-хамд ℓ-laṭāh, li-l-l_ah-i ℓ-qA'il-
u ℓ-хамд ℓ-laṭāh ℓ-qa'ìl.

The conjunctive اللّٰهُ Although it is compounded of the article الله, the demonstrative letter ل and the demonstrative pronoun ذا, both masculine and feminine forms that are written defectively are encoded alla_di and allatī respectively. Forms starting with the connective ʾalif are encoded للا_di and للا-tī:—

'a_hAfu mina 'l-malik-i 'lla_di ya.zlimu 'l-nAs-a
أُحَافُ مِنُ الْمَلِكِ الْأَذِي
la-ta.g`alanna
bi-'l-l_ah-i
bi-'l-
laṭāh
la-ta.g`alanna,

30 Note the “pipe” character '|' here after ɣا and below after ɣa before footnote mark 32: 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. ʿalā ʿl-ṭarīq. See sect. 4.5 on page 21 on the “pipe” and sect. 8 on page 37 on dmg mode.

31 See 30.
The other forms are encoded regularly as al-1 or '1-1:—

fa-ʾinnā naḏkur ʿṣ-sawt ʿayn 'l-laḏaynī rawaynā-humā ʿan Ǧaḥzatān.

And also: al-laḏāni al-lātāni, al-latayni al-lāʾī, al-lāʾāti al-lāʾī, and so forth.

4.4 Quoting

It is here referred to “quoting” after the arabtex package.31 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:

ةءيهلﺍملعمّلع

; is

َمِّلُع

, he was taught the science of astronomy, or

َمَّلَع

, he taught the science of astronomy? In order to disambiguate this clause, it may be sensible to put a ḍammah above the first ﻉ, 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 (") 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 ḍammah, fatḥah or kasrah appear above the appropriate consonant:—

"ullima ʿilm-a ʿl-hayʿat-i

ya gz"UA

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

wa-ʾinnA sawfa tudriku-nA ʿl-manAyA muqaddarat an

wa ʾinnā sawfa tudriku-na ʿl-manāyā muqaddarat an.

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

31Note here the “pipe” character ‘|’: as already stated on page 15, the sequence ‘A usually encodes ʿalif with hamzah followed by ʿalif of prolongation, which is represented in writing ʿalif with maddah: ٰ. The “pipe” character prevents this rule from being applied. See sect. 4.5 on page 21.

At the commencement of a word, the straight double quote is interpreted as ‘ālif’ ‘l-waṣl’:

\[ \text{wa-“ust\textsuperscript{u}mila} \quad \text{wa-“stu\textsuperscript{u}mila, huwa “inhazama, al-“intiqād-u.} \]

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:

\[ \text{q”Ala q“A’iluN} \quad \text{“qāla qā’il} , ibn-u ‘aBI \text{ ‘u.saybi\textsuperscript{a}T-} \text{“a \quad \text{Ibn” Abī Uṣaybi’at”}.} \]

(b) tanwīn:

\[ \text{madīnat\textsuperscript{a}} \quad \text{bāb\textsuperscript{a}n} \quad \text{hudā\textsuperscript{A}} \quad \text{šay’iN} . \]

One may more usefully “quote” the initial vowels to write the waṣlah above the ‘ālif or insert a straight double quote after a consonant not followed by a vowel to make the sukūn appear:

(a) ‘ālif’ ‘l-waṣl’:

\[ \text{fi “istiqṣā’-iN} \quad \text{fi “stiqṣā’-“a \quad \text{wa-“stiqṣā’-“a} \quad \text{qAla “uhrub fa-lan tuqtala \quad qāla ‘hrub fa-lan tuqtala.} \]

(b) sukūn:

\[ \text{qAla “uqtul” fa-lan tuqtala \quad qāla ‘qṭul fa-lan tuqtala, mā J阿’t} \text{’ mini imra’aTiN} \quad \text{mā gā’at mini ‘mra’at” \quad \text{qad” ma.dat” min” laylaTiN} . \]

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

\[ \text{al-jamr-u ‘l-.sayfiyy-u ‘lla_dI kAna bi-“rAn”|nUn-a} \quad \text{al-“gamr” ‘Ş-sayfiyy” “llağı kāna bi-Qrānun”.} \]

### 4.4.1 Quoting the hamzah

As said above in sect. 4.2 on page 13, 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 on the next page gives the equivalents for all the possible carriers the hamzah may take.
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, ٙ is encoded ⟨w'''a⟩, while ٚ 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 19–20.

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 sect. 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 30 on page 18 and in 32 on page 19. One more example follows:—

bi-Qrān|nUn-a َنوُننﺍَرقِب bi-Qrānnūn, “in Crannon” (Thessaly, Greece).35

As one can see, the “pipe” character between the two ⟨n⟩ prevents the necessary tašdīd rule ( on page 15) 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 LaTeX commands. To take an example, consider the following:—

\begin{arab}[fullvoc]
fa-lammA ra'aw\LRfootnote{A footnote which interferes with the contextual analysis.} 'l-na^gma...
\end{arab}

1 This is wrong:
2 \begin{arab}[fullvoc]
3 fa-lammA ra'aw\LRfootnote{A footnote which interferes with the contextual analysis.} 'l-na^gma...
4 \end{arab}

34 See below sect. 8 on page 37.
35 See more context on the previous page.
According to the rule stated on page 17, the diphthong in raʾaw must be resolved into two simple vowels before the 'alif 'l-waṣl', as َﺭﺍَّمَلَف ﺃ...َﻡْجَّنلٱ.

The \arbnull 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, \arbnull may be found just after or before Arabic words.

In any case, no space must be left after or before the Arabic word that \arbnull 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:

\begin{arab}[fullvoc]
fa-lammA raʾaw \arbnull{1-naʾgma} \LRfootnote(A footnote which interferes with the contextual analysis.)
'l-naʾgma...
qAla \LRfootnote(A footnote which interferes with the contextual analysis.) \arbnull(qAla) uhrub fa-lan tuqtala.
\uc(z)aydu\arbnull(ibnu) \LRfootnote(A footnote which interferes with the contextual analysis.) \arbnull(zaydu) ibn-u \uc('a)mruNU \LRfootnote(See \vref{fn:zayd-is-son}.)
\end{arab}

\begin{arab}[trans]
\uc(z)aydu\arbnull(ibnu) \LRfootnote(A footnote which interferes with the contextual analysis.) \arbnull(zaydu) ibn-u \uc('a)mruNU \LRfootnote(See \vref{fn:zayd-is-son}.)
\end{arab}

ZaydunībnuʿAmr

\begin{arab}
\end{arab}

\footnote{A footnote which interferes with the contextual analysis.}
\footnote{A footnote which interferes with the contextual analysis.}
\footnote{A footnote which interferes with the contextual analysis.}
\footnote{See 28 on page 17.}
\footnote{A footnote which interferes with the contextual analysis.}
\footnote{See 28 on page 17.}
4.7 Stretching characters: the taʾwil

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:

\text{hunayn}-u bn-u 'is-h--_aq-a Ḥunayn" bn" Ḣaqq"

4.8 Digits

4.8.1 Numerical figures

The Indian numbers, ar-raqa‘u "l-hindiyyа", are ten in number, and they are compounded in exactly the same way as our numerals:

1874 ١٨٧٤, 123-456,789 ١٣٣-٤٥٦,٧٨٩, fI sanaT-i 1024 ١٠٢٤ِةَنَسيِف

\SetArbNumbers\SetArbNumbers{Indian|Arabic} \SetArbNumbers\SetArbNumbers{Arabic}

As described above, arabluatex prints Indian numbers by default. \SetArbNumbers{Arabic} can be used at any point of the document to have Arabic numbers printed. Furthermore, \SetArbNumbers{Arabic} gives control over the way numbers are to be printed by means of the \anum font feature, like so:

\usepackage{arabluatex}
\SetArbNumbers{Arabic}
% use '+anum' for Arabic numbers or '-anum' for Indian numbers:
\newfontfamily\arabicfont{Amiri}[Script=Arabic, RawFeature={+anum}]
\SetArbNumbers must be used \textit{outside} Arabic environments. Once used, this command operates on subsequent Arabic environments.

4.8.2 The ʾabǧad

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{A Grammar of the Arabic Language}, i. 28 B–C). The ʾabǧad numbers are usually distinguished from the surrounding words by a stroke placed over them.

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

\abjad{45} kitAbu-hu fI 'l-ʾAdAt-i 1-ʾAbjad{45} kitābu-hu fi 'l-
\abjad{45} kitAbu-hu fI 'l-ʾAdAt-i 1-ʾAbjad{45} kitābu-hu fi 'l-
\abjad{45} kitAbu-hu fI 'l-ʾAdAt-i 1-ʾAbjad{45} kitābu-hu fi 'l-
\abjad{45} kitAbu-hu fI 'l-ʾAdAt-i 1-ʾAbjad{45} kitābu-hu fi 'l-

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

Rem. \textit{b} \textit{abjad} may also be found outside Arabic environments. In that case, arabluatex 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 \aoline* command that is described below in sect. 4.10.1 on page 25 or insert the \abjad number in \arb{novoc}:

\text{\\arb{trans}\{\text{abjad}\} number for 1874 is \text{\aoline*\{\text{abjad()}\}}} The \text{abjad number for 1874 is} ١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧ legality.

The \text{\\arb{trans}\{\text{abjad}\} number for 1874 is \text{\aoline*\{\text{abjad()}\}}} The \text{abjad number for 1874 is} ١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨٧٤١٨股权
\newfeature{v1.12}{abjad} may also be used to convert values of counters into 'abjad numbers, like so:—

\begin{itemize}
\item The \texttt{\arbran\{\abjad\}} number for the current page (\texttt{\thepage}) is \texttt{\abjad\{\thepage\}}.
\end{itemize}

The \texttt{\abjad} number for the current page (23) is جک.

This technique can be used to produce abjad-numbered lists as will be demonstrated on page 48.

### 4.9 Additional characters

In the manuscripts, the unpointed letters, \textit{al-ḥurūf} ‘l-muhmalat”, are sometimes further distinguished from the pointed by various contrivances, as explained in Wright.\textsuperscript{36} 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 \texttt{arabluatex} will mature.

For the time being, the following is included:—

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration\textsuperscript{37}</th>
<th>\texttt{dmg+}</th>
<th>\texttt{loc}</th>
<th>\texttt{arabica}</th>
<th>\texttt{Arab\TeX} notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ت</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>.b</td>
<td></td>
</tr>
<tr>
<td>د</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>^d</td>
<td></td>
</tr>
<tr>
<td>س</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>.f</td>
<td></td>
</tr>
<tr>
<td>غ</td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>.q</td>
<td></td>
</tr>
<tr>
<td>خ</td>
<td>k</td>
<td>k</td>
<td>k</td>
<td>.k</td>
<td></td>
</tr>
<tr>
<td>ن</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>.n</td>
<td></td>
</tr>
<tr>
<td>ه</td>
<td>ah</td>
<td>ah</td>
<td>a</td>
<td>H\textsuperscript{38}</td>
<td></td>
</tr>
<tr>
<td>م</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>.y\textsuperscript{39}</td>
<td></td>
</tr>
<tr>
<td>ﭛ</td>
<td>(</td>
<td>(</td>
<td>(</td>
<td>(</td>
<td>)</td>
</tr>
<tr>
<td>ﭜ</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
</tr>
</tbody>
</table>

Table 6: Additional Arabic codings

\textit{'afAman.tUs Gal.(M) .fmn.n.ts (sic) Gal.(E1), ʾafāmanṭūs Gal.(M) fmn₄ts (sic) Gal.(E1).}

### 4.10 Arabic emphasis

As already seen in sect. 4.8.2 on the preceding page, the \texttt{\abjad} 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.

