arabluatex
ArabTEX for LuaL
Robert Alessi
mailto:alessi@robertalessi.net
2023/05/21 – v1.21-dev

Contents

License and disclaimer 2

1 Introduction 3
1.1 arabluatex is for LuaL ... 4

2 The basics of arabluatex 4
2.1 Activating arabluatex ... 4
2.2 Font setup ... 4
2.2.1 Classic contrasted with modern typesetting of Arabic 5
2.3 Typing Arabic ... 7
2.3.1 Local options ... 8

3 Standard ArabTEX input 8
3.1 Consonants ... 8
3.2 Additional characters ... 9
3.3 Vowels ... 10
3.3.1 Long vowels ... 10
3.3.2 Short vowels ... 11

4 arabluatex in action 11
4.1 The vowels and diphthongs ... 11
4.1.1 Underlining words or numbers ... 25
4.2 Other orthographic signs ... 13
4.2.1 The 'pipe' character (|) ... 21
4.3 Special orthographies ... 18
4.4 Quoting ... 19
4.4.1 Quoting the hamzah ... 20
4.5 The ‘pipe’ character (|) ... 21
4.6 Putting back on broken contextual analysis rules ... 21
4.7 Stretching characters: the ḍaffles ... 23
4.7.1 Numerical figures ... 23
4.8 Digits ... 23
4.8.2 The ābjad ... 23
4.9 Additional characters ... 24
4.10 Arabic emphasis ... 24

5 Arabic poetry 25
5.1 Example ... 25

6 Special applications 30
6.1 The Qur’an ... 32
6.2 Linguistics ... 30
6.3 Brackets ... 30
6.4 Additional Arabic marks ... 30
6.5 The ‘Zero width joiner’ character (U+200D) ... 31

7 Color 33
7.1 Tricks of the trade ... 33
7.2 Diacritics ... 35
7.3 The name of God ... 36
7.4 The conjunctive لَوْ ... 36
7.5 The definite article and the euphonic tašdīd ... 36

8 Transliteration 37
8.1 Additional note on ḏg convention ... 39
8.2 Examples ... 41

9 Buckwalter input scheme 41
9.1 ‘base’, ‘xml’ and ‘safe’ schemes ... 42
9.2 Transliteration ... 43

10 Unicode Arabic input 43

11 LATEX Commands in Arabic environments 44
11.1 New commands ... 46
11.2 Environments ... 47
11.2.1 Lists ... 47
11.3 Pronoun 'I' ... 48

2023/05/21 – v1.21-dev
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.¹

arabluatex -- ArabTeX for LuaLaTeX
----------------------------------------------------------

Copyright (c) 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023
Robert Alessi <alessi@robertalessi.net>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF

¹More information about the OpenBSD policy to which arabluatex adheres: https://www.openbsd.org/policy.html.
1 Introduction

In comparison to Prof. Lagally’s outstanding Arab\TeX,\textsuperscript{2} ArabLua\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 LuaL\TeX users.

\texttt{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 L\TeX. To achieve that, \texttt{arabtex} provided—and still does—an Arabic font in \textit{Nasḥī} 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”\textsuperscript{3}. Even if the same can be said about Unicode, Arab\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 particularly abundant. It must also be said that most text editors have trouble in displaying Arabic script connected with preceding or following L\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 LEFT-TO-RIGHT or RIGHT-TO-LEFT MARKS (U+200E, U+200F), RTL/LTR “embed” characters (U+2028, U+202A) and RLO/LRO “bidi-override” characters (U+202E, U+202D).\textsuperscript{4} Nonetheless, it remains that inserting all the time these invisible characters in complex documents rapidly becomes confusing and cumbersome.

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

\textsuperscript{2}See \url{http://ctan.org/pkg/arabtex}


\textsuperscript{4}Gáspár Sinai’s Yudit probably has the best Unicode support. See \url{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, ArabTeX input can be full vocalized Arabic (scriptio plena), vocalized Arabic or non-vocalized Arabic (scriptio defectiva); it further can be transliterated into whichever romanization standard the user may choose.

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

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

1.1 arabluatex is for LuaLaTeX

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

In comparison to arabxetex, arabluatex works in a very different way. The former relies on the TECkit engine which converts ArabTeX input on the fly into Unicode Arabic script, whereas the latter passes ArabTeX input on to a set of Lua functions. At first, LaTeX commands are taken care of in different ways: some, as \textbf, \textit and the like are expected to have Arabic text as arguments, while others, as \LR, for “left-to-right text”, are not. Then, once what is Arabic is carefully separated from 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 ArabTeX input yet to be processed—for further processing.

2 The basics of arabluatex

2.1 Activating arabluatex

arabluatex is loaded the usual way:

\usepackage{arabluatex}

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

Font setup Any Arabic font can be defined to be used with arabluatex. For example, assuming that fontspec is loaded, this line may be inserted in the preamble, just above the line that loads arabluatex:

\usepackage{fontspec}

\setmainfont{Your Font Name}

To date, both packages support Arabic, Maghribi, Urdu, Pashto, Sindhi, Kashmiri, Uighuric and Old Malay; in addition to these, arabtex also has a Hebrew mode, including Judeo-Arabic and Yiddish.
\newfontfamily\arabicfont{⟨fontname⟩}[Script=Arabic]

where ⟨fontname⟩ is the standard name of the Arabic font to be used.

By default, if no Arabic font is selected, arabluatex will issue a warning message and attempt to load the Amiri font like so:

\newfontfamily\arabicfont{Amiri}[Script=Arabic]

Rem. a By default Amiri places the kasrah in combination with the taṣlid below the consonant, like so: ِّـ.

That is correct, as at least in the oldest manuscripts ّـِ may stand for َّـِ as well as ِّـِ. See Wright. 7

The placement of the kasrah above the consonant may be obtained by selecting the ss05 feature of the Amiri font, like so:—

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

Other Arabic fonts may behave differently.

Rem. b \newfontfamily can be used to have either Indian or Arabic numbers printed. See on page 23 for more information.

2.2 Options

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

\textbf{voc}

In this mode, which is the one selected by default, every short vowel written generates its corresponding diacritical mark: dammah (ُـ), fatḥah (َـ) and kasrah (ِـ). If a vowel is followed by N, viz. ⟨uN, aN, iN⟩, then the corresponding tanwīn (ٌـ, اًـ, ًﺓ, ىًَـ or ٍـ) is generated. Finally, ⟨u, a, i⟩ at the commencement of a word indicate a “connective ʾalīf” (ʿalifu ʾ-waṣli), but voc mode does not show the waṣlah above the ʾalīf; instead, the accompanying vowel may be expressed at the beginning of a sentence (ِﺍَﺍُﺍ).

\textbf{fullvoc}

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

\textbf{novoc}

None of the diacritics is showed in novoc mode, unless otherwise specified (see “quoting” technique below sect. 4.4 on page 19).

\textbf{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):

\textbf{dmg/dmg+} Deutsche Morgenländische Gesellschaft dmg is selected by default;

\textbf{loc} Library of Congress;

\textbf{arabica} Arabica.

More standards will be included in future releases of arabluatex.

\textbf{export} export=true|false

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

6Khaled Hosny, \textit{Amiri} (Dec. 13, 2017), \url{http://www.amirifont.org/}.


8See the documentation of amiri, \textit{ibid.}, 6.
2.2.1 Classic contrasted with modern typesetting of Arabic

*New feature v1.2* By default, arablatex 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 (Ł), then a mere hamzah is written on the line, and a maddah is placed over the ‘alif, like so:—

  \[\text{samā'ūn: } \text{samā'} \wedge, \text{ŷā'a, yatās'ūn: yatāsā'ūn} \]

  (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 ‘e‘mēz and the like, the hamzah alone is not written over the letter yā‘ with no diacritical points below as in ‘e‘mēz, 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, \SetArbDflt between \SetArbEasy and its ‘starred’ version \SetArbEasy* is that the former keeps the sukūn that is generated by the fullvoc mode, while the latter 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.\footnote{For an example, see sect. 5.1 on page 28.}

Examples follow:—

(a) \SetArbDflt:

i. voc \[\text{wāmat ṣustas'ā: qabla an yūtimma kitābahu fī nuqūm 's-samā'}\]

ii. fullvoc \[\text{wāmat ṣustas'ā: qabla an yūtimma kitābahu fī nuqūm 's-samā'}\]

iii. trans wa-māta 'stisqā'\wedge gabla an yutimma kitābahu fī nuqūm 's-samā'\wedge

(b) \SetArbDflt*:

i. voc \[\text{wāmat ṣustas'ā: qabla an yūtimma kitābahu fī nuqūm 's-samā'}\]

ii. fullvoc \[\text{wāmat ṣustas'ā: qabla an yūtimma kitābahu fī nuqūm 's-samā'}\]

iii. trans wa-māta 'stisqā'\wedge gabla ay yutimma kitābahu fī nuqūm 's-samā'\wedge

(c) \SetArbEasy:

i. voc \[\text{wāmat ṣustas'ā: qabla an yūtimma kitābahu fī nuqūm 's-samā'}\]

ii. fullvoc \[\text{wāmat ṣustas'ā: qabla an yūtimma kitābahu fī nuqūm 's-samā'}\]

iii. trans wa-māta 'stisqā'\wedge gabla an yutimma kitābahu fī nuqūm 's-samā'\wedge

(d) \SetArbEasy*:

i. voc \[\text{wāmat ṣustas'ā: qabla an yūtimma kitābahu fī nuqūm 's-samā'}\]

\footnote{Note that in old mss. such forms as ‘e‘mēz, ‘e‘mēz 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), i. 24 D.}
2.3 Typing Arabic

\arb 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 (\arb{.hurUf-u 'l-hijA'-i}, \arb{.hurUf-u 'l-tahajjI}, \arb{al-.hurUf-u 'l-hijA'iyyaT-u}, or \arb{.hurUf-u 'l-mu`jam-i}) 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:—

XVI. ُلِعﺍَوَف .*

1. ٌلَعاَف; as ٌمَتاَخ a signet-ring, ...