\textsuperscript{36}Wright, see n. 7, i. 4 B–C.  
\textsuperscript{37}See below sect. 8 on page 37.  
\textsuperscript{38}This letter can be used to encode the \texttt{\tā’ marbūtah} devoid of diacritical points as it is found in some manuscripts, with the same \texttt{tanwin} and the same short vowels as the standard \texttt{\tā’ marbūtah} with two points above, e. g. \texttt{al-madinah}, \texttt{madInaHa}.  
\textsuperscript{39}See above Rem. \textit{b} on page 9.
One may use the \texttt{\textbackslash emph}(\textit{Arabic text}) command to use the same technique to emphasize words, like so:—

\textcode{\abjad{45}: kitAbu-hu \texttt{\textbackslash emph}{fI 'l-`AdAt-i}.

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

\texttt{\textbackslash emph*} is also provided should one wish to always have the horizontal stroke printed over the emphasized words, like so: \textcode{\abjad{45}: kitAbu-hu \texttt{\textbackslash emph*}{fI 'l-`AdAt-i}.

4.10.1 Underlining words or numbers

Three additional, non context-sensitive commands are provided to distinguish words or numbers:—

(a) \texttt{\textbackslash aoline}, which is equivalent to \texttt{\textbackslash emph*} described above.

(b) \texttt{\textbackslash aoline*}, which is the same as \texttt{\textbackslash aoline}, but better suited for 'abǧad numbers.\textsuperscript{40}

(c) \texttt{\textbackslash aoline}, which can be used to underline Arabic words.

5 Arabic poetry

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

\texttt{arabverse} (env.) The \texttt{arabverse} environment may take up to eight optional ‘named arguments’ each of which is set using the syntax \texttt{⟨key⟩=⟨value⟩}, like so:—

\begin{verbatim}
1 \begin{arabverse}[key1=value1, key2=value2, ...]
2 <verses>
3 \end{arabverse}
\end{verbatim}

The description of the optional arguments follows:—

\texttt{mode} \texttt{mode=⟨mode⟩}, either \texttt{voc}, \texttt{fullvoc}, \texttt{novoc} or \texttt{trans}. The default mode is the one that is set at load time as already seen sect. 2.2 on page 5.

\texttt{width} \texttt{width=⟨length⟩} \textbf{Default:} 0.3\textbackslash linewidth

The default width of each hemistich that the verse consists of. It may be expressed in any accepted unit of measurement, such as 4\textbackslash cm or 2\textbackslash in. However, one must keep in mind that the total length of the two hemistichs added to the one of the gutter that separates them must not exceed the length of the base line, unless one wishes to have the hemistichs distributed on subsequent lines.

\texttt{gutter} \texttt{gutter=⟨width⟩} \textbf{Default:} 0.15 x (hemistich width)

The gutter consists of the blank space that is between the two hemistichs. By default, it is commensurate with the width of the hemistich, but it may be expressed in any accepted unit of measurement as well.

\texttt{metre} \texttt{metre=⟨name⟩} \textbf{Default:} none

If the name of the metre is expressed, it is printed after the lines and set flush left in \texttt{voc}, \texttt{fullvoc} and \texttt{novoc} modes or flush right in \texttt{trans} mode.

\texttt{delim} \texttt{delim=⟨true|false⟩} \textbf{Default:} false

\textsuperscript{40} See the example provided above sect. 4.8.2 on page 23.
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.

\SetHemistichDelim 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 Arab\TeX{} or Buckwalter input schemes. See sect. 10 on page 43 for more details.

\SetHemistichDelim color color=\{color name\}

New feature v1.13 The color in which lines of poetry are to be rendered.

\SetHemistichDelim export export=true|false

New feature v1.13 This named argument does not need a value as it defaults to true if it is used. If export is set as a global option as well (see above on page 5), all the lines will be converted to Unicode and exported to the external selected file. See below sect. 12 on page 51 for more details.

\SetHemistichDelim \bayt \bayt \{⟨sadr⟩\} \[⟨tadwîr⟩\] \{⟨‘aġuz⟩\} \`

\Bayt{⟨sadı̂r⟩}\[⟨tadwîr⟩\] {⟨‘aģız⟩}\`

That two subsequent hemistichs should be connected with one another is technically named tatwîr. Should that happen, either the šadr or the ‘aţuz 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 tadwîr. 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 tatwîl should be used, like so: [---] (see sect. 4.7 on page 23). Of course, the ending word of the šadr and the word at the commencement of the ‘aţuz must have the tatwīl too so that the proper shapes of the letters be selected. Consider for example the following:

\Begin{arabverse}[mode=fullvoc, width=.3\linewidth]
\Bayt{1A ‘ar_A man ‘ahidtu fI-hA fa-‘abkI ‘l---}[---]{---yawma}
dalhA wa-mA yaruddu ‘l-bukA’u)
\End{arabverse}

As one can see, triple hyphens have been used. In the šadr, the first hyphen triggers the rules that are related to the definite article and the ’alif ’l-wasîl, while the following two select the figure of the letter làm connected with a following letter. In the ‘aţuz, the last two hyphens select the letter yā 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:

41 A ‘starred’ version \Bayt* is also defined. arab\LaTeX{} 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.

42 See sect. 4.2 on page 16.
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:

In the first line, the word *مَهلِّا* should be split into *مَمَهلِّا* as the first part of it belongs to the *ṣadr* and the second to the *ʿaǧuz*. 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 *مْم* should be split into *مْمْ* 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.

**Scaling and distortion of characters** The `arabverse` environment and the \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.

\StretchBayt \StretchBayt[true|false] \textbf{New feature v1.20} \textbf{Default: true} \StretchBayt takes one optional argument, either `true` or `false` and can be used to remove the stretching form lines of Arabic poetry. As a side effect, there will be more space between words, but this can be compensated by inserting double hyphens between letters (on this technique, see sect. 4.7 on page 23). Should it be desired to extend further the strokes, four hyphens may be inserted (`----`), viz. a multiple of two. \StretchBayt may be used at any point of the document, even between two subsequent lines of poetry. Note that \StretchBayt[false] may require to carefully adjust the width of the hemistichs to avoid overlapping.

**Footnotes** Footnotes are not set by default inside the \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 \footnotemark{} (please mind the braces or "declare" footnotemark using \MkArbBreak to take it out of the Arabic environment) in the argument of the \bayt command and \footnotetext outside the \bayt command.

If the footnotes are abundant in number, it is advised to load the footnotehyper package which arabluatex will then use to typeset any kind of footnote that is called from the arguments of the \bayt command.

\footnotetext{43}{See sect. 11.1 on page 46.}
\footnotetext{44}{The footnote package can also be used for the same effect. However, it must be loaded after arabiutext.}
\footnotetext{45}{Robert Alessi, The Ekdosis package: Typesetting TEI-xmd compliant Critical Editions (version 1.4) (Nov. 21, 2021), http://www.ekdosis.org.}
\footnotetext{46}{Ibid., see "Arabic Poetry".}
\footnotetext{47}{Please note that for the time being only the assimilation rules that are laid on (b) on page 15 are applied. See sect. 2.2.1 on page 5 for more information. None of the editions of the \textit{Muʿallaqāt} that I know of feature the ʾidgām in the Arabic text, although it is often strongly marked in declamation.}

\footnotetext{v1.21}{New feature}

\begin{itemize}
\item \textbf{Critical Notes} If the ekdosis package be loaded, the \bayt command also accepts a + \bayt+ optional argument that can be used to let critical notes be inserted in lines of poetry. Details on how to use this command are provided in the relevant section of the documentation of the ekdosis package.
\item \textbf{Line numbering} Inside the arabverse environment, the linenumbers environment of the 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.
\end{itemize}

\section{Example}

Here follow the first lines of Imruʿu’l-Qaysi’s \textit{Muʿallaqah}. In this example, \SetArbDflt* has been selected so as to mark the ʾidgām that is fit to this declamatory poetry;—

\begin{verbatim}
\begin{arab}[fullvoc]
qAla imru'u 'l-\uc{q}aysi fI mu'allaqati-hi:
\end{arab}
\begin{arabverse}[mode=fullvoc, metre={\uc{al}-.darbu 'l-\uc{t}AnI mina 'l-.\uc{t}awIli}]
\SetArbDflt*
\begin{linenumbers*}
\bayt{qifA nabki min _dikr_A .habIbiN wa-manzili}{bi-siq.ti 'l-lw_A bayna \uc{'l-d}a_hul\uc{li} fa-\uc{h}awmi-hu}\bayt{fa-\uc{h}awmi-hu lam ya'fu}{rasmu-h\uc{h}A min 'gam\uc{U}biN wa-'sam'alI}\\
\bayt{\uc{t}U.di.ha fa-'l-miqrATi lam ya`fu}{samur\uc{u}ti 'l-.\uc{r}a\uc{m}i-hA l\uc{d}a\uc{h}Uli fI 'ara.s\uc{u}ti-hA}{\uc{h}a\uc{u}ta\uc{a}nu .habbu fulfulu}\\
\bayt{ka-\uc{h}am\uc{u}nI .\uc{g}ad\uc{A}Ata \uc{y}a\uc{m}na ta.hammalU\uc{A}nA}{lad\uc{A}}\\
\bayt{\uc{h}am\uc{u}nI .\uc{g}ad\uc{A}Ata \uc{y}a\uc{m}na ta.hammalU\uc{A}nA}{lad\uc{A}}\\
\bayt{\uc{h}am\uc{u}nI .\uc{g}ad\uc{A}Ata \uc{y}a\uc{m}na ta.hammalU\uc{A}nA}{lad\uc{A}}\\
\end{linenumbers*}
\end{arabverse}
\end{verbatim}

\texttt{\stretchbayt* \texttt{true}} (Default):—
قال أمير الْقِنِس في مَعْلِمَة:

1. قَالَ ِبِنُكِمْ مِنْ ذَكَرٍ حَيْبٍ وَمَزَّلَ بِسَطِّ اللَّوَى بِذِنَّ الدُّخُولِ حَوْلَمۡ.
2. فُنُجِحَ أَفْلَامُهُ لَوْ عِفْ رَجَحَهَا مَلَّا نُسْجِيَهَا مِنْ جَنَّوٍ وَمُتَّنَّ.
3. تَرَى بِعْرَ الأَلَّامِ فِي عَرْسِهَا وَقِيَمَهَا كَأَنَّهُ حَبٌّ فَقُلُ لَأَنْ قَدْ عَرَّفَ النَّسَقَةَ ۖ لَا يُذْكِرَ أَيْضًا وَجَلْعُهَا.
4. كَأَنَّ عَناْ دَابَاً يَوْمًا يَمِّنَحُوا وَقَفَّ فِي هُدَى عَلَى مَطْبَعٍ قَلْحُ لاَ يُتَّكِلَ أَيْضًا وَجَلْعُهَا فِي هُدَى عَنْدَ رَسُمٍ دَارِسٍ مِنْ مَعْوَلٍ.
5. وَإِنَّ شَفَاطٌ عَبْرَةَ مُهَرَاقَةٍ فِي هُدَى مَعْوَلٍ دَارِسٍ مِنْ مَعْوَلٍ.

(الْعَرْضُ الثَّانِي مِنْ العَرْضَ الْأَوَّلِ مِنْ الْطَّوْلِ)

قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:

1. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
2. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
3. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
4. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
5. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
6. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:

(عَلَيْهِ مَعْوَلٌ دَارِسٍ مِنْ مَعْوَلٍ)

In what follows, \\texttt{width} has been set to 0.3\texttt{\linewidth} and double hyphens have been inserted between some letters to prolong their horizontal strokes.

1. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
2. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
3. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
4. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
5. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:
6. قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:

(عَلَيْهِ مَعْوَلٌ دَارِسٍ مِنْ مَعْوَلٍ)

In what follows, \\texttt{width} has been set to 0.375\texttt{\linewidth} and \\texttt{scriptsize} has been used so as to avoid overlapping.

قَالَ ِمْرُوْعُ الْقَيْسِي فِي مَعْلِمَةٍ:

qifā nabki min dikrā habīb wā-wa-manzili bi-siqi 'l-ībā bayna 'l-Dāhūli fa-Haemali
fa-Tūdīha fa-'l-Mgrāṭ lam yā-fa rasnu-hā limā nasāqat-hā min ganiḥ wā-wa-samali
tārā ba'ara 'l-arāmī fi 'arāṣātī-hā wa-qātī-hā ka-anna-hu habbu ṣafalā
ta-anna gudāra 'l-bayni yuwa la-bahamalā laḏā samurāri 'l-hayyi nāqifu hanziala
wugāth bi-hā saẖṭī 'l-ayyā matiyya-hum yaqūlinā lā tahlik ṣaq wā-wa-taẖāmnalā
wa-ābīna 'rbrat wā-marqat wa-hali 'l-ḥayyi nāqifu ḥanẓali
wuqūf an bi-hā ṣahā bi-ʿalayya maṭiyya-hum yaqūlūna lā tahlik ʾasā wa-taḏammali
wa-ʾinna šifāʾī ṣawrat um muḥaraqat wa-hali ʿinda rasm dāris mim muwawalā
(wa-duḥbū 'l-fānī mina 'l-arūdī 'l-ulā mina 'l-ṣawrī)

(الْعَرْضُ الثَّانِي مِنْ العَرْضَ الْأَوَّلِ مِنْ الْطَّوْلِ)
6 Special applications

**Linguistics** The same horizontal stroke as the taʾtwīl (see sect. 4.7 on page 23) may be encoded ⟨B⟩; ⟨BB⟩ will receive the tašdīd. This is useful to make linguistic annotations and comments on vowels:

\[ Bu \quad Ba \quad Bi \quad BuN \quad BaN \quad BiN \quad \ldots \quad u \quad a \quad i \quad \uparrow \quad an \quad \uparrow \quad in, \quad BBu \quad BBa \quad BBi \quad \ldots \quad u \quad a \quad i, \quad B-\quad aN \quad \downarrow \quad an, \quad B^n. \]

**New feature v1.4.3 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. arablatex will automatically fit those symbols to the direction of the text. For the time being, the following symbols are supported:

- parentheses: ⟨⟩
- square brackets: [ ]
- angle brackets: < >
- braces: { }

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 \abracess command:

\begin{arab}
\abracess{wa-qAla} 'inna 'abī kāma mina 'l-muqātilaTi
\wa-kāma\textless \texttt{-t}\r:\texttt{immī min 'u.zamA'i buyUṭi 'l-zamĀzimaTi}.
\end{arab}

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.

**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\arbmark[(rl|lr)]\{}\{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>لَهِيْنِيْ َرَحْمَتَيْ</td>
<td>bi'-'smi 'Llāhi 'r-raḥmāni 'r-raḥīmi</td>
</tr>
<tr>
<td>FDF5</td>
<td>salam</td>
<td>صَلَّم</td>
<td>ṣallā 'Llāhu 'alay-hi wa-sallama</td>
</tr>
<tr>
<td>FDF6</td>
<td>slm</td>
<td></td>
<td>ṣallā 'Llāhu 'alay-hi wa-sallama</td>
</tr>
<tr>
<td>FDFB</td>
<td>jalla</td>
<td>ġalla</td>
<td>ġalla ġalāla-hu</td>
</tr>
</tbody>
</table>

Table 7: Additional Arabic marks

**New feature v1.11**
\newarbmark

\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{\langle shorthand\rangle}{\langle RTL codepoint\rangle}{\langle LTR rendition\rangle}

As regards the right-to-left codepoint, it may be either typed in Unicode or selected as Unicode codepoint. To that end, the \LaTeX command \texttt{\symbol{"XYZT}} or its plain \TeX variant \texttt{\char"XYZT\relax} may be used, where \texttt{XYZT} are uppercase hex digits (0 to 9 or A to F).

It is also possible to use the so-called ‘\\\\’ notation’ like so: ‘\\\\xzyt’, where xzyt 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 \texttt{\arb[trans]\{arabtex code\}} so as to have it printed in romanized Arabic.

It must be noted that \texttt{\newarbmark} expects Arab\TeX input scheme inside \texttt{\arb[trans]} to the exclusion of buckwalter input scheme.

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

\begin{verbatim}
\SetArbDflt*
\newarbmark{sly}{\arbcolor[red]{\\\\06d6}}{}
\newarbmark{jim}{\arbcolor[red]{\\\\06da}}{}
\begin{arab}
sUraTu 'l-nis'A'i, 19:
\end{arab}
\begin{center}
\begin{arab}
\arbmark{bismillah}
\end{arab}
\end{center}
\begin{arab}[fullvoc]
y_a'ayyuhA 'lla_dIna 'a'manUA lA ya.hillu la-kum 'an tari_tUA
'l-nis'A'a karhaN\arbmark{sly} wa-lA ta'.duU-hunna li-ta_dhabUA
bi-ba'.di mA 'a'tayyunU-hunna 'illa 'an ya'tina bi-fA.hi^saTiN
mubayyinATn\arbmark{jim} wa-'A^sirU-hunna
bi-'l-ma`rUf\arbmark{jim} fa-'in karhutU-hunna fa-'as_A_a
'an takrahUA 'say'aN wa-ya^g`ala 'l-l_ahu fI-hi _hayraN
ka_tIr\arbmark{bismillah} ((19))
\end{arab}
\end{verbatim}

New feature v1.18 The ‘Zero width joiner’ character (U+200D) The ‘Zero width joiner’ character (U+200D)
belongs to the ‘General Punctuation’ block (range 2000–206F) of the Unicode standard. It is a non-printing character which, when it is placed between two characters that would for some reason not be connected, causes them to be printed in their connected forms.

It is encoded & in ArabTgX scheme.

In elegantly printed books where many of the letters are interwoven with one another so as to form ligatures, it may be convenient to bring the letters into line in some instances. In the following example, the ‘zero width joiner’ is used to prevent two adjacent letters, viz. س and ح, from standing one above the other in the name of ʾIshāq (אִשְׁעַq):

48

6.1 The Qur'ān

This sub-part is destined to become a part of its own, as fine typesetting of Qur'ānic 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 ʾāyah that it is referred to inside the dedicated mark—Unicode U+06DD: ﭢ—in Arabic script or inside parentheses in romanized Arabic:

\ayah{123} ﭬ (123).

An example follows:—

49\underline{underline} and \highlight{highlight} are taken from the lua-ul package which is loaded by arabluatex. See Marcel Krüger, The Lua-ul package: Underlining for LuaLaTeX (version 0.0.1) (Mar. 12, 2020), http://www.ctan.org/pkg/luatex.
min (sūrati 'l-Baqarati):
ʾalif lām mim (1) dālika 'l-kitābu lā rayba fī-hi hudā' li-l-muttaqīna (2) 'llaḏīna yuʾminūna bi-ʾl-ġaybi wa-yuqīmūna ʾṣ-ṣalāta wa-mimmā razaqnā-hum yun-fiqūna (3)

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

New feature v1.12 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 (, sect. 4.5 on page 21);
(b) ‘Quoting’ technique (sect. 4.4 on page 19), and more specifically ‘quoting the hamzah’ ( on page 20);
(c) Putting back on broken contextual analysis rules (sect. 4.6 on page 21);
(d) Arabic marks (sect. 6 on page 30).

\arbcolor \arbcolor[(color)]{(Arabic text)}

---

Regarding 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.
As this example shows, \textcolor{red}{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 syllables inside words. \textcolor{red} {arabluatex} takes care of selecting the appropriate shape of the letters while coloring them:—

\textit{‘voc’ mode:}
\begin{verbatim}
  i"stara\arbcolor[brown]{y}tu-hu bi-_tama\arbcolor[red]{niN}
  'a\arbcolor[blue]{ga}ba-ka \textit{а́шьры́й у ма́й j̱а̨̣̯и́ а̯ж̱ы́ќ luš‘aytu-hu bi-_taman} \textit{ин̱̣̯и́ а̯ж̱ы́ќ} a̯̣̯а̯б̱а̯-ка.
\end{verbatim}

\textit{‘fullvoc’ mode:}
\begin{verbatim}
  i"stara\arbcolor[brown]{y}tu-hu bi-_tama\arbcolor[red]{niN}
  'a\arbcolor[blue]{ga}ba-ka \textit{а́шьры́й у ма́й j̱а̨̣̯и́ а̯ж̱ы́ќ luš‘aytu-hu bi-_taman} \textit{ин̱̣̯и́ а̯ж̱ы́ќ} a̯̣̯а̯б̱а̯-ка.
\end{verbatim}

### 7.1 Tricks of the trade

**Diacritics** Depending on the mode selected, either \textit{voc}, \textit{novoc} or \textit{fullvoc}, coloring the diacritics requires more attention for the insertion of \textcolor{red}{arbcolor} may prevent contextual analysis from being applied.

Furthermore, depending on the surrounding letters, the standard encoding of short vowels \langle u, a, i \rangle may result either in diacritics or in a connective ‘\textit{alif}’ with the \textit{waṣlah} or its accompanying vowel. As for the \textit{sukūn}, it is generated by contextual analysis. Thus applying colors to bare diacritics requires them to have specific encodings.

Table 8 gives the \textit{ArabTEX} equivalents for the diacritics to be printed inside or just after \textcolor{red}{arbcolor}.

\begin{table}[h]
\centering
\begin{tabular}{llll}
\hline
\textbf{Diacritic} & \textbf{Transliteration} & \textbf{ArabTEX notation} \\
\hline
\textit{'} & \textit{a} & \textit{a} & \textit{a} & \textit{.a} \\
\textbackslash{} & \textit{u} & \textit{u} & \textit{u} & \textit{.u} \\
\textbackslash{} & \textit{i} & \textit{i} & \textit{i} & \textit{.i} \\
\textbackslash{} & \textit{0} & \textit{0} & \textit{0} & \textit{0} \\
\hline
\end{tabular}
\caption{ArabTEX diacritics for \textcolor{red}{arbcolor}}
\end{table}

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:—

\textit{‘voc’ mode:}
\begin{verbatim}
  i"staraytu-hu bi-_taman\arbcolor[red]{iIN} \textit{а́шьры́й у ма́й j̱а̨̣̯и́ а̯ж̱ы́ќ luš‘aytu-hu bi-_taman} \textit{ин̱̣̯и́ а̯ж̱ы́ќ} a̯̣̯а̯б̱а̯-ка.
\end{verbatim}

\footnote{\textit{See below sect. 8 on page 37.}}
As can be seen, `fullvoc` required the letters y, n and `g before \arbc olor` to be ‘quoted’. Otherwise, unwanted sukūns would have been generated because of the absence of a vowel after those consonants.