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

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

\begin{arab}
[...]
\end{arab}

\begin{arab}
'at_A .sadIquN 'il_A ju.hA ya.tlubu min-hu .himAra-hu
\end{arab}
Àنَّ صَدِيقٍ إِلَى بِهِ يَا عِلَيْكُم وَقَلِ عِنْدَكُمْ أَجْرَةٍ. قَالَ لَهُ: "أَنَا أَسْتَقِيمُ أَنْ أُحْكَمَ لَكُمْ وَأُجْرِكُمْ حَيَاةٌ فُرُدٌ تَّيْمُهَا لِيْسَ هَذَا الْيَوْمُ". وَقَبْلَ أَنْ يَنْحَأَ لَهُ عِنْدَكُمْ كَلَّامَهُ بِأَذْلِكَ الْحَامِرَ يَنْبِيِّقُ إِلَيْهِ، قَالَ لَهُ صَدِيقُهُ: "إِنَّى أَحْمَرُ حَاكِمَكَ بِبَيْنِكُمْ، أَنْحَأَ لَهُ حَيَاةٌ غَرِيبٌ أَرْكِزُ إِلَى أَذْلِكَ الحَامِرَ وَكَبِيرِيْن".

### 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—

```latex
\arb[voc]{{Arabic text}};
\arb[fullvoc]{{Arabic text}};
\arb[novoc]{{Arabic text}};
\arb[trans]{{Arabic text}}.
```

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

### 3 Standard ArabTEx input

#### 3.1 Consonants

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

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration(^{12})</th>
<th>ArabTEx notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ٌّ</td>
<td>'u, a, i</td>
<td>'u or 'a or 'i</td>
</tr>
<tr>
<td>ب</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>ت</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

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

\(^{13}\)See below, Rem. a. For ʾalif as a consonant, see Wright, A Grammar of the Arabic Language, i. 16 D. The hamzah itself is encoded \(<’>\) and may be followed by either (u, a) or (i). See below sect. 4.2 on page 13.
Table 1: Standard ArabTeX (consonants)

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>ArabTeX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ج</td>
<td>ḡ</td>
<td>g or j</td>
</tr>
<tr>
<td>ح</td>
<td>ḥ</td>
<td>h</td>
</tr>
<tr>
<td>خ</td>
<td>ḫ</td>
<td>h</td>
</tr>
<tr>
<td>د</td>
<td>ḍ</td>
<td>d</td>
</tr>
<tr>
<td>ز</td>
<td>ṯ</td>
<td>d</td>
</tr>
<tr>
<td>ر</td>
<td>ṛ</td>
<td>r</td>
</tr>
<tr>
<td>س</td>
<td>š</td>
<td>s</td>
</tr>
<tr>
<td>ك</td>
<td>ƙ</td>
<td>s</td>
</tr>
<tr>
<td>ل</td>
<td>ḫ</td>
<td>s</td>
</tr>
<tr>
<td>ن</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>ه</td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>و</td>
<td>w</td>
<td>w</td>
</tr>
<tr>
<td>ي</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>a</td>
<td>ah</td>
<td>a</td>
</tr>
</tbody>
</table>

Rem. a Please note that in all cases of elision, the 'alif ‘l-wasli 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 ‘l-wasli see sect. 4.2 on page 16.

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

Rem. b The letter ی with two points below, ی�, may also be written without diacritical points as ی. When it is used as a consonant, it is encoded یَ, where a recalls the fatḥah placed above the preceding letter in vocalized Arabic, like so: qaY'uN ṭan qaY'uN, saY'uN ṭan saY'uN, saY'aN ṭan saY'aN.

The same result may be achieved by encoding this letter as ṭ, like so: qa.ṭan qa.ṭan, qa.ṭan qa.ṭan.

3.2 Additional characters

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

\[14\text{For the letter ى with no diacritical points below, see Rem. b below.}\]
### 3.3 Vowels

#### 3.3.1 Long vowels

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

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>ArabTEX notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ا</td>
<td>ā ā ā</td>
<td>A</td>
</tr>
<tr>
<td>و</td>
<td>ā ā ā</td>
<td>U</td>
</tr>
<tr>
<td>ی</td>
<td>ĕ ĕ ĕ</td>
<td>Ī</td>
</tr>
<tr>
<td>ی</td>
<td>ā ā ā</td>
<td>Ī</td>
</tr>
<tr>
<td>ی</td>
<td>ā ā ā</td>
<td>_a</td>
</tr>
<tr>
<td>ی</td>
<td>ĕ ĕ ĕ</td>
<td>_i</td>
</tr>
</tbody>
</table>

Table 3: Standard ArabTEX (long vowels)

Rem. a The long vowels ā, ĕ, ī, otherwise called *ḥurūf 'l-madda‘, the letters of prolongation, involve the placing of the short vowels a, u, i before the letters ḩ, ṭ, ṭ respectively. arabluatex does that automatically in case any from voc, fullvoc or trans modes is selected e.g. ُلاَق qāla, ُليِق qīla, ُلوُقَي yaqūlu.

---

15 See below sect. 8 on page 37.
16 The characters that are listed in this table are not included in this standard. However, as arabica is based on dmg, the dmg equivalents have been used here.
18 See 17.
19 See below sect. 8 on page 37.
20 For the letter ی with no diacritical points, see Rem. c. below.
21 = al-ʾalif ‘l-maqṣūrat.
Defective writings, such as ـٰ, al-ʾalif and ʾl-mahḏūfat, or defective writings of ā and ī are encoded ٰ and ٰ respectively, e.g. َْٰ лকُّةٰ اَلِّيُّمُعِبَدٰ for Ḥuḏayfat bn ʾl-Yamānī, etc.

The letter ی 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 i recalls the kasrah placed below the preceding letter in vocalized Arabic, like so: یِل lī, yamšī.

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></td>
<td>dmg+ loc arabica</td>
<td></td>
</tr>
<tr>
<td>َـ</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>ُـ</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>ِـ</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>ًـ</td>
<td>an</td>
<td>an</td>
</tr>
<tr>
<td>ِـ</td>
<td>un</td>
<td>un</td>
</tr>
<tr>
<td>ٍـ</td>
<td>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 fātihah, dāmmah 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⟩:

---

22See below sect. 8 on page 37.
23See below sect. 4.4 on page 19.
Long vowels They are written ⟨U, A, I⟩:

qāla َلاَق qāla, bīʿa َعيِب bīʿa, ṭūr ٌﺭوُط ṭūr, ṭīn ٌنيِط ṭīn, murūʿat ٌﺓَﺀوُرُم murūʿat.

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

al-fat ىَتَفلَﺍ al-fatā, al-maqh ىَهقَملَﺍ al-maqhā,ʾil ﺇ ىَل ʾilā.

ʾalif otiosum Said ʾalif “ʾl-wiqāyat”, “the guarding ʾalif”, after و at the end of a word, both when preceded by dammah and by fatha is written (UA) or (aW, aWA):

naṣarū ﺍوُرَصَن naṣarū, katabū ﺍوُبَتَك katabū, yaġzū ﺍوُزغَي yaġzū, ramaw ﺍْوَمَﺭ ramāw, banaw ﺍْوَنَب banāw.

ʾalif maḥḏūfah and defective ِٰ They are written (_a, _i _u):


Silent Some words ending with ِٰ are usually written ِٰوَ or ِٰوَة instead of ِٰة ِ ظاَءُهَٰة for ِٰة ِ ظاَءُهَٰة. Long vowels ⟨U, I⟩ shall receive no sukūn after a ʾalif maḥḏūfah and are discarded in trans mode:

ḥayāt ِٰوَة ḥayāt, ṣalāt ِٰوَة ṣalāt, miškāt ِٰوَة miškāt, tawrāt ِٰوَة tawrāt.

And so also: ar-ribāt ُةيٰبِّرلَﺍ ar-ribāt.