\textit{tanwīn} \arbn ul must be used with \textit{fathalān} (٧) so as to put back on contextual analysis rules:—

\begin{itemize}
\item mu’allim\arbc olor[red]{\\arbn ul\{m\}aN} \textit{mu’allim} an,
\item istisqā’\arbc olor[red]{\\arbn ul\{A’\}aN} \textit{istikhā}\textit{ā} an,
\item šay\arbc olor[red]{\arbn ul\{ay’\}aN} \textit{šay} an,
\item ġāmi`aT\arbc olor[red]{\\arbn ul\{T\}aN} \textit{ġāmi`aT} an.
\end{itemize}

Rem. Note that in the last example (ġāmi`aT an), the ‘pipe’ character has been inserted before \arbc olor. Otherwise, the \textit{dang} mode of the transliteration rules would have interpreted the ِta’ marbūṭah as final (e.g. \textit{h} instead of the expected \textit{t}).\textsuperscript{51}

The \textit{tanwīn} preceding a ى 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 ى in Arabic script and not a thing in transliteration. It is to be appended after \arbc olor, like so:—

\texttt{
{\newarbmark{Y}{}
\arb\{hud\}\arbc olor[red]{\\arbn ul\{\_A\}}\arbrmark{Y}\}
\arb\{trans\}\{hud\}\arbc olor[red]{\\arbn ul\{\_A\}}\arbrmark{Y}\}

\texttt{ hud\textsuperscript{a}n}

\textsuperscript{51}See also on page 40 “Discarding the ʾīrāb” for more information.
waṣlah and maddah  Both can be generated with the help of \arbnull:—

\arbcolor[red]{\arbnull{wa}i}stis qa‘un
ff "al"-i-\arbcolor[red]{\arbnull{l}-i}bitid‘i
\arbcolor[red]{\arbnull{a}'}kulu,
\arbcolor[red]{\arbnull{A}\arbnull{k}}il un.

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

1 \newarbmark{alifmaddahred}{\arbnull{al}i}{\arbnull{stisqa‘un}}
2 \arb[trans]{\arbnull{a}'kulu}
3 \arb[mark]{alifmaddahred}{kulu}
4 \arb[trans]{alifmaddahred}{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 تَلْعِبُونَ, al-mu‘allimūna, 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:—

1 \newarbmark{lamshaddah}{\arbnull{al}i}{\arbnull{mu‘allimuna}}
2 \arb[fullvoc]{al-mu‘al"i}{\arbnull{shaddah}}.imUna
3 \arb[trans]{al-mu‘al"i}{\arbnull{shaddah}}.imUna.

The definite article and the euphonic tašdīd  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 ff ‘l-nAsiِ ‘fi ḥa‘sās, the initial ʾalif ‘l-waṣl can be rendered in red like so: \arbcolor[red]{\arbnull{al}-\arbnull{a}}fi ‘n-nāsi. Then, the following two letters, namely 1-n, must print the string ʾālām + ʾnūn + šaddah in Arabic, and exactly n-n in transliteration. Thus an Arabic mark is needed:—

1 \newarbmark{lnn}{\arbnull{al}i}{\arbnull{mu‘allimuna}}
2 \arb[fullvoc]{al-mu‘al"i}{\arbnull{shaddah}}.imUna
3 \arb[trans]{al-mu‘al"i}{\arbnull{shaddah}}.imUna.

52 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 21:

\[ yataš\begin{arbc} \text{a}\end{arbc} \text{null}\{\text{‘a}\}\text{.alUna} yataš\begin{arbc} \text{a}\end{arbc} \text{ulluna}, \]
\[ "\text{say}\{\text{‘a}\}\text{.anulla} \text{say} "\text{an}, \text{say}\{\text{‘a}\}\text{.alUna} \text{say} \text{in}, \text{anulla} \text{as} \text{.ilaTuN} \text{as} \text{.as} \text{.ilaTuN}. \]

8 Transliteration

It may be more appropriate to speak of “romanization” than “transliteration” of Arabic. As seen above in sect. 2.2 on page 5 on pages 5–8, 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 5, three standards are supported at present:

- **dmg** Deutsche Morgenländische Gesellschaft, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.\(^{53}\) dmg transliteration convention is selected by default;
- **dmg+** This is the same as above, with the difference that the hamza is always printed, even in words preceded by the definite article, as in al-\(\text{asadu}\).
- **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;\(^{54}\)

**New feature v1.8 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.\(^{55}\)

More standards will be included in future releases of arabluatex.

\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, dmg+, loc or arabica. This command is also accepted in the preamble should one wish to set the transliteration mode globally, e.g.:—

\begin{verbatim}
\usepackage{arabluatex}
\SetTranslitConvention{loc}
\end{verbatim}

\SetTranslitStyle**Style** 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. upshape, \texttt{\itshape}, \texttt{\slshape}, and so forth.

**New feature v1.4 Font** \SetTranslitFont{⟨font selection command⟩} allows any specific font to be selected for rendering transliterated text with the font-selecting commands of the fontspec or luatfload package. Of course, this font must have been defined properly. To take one example, here is how the Gentium Plus font can be used for rendering transliterated text:—

\begin{verbatim}
\SetTranslitFont{Gentium Plus}
\end{verbatim}

\(^{53}\)See Brockelmann et al., “Die Transliteration der arabischen Schrift”.
\(^{54}\)See http://www.loc.gov/catdir/cpso/roman.html for the source document concerning Arabic language.
\(^{55}\)See http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf.
\texttt{Proper names} Proper names or book titles that must have their first letters upcased 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 upcased, viz. ‘or’, the letter next to it will be upcased:—

\begin{verbatim}
\uc{'hunayn-u} bn-u \uc{'is.h_aq-a} Hunayn" bn" Ishāq", \\
\uc{'u_tm_an-u} Umān", .daraba \uc{zayd-u} bn-u \uc{'h_alidin} \\
\uc{sa'd-a} bn-a \uc{'awf-i} bn-i \uc{'l-lah-i} Daraba Zayd" bn" Hālid" Sa'd" bn" 'Awf" bn" Abd" 'Lāh".
\end{verbatim}

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 18, the transliteration of \textit{يِبَّنلٱٌدَّمَحُم} must be \textit{Muḥammaduni 'n-nabī}, as nouns having the \textit{tanwīn} must be \textit{Muḥammaduni 'n-nabī}, as nouns having the \textit{tanwīn} take a \textit{kasrah} in pronunciation before \textit{ʾalifu 'l-waṣli}. In that case, encoding like so: \begin{verbatim}\uc{mu.hammaduN}\end{verbatim} is wrong, because the closing brace would prevent \texttt{arabluatex} from detecting the sequence \texttt{’uN} immediately followed by \texttt{’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 upcased, like so: \begin{verbatim}\uc{m}u.hammaduN\end{verbatim}.

\texttt{Hyphenation} 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:—

\begin{verbatim}
\uc{'abU} \uc{bakriN} \uc{mu-.ham\-madu} bnu \uc{za\-ka \-riy\-ya'a} \uc{'l-rahziyyu} \textit{Abū Bakr" Muḥammad" bn" Zakariyya" 'r-Rāziyyu}.
\end{verbatim}

\texttt{New feature v1.10} ‘\textit{Long}’ proper names \texttt{\uc} is also able to process proper names consisting of several subsequent words:—

\begin{verbatim}
\arb[trans]{\uc{'abU zaydiN .hunaynu bnu 'is.h_aq-a 'l-`ibAdiyyu}}
\textit{Abū Zayd" Hunaynu bnu Ishāqa 'l-`AbdAdiyyu}.
\end{verbatim}

\texttt{Proper names outside Arabic environments} Transliterated proper names inserted in paragraphs of English text should be printed in the same typeface as the surrounding text. \texttt{\prname{(Arabic proper name)}} is provided to that effect:36—

36Just as \texttt{\uc}, \texttt{\prname} is also able to process proper names consisting of several subsequent words.
From Wright:—If the name following ُنْبِا be that of the mother or the grandfather, the ُنْ is retained; as َمَيْرَمُنْبٱىَسيِع "Jesus the son of Mary"; ٍﺭوُصْنَمُنْبٱُﺭاَّمَع ʿAmmār the (grand)son of Manṣūr".

\begin{nameauth}
\Hunayn & \prname{ʾabU zayd} & \prname_.hunayn}, \prname{(i)bn 'is_h_aq al-ʿibādiyy} & > %
\Razi & \prname{ʾabU bakr mu.hammad ibn zakariyyA'} & > %
\prname(al-ʿaziyy) & > %
\end{nameauth}

On first occurrence, proper names are printed as \Hunayn, \Razi.
Then as \Hunayn, \Razi.

\begin{nameauth}
\Hunayn & \prname{ʾabU zayd} & \prname_.hunayn}, \prname{(i)bn 'is_h_aq al-ʿibādiyy} & > %
\Razi & \prname{ʾabU bakr mu.hammad ibn zakariyyA'} & > %
\prname(al-ʿaziyy) & > %
\end{nameauth}

On first occurrence, proper names are printed as ʾAbū Zayd Ḥunayn ibn ʾIsḥāq al-ʿIbādi, ʾAbū Bakr Muḥammad ibn Zakariyyāʾ ar-Rāzī. Then as Ḥunayn, ar-Rāzī.

8.1 Additional note on dmg convention

New feature vi.3 According to Brockelmann et al., ʿiʿrāb may be rendered into dmg in three different ways:
(a) In full: Aʾmrun;
(b) As superscript text: ʾAmr“n; ʾAmr
(c) Discarded: Aʾmr.
\arbup By default, arablatex applies rule (b). Once delimited by a set of Lua functions, ʿiʿrāb is passed as an argument on to a \arbup command which is set to \textsuperscript.

\NoArbUp \NoArbUp may be used either in the preamble or at any point of the document in case one wishes to apply rule (a). The default rule (b) can be set back with \ArbUpDflt at any point of the document.
\SetArbUp Finally, \SetArbUp{formatting directives} can be used to customize the way ʿiʿrāb is displayed. To take one example, here is how Arabic ʿiʿrāb may be rendered as subscript text:—

57See the documentation of nameauth for more details: https://ctan.org/pkg/nameauth
58See above sect. 12 on page 51 for more details.
As shown in the above example, \#1 is the token that is replaced with the actual tanwīn in the formatting directives of the \SetArbUp command.

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

\textit{munṣarif}:
\begin{itemize}
  \item muʿallīn ñ\textsuperscript{\textit{an}}
  \item kā\textsuperscript{\textit{an}}\textit{nu}\textsubscript{\textit{an}}
  \item fātā\textsuperscript{\textit{an}}
\end{itemize}

\textit{ġayr munṣarif}:
\begin{itemize}
  \item al-muʿallīn ñ\textsuperscript{\textit{an}}
  \item kītāb\textsuperscript{\textit{an}}
  \item rāsā\textsuperscript{\textit{an}}
  \item sāriq\textsuperscript{\textit{an}}
  \item al-.zulm\textsuperscript{\textit{an}}
\end{itemize}

\textbf{Rem. a}  
As the tanwīn is passed over in pronunciation when it is followed by the letters ē, ī, ū, and ū, it may be desirable to further distinguish it by putting it above the line, but not to do the same for \textit{ġayr munṣarif} terminations. This can be achieved by simply omitting the hyphen before any \textit{ġayr munṣarif} termination:—

\begin{itemize}
  \item kānā ġanīyy\textsuperscript{\textit{an}}
  \item nās
  \item aṣ-
\end{itemize}

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

\begin{itemize}
  \item manaʿa l-nās ñ\textsuperscript{\textit{an}}
  \item kuff\textsuperscript{\textit{an}}
  \item min muḥāṭābati-hi ʾaḥad\textsuperscript{\textit{an}}
\end{itemize}

\textbf{Discarding the ʿiʿrāb}  
As said above ((c) on the previous page), the ʿiʿrāb may be discarded in some cases, as in transliterated proper names or book titles. \textit{arabluatex} is able to render words ending with \textit{tāʾ marbūṭah} in different ways, depending on their function:—

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

(b) Nouns followed by another noun in the genitive (construct state): hikmat Allāh, fī dāt al-dārāhim fī dāt al-dārāhim.
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: عَفََلِ. In transliteration, the uncertain vowels go between slashes and are separated by commas: fa`uaila َُ fa/u,a,i/la.

8.2 Examples

Here follows in transliteration the story of Ǧuḥā and his donkey (ُهُﺭﺍَمِحَواَحُج). See the code on page 7:—


9 Buckwalter input scheme

New feature v1.4 Even though arabluatex is primarily designed to process the ArabTeX notation, it can also process the Buckwalter input scheme to a large extent.60 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 sect. 2.2 on page 5.

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

| \usepackage{arabluatex} |
| \SetInputScheme{buckwalter} |

60 See http://www.qamus.org/transliteration.htm
‘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.\(^{61}\) However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces \{ and \}, which are used to encode \ُ and \ُ, 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 24 are also available.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration(^{62})</th>
<th>Buckwalter notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ء</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>﴾</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>ﴽ</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>ﴺ</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>ﴹ</td>
<td>j</td>
<td>j</td>
</tr>
<tr>
<td>ﴻ</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>﴾</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ﴬ</td>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>ﴭ</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>ﴮ</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>ﴯ</td>
<td>z</td>
<td>z</td>
</tr>
<tr>
<td>ﴰ</td>
<td>s</td>
<td>s</td>
</tr>
<tr>
<td>ﴱ</td>
<td>ṣ</td>
<td>ṣ</td>
</tr>
<tr>
<td>ﴲ</td>
<td>ḥ</td>
<td>ḥ</td>
</tr>
<tr>
<td>ﴳ</td>
<td>ḫ</td>
<td>ḫ</td>
</tr>
<tr>
<td>ﴴ</td>
<td>ḳ</td>
<td>ḳ</td>
</tr>
<tr>
<td>ﴵ</td>
<td>Ḳ</td>
<td>Ḳ</td>
</tr>
<tr>
<td>ﴶ</td>
<td>ḹ</td>
<td>ḹ</td>
</tr>
<tr>
<td>ﴷ</td>
<td>ḿ</td>
<td>ḿ</td>
</tr>
<tr>
<td>ﴸ</td>
<td>ṁ</td>
<td>ṁ</td>
</tr>
<tr>
<td>ﴹ</td>
<td>Ṃ</td>
<td>Ṃ</td>
</tr>
<tr>
<td>ﴺ</td>
<td>ṅ</td>
<td>ṅ</td>
</tr>
<tr>
<td>ﴻ</td>
<td>ṙ</td>
<td>ṙ</td>
</tr>
<tr>
<td>ﴼ</td>
<td>ṝ</td>
<td>ṝ</td>
</tr>
<tr>
<td>ﴴ</td>
<td>ṙ</td>
<td>ṙ</td>
</tr>
<tr>
<td>ﴵ</td>
<td>ḡ</td>
<td>ḡ</td>
</tr>
<tr>
<td>ﴶ</td>
<td>ḫ</td>
<td>ḫ</td>
</tr>
<tr>
<td>ﴷ</td>
<td>ḧ</td>
<td>ḧ</td>
</tr>
<tr>
<td>ﴸ</td>
<td>ṉ</td>
<td>ṉ</td>
</tr>
<tr>
<td>ﴹ</td>
<td>ḟ</td>
<td>ḟ</td>
</tr>
<tr>
<td>ﴺ</td>
<td>ṛ</td>
<td>ṛ</td>
</tr>
<tr>
<td>ﴻ</td>
<td>Ṗ</td>
<td>Ṗ</td>
</tr>
<tr>
<td>ﴼ</td>
<td>ḕ</td>
<td>ḕ</td>
</tr>
</tbody>
</table>

Table 9: Buckwalter input scheme


\(^{62}\) I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in arabluatex.

\(^{63}\) See sect. 8 on page 37.
Transliteration  The Buckwalter notation can also be transliterated into any accepted romanization standard of Arabic. See above sect. 8 on page 37 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:

Al-‘Alamū al-‘ālamū, Al-camsu al-shamsū, bi-SinaAEapi Al-Tibī, bi-sināʿat l-ṭibbī.


Similary, 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, <ilY Al-LntiqADi gives al-ḥamdū as expected, but only <ilY Al-intiqADi can be transliterated as al-ḥamdī with the correct vowel ⟨i⟩ in place of the ‘alif’ ‘l-waṣlī.

10 Unicode Arabic input

New feature vi.5 As said above in sect. 9 on page 41 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:

\texttt{\textbackslash txarb\{\textit{Unicode Arabic}\}} command for inserting Unicode Arabic text in paragraphs;

\texttt{txarb\ (env.)} (b) The \texttt{txarab} environment for inserting running paragraphs of Arabic text, like so:—

\begin{verbatim}
\begin{txarab}
  \textit{<Unicode Arabic text>}
\end{txarab}
\end{verbatim}

\section{\LaTeX\ Commands in Arabic environments}

\subsection*{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: \texttt{\textbackslash command\{\textit{opt1}\}\{\textit{opt2}\}\{\textit{arg}\}}—such as \texttt{\textbackslash emph\{\textit{text}\}}, \texttt{\textbackslash textbf\{\textit{text}\}} and the like, is assumed to have Arabic text in its mandatory argument:—

\begin{verbatim}
\abjad{45} kitAbu-hu \textit{fI }'l-\uc{`AdAt-i} \textsuperscript{45}
\end{verbatim}

\begin{verbatim}
\arb{\abjad{45} \rlframebox[1in][s]{kitAbu-hu fI 'l-`AdAti}} \textsuperscript{65}
\end{verbatim}

The same applies to footnotes:—

\begin{verbatim}
\renewcommand{\footnoterule}
\flushleft\rule{.4\textwidth}{.15mm}
\begin{arab}
\textit{'inna 'abI kAna mina '1-muqAtilaT-i:\footnote{al-muqAtilaT-i:}
al-muqAtilaT-i: \textit{wa-kAnat 'ummI min \textit{u.zamAzimaT-i buyUt-i}} \textit{'}l-zamAzimaT-u: \textit{.tA'ifaT-u mina}}\textit{'}l-furs-i.}\footnote{al-zamAzimaT-u: \textit{.tA'ifaT-u mina}}\textit{'}l-furs-i.\footnote{al-muqAtil-Ina.}, wa-kAnat 'ummI min \textit{u.zamAzimaT-i buyUt-i}}\textit{'}l-zamAzimaT-u: \textit{.tA'ifaT-u mina}}\textit{'}l-furs-i.\end{arab}
\end{verbatim}

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.

\texttt{\textbackslash LR\{\textit{arg}\}} is designed to typeset its argument from left to right. It may be used in an Arabic environment, either \texttt{\textbackslash arab\{Arabic text\}} or \texttt{\begin{arab}\textit{Arabic text}} \texttt{\end{arab}}, for short insertions of left-to-right text, or to insert any \LaTeX\ command that would otherwise be rejected by \texttt{arabluatex}, such as commands the argument of which is expected to be a dimension or a unit of measurement.

\texttt{\textbackslash RL\{\textit{arg}\}} does the same as \texttt{\textbackslash LR\{\textit{arg}\}}, but typesets its argument from right to left.

---

\textsuperscript{64}This is odd in Arabic script, but using such features as \texttt{\textbackslash emph} or \texttt{\textbackslash textbf} is a matter of personal taste.

\textsuperscript{65}\texttt{\textbackslash rlframebox} has been adapted from \texttt{\textbackslash framebox} for insertions of right-to-left text.
Even in an Arabic environment, this command may be useful. \LRfootnote{(text)} and \RLfootnote{(text)} typeset left-to-right and right-to-left footnotes respectively in Arabic environments. Unlike \footnote{(text)}, the arguments of both \LRfootnote and \RLfootnote are not expected to be Arabic text. For example, \LRfootnote can be used to insert English footnotes in running Arabic text:

\begin{arab}[fullvoc]
\uc{z}ayd-u\arbnull{ibnu}
\enquote{\arb[trans]{\uc{z}ayd} is the son of \arb[trans]{{`a}mr}}: the second noun is not in apposition to the first, but forms part of the predicate\ldots
\end{arab}

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. arabluatex is not responsible for that, but should it happen, it may be necessary to redefine in the preamble the LaTeX macro \thefootnote like so:—

\renewcommand*{\thefootnote}{\textsuperscript{\LR\arabic{footnote}}}

Another solution is to put in the preamble, below the line that loads arabluatex, the \FixArbFtnmk command. However, for more control over the layout of footnotes marks, it is advisable to use the \FixArbFtnmk package.

The \LRmarginpar command does for marginal notes the same as \LRfootnote does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that \LRmarginpar also works in Arabic environments, but it acts as any other single-argument command inserted in Arabic environments. The general principle laid on the preceding 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{arab}
q\={a}la barzawayhi bn-u `azhar-a, ra`a-u `a.tibbA`i f\={A}ris-a\ldots
\end{arab}

66 See http://ctan.org/pkg/koma-script; read the documentation of KOMA-script for details about the \deffootnotemark and \deffootnote commands.
11.1 New commands

New feature v1.9 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 44, it follows that any command that is found inside an Arabic environment is assumed to have Arabic text in its argument which arabluatex 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{\text{fullvoc}}{#1}}

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

\texttt{\MkArbBreak{\texttt{⟨csv list of commands⟩}}}

The \texttt{\MkArbBreak} command can be used in the preamble to give any command—either new or already existing—the precedence over arabluatex 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:—

\texttt{\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:—

1 \texttt{\MkArbBreak{fvred}}
2 \texttt{\newcommand{\fvred}[1]{\arbcolor{red}{\arb{\text{fullvoc}}{#1}}}}
3 \texttt{\begin{arab}voc}
4 _tumma "intalaqa _dU 'l-qarn-ayni 'il_A 'ummaT-iN 'u_hr_A fI
5 \fvred{((ma.tli`-i 'l-^sams-i))} wa-lA binA'-a la-hum
6 _u'amminu-hum mina 'l-^sams-i.
7 \end{arab}

It must be noted that the arguments, either optional or mandatory, of commands declared with \texttt{\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 \texttt{arb}. These commands themselves may have up to two optional and/or mandatory arguments followed by one optional argument, like so:—

(a) \texttt{\command} (no argument, lowermost combination)
(b) \texttt{\command{⟨opt⟩}} (one optional argument)
(c) \texttt{\command{⟨arg⟩}} (one mandatory argument)
(d) \texttt{\command{⟨opt⟩}{⟨arg⟩}} (one optional and one mandatory argument)
(e) [...] 
(f) \texttt{\command{⟨opt⟩}{⟨opt⟩}{⟨arg⟩}}
(g) \texttt{\command{⟨opt⟩}{⟨opt⟩}{⟨arg⟩}{⟨opt⟩}{⟨opt⟩}} (uppermost combination)

\texttt{\MkArbBreak*}

As said above, \texttt{\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—\texttt{\MkArbBreak*{⟨csv list of commands⟩}}—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.