ʿAmr and the silent To that name a silent ِٰ is added to distinguish it from ʿUmar: 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-uU bn-u mu.hammadiN ʿAmr u bn Muhammad in Muḥammad u bn ʿAmr i bN iḪālidN.

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


4.2 Other orthographic signs

tāʾ marbūṭah It is written ⟨T⟩:

hamzah It is written ⟨⟩, its carrier being determined by contextual analysis. In case one wishes to bypass this mechanism, he can use the “quoting” feature that is described below in sect. 4.4 on page 19.

Initial hamzah: ʿasaduN ʾāsaduN, ʿu ihtuN ʾ iḥtuN, ʾiqlīduN ʾiqlīduN, ʾiqlīduN ʾiqlīduN, ʾinna ʾiṭṭa, ʾinna ʾinna. hamzah followed by the long vowel و is encoded ' _U ' _UL_A 'ūlā, ' _ULU 'ūlā, ' _UL_A ʿika 'ūlā ʿiika.

hamzah followed by the long vowel ي is encoded ' _I: ' _ImANuN ʾīmānun26 .

26 For another way of encoding the initial hamzah followed by a long vowel, see the taḥfīf ' l-hamzat' on the next page.
Middle hamzah: xa.ta'ī'-Ina 汉语 ina ხათი ხათი, ru'UsuN ရုလ် ရုလ်, xa.ta'iN რაბუ და ხათ, su'ila სუ ილა, 'as'ilaTuN ას ილა თუნ, mas'alaTuN მას ალა თუნ, ta'xIruN თა ხი რუ, ta'axara თახ ახ რა, 27: with the preceding vowel; hence: ta'axara თახ ახ რა, ta'xIruN თა ხი რუ, ta'axara თახ ახ რა.

From Wright: 27— All consonants, whatever, not even 'ali il hēmzatun excepted, admit of being doubled and take tashdīd. Hence we speak and write ra'AsuN რა ა ს, sa'AluN სა ალ ნ, na' 'AjuN ნა ა ჯ უ ნ.

Final hamzah: xa.ta'uN ხათი ხათი, xa.ta'AN ხათ ან, xa.ta'iN ხათ აი.

the preceding vowel; hence: ta'xIruN თა ხი რუ, ta'axara თახ ახ რა, 27: with the preceding vowel; hence: ta'axara თახ ახ რა, ta'xIruN თა ხი რუ, ta'axara თახ ახ რა.

From Wright: 27— All consonants, whatever, not even 'ali if hēmzatun excepted, admit of being doubled and take tashdīd. Hence we speak and write ra'AsuN რა ა ს, sa'AluN სა ალ ნ, na' 'AjuN ნა ა ჯ უ ნ.

the preceding vowel; hence: ta'xIruN თა ხი რუ, ta'axara თახ ახ რა, 27: with the preceding vowel; hence: ta'axara თახ ახ რა, ta'xIruN თا ხი რუ, ta'axara თახ ახ რა.

From Wright: 27— All consonants, whatever, not even 'ali if hēmzatun excepted, admit of being doubled and take tashdīd. Hence we speak and write ra'AsuN რა ა ს, sa'AluN სა ალ ნ, na' 'AjuN ნა ა ჯ უ ნ.

the preceding vowel; hence: ta'xIruN თა ხი რუ, ta'axara თახ ახ რა, 27: with the preceding vowel; hence: ta'axara თახ ახ რა, ta'xIruN თა ხი რუ, ta'axara თახ ახ რა.

From Wright: 27— All consonants, whatever, not even 'ali il hēmzatun excepted, admit of being doubled and take tashdīd. Hence we speak and write ra'AsuN რა ა ს, sa'AluN სა ალ ნ, na' 'AjuN ნა ა ჯ უ ნ.
The strange spelling of miʾat:  miʾaTuN مِئَاتُ، miʾatun مِئَاتٌ، miʾatAni مِئَاتِي، miʾatyNi مِئَاتِي، miʾaUna مِئَانَة، miʾāuN مَئَانَةً، miʾätun مِئَاتٍ. Of course, the ‘pipe’ character can be used to prevent this rule from being applied (see sect. 4.5 on page 21): miʾalTuN مِئَاتِ. Of course, the ‘pipe’ character can be used to prevent this rule from being applied (see sect. 4.5 on page 21):

Hence one should keep to this distinction and encode 'aʾkulu أَكِلُو and 'AkiluN أَكِلٌ respectively.

maddah At the beginning of a syllable, ʾalif with hamzah (_holder) followed by ʾalifu ʾl-maddi (ʾalif of prolongation) or ʾalif with hamzah and ǧazmah (_holder) are both represented in writing ʾalif with maddah: ʿ (see Wright, A Grammar of the Arabic Language, i. 25 A–B).

Hence one should keep to this distinction and encode 'aʾkulu أَكِلُو and 'AkiluN أَكِلٌ respectively.

Hence one should keep to this distinction and encode 'aʾkulu أَكِلُو and 'AkiluN أَكِلٌ respectively.

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

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

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

(a) With the solar letters ن, ب, ت, ض, ص, ش, س, ز, د, ث, ث, ت, after the article أُل:—

Unlike arabtex and arabxetex, arabluatex never requires the solar letter to be written twice, as it automatically generates the euphonic tašdīd above the letter that carries it, whether the article be written in the assimilated form or not, e.g. al-“sams-u أَلسَّمَّسُ، or aʾs-“sams-u أَسَّسَمُ.

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

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

(a) With the solar letters ن, ب, ت, ض, ص, ش, س, ز, د, ث, ث, ت, after the article أُل:—

Unlike arabtex and arabxetex, arabluatex never requires the solar letter to be written twice, as it automatically generates the euphonic tašdīd above the letter that carries it, whether the article be written in the assimilated form or not, e.g. al-“sams-u أَلسَّمَّسُ، or aʾs-“sams-u أَسَّسَمُ.

(b) With the letters ي, و, م, ل, ر after the ǧazmah, and also after the tanwīn:—

Unlike arabtex and arabxetex, arabluatex never requires the solar letter to be written twice, as it automatically generates the euphonic tašdīd above the letter that carries it, whether the article be written in the assimilated form or not, e.g. al-“sams-u أَلسَّمَّسُ، or aʾs-“sams-u أَسَّسَمُ.

(b) With the letters ي, و, م, ل, ر after the ǧazmah, and also after the tanwīn:—

Unlike arabtex and arabxetex, arabluatex never requires the solar letter to be written twice, as it automatically generates the euphonic tašdīd above the letter that carries it, whether the article be written in the assimilated form or not, e.g. al-“sams-u أَلسَّمَّسُ، or aʾs-“sams-u أَسَّسَمُ.

(b) With the letters ي, و, م, ل, ر after the ǧazmah, and also after the tanwīn:—

Unlike arabtex and arabxetex, arabluatex never requires the solar letter to be written twice, as it automatically generates the euphonic tašdīd above the letter that carries it, whether the article be written in the assimilated form or not, e.g. al-“sams-u أَلسَّمَّسُ، or aʾs-“sams-u أَسَّسَمُ.
Note the absence of sukūn above the passed over ن in the following examples, each of which is accompanied by a consistent transliteration:

min_rabbi-hi ِهِّبَّریٰنِم, mir rabbi-hi, min layliN ٍلْيَّلنِم, mil laylim, 'an yaqtula ِلْتُقَّین ʾay yaqtula.

With tanwīn: kitAbuN mubInuN ٌنيِبُّمٌﺏاَتِک, kitāb um mubīn un.

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

(c) With the letter ت after the dentals ث,ḍ,ﺫ,ﺽ,ﻁ,ﻅ in certain parts of the verb: this kind of assimilation, e.g.ُّتثِبَل for ُّتثِبَل labiṯtu, will be discarded here, as it is largely condemned by the grammarians (see Wright, A Grammar of the Arabic Language, i. 16 B–C).

The definite article and the ʿalif ʾl-waṣli At the beginning of a sentence, ِی is never written, as ُّتثِبَل; instead, to indicate that the ʿalif is a connective ʿalif (ʿalif ʾl-waṣli), the hamzah is omitted and only its accompanying vowel is expressed:

al-.hamd-u li-l-ah-i ِةَسَرْدَمْلٱُﺏاَب, al-ḥamd-u li-l-lāh-u.

As said above on on page 5, fullvoc is the mode in which arabluatex expresses the sukūn and the waṣlah. arabluatex will take care of doing that automatically provided that the vowel which is to be absorbed by the final vowel of the preceding word be properly encoded, like so:

(a) Definite article at the beginning of a sentence is encoded

al-, or [a<solar letter>-

if one wishes to mark the assimilation—which is in no way required, as arabulatex will detect all cases of assimilation.
(b) Definite article inside sentences is encoded

'l- or [<solar letter>-.

(c) In all remaining cases of elision, the ʿalifu ʾl-waṣli is expressed by the vowel that accompanies the omitted hamzah: ⟨u, a, i⟩.


Particles:—

(a) ʿalif ʾl-waṣli is omitted in the article ِن when it is preceded by the preposition ِل: ِلْلْرجِلِ ِلْرِجْلِ sunt-i, li-r-rağul-i.

If the first letter of the noun be ل, then the ل of the article also falls away, but arabluatex is aware of that: li-l-laylaT-i لِلْلَّيْلَةِ, li-l-laylat-i.
(b) \(la\)-: the same applies to the affirmative particle 
\(la\)-haqq\(u\).

c) With the other particles, \(alif\) \(l\)-waṣ\(l\) is expressed: \(fi\) \(l\)-madīnāT-i in ِلَمْدَلَّي l-madīnat\(w\), wa-\(l\)rajul-\(w\) l-rajul\(w\), bi-\(l\)-qalam\(i\) ِبَقَلَّي bi-\(l\)-qalam\(i\), bi-\(l\)-ru\(b\)-i ِبَرِي bi-\(l\)-ru\(b\)-i.

Perfect active, imperative, nomen actionis: qAla isma\(`\) l-qa\(l\)a `sma\(`\), qAla uqtul ِلَتْقَلَّي qāla `qtul, huwa inhazama ِمَزَهْنَي huwa `nhazama, law istaqbala ِلَبْقَتْسَي lawi `staqbala.


\(alif\) l-waṣ\(l\) preceded by a long vowel The long vowel preceding the connective \(alif\) 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:—

\(fi\) \(n\)-nas\(i\), \(abU\) `l-wazIr-i ِوْمْيَرْسَي `abu `l-wazīr, \(fi\) \(l\)-ibtidāʾ-i ِمَرْسَيَلْسَي \(fi\) `l-ibtidāʾ, dU`l-I`lAl-i ِمَرْسَيَلْسَي du `l-i`lāl, maqh\(A\) ِمَيْلاَقْسَي maqh\(A\) ِمَلَقْلَي `amIr-i.

\(alif\) l-waṣ\(l\) preceded by a diphthong The diphthong is resolved into two simple vowels (ibid., i. 21 D–22 A) viz. ay → aï and aw → aï. arabluatex detects the cases in which this rule applies:—

\(fi\) `aynay `l-malik-i ِوْمْيَرْسَي `aynayi `l-malik, ixṣay `l-qawm-a ِمَيْلاَقْسَي ixṣayi `l-qawm, mu.s.tafaw `l-I`lAl-i ِمَيْلاَقْسَي muṣṭafawu `n-lāh.

ramaW `l-.hijAraT-a ِوْمْيَرْسَي ramaWu `l-hiǧārat, fa-lammA ra`aw\(A\) ِوْمْيَرْسَي fa-lammā ra`awu `n-naǧm.

\(alif\) l-waṣ\(l\) preceded by a consonant with sukūn The vowel which the consonant takes is either its original vowel, or that which belongs to the connective \(alif\) 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 ‘Amr”: the second noun is not in apposition to the first, but forms part of the predicate.

Hence زَيْدُ بْنُ عَمْรِرٌدْيَز زَيْدُ بْنُ عَمْرِرٌدْيَز, “Zayd, son of ‘Amr”.

29 “‘Umar is the son of al-Haṭṭāb” (see n. 28).

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

mu.hammaduN'l-nabī Muḥammad uni 'n-nabī, salAmuN 'dḥulū salām 'n-dḥulū, qaṣīdata-hu fī qatl-i ʾAbī Muslim inī 'llatī yaqūlu fī-hā.

4.3 Special orthographies

The name of God The name of God, اللّاهُ, is compounded of the article الّ (noted لّ (لّ) with the defective ‘alif) so that it becomes اللّ (لّ) لّ; then the hamzah is suppressed, its vowel being transferred to the ل before it, so that there remains للهُ (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 lašdid 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 al-lāh-u:—


The conjunctive الّ Although it is compounded of the article الّ, the demonstrative letter ل and the demonstrative pronoun ذا, both masculine and feminine forms that are written defectively are encoded alla_dī and allatī respectively. Forms starting with the connective ‘alif are encoded 'lla_dī and 'llatī:—

'a_hAfu mina 'l-malik-i 'lla_dī ya.zlimu 'l-nAs-a ' glu'af mina 'l-malik-i 'lla_dī yazlimu 'n-nās-a, udu 'l-'say_h-a 'lla_dī huwa marī-duN, mA 'anA bi-'lla_dī qa'ilaN la-ka 'say'ah, ma ṣānA bi-1-llatī qa'ilaN la-ka 'say'ah.'

Note the “pipe” character ‘|’ here after yA and below after fa before footnote mark 33: 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. ʿala ʿl-ṭarīq'. See sect. 4.5 on page 21 on the “pipe” and sect. 8 on page 37 on dmg mode.

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

fa-ʾinnā naḏkuru 'ṣ-ṣawt ayni 'l-laḏayni rawaynā-humā ʿan Ǧaḥẓat a.


4.4 Quoting

It is here referred to “quoting” after the arabtex package.33 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 he taught the science of astronomy, or he taught the science of astronomy? In order to disambiguate this clause, it may be sensible to put a dammah above the first ʿullm, 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 dammah, fatha or kasrah appear above the appropriate consonant:—

`ullima ʿilm-a ’l-hayʿat-i ʿullima ʿilm “l-hayʿat”, ya.gz"UA

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

wa-ʾinnA sawfa tudrik-nA 'l-manAyA muqadd"araT"aN samaaYa maqade raN.

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

32Note here the “pipe” character '|': as already stated on page 15, the sequence 'A usually encodes 'alif with hamzah followed by 'alif of prolongation, which is represented in writing ‘alif with maddah: ٰ. The “pipe” character prevents this rule from being applied. See sect. 4.5 on page 21.

33See Lagally, ArabTEX, 22
qAla.isma" : qala 'isma', jA'at hinduN َجَاءَتُ هَنَذَٰهُ gā'at Hindnn,
'sabiḥuN bi-man q"u.ti`at qadamA-hu šabihnn bi-
man quṭiʿat qadamā-hu.

(d) At the commencement of a word, the straight double quote is interpreted as 'alif" ʾ-
l-ʾaṣla!':—
wa="ust'u.mila wa-'stuʿmilah, huwa "inhazama huwa
'nhazama, al="intiqA.d-u al-intiqād".

voc In accordance with the general rule, in this mode, “quoting” makes the vowels and
the tanwīn disappear, should this feature be required for some reason:—
(a) Short and long vowels:—
q"Ala.qA'iluN qāla qāʾil uN, ibn-u._'abI_u.'u.saybiʿaT="a ʾا
behavior Ibn "Abī Uṣaybiʿat".
(b) tanwīn:—
madInaT"aN madīnat" an, bAb"aN bāb" an, hud"aN_A hudạ
in.

One may more usefully “quote” the initial vowels to write the waṣlah above the 'alif or
insert a straight double quote after a consonant not followed by a vowel to make the sukūn
appear:—
(a) 'alif" ʾ-l-ʾaṣla!':—
fi."istiq.sA'-iN in fi 'stiqṣāʾ, wa="istiq.sA'-uN wa-
'stiqṣāʾun", qAla "uhrub fa-lan tuqtala َلَتْقُتْنَلَفْبُرهٱَلاَق qāla 'hrub fa-lan
tuqtala.
(b) sukūn:—
qAla "uqtul" fa-lan tuqtala َلَتْقُتْنَلَفْلُتقٱَلاَق qāla 'qtul fa-lan
tuqtala, mA jA'at" mini imra'aTiN َرَمَآِمْﺕَﺀآَجَم mā ǧāʾat mini 'mraʾat". kam"
qad" ma.dat" min" laylaTiN ِةَلْيَلْنِمْتَضَمْدَقْمَك kam qad maḍat min
laylatiN.

fullvoc In this mode, “quoting” can be used to take away any short vowel (or tanwīn,
as seen above) or any sukūn:—
al-jamr-u 'l-.sayfiyy-u 'lla_di kAna bi-q"rAn" |nUn-a َأَخَرَ ٱلصَّبِيحِ ٱلَّذِي
kAn ʿyaraṭun َكَانَ يُقَرَّأُونَ al-ğamr" ʿṣ-.sayfiyy" ʾllaḍi kāna bi-Qrānnūn".

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’⟩, 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:—

\begin{arab}[	ext{fullvoc}]\begin{LRfootnote}{A footnote which interferes with the contextual analysis.}\end{LRfootnote}fa-lammā ra‘ā’W\end{arab}

As one can see, the “pipe” character between the two ⟨⟩ prevents the necessary *tašdīd* rule ( on page 15) from being applied.

4.6 Putting back on broken contextual analysis rules

*New feature v1.7* 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:—

1. This is wrong:
2. \begin{arab}[	ext{fullvoc}]\begin{LRfootnote}{A footnote which interferes with the contextual analysis.}\end{LRfootnote}fa-lammā ra‘ā’W\end{arab}

34 See below sect. 8 on page 37.
35 See more context on the previous page.
This is wrong:

\begin{arab}
فَلَا رَأِيََّ الْحَجِّ
\end{arab}

*A footnote which interferes with the contextual analysis.