\texttt{\MkArbBreak*}

It must be noted that \texttt{\MkArbBreak*} must be used with the utmost care and should never be used if \texttt{\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 \texttt{arab} environment. Up to one optional argument may be passed to each nested environment, like so:

\begin{verbatim}
\begin{arab}
\begin{<environment>}[<options>]
<Arabic text>
\end{<environment>}
\end{arab}
\end{verbatim}

In the following example, the quoting package is used:

\begin{verbatim}
\setquotestyle{arabic}
\begin{arab}[fullvoc]
\begin{<environment>}
\texttt{kuAna uc(\texttt{\textquoteleft abu\textquoteright}) uc(\texttt{'1-hu_dayli}) 'ahd_A 'il_A uc(muwaysiN})
\endtext\begin{<environment>}
dajAjatu-'l-latI 'ahdA-ha duUna ma kana
\endtext\begin{<environment>}
yu'rafu bi-'l-'imsAl 'l-sadditi. fa-qala: \texttt{\textquoteleft wa-kayfa va-ayyi \textquoteleft wa-kayfa ya\textquoteleft rifu-hu na\textquoteleft hnu wa-l\textquoteleft a ya\textquoteleft rifu-hu \texttt{\textquoteleft abu\textquoteright}) uc(\texttt{'1-hu_dayli}).}
\endtext\begin{arab}
\end{verbatim}

11.2.1 Lists

Lists environments are also accepted inside the \texttt{arab} environment. One may either use any of the three standard list environments, viz. \texttt{itemize}, \texttt{enumerate} and \texttt{description} or use packages that provide additional refinements such as \texttt{paralist} or \texttt{enumitem}.

To take a first example, should one wish to typeset a list of manuscripts, the \texttt{description} environment can be used like so:
As a second example, the contents of a treatise may be typeset with the standard list environments, like so:—

```latex
\begin{itemize}
\item \textbf{al-fannu 'l-'awwalu} fI .haddi 'l-.tibbi wa-maw.dU`Ati-hi mina 'l-'umUri 'l-.tabI`iyyaTi wa-ya^stamilu 'al_A sittaTi ta`AlImiN
\begin{itemize}
\item \textbf{al-ta`lImu 'l-'awwalu} [wa-huwa fa.slAni]
\end{itemize}
\end{itemize}
\end{itemize}
```

As a third example, abjad-numbered lists can be typeset in conjunction with the enumitem package, like so:—

```latex
\begin{enumerate}
\item \textbf{al-qAnUnu fI 'l-.tibbi}
\item \textbf{al-fannu 'l-'awwalu} fI .haddi 'l-.tibbi wa-maw.dU`Ati-hi mina 'l-'umUri 'l-.tabI`iyyaTi wa-ya^stamilu 'al_A sittaTi ta`AlImiN
\item \textbf{al-ta`lImu 'l-'awwalu} [wa-huwa fa.slAni]
\end{enumerate}
```

\textsuperscript{67}See the documentation of enumitem for more details: \url{https://ctan.org/pkg/enumitem}
From Wright:— 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.

\begin{arab}[fullvoc]
\begin{enumabjad}
\item fa`ala
\item fa``ala
\item fA`ala
\item 'af`ala
\item tafa``ala
\item tafA`ala
\item infa`ala
\item ifta`ala
\item if`alla
\item istaf`ala
\item if`alla
\item if`aw`ala
\item if`awala
\item if`anlala
\item if`anl_A
\item
\end{enumabjad}
\end{arab}
\end{multicols}

From Wright:— 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.

\begin{arab}[fullvoc]
\begin{enumabjad}
\item fa`ala
\item fa``ala
\item fA`ala
\item 'af`ala
\item tafa``ala
\item tafA`ala
\item infa`ala
\item ifta`ala
\item if`alla
\item istaf`ala
\item if`alla
\item if`aw`ala
\item if`awala
\item if`anlala
\item if`anl_A
\item
\end{enumabjad}
\end{arab}
\end{multicols}

\begin{paralist}
\item fa`ala
\item fa``ala
\item fA`ala
\item 'af`ala
\item tafa``ala
\item tafA`ala
\item infa`ala
\item ifta`ala
\item if`alla
\item istaf`ala
\item if`alla
\item if`aw`ala
\item if`awala
\item if`anlala
\item if`anl_A
\end{paralist}

*Wright, see n. 7, i. 29 B–C.

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:—

\usepackage[french]{babel}
\frenchsetup{StandardLists=true}

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:

```latex
\usepackage{csquotes}
\DeclareQuoteStyle{arabic}{\textquotedblright}{\textquotedblleft}{\textquoteleft}{\textquotationmark}
```

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

```latex
\setquotestyle{arabic}
\begin{arab}fa-qAla la-bu ju.hA: \enquote{garIb-uN 'amru-ka yA .sadiqI
't-a-tu.saddiqu '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:

```latex
\begin{arab}
\textcolor{red}{\uc{m}uha_d_dabu \uc{a}bdu \uc{r}a.hImi bnu \uc{a}liyyiN} huwa ^say_hu-nA 'l-'imAmu 'l-.sadru 'l-kabIru 'l-`Alimu 'l-fA.dilu \uc{m}uha_d_dabu \uc{r}a.hImi bnu \uc{a}liyyi bni \uc{h}AmidiN wa-yu`rafu bi-\uc{d}a_hwari.
\end{arab}
```

\arbrtex{provides its own \arbcolor command which is able to render syllabes or diacritics in colors. See sect. 7 on page 33.}
Muhaddabu 'd-Dīni 'Abdu 'r-Raḥīmi bnu ʿAliyy on huwa šayḫu-na l-ʾimāmu ʿz-sādru l-kabīru l-ʾālimu l-fāḍilu Muhaddabu 'd-Dīni 'Abū Muḥammad on 'Abdu 'r-Raḥīmi bnu ʿAliyyi bni Ḥāmid on wa-γu raʃu bi-'d-Daẖwari.

reledmac The two-argument command \edtext{⟨lemma⟩}{⟨commands⟩} is supported inside \begin{arab} \end{arab}.\(^{69}\) As an example, one may get arablatetex and reledmac to work together like so:

1 \begin{numbering}
2 \pstart
3 \begin{arab}
4 wa-ya.ʃrū ta.hta 'l-jild-i
5 \edtext{arb(.sadId-uN)}{\Afootnote{M: \arb(.sadId-aN) E1}}
6 \end{arab}
7 \pend
8 \end{numbering}

11.5 quran

arablatetex 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 sect. 10 on page 43 for more details). Please note that arablatetex takes care of formatting the Arabic: therefore, it is recommended to load the quran package with the nopar option, after arablatetex itself has been loaded, like so:

1 \usepackage{arablatetex}
2 \usepackage[nopar]{quran}

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

1 \begin{txarab}
2 \quransurah[1]
3 \end{txarab}

12 Exporting Unicode Arabic to an external file

New feature v.1.13 arablatetex is able to produce a duplicate of the original .tex source file in which all arbcutx

\(^{69}\)\pstart and \pend are also supported inside the arab environment.
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 require LuaLaTeX to be run several times as will be explained below.

### 12.1 Commands and environments

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

```
\usepackage[export]{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 (enc.) 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 `\txarb` or `\txtrans` 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*`.

**\ArbOutFile** **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.

---

70 See above on page 5 for more information.
71 See above sect. 10 on page 43.
72 `\txtrans` is used internally by several Lua functions to format transliterated Arabic. Therefore, it is not documented.
73 See above on page 45.
74 See above on page 39.
**Exporting Arabic poetry**  Lines of Arabic poetry are exported as described above on page 26 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:—

```
\PackageWarning{arabluatex}{There are still `arabtex` strings to be converted. Please open \⟨jobname⟩\⟨suffix⟩.tex and compile 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):—

```
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:—

```
% preamble:
\usepackage{arabluatex} \% note that `export` has been removed
\renewcommand{\txarb}[1]{#1}
\renewcommand{\txtrans}[1]{\emph{#1}}
\renewcommand{\arbup}[1]{\textsuperscript{#1}}
\% 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 `arabluatex` to come follows:

(a) **Short-term:**

---

75 See above 41 on page 26 for more information.
76 See http://pandoc.org/
i. **TEI xml support**: 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 arabtex; see 5 on page 4.
   ii. Formulate propositions for extending the ArabTeX notation and the transliteration tables. Include them in arabluatex. See sect. 4.9 on page 24.

### 14 Implementation

The most important part of arabluatex relies on Lua functions and tables. Read the .lua files that accompany arabluatex for more information.

```
\RequirePackage{iftex}
```

arabluatex requires LuaLaTeX of course. Issue a warning if the document is processed with another engine.

```
\RequireLuaTeX
```

Declare the global options, and define them:

```
\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}}
\AtEndDocument{\luadirect{arabluatex.closestream()}}
\else\fi
\ExecuteOptionsX{voc}
\ProcessOptionsX\relax
\def\al@mode{voc}
\def\al@mode{fullvoc}
\def\al@mode{novoc}
\def\al@mode{trans}
```

Packages that are required by arabluatex:

```
\RequirePackage{xcolor}
\RequirePackage{luacolor}
\RequirePackage{etoolbox}
\RequirePackage{arabluatex-patch}
\RequirePackage{fontspec}
\RequirePackage{luacode}
\RequirePackage{xparse}
\RequirePackage{adjustbox}
\RequirePackage{xstring}
\RequirePackage{lua-ul}
```

The following boolean will be set to true in RL mode:

```
\providebool{al@rlmode}
```

Here begins the real work: load arabluatex.lua:

```
\luadirect{dofile(kpse.find_file("arabluatex.lua"))}
```
Font setup. If no Arabic font is selected, issue a warning message and attempt to load the Amiri font which is included in TeXlive:

```latex
\AtBeginDocument{\ifedefined\arabicfont\relax\else
PackageInfo{arabluatex}{% string\arabicfont is not defined.\MessageBreak arabluatex will try to load Amiri}% newfontfamily\arabicfont{Amiri}[Script=Arabic]\fi}%
```

\setRL This neutralizes what may be defined by other packages:

```latex
\AtBeginDocument{\def\setRL{\booltrue{al@rlmode}\pardir TRT% \textdir TRT}}
```

\setLR The same applies to \setLR:

```latex
\AtBeginDocument{\def\setLR{\boolfalse{al@rlmode}\pardir TLT% \textdir TLT}}
```

\LR 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:

```latex
\AtBeginDocument{\ifdef{\LR}{\RenewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\reset@font#1\egroup}}{\NewDocumentCommand{\LR}{m}{\bgroup\textdir TLT#1\rmfamily\egroup}}}\SetArbNumbers
```

```
\NewDocumentCommand{\SetArbNumbers}{m}{% luadirect{arabluatex.setnums(\luastringN{#1})}%
\atRL
\LR
\SetArbNumbers
```

\MkArbBreak The \MkArbBreak{⟨csv list of commands⟩} command can be used to give any command—either new or already existing—the precedence over arabluatex inside Arabic environments. It is actually coded in Lua.

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

```latex
\NewDocumentCommand{\MkArbBreak}{s m}{% IfBooleanTF{#1}{\luadirect{arabluatex.mkarbbreak(\luastringN{#2}, "out")}}{\luadirect{arabluatex.mkarbbreak(\luastringN{#2}, "df1t")}}
```

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

```
\NewDocumentCommand{\aemph}{s}{% The ‘starred’ version of this command alway puts the stroke over its argument. As of v1.19, arabluatex uses lua-ult to render the strokes, thus allowing line breaks and manual hyphenation for transliterated Arabic.
\NewDocumentCommand{\aoline}{s}{% \aoline and \auline derive from \newunderlinetype provided by the lua-ult package whereas \aoline*, which uses \overline in math-mode, is better suited for so-called abjad numbers.
```

```
\SetArbNumbers
```

```
\NewDocumentCommand{\SetArbNumbers}{m}{% \luadirect{arabluatex.setnums(\luastringN{#1})}%
```

```
\NewDocumentCommand{\SetArbNumbers}{m}{% \luadirect{arabluatex.setnums(\luastringN{#1})}%
```

```
\MkArbBreak*
```

```
\MkArbBreak*
```

```
\MkArbBreak*
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```

```
\SetArbNumbers
```

```
\atRL
```

```
\LR
```
\newunderlinetype\@aoline{\leaders\vrule height 3ex depth -2.9ex}
\def\aoline{\@ifstar\@aoline\@@aoline}
\def\@aoline#1{\ensuremath{\overline{\mbox{#1}}}}
\newunderlinetype\@auline{\leaders\vrule height -.65ex depth .75ex}
\def\auline#1{\@auline#1}
\AtBeginDocument{\ifdef\aemph{%}
\RenewDocumentCommand{\aemph}{s m}{%}
\IfBooleanTF{#1}{%}
\luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},\textquote{over}))}}%
\else{\luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},\textquote{dflt}))}}%}
\} }
\NewDocumentCommand{\SetInputScheme}{m}{\def\al@input@scheme{#1}}
\SetInputScheme{arabluatex}
\NewDocumentCommand{\arbcolor}{o m}{\textcolor{#1}{#2}}
\arbcolor[⟨color⟩]⟨Arabic text⟩ takes the Arabic text to be colored as argument.
\NewDocumentCommand{\SetTranslitFont}{m}{\def\al@trans@font{#1}}
\SetTranslitFont{\rmfamily}
\SetArbEasy By default, arabluatex applies complex rules to generate euphonic taṣdīd, ‘alif mamdūdah
\SetArbEasy* 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 (b)
on page 15; a new starred version \SetArbDflt* is now available to the user should he
wish to apply them.
\NewDocumentCommand{\SetArbDflt}{s}{%}
\SetArbDflt{dflt}
\NewDocumentCommand{\SetArbEasy}{s}{%}
\SetArbEasy{easy}
\NewDocumentCommand{\SetArbEasy*}{s}{%}
\SetArbEasy*{easynosukun}
\NewDocumentCommand{\SetArbDflt*}{s}{%}
\SetArbDflt*{idgham}
\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.
\NewDocumentCommand{\SetTranslitFont}{m}{\def\al@trans@font{#1}}
\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:

```
\def\al@trans@style{\itshape}%
\NewDocumentCommand{\SetTranslitStyle}{m}{\def\al@trans@style{#1}}
```

\altrfont Finally \altrfont is used internally by arabluatex to store family and shape information about the font to be used for transliterated Arabic.

```
\def\altrfont{\al@trans@font\al@trans@style}
```

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

```
\def\al@trans@convention{dmg}
\NewDocumentCommand{\SetTranslitConvention}{m}{\def\al@trans@convention{#1}}
```

\arbup \NoArbUp \ArbUpDflt \SetArbUp By default, \arbup is set to \texttt{textsuperscript}. This is how the \texttt{tanwîn} that takes place at the end of a word should be displayed in \texttt{dmg} mode. \NoArbUp may be used either in the preamble or at any point of the document in case one wishes to have the \texttt{tanwîn} 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 \texttt{tanwîn} is displayed: this command takes the formatting directives as argument, like so: \SetArbUp\{\texttt{code}\}.

```
\NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
\NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt}
\NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\al@arbup}{m}{##1}}
\NewDocumentCommand{\SetArbUp}{m}{\RenewDocumentCommand{\al@arbup}{m}{#1}}
```

\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.

```
\NewDocumentCommand{\uc}{m}{{\luadirect{\tex.sprint(arabluatex.uc(\luastringN{#1}))}}}
```

\uc \uc can be used safely in all of the modes that are provided by 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. 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.

```
\NewDocumentCommand{\prname}{s m}{%
  \bgroup\SetTranslitStyle{\relax}%
  \txtrans{#2}%(#1)}%  \egroup}
```

\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.

```
\NewDocumentCommand{\prname*}{s m}{%  \bgroup\SetTranslitStyle{\relax}%(#1)}%  \IfBooleanTF{#1}{\txtrans{#2}%(#2)}%  \arb{#2}%(#2)}%  \egroup}
```

57
\texttt{txarb} \texttt{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. \texttt{txtrans} \texttt{txtrans} is used internally by several Lua functions to insert transliterated Arabic text. Therefore, it is not documented.

\begin{verbatim}
108 \NewDocumentCommand{\txarb}{+m}{% 
109 \ifvmode\leavevmode\fi% 
110 \bgroup\textdir TRT\arabicfont#1\egroup}
111 \NewDocumentCommand{\txtrans}{+m}{% 
112 \bgroup\textdir TLT\altrfont#1\egroup}
\end{verbatim}

\texttt{txarab (env.)} The \texttt{txarab} environment does for paragraphs the same as \texttt{txarb} does for short insertions of utf8 Arabic text.

\begin{verbatim}
113 \NewDocumentEnvironment{txarab}{}{% 
114 \par%
115 \booltrue{al@rlmode}%
116 \pardir TRT\textdir TRT\arabicfont}{\par}
\end{verbatim}

\texttt{txarabtr (env.)} \texttt{txarabtr} environment is used internally by several Lua functions to insert running paragraphs of transliterated Arabic text Therefore, it is not documented.

\begin{verbatim}
117 \NewDocumentEnvironment{txarabtr}{}{% 
118 \par%
119 \pardir TLT\textdir TLT\% 
120 \altrfont}{\par}
\end{verbatim}

\texttt{\arb} The \texttt{\arb} command detects which Arabic mode is to be used, either globally if no option is set, or locally, then passes its argument to the appropriate Lua function.

\begin{verbatim}
121 \NewDocumentCommand{\arb}{O{\al@mode} +m}{% 
122 \edef\@tempa{#1}% 
123 \ifx\@tempa\al@mode@voc%
124 \ifvmode\leavevmode\fi% 
125 \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont% 
126 \luadirect{tex.sprint(arabluatex.processvoc(\luastringN{#2},
127 \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup% 
128 \else%
129 \ifx\@tempa\al@mode@fullvoc%
130 \ifvmode\leavevmode\fi% 
131 \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont% 
132 \luadirect{tex.sprint(arabluatex.processfullvoc(\luastringN{#2},
133 \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup% 
134 \else%
135 \ifx\@tempa\al@mode@novoc%
136 \ifvmode\leavevmode\fi% 
137 \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont% 
138 \luadirect{tex.sprint(arabluatex.proc
139 \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))\egroup% 
140 \else%
141 \ifx\@tempa\al@mode@trans%
142 \bgroup\textdir TLT\%
143 \luadirect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
144 \luastringO{\al@trans@convention},
145 \luastringO{\al@arb@rules},
146 \luastringO{\al@input@scheme}))}\egroup% 
147 \else%
\end{verbatim}
\arbmark \arbmark\{\langle\textit{shorthand}\rangle\} 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 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})}}

\texttt{arab (env.)} The \texttt{arab} environment does for paragraphs the same as \texttt{arb} does for short insertions of Arabic text.

\NewDocumentEnvironment{arab}{!O{\al@mode} +b}{}% 
{\par\edef\@tempa{#1}{}% 
  \ifx\@tempa\al@mode@voc% 
    \booltrue{al@rlmode}% 
    \bgroup\pardir\textit{TRT}\textdir\textit{TRT}\arabicfont% 
    \luadirect{tex.sprint(arabluatex.processvoc(\luastringN{#2}, \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup% 
  \else% 
    \ifx\@tempa\al@mode@fullvoc% 
      \booltrue{al@rlmode}% 
      \bgroup\pardir\textit{TRT}\textdir\textit{TRT}\arabicfont% 
      \luadirect{tex.sprint(arabluatex.processfullvoc(\luastringN{#2}, \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup% 
    \else% 
      \ifx\@tempa\al@mode@novoc% 
        \booltrue{al@rlmode}% 
        \bgroup\pardir\textit{TRT}\textdir\textit{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\pardir\textit{TLT}\textdir\textit{TLT}% 
          \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\fi\fi\fi\fi\fi\fi% 
\par}

\texttt{arabverse (env.)} The \texttt{arabverse} environment may receive different options: \texttt{mode}, \texttt{width}, \texttt{gutter}, \texttt{metre}, \texttt{color}, \texttt{utf}, \texttt{delim} and \texttt{export}; all of them are defined here just before the \texttt{arabverse} environment.

\newlength{\al@bayt@width} 
\newlength{\al@gutter@width}
Then follows the environment itself:

\NewDocumentEnvironment{arabverse}{!O{}}{
  \bgroup\setkeys[al]{verse}[width, gutter, color, utf, delim, metre]{#1}\n  \if@pkg@export\ifal@verse@export\ArbOutFile{\begin{arabverse}}\n    % \ifx\al@mode\al@mode@trans\n      \luadirect{ arabluatex.tooutfile(\luastringN{[#1]})}\n    % \else\setRL\fi%\n    \IfSubStr[1]{#1}{utf}\n      {\luadirect{arabluatex.tooutfile(\luastringN{[#1, utf]})}}\n    % \fi\n  \else\fi\n  \egroup\centering\noindent\bgroup\setkeys[al]{verse}[metre]{#1}\n  % \ifx\al@mode\al@mode@trans\n    % \ifal@verse@utf \setRL\else\setLR\fi%\n    \ifal@verse@utf\n      \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi%\n    \else\n      \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi%\n    \fi\n  \addtolength{\al@verse@twidth}{2\al@bayt@width}\n  \addtolength{\al@verse@twidth}{\al@gutter@width}\n  \arab@v@export[#1]\n  \egroup\setkeys[al]{verse}[width, gutter, color, utf, delim, mode, metre]{#1}\n  \if@pkg@export\ifal@verse@export\ArbOutFile{\end{arabverse}}\n  \else\fi\n  \else\fi\n}\n
\presetkeys[al]{verse}{utf=false, delim=false}{

\setkeys[al]{verse}[width, gutter, color, utf, delim, mode, export]{#1}\n  %\ifdefined\al@verse@metre@value\hfill\al@verse@metre@value\fi\n\egroup\bgroup\setkeys[al]{verse}[width, gutter, color, utf, delim, mode, metre]{#1}\n  \if@pkg@export\ifal@verse@export\ArbOutFile{\end{arabverse}}\n  \else\fi\n  \else\fi\else\fi\egroup}
ayt Each verse consists of two hemistichs; therefore the \bayt command takes two arguments, the first receives the \textit{sadr} and the second the \textit{ajuz}. That two subsequent hemistichs should be connected with one another is technically named \textit{tadwir}. In some of these cases, the hemistichs may be connected by a prominent horizontal flexible stroke which is drawn by the \texttt{\alverse\stroke} command.