According to the rule stated on page 17, the diphthong in *raʾaw* must be resolved into two simple vowels before the 'alif 'uwaṣl', as ﹾﺭﺃﺍَّ ﺍْوُمْجَّنلٱﺍُو. The \texttt{\textbackslash{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, \texttt{\textbackslash{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 \texttt{\textbackslash{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 \texttt{\textbackslash{arbnull}}{l-na^gma}\LRfootnote{A footnote which interferes with the contextual analysis.}
\texttt{\textbackslash{arbnull}}{l-na^gma...}
qAla\LRfootnote{A footnote which interferes with the contextual analysis.}\texttt{\textbackslash{arbnull}}(qAla)uhrub fa-lan tuqtala.
\texttt{\textbackslash{arbnull}}(zaydu)\LRfootnote{A footnote which interferes with the contextual analysis.}\texttt{\textbackslash{arbnull}}(ibnu)\LRfootnote{A footnote which interferes with the contextual analysis.}\texttt{\textbackslash{arbnull}}(zaydu)ibn-\texttt{\textbackslash{uc}}(a)miNU\LRfootnote{See \vref{fn:zayd-is-son}.}
\end{arab}

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

Zayd\LRfootnote{A footnote which interferes with the contextual analysis.} unie\LRfootnote{A footnote which interferes with the contextual analysis.} 'bn\LRfootnote{A footnote which interferes with the contextual analysis.} 'Amr\LRfootnote{See 28 on page 17.} in.\LRfootnote{See 28 on page 17.}
4.7 Stretching characters: the \textit{ta\'t\w til}

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

\textit{Hunayn} bn \textit{Ish\'aq}

4.8 Digits

4.8.1 Numerical figures

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

\textit{1874} \textit{123-456,789} \textit{fI sanaT-i 1024}

\texttt{% Usepackage(arabluatex)}
\texttt{% SetArbNumbers(Arabic)}
\texttt{% Use 'anum' for Arabic numbers or '-anum' for Indian numbers:}
\texttt{\newfontfamily{arabicfont}{Amiri}[Script=Arabic, RawFeature={+anum}]}
\abjad may also be used to convert values of counters into ʾabǧad numbers, like so:

1. The \arb{\text{trans}}{ʾabǧad} number for the current page (\thepage) is
2. \abjad{\thepage}.

The ʾabǧad number for the current page (23) is جك.

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, al-ḥurūf" l-muhmalat", are sometimes further distinguished from the pointed by various contrivances, as explained in Wright.\footnote{Wright, see n. 7, i. 4 B–C.} One may find these letters written in a smaller size below the line, or with a dot or another mark below. As representing all the possible contrivances leads to much complexity and also needs to be agreed among scholars, new ways of encoding them will be proposed and gradually included as arabluatex will mature.

For the time being, the following is included:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration\footnote{This letter can be used to encode the tāʾ marbūtah devoid of diacritical points as it is found in some manuscripts, with the same tanwīn and the same short vowels as the standard tāʾ marbūtah with two points above, e. g. al-madInaHa, madInaHa.}</th>
<th>ArabTEX notation</th>
</tr>
</thead>
</table>
| ت | b |有所不同 
| ث | d |有所不同 
| س | f |有所不同 
| ل | g |有所不同 
| م | k |有所不同 
| ن | n |有所不同 
| ح | ah |有所不同 
| ن | y |有所不同 |

Table 6: Additional Arabic codings


4.10 Arabic emphasis

As already seen in sect. 4.8.2 on the preceding page, the ʾabǧad 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.

\footnote{See below sec. 8 on page 37.} \footnote{See above Rem. b on page 9.}
One may use the \texttt{\textbackslash emph{⟨Arabic text⟩}} command to use the same technique to emphasize words, like so:—

\begin{verbatim}
\abjad{45}: kitAbu-hu \texttt{\textbackslash emph{fI 'l-ʿAdAt-i}}.
\end{verbatim}

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

\textbf{4.10.1 Underlining words or numbers}

\begin{itemize}
\item \texttt{\textbackslash aoline}, which is equivalent to \texttt{\textbackslash emph*} described above.
\item \texttt{\textbackslash aoline*}, which is the same as \texttt{\textbackslash aoline}, but better suited for ʿabǧad numbers.\footnote{See the example provided above sect. 4.8.2 on page 23.}
\item \texttt{\textbackslash aoline}, which can be used to underline Arabic words.
\end{itemize}

\section{Arabic poetry}

\textit{New feature v1.6} arabluatex provides a special environment for typesetting Arabic poetry. Every line in this environment must end with \texttt{\textbackslash \}.

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

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

The description of the optional arguments follows:—

\begin{description}
\item[\texttt{mode}] \texttt{mode=} ⟨\texttt{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.
\item[\texttt{width}] \texttt{width=} ⟨\texttt{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 \texttt{4cm} or \texttt{2in}. 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.
\item[\texttt{gutter}] \texttt{gutter=} ⟨\texttt{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.
\item[\texttt{metre}] \texttt{metre=} ⟨\texttt{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.
\item[\texttt{delim}] \texttt{delim=} ⟨true|false⟩ \textbf{Default: false}
\end{description}
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 `*` character. The \SetHemistichDelim{⟨delimiter⟩} command may be used at any point of the document to change this default setting.

```latex
utf utf=true|false
```

As the preceding one, this named argument does not need a value as it defaults to `true` if it is used. If so, Unicode Arabic input is expected in the `arabverse` environment instead of `ascii` ArabTEX or Buckwalter input schemes. See sect. 10 on page 43 for more details.

```latex
\color\color=(color name)
```

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

```latex
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.

```latex\bayt\}
\bayt\{⟨ṣadr⟩\}\{⟨tadwir⟩\}\{⟨ʿaḡuz⟩\}\"
```

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

```latex
\begin{arabverse}[mode=fullvoc, width=.3\linewidth]
\bayt(lA \text{ 'ar}_A \text{ man } \text{ 'ahidtu } fi-hǎ fa-‘abkI } l---)[---]{---yawma
dalhaK wa-mA yaruddu } l-bukA’]\"
\end{arabverse}
```

As one can see, `triple hyphens` have been used. In the `ṣadr`, the first hyphen triggers the rules that are related to the definite article and the ‘`alif` ‘awl’, while the following two select the figure of the letter `lām` connected with a following letter. In the `ʿaḡuz`, the last two hyphens select the letter `yā` connected with a preceding letter, while the first one is simply discarded in this mode, but still may appear as it should, if the `trans` mode be selected:—

---

41A `starred` version `\bayt*` is also defined. `arabluatex` uses it internally when `export` is set to `true` to instruct some Lua functions that lines of poetry have already been processed. That aside, `\bayt` and `\bayt*` do the same, and only `\bayt` should be used.

42See 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 ِّمَهلَﺍ should be split into ِمــْمَهلَﺍ as the first part of it belongs to the șadr 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 ُﺃُّم should be split into ُﺃُمـْم، 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\StretchBayt[true|false] 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.

\bayt+ Critical Notes If the ekdosis package be loaded, the `\bayt` command also accepts an 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.

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.

5.1 Example

Here follow the first lines of Imruʾu ’l-Qaysi’s *Muʿallaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the ʾidɡām that is fit to this declamatory poetry:—

\begin{verbatim}
1 \begin{arab}[fullvoc]
2 qAla imru'u 'l-\uc{q}aysi fi mu'allaqati-hi:
3 \end{arab}
4 \end{arabverse}
5 \SetArbDflt*
6 \begin{linenumbers*}
7 \bayt{qifA nabki min _dikr_A .habIbI N wa-manzili}{bi-siq.ti 'l-1w_A bayna \uc{'l-d}_a_hulI fa-\uc{h}awmalI}\\
8 \bayt{fa-\uc{m}iqATI lam ya'fu rasnu-hA}{limA nasa'-gat-hA min 'ganUbI N wa-'sam'alI}\\
9 \bayt{tar_A ba'ara 'l-az_A'Ami fI 'ara.sAki-hA}{wa-qI'Ani-hA ka-'anna-hu .habbu fulfulI}\\
10 \bayt{ka-'annI .gadATa 'l-bayni yawa ta.hammalUA}{lad_A samurAti 'l-.hayyi naqifu .han.zali}\\
11 \bayt{wuqUfA N bi-hA .sa.hbI alayya ma.tiyya-hum}{yaqulUna lA tahlik 'asanA A wa-ta'gammali}\\
12 \bayt{wa-'inna sifA'I 'abraTuN muharAqaTuN}{fa-hal `inda rasmiN dArisiN min mu'awwali}\\
13 \end{linenumbers*}
14 \end{arabverse}
\end{verbatim}

\StretchBayt{true} (Default):—

43 See sect. 11.1 on page 46.
44 The footnote package can also be used for the same effect. However, it must be loaded after `arabluatex`.
46 [Ibid., see “Arabic Poetry”.
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 *Muʿallaqat* that I know of feature the ʾidɡām in the Arabic text, although it is often strongly marked in declamation.
قال أمير الفِنِس في معلَّمته:

(qała 'mrū'u 'l-Qaysi fī muʿallaqati-hi):

gifā nabki min dikrā habīb wa-manzili bi-sīgī 'l-līwā bayna 'l-Dahūli fa-Hawmali
fa-Tudākh fa-'l-Muğrāt lam yaṣṣa rasmu-hā limā nasāqāt-hā min ganīb wa-samali
tarā ba'ara 'l-arāmi fī 'arasātī-hā wa-qāna-hā ka-'anna-hu habbu ṣadūf
ka-'annī ḍādāla 'l-bayn yuwa lahammali laḍā samurātī 'l-hayyī nāqīfī hanzali
wugāf bi-hā saḥḥī alayya mutiyya-hum yaqūlūna lā tahlik asqī wa-lāqamani
wa-'inna siyā'i 'abrātī muḥarragat fa-hal 'inda rasμ tārismin min muwawalī

(الضرب الثاني من العروض الأولى من الطويل)

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

(gała 'mrū'u 'l-Qaysi fī muʿallaqati-hi):

qāla 'mrū'u 'l-Qaysi fī muʿallaqati-hi:

(qała 'mrū'u 'l-Qaysi fī muʿallaqati-hi):

wild has been set to 0.375\linewidth and \scriptsize has been used so as to avoid overlapping.

qāla 'mrū'u 'l-Qaysi fī muʿallaqati-hi:

(الضرب الثاني من العروض الأولى من الطويل)

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

qāla 'mrū'u 'l-Qaysi fī muʿallaqati-hi:

(الضرب الثاني من العروض الأولى من الطويل)
6 Special applications

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

\[
\begin{align*}
Bu & \quad Ba & \quad Bi & \quad BuN & \quad BaN & \quad BiN & \quad \text{u} & \quad a \quad i \quad \text{u} & \quad an & \quad \text{in} & \quad \text{ Bu } & \quad \text{ Ba } & \quad \text{ Bi } & \quad \text{ BuN } & \quad \text{ BaN } & \quad \text{ BiN } & \quad \text{ Bu } & \quad \text{ Ba } & \quad \text{ Bi } \\
\text{ Bu } & \quad \text{ Ba } & \quad \text{ Bi } & \quad \text{ BuN } & \quad \text{ BaN } & \quad \text{ BiN } & \quad \text{ u } & \quad \text{ a } & \quad \text{ i } & \quad \text{ u } & \quad \text{ an } & \quad \text{ in } & \quad \text{ Bu } & \quad \text{ Ba } & \quad \text{ Bi } & \quad \text{ BuN } & \quad \text{ BaN } & \quad \text{ BiN } & \quad \text{ Bu } & \quad \text{ Ba } & \quad \text{ Bi }
\end{align*}
\]

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 `\{abraces\}` command:

\[
\begin{align*}
\\{\text{begin}\{arab\}\} & \quad \text{inna } \text{'abI k\={a}na } \text{min } \text{l-muq\={a}tilaTi} \\
\quad \text{wa-} & \quad \text{k\={a}na} \quad \text{\textlangle-\textrangle-\textlangle-t\rangle} \quad \text{\textquoteright ummi } \text{\textquoteright min } \text{'u.zamA\={i} buyUti } \text{\textquoteright l-zam\={a}zimaTi}.
\end{align*}
\]

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 Qur\={a}n or in classical poetry. `\{\text{arbbox}\}` `\{\text{arbbox}\}\{\text{\textbar{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>لَبَسْتَ عَلَى نَفْسِكَ &quot;bi-'smi 'Llāhi 'r-raḥmāni 'r-raḥīmi&quot;</td>
<td></td>
</tr>
<tr>
<td>FDF5</td>
<td>salam</td>
<td>صُلُم  &quot;salla 'Llāhu 'lay-ki wa-sallama&quot;</td>
<td></td>
</tr>
<tr>
<td>FDF6</td>
<td>slm</td>
<td>ضُعِ &quot;salla 'Llāhu 'lay-ki wa-sallama&quot;</td>
<td></td>
</tr>
<tr>
<td>FDFB</td>
<td>jalla</td>
<td>¡اللَا &quot;={a}lла йalла-hu&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Additional Arabic marks

New feature v1.13 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.
is also provided should one wish to define new marks in addition to the marks defined above. This command takes three arguments, like so:—

```
\newarbmark{⟨shorthand⟩}{⟨RTL codepoint⟩}{⟨LTR rendition⟩}
```

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

It is also possible to use the so-called ‘\\\\’ notation like so: `\\\\xyzt`, where xyzt are lowercase hex digits (0 to 9 or a to f).

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

It must be noted that `\newarbmark` expects Arab\LaTeX input scheme inside `\arb[trans]` to the exclusion of buckwalter input scheme.

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