\texttt{\StretchBayt[(true|false)]} Allows to remove stretching and undesirable warping effect from Arabic lines of poetry. This command accepts one fixed optional argument, either \texttt{true} or \texttt{false}, and may be used either in the preamble or at any point of the document. By default, it is set to \texttt{true}.

\texttt{\SetHemistichDelim} A hemistich delimiter also may be defined. By default, it is set to the ‘star’ character: \texttt{*}. The \texttt{\SetHemistichDelim{delimiter}} command can be used at any point of the document to change this default setting.

\begin{lstlisting}[language=TeX]
\newif\ifal@warp@bayt
\al@warp@bayttrue
\NewDocumentCommand{\StretchBayt}{O{true}}{
  \edef\oarg@true{true}
  \edef\oarg@false{false}
  \edef@tempa{#1}
  \ifx@tempa\oarg@true\al@warp@bayttrue
  \else
    \ifx@tempa\oarg@false\al@warp@baytfootalse
    \else
      \PackageError{arabluatex}{\string\StretchBayt\space must be either 'true' or 'false'}{}
    \fi
  \fi
}
\NewDocumentCommand{\arb@utf}{m}{% 
  \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
\def\al@hemistich@delim{*}
\NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
\def\al@verse@stroke{\leavevmode\xleaders\arb{--}\hfill\kern0pt}
\providebool{ekd@state}
\NewDocumentCommand{\bayt}{t+s+m+om}{% 
  \IfBooleanTF{#1}{% 
    \ifekd@state 
    \leavevmode 
    \stepcounter{ekd@lab} 
    \zlabel{ekd:\theekd@lab} 
    \luadirect{ekdosis.storeabspg(} \luastring{\zref@extract{ekd:\theekd@lab}{abspage}}) \}% 
  }{\relax}% 
  \IfBooleanTF{#2}{\relax}{\relax}\
  \ifdefined\savenotes\savenotes\else\fi%
  \edef\al@tatweel{--}%
  \ifal@warp@bayt%
    \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
  \else%
    \makebox[\al@bayt@width][s]{\arb@utf{#3}}%
  \fi%
  \IfNoValueTF{#4}{% 

\end{lstlisting}
\arind \arind{(\textit{root})} is a command specialized in the construction of indexes. As a mandatory argument, it takes the Arabic root under which a given word is to be indexed. Additionally, it may receive three optional ‘named’ arguments: \textit{index}, \textit{root} and \textit{form}. 

\NewDocumentCommand{\SetDefaultIndex}{m}{
  \edef\@tempa{#1}
  \ifx\@tempa\empty
    \def\al@default@index{\jobname}
  \else
    \def\al@default@index{#1}
  \fi
}

\def\al@index@mode{\al@mode}
\NewDocumentCommand{\SetIndexMode}{m}{
  \def\al@index@mode{#1}
}
\define@cmdkeys[al]{index}{alind@}{index,root,form,pipe}
\NewDocumentCommand{\arind}{o m}{
  \IfNoValueTF{#1}{
    \ifdefined\al@default@index
      \csname index\endcsname[\al@default@index]{#2}
    \else
      \csname index\endcsname{#2}
    \fi
  }{
    \bgroup
    \IfNoValueTF{#1}{
      \\csname index\endcsname[\al@default@index]{#2}
    }{
      \csname index\endcsname[\al@index@mode]{#2}
    }
  }
}
\abjad{\abjad{\langle number\rangle}} 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 \texttt{arabluatex.lua} for more information.
\ayah \ayah{⟨number⟩} prints up to 3-digit numbers inside ‘end of Ayah’ sign (U+06DD) or inside parentheses depending on the mode which is selected.

\arbn\null The \arbn\null 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.

\abraces \abraces{⟨Arabic text⟩} puts its argument between braces. This macro is written in Lua and is dependent on the current value of \texttt{tex.textdir}.

\LR\marginpar \LR\marginpar is supposed to be inserted in an Arabic environment. It typsets his argument in a marginal note from left to right.

\LR\footnote \LR\footnote and \RL\footnote are supposed to be used in Arabic environments for insertions of non Arabic text. \LR\footnote typsets its argument left-to-right...

\RL\footnote while \RL\footnote typsets its argument left-to-right.

\FixArbFtnmk In the preamble, just below \texttt{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.

\SetArbOutSuffix By default, \texttt{_out} is the suffix to be appended to the external file in which \texttt{arabluatex} exports Unicode in place of \texttt{arabtex} or \texttt{buckwalter} strings. Any other suffix may be set with \SetArbOutSuffix{⟨suffix⟩}.

\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}. 

64
The \arb exporting environment processes and prints its argument unchanged to the current .pdf file. Additionally, if \arblatex is loaded with the export option, this argument is exported to the external selected .tex file with Unicode in place of the original arabtex or buckwalter strings.

\arb exporting (env.) The \arb exporting environment does for \arbverse the same as \arb exporting. It is used internally by \arbverse.

\arb@v@export (env.) The \arb@v@export environment does for \arbverse the same as \arb exporting. It is used internally by \arbverse.
\arbpardir \arbpardir is automatically inserted by \arabluatex at the beginning of Arabic paragraphs converted to Unicode so that they are printed in the right direction.

\NewDocumentCommand{\arbpardir}{}{{\ifx\al@mode\al@mode@trans\setLR\else\setRL\fi}}

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)\string\footnote\space is not allowed} inside \string\RL{} and \string\RL{} commands}{Get rid of the surrounding \string\RL{} or \string\LR{} command.}
\newcommand{\al@wrong@mark}{\al@warning{Unknown Arabic mark in \string\arbmark{}. Replaced with\MessageBreak <??>. Please check your code}}

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 .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 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 Lua\TeX primitives \bodydir and \pagedir will eventually allow us to get rid of this:
\def\list#1#2{%}
\ifnum \@listdepth >5\relax
\@toodeep
\else
\global \advance \@listdepth \@ne
\rightmargin \z@ \listparindent \z@ \itemindent \z@ 
\csname @list\romannumeral \the \@listdepth \endcsname \def \@itemlabel {#1}%
\let \makelabel \@mklab \@nmbrlistfalse
#2\relax
\@trivlist \parskip \parsep \parindent \listparindent
\advance \linewidth -\rightmargin \advance \linewidth -\leftmargin
patch begins:
This is adapted from Vafa Khalighi’s bidi package. Thanks to him.

15 References


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


16 Change History

\[V1.0\]
General: Initial release .................. 1

\[V1.0.1\]
General: Minor update of the documentation .................. 1

\[V1.1\]
\textbackslash abjad: New and more flexible \textbackslash abjad command. .................. 63

\[V1.2\]
\textbackslash SetArbEasy: New \textbackslash SetArbEasy/\textbackslash SetArbDflt for ‘modern’ or ‘classic’ Arabic styles. 56

\[V1.3\]
\textbackslash arabup: \textit{ṭārāb} is now written as superscript text in \texttt{dng} mode by default. .................. 57

\[V1.4\]
\textbackslash SetInputScheme: \textbackslash SetInputScheme can be used to process other input schemes such as ‘Buckwalter’ .... 56
\textbackslash SetTranslitFont: For selecting a specific font for transliterated texts 56

\[V1.4.3\]
\textbackslash abraces: New \textbackslash abraces command which expresses its argument between braces. .................. 64

\[V1.4.4\]
\textbackslash SetArbEasy*: this starred version discards the sukūn in addition to what is already discarded by \textbackslash SetArbEasy. .................. 56

\[V1.5\]
General: Compatibility with the \texttt{quran} package .................. 51
Environments may be nested inside the \texttt{arab} environment ........... 47
\texttt{txarab}: New \texttt{txarab} environment for typesetting running paragraphs in Unicode Arabic .................. 58

\[V1.6\]
\texttt{arabverse}: New environment \texttt{arabverse} for typesetting Arabic poetry .................. 59
\textbackslash bayt: New macro \textbackslash bayt for typesetting each verse inside the \texttt{arabverse} environment .................. 61
\textbackslash SetArbDflt*: This starred version applies the assimilation rules in addition to what \textbackslash SetArbDflt already does. .................. 56
\textbackslash SetHemistichDelim: New \textbackslash SetHemistichDelim command for changing the default delimiter between hemistichs .................. 61
\[V1.7\]
\textbackslash arbnul: New \textbackslash arbnul command for putting back on any contextual analysis rule broken by other commands. .................. 64

\[V1.8\]
General: \texttt{arabica} transliteration standard is now supported .... 37

\[V1.8.5\]
General: Six additional Persian characters are now available .... 9

\[V1.9\]
\textbackslash MkArbBreak: New \textbackslash MkArbBreak command for inserting user-defined macros in Arabic environments .......... 55
\[V1.9.2\]
\textbackslash \texttt{aemph}*: Starred version which always puts the stroke over its argument ........ 55
\[V1.10\]
General: \texttt{\uc} supersedes \texttt{\cap} ........ 38
\texttt{\prname}: New command for typesetting Arabic proper names in transliteration .................. 57

\[V1.11\]
\texttt{\arbmark}: New command for inserting additional marks in Arabic environments .................. 59
\texttt{\newarbmark}: Allows defining additional sets of Arabic marks .................. 59

\[V1.12\]
General: \texttt{\abjad} can now process \texttt{L\LaTeX} counters .................. 24
\texttt{\arbcolor}: Standard color command for Arabic environments .......... 56
\texttt{\MkArbBreak*}: ‘starred’ version which closes Arabic environments before processing declared commands ........ 55

\[V1.13\]
\texttt{\arabexport}: Processes and print its argument in the current file and exports it in full Unicode in the external selected \texttt{.tex} file .......... 65
\texttt{\arabverse}: New options \texttt{color} and \texttt{export} to \texttt{arabverse} environment .......... 59
\texttt{\arbmark}: New optional argument: either \texttt{rl} or \texttt{lr} ........ 59
\texttt{\ArbOutFile}: Silently exports its argument in the selected external file. 64
\arbpardir: Sets the direction of Arabic paragraphs once they are converted to Unicode. 66
\prname: Renders proper names already converted to Unicode in upright roman style. 57
\SetArbOutSuffix: Sets a suffix to be appended to the filename of the external Unicode file. 64
\ayah: Prints End of Ayah sign. 63
\aemph: Now uses ulem. 55
\arind: New command \arind for building indexes. 62
\SetArbNumbers: selects Indian or Arabic numbers. 55

17 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.