```
\SetArbDflt*
\newarbmark{sly}{\arbcolor[red]{\\\\06d6}}{}
\newarbmark{jim}{\arbcolor[red]{\\\\06da}}{}
\begin{arab}
sUraTu 'l-nisA'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_UA
'l-nisA'a karhaN\arbmark{sly} wa-lA ta'_.duU-hunna li-ta_dhabUA
bi-ba`.di mA 'a'taytU-hunna 'illA 'an ya'tina bi-fA.hi^saTiN
mubayyinaTi\arbmark{ jim} wa-'A^sirU-hunna
bi-'l-ma`rUfi\arbmark{ jim} fa-'in karihtU-hunna fa-'as_A_a
'an takrahUA 'say'aN wa-ya^g`ala 'l-l_ahu fI-hi _hayraN
ka_tIraN ((19))
\end{arab}
```

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 ArabTFpX 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 ‘Ishāq (ع)’:

\begin{arab}[fullvoc]
\text{\texttt{huwa 'abU zaydiN .humaynu bnu 'is\&\underlineLine{&.h.a}qa}}
\text{\texttt{'l-'a\underlineLine{&.h.a}diyyu bi-fat.hi 'l-'a\&\highlight{&.h.a}ny wa-ta_hfIfi 'l-bA'i.}}
\end{arab}

An example follows:

\begin{arab}
\text{\texttt{\underlineLine{&.h.a}huwa 'abU zaydiN .humaynu bnu 'is\&\highlight{&.h.a}qa}}
\text{\texttt{'l-'a\highlight{&.h.a}diyyu bi-fat.hi 'l-'a\&\highlight{&.h.a}ny wa-ta_hfIfi 'l-bA'i.}}
\end{arab}

6.1 The Qurʾān

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

For the time being, \texttt{\texttt{\aya\{3-digit number\}}} is provided so as to typeset the number of \texttt{\aya} that it is referred to inside the dedicated mark—Unicode U+06DD: ﺔ—in Arabic script or inside parentheses in romanized Arabic:

\texttt{\aya{123} ﺔ (123)}.

An example follows:

\begin{arab}
\text{\texttt{\aya{1}}}
\end{arab}
min (sūrati 'l-Baqarati):
ʾalif lām mim (1) dālika 'l-kitābu lā rayba fi-hi hudạ li l-muttaqīna (2) 'llaḍīna yuʾminūna bi l-ġaybi wa-yuqīmūna ʿṣ-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)}

1 \begin{arab}
2 \arbcolor[red]{al-bAbu 'l-_hAmisu} fI .tabaqAti 'l-'a.tibBā'i
3 'lla_dīna kAnUa mun_du zamAni \uc{gAlInUsa} wa-qarIbAn
4 min-hu. \arbcolor[red]\{\uc{gAlInUsu}\}: wa-l-na.da 'awwalaN
5 kalāmāN kuliyyaN fI 'a_hbāri \uc{gAlInUsa} wa-mā kāna
6 'alay-hi...
7 \end{arab}
8 \begin{arab}[trans]
9 \arbcolor[red]{al-bAbu 'l-_hAmisu} fI .tabaqAti 'l-'a.tibBā'i
10 'lla_dīna kAnUa mun_du zamAni \uc{gAlInUsa} wa-qarIbAn
11 min-hu. \arbcolor[red]\{\uc{gAlInUsu}\}: wa-l-na.da 'awwalaN
12 kalāmāN kuliyyaN fI 'a_hbāri \uc{gAlInUsa} wa-mā kāna
13 'alay-hi...
14 \end{arab}

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}{} 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. \texttt{arabluatex} takes care of selecting the appropriate shape of the letters while coloring them:—

`voc` mode:
\begin{verbatim}
\texttt{i^staraytu-hu bi-_taman} \textcolor{red}{\{niN\}}
\texttt{\{ga\}ba-} \textcolor{red}{\{a\}}ba-\textcolor{red}{\{ka\} in ʾaʿğaba-ka}.
\end{verbatim}

`fullvoc` mode:
\begin{verbatim}
\texttt{i^staraytu-hu bi-_taman} \textcolor{red}{\{niN\}}
\texttt{\{ga\}ba-} \textcolor{red}{\{a\}}ba-\textcolor{red}{\{ka\} in ʾaʿğaba-ka}.
\end{verbatim}

### 7.1 Tricks of the trade

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

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

Table 8 gives the \texttt{ArabTeX} equivalents for the diacritics to be printed inside or just after \texttt{\textcolor{red}{\{\}}}).

<table>
<thead>
<tr>
<th>Diacritic</th>
<th>Transliteration ( ^50 )</th>
<th>\texttt{ArabTeX} notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>( a ) ( a ) ( a )( .a )</td>
<td>( \texttt{{a}} )</td>
</tr>
<tr>
<td>( u )</td>
<td>( u ) ( u ) ( u )( .u )</td>
<td>( \texttt{{u}} )</td>
</tr>
<tr>
<td>( i )</td>
<td>( i ) ( i ) ( i )( .i )</td>
<td>( \texttt{{i}} )</td>
</tr>
<tr>
<td>( o )</td>
<td>( 0 )</td>
<td></td>
</tr>
</tbody>
</table>

\( ^50 \) See below sect. 8 on page 37.

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

`voc` mode:
\begin{verbatim}
\texttt{i^staraytu-hu bi-_taman} \textcolor{red}{\{\}} \texttt{a\{ga\}ba-ka in ʾaʿğaba-ka}.
\end{verbatim}
ištaraytu-hu bi-ṯamanin aʿğaba-ka.
ištaraytu-hu bi-ṯamanin aʿğaba-ka.

‘fullvoc’ mode:
ištaray"tu-hu bi-_taman"u bi-ṯamanin aʿğaba-ka.
ištaray"tu-hu bi-_taman"u bi-ṯamanin aʿğaba-ka.
ištaray"tu-hu bi-_taman"u bi-ṯamanin aʿğaba-ka.

As can be seen, fullvoc required the letters y, n and ʿg before \arbc color to be ‘quoted’. Otherwise, unwanted sukūns would have been generated because of the absence of a vowel after those consonants.

\tanwîn \arbn ull must be used with fathalân (ʼ) so as to put back on contextual analysis rules:—

\mu`allim\arbc olor\{red\}\{\\arbn ul\{m\}aN\} \text{ـيّمً} mu`allim\text{ًّن},
\text{ـيّمً} istisq\text{ًّن},
\text{ـيّمً} say\\arbc olor\{\arbn ul\{\text{ًّت\}aN\} \text{ـيّمً} sha\text{ًّن},
\text{ـيّمً} \text{ـيّمً} jāmiʿat\text{ًّن}.

Rem. Note that in the last example (jāmiʿat\text{ًّن}), the ‘pipe’ character has been inserted before \arbc olor. Otherwise, the \dmg\ mode of the transliteration rules would have interpreted the tāʾ marbūṭah as final (e.g. h instead of the expected t).\text{٥١}

The \tanwîn 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:—

\newarbmark{Y}{^0649}\{
\text{٥١}See also on page 40 “Discarding the ʻṬrab” for more information.

35
**waṣlah and maddah** Both can be generated with the help of `\arbnull`:—

wa-\arbc[red]{\arbnull(wa)i}stisqA'uN \wassala\wassala \waṣlah
\fi "al".i-\arbc{\arbnull('l'-i)btidA'i fi 'l-\btidā'i
\arbc[red]{('a'\arbnull(k)})kulū fi ākulu,
\arbc[red]{('A'\arbnull(k)})kulū fi ākil''n.

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

\begin{verbatim}
1 \newarbmark{alifmaddahred}{^^^0627}{^^^^0653}
2 \arbc{al-mu`al"imUna}\arbmark{lamshaddah}
3 \arbc{al-mu`al"imUna}.
\end{verbatim}

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 لَوُمِّلَعُمْلَﺍ, al-muʿallimūna, is to be rendered in red. Thus the Arabic mark must generate the šaddah alone—of which the Unicode codepoint is 0651—in Arabic script and the letter `l` in transliteration:—

\begin{verbatim}
1 \newarbmark{lamshaddah}{^^^0651}{l}
2 \arbc{al-mu`al"imUna}\arbmark{lamshaddah}.imUna}
3 \arbc{al-mu`al"imUna}.
\end{verbatim}

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 ﬂI,I-\l{nAsi} fi \l{n-nāsi}, the initial `ʾalif``l-waṣl'' can be rendered in red like so: \arbc{\arbnull(al-)a}\i. Then, the following two letters, namely `l-n`, must print the string lām + nūn + šaddah in Arabic, and exactly `n-n` in transliteration. Thus an Arabic mark is needed:—

\begin{verbatim}
1 \newarbmark{lnn}{^^^0644^^^^0646^^^^0651}{n-n}
2 \arbc{al-mu`al"imUna}\arbmark{lnn}Asi}
3 \arbc{al-mu`al"imUna}.
\end{verbatim}

52To 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:

\begin{verbatim}
yatas\'a\arbnull\{"\}'\arbcolor[red]{|'}.alUna 
\"\textit{say}\arbcolor[red]\{"\}'\arbnull\{\}'\arbcolor[red]{\textit{say} "}.\arbcolor[red]\{"\}'\arbcolor[red]{\textit{say} "}.\arbcolor[red]\{\}'\arbcolor[red]{\textit{say} "}.\arbcolor[red]{\textit{say} "}.\arbcolor[red]{{|'}.ilaTuN 
\textit{as\'ilat}". 
\end{verbatim}

\section{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:

\begin{itemize}
  \item \texttt{dmg} \textit{Deutsche Morgenländische Gesellschaft}, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.\footnote{See Brockelmann et al., “Die Transliteration der arabischen Schrift”.}
  \texttt{dmg} transliteration convention is selected by default;
  \item \texttt{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 \textit{al-asadu}.\footnote{See \url{http://www.loc.gov/catdir/cpso/roman.html} for the source document concerning Arabic language.}
  \item \texttt{loc} \textit{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;\footnote{See \url{http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf}.}
\end{itemize}

\begin{itemize}
  \item \texttt{arabica} \textit{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.\footnote{More standards will be included in future releases of arabluatex.}
\end{itemize}

\begin{itemize}
  \item \texttt{Convention} The transliteration mode, which is set to \texttt{dmg} by default, may be changed at any point of the document by the \texttt{\SetTranslitConvention\{mode\}} command, where \texttt{(mode)} may be either \texttt{dmg}, \texttt{dmg+}, \texttt{loc} or \texttt{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}
  \item \texttt{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 \texttt{\SetTranslitStyle\{style\}} command, where \texttt{(style)} may be any font shape selection command, e.g. \texttt{\upshape}, \texttt{\itshape}, \texttt{\slshape}, and so forth.
  \item \texttt{Font} \texttt{\SetTranslitFont\{font selection command\}} allows any specific font to be selected for rendering transliterated text with the font-selecting commands of the fontspec or luaotfload package. Of course, this font must have been defined properly. To take one example, here is how the \textit{Gentium Plus} font can be used for rendering transliterated text:—
\end{itemize}
Proper names Proper names or book titles that must have their first letters uppercased may be passed as arguments to the \uc{⟨word⟩} command. \uc is a clever command, for it will give the definite article al- in lowercase in all positions. Moreover, if the initial letter, apart from the article, cannot be uppercased, viz. ‘or’, the letter next to it will be uppercased:

\uc{.hunayn-u} bn-u \uc{‘is_h_aq-a} Hunayn\textsuperscript{a} bn\textsuperscript{a} Iśāq\textsuperscript{a},
\uc{‘u_tmn-an-u} Umān\textsuperscript{a}, daraba \uc{zayd-u} bn-u \uc{‘h_alidiN} Zayd\textsuperscript{a} bn\textsuperscript{a} Ḥālid\textsuperscript{a},
\uc{sa’d-a} bn-a \uc{‘awf-i} bn-i \uc{‘abd-i} \uc{‘l-l_ah-i} Sa’d\textsuperscript{a} bn\textsuperscript{a} ‘Awf\textsuperscript{a} bn\textsuperscript{a} ‘Abd\textsuperscript{a} ‘Llāh\textsuperscript{a}.

However, \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 یِبَّنلٱٌدَّمَحُم must be Muḥammad\textsuperscript{a} ’n-nabī, as nouns having the tanwīn take a kasrah in pronunciation before ʾalifu ‘l-waṣli. In that case, encoding ٌدَّمَحُم like so: \uc{mu.hammaduN} is wrong, because the closing brace would prevent arablutex from detecting the sequence ⟨-uN⟩ immediately followed by ⟨‘⟩. Fortunately, this can be circumvented in a straightforward way by inserting only part of the noun in the argument of \uc vz. up to the first letter that is to be uppercased, like so: \uc{m}u.hammaduN.

Hyphenation In case transliterated Arabic words break the \TeX hyphenation algorithm, one may use the \- command to insert discretionary hyphens. This command will be discarded in all of the Arabic modes of arablutex, but will be processed by any of the transliteration modes:—

\uc{‘abU} \uc{bakriN} \uc{mu-.ham\-madu} bnu \uc{za\-ka \-riy\-ya’a} \uc{‘l-raziyyu} Abū Bakr\textsuperscript{a} Muḥammad\textsuperscript{a} bn\textsuperscript{a} Zakariyyā\textsuperscript{a} ‘r-Rāziyyu.

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

\arb\[trans\]{\uc{‘abU zaydiN .hunaynu bnu ‘is_h_aqa ‘l-’ibAdiyyu}}
Abū Zayd\textsuperscript{a} Hunaynu bnu Išāqa ‘l-Ībādiyyu.

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. \prname{⟨Arabic proper name⟩} is provided to that effect:

\textcite[i. 23 C]{Wright}:--- If the name following
\arf{fullvoc}{ibnuN} be that of the mother or the grandfather, the
\arb\[fullvoc\]{‘a} is retained; as \arb\[fullvoc\]{‘Is_A ibnu maryama},
\enquote{Jesus the son of Mary}; \arb\[fullvoc\]{‘ammAru ibnu

\textsuperscript{56}Just as \uc, \prname is also able to process proper names consisting of several subsequent words.
From Wright:— If the name following انْبِاَبَٰٴٖ be that of the mother or the grandfather, the ی is retained; as ِمُعْرََٰاَبَٰٖ انْمُرَٰٖ, “Jesus the son of Mary”; َمَيْرَٰمُنْبٱىَسيِع, “Ammār the (grand)son of Manṣūr”.

\begin{nameauth}
\begin{quote}
\begin{itemize}
\item Hunayn & \prname{"abU zayd} & \prname{.hunayn}, \prname{(i)bn 'is.h_aq al-`ibAdiyy} & > %
\item Razi & \prname{"abU bakr mu.hammad ibn zakariyyA'} & \prname{al-rAziyy} & > %
\end{itemize}
\end{quote}
\end{nameauth}

8.1 Additional note on dmg convention

According to Brockelmann et al.,

Arabic ʾiʿrāb may be rendered into dmg in three different ways:

(a) In full: 'Amr\textsuperscript{u}\n;
(b) As superscript text: 'Amr\textsuperscript{u}n;
(c) Discarded: Amr.

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

\verb|\NoArbUp| \verb|\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 \verb|\ArbUpDflt| at any point of the document.

\verb|\SetArbUp| Finally, \verb|\SetArbUp| can be used to customize the way ʾiʿrāb is displayed. To take one example, here is how Arabic ʾiʿrāb may be rendered as subscript text:—

\begin{quote}
\begin{itemize}
\item \prname* Rum. arabluate also provides \prname* which only renders in upright roman style already transliterated proper names without applying any further processing. It is mostly used internally and applied to proper names exported in Unicode to an external selected file.\textsuperscript{58}
\end{itemize}
\end{quote}

\textsuperscript{57}See the documentation of nameauth for more details: \url{https://ctan.org/pkg/nameauth}

\textsuperscript{58}See below 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.

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

\[
\text{munṣarif: } \text{mu`allimuN} \text{\(\text{m} \rightarrow \text{m} \), } \text{kA`inn} \text{\(\text{m} \rightarrow \text{m} \), } \text{ka`inAtuN} \text{\(\text{k} \rightarrow \text{k} \),} \text{\(\am{\text{`amraNU}}\) \(\text{k} \rightarrow \text{k} \)}}
\]

\[
\text{ġayr munṣarif: } \text{al-mu`allim-u} \text{\(\text{m} \rightarrow \text{m} \), } \text{kitAb-Ani} \text{\(\text{k} \rightarrow \text{k} \), } \text{ra^sa`-Ani} \text{\(\text{r} \rightarrow \text{r} \), } \text{sAriq-Una} \text{\(\text{s} \rightarrow \text{s} \), } \text{qA.d-Una} \text{\(\text{q} \rightarrow \text{q} \),} \text{\(\am{\text{`amraNU}}\) \(\text{k} \rightarrow \text{k} \)}}
\]

Rem. a As the tanwīn is passed over in pronunciation when it is followed by the letters \(\text{r} \), \(\text{l} \), \(\text{m} \), \(\text{w} \), \(\text{y} \) (see (b) on page 15), it may be desirable to further distinguish it by putting it above the line, but not to do the same for ġayr munṣarif terminations. This can be achieved by simply omitting the hyphen before any ġayr munṣarif termination:—

\[
\text{kA`ina \(\text{g} \rightarrow \text{g} \), } \text{\(\am{\text{`amraNU}}\) \(\text{k} \rightarrow \text{k} \)}
\]

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

\[
\text{mana`a'l-nAs-akAffaT-aNminmu_hA} \text{\(\text{n} \rightarrow \text{n} \), } \text{\(\am{\text{`amraNU}}\) \(\text{k} \rightarrow \text{k} \)}}
\]

It may so happen, as in the absence of the article before the annexed word, that arabluatex be unable to determine which of the above two cases the word ending with tāʾ marbūṭah falls into. The ‘pipe’ character (see sect. 4.5 on page 21) may be appended to that word to indicate that what follows is in the construct state:—

\[
\text{Risālah fī tartīb qirāʾat kutub \(\text{g} \rightarrow \text{g} \), } \text{\(\am{\text{`amraNU}}\) \(\text{k} \rightarrow \text{k} \)}}
\]
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: َُلِ. In transliteration, the uncertain vowels go between slashes and are separated by commas: fa`uaila َُلِ.

8.2 Examples

Here follows in transliteration the story of Ǧuḥā and his donkey (ُهُrah). 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 arabtex by default, may be changed at any point of the document by the \SetInputScheme{⟨scheme⟩} command, where ⟨scheme⟩ may be either arabtex or buckwalter. This command is also accepted in the preamble should one wish to set the input scheme globally, like so:—

\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. However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode ٱ and ِ respectively, 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</th>
<th>Buckwalter notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ٰ</td>
<td>a</td>
<td>A</td>
</tr>
<tr>
<td>ٰ</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>َ</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>َ</td>
<td>ٰ</td>
<td>v</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>j</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>H</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>H</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>x</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>d</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>d</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>* V</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>r</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>z</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>s</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ș</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ș</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ă</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ă</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ḍ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ḍ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ṭ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ṭ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ẓ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ẓ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ʿ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ʾ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ʾā</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ʾā</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ʾ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ʾ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>ʾ</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>C</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>M</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>O</td>
</tr>
<tr>
<td>َ</td>
<td>َ</td>
<td>&amp;W</td>
</tr>
</tbody>
</table>

Table 9: Buckwalter input scheme

62I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in arabluatex.
63See sect. 8 on page 37.
Table 9: Buckwalter input scheme

<table>
<thead>
<tr>
<th>Letter</th>
<th>Transliteration</th>
<th>Buckwalter notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ٍـ</td>
<td>a a a a</td>
<td>a a</td>
</tr>
<tr>
<td>َـ</td>
<td>u u u u</td>
<td>u u</td>
</tr>
<tr>
<td>ُـ</td>
<td>i i i i</td>
<td>i i</td>
</tr>
<tr>
<td>ِـ</td>
<td>an an an</td>
<td>F F</td>
</tr>
<tr>
<td>ًـ</td>
<td>un un un</td>
<td>N N</td>
</tr>
<tr>
<td>ٌـ</td>
<td>in in in</td>
<td>K K</td>
</tr>
<tr>
<td>ْـ</td>
<td>o o</td>
<td>o o</td>
</tr>
<tr>
<td>ةـ</td>
<td>ā ā ā `</td>
<td>ā ā ā `</td>
</tr>
<tr>
<td>(ta肆wil)</td>
<td>— — —</td>
<td>— —</td>
</tr>
</tbody>
</table>

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-EaAlamu ُمَلاَعالَﺍ, Al-camsu ُسمَّشلَﺍ aš-šams, bi-SinaAEapi Al-T~ib~iِّبِّبطلﺍِةَعاَنِصِب bi-ṣināʿat i’ṭ-ṭibb.


Similarly, it is not advisable to use | and [ (‘base’ and ‘xml’ schemes) or M and L (‘safe’ scheme) to encode the ‘alif’ ‘l-mamdūdat’ and the ‘alif’ ‘l-waṣl’ for such signs are supposed to be generated by arabluatex internal functions. Besides, as they do not per se convey any morphological information on what they are derived from, they cannot be transliterated accurately. To take one example, <ilY Al-LntiqADi gives ِأمْضِّ أَلْحَمَدَ “Al-ḥamd” li-llāh’.

10 Unicode Arabic input

New feature v1.5 As said above in sect. 9 on page 41 about the Buckwalter input scheme, even though arabluatex is primarily designed to process the ArabT\TeX 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:

- The `\txarb{Unicode Arabic}` command for inserting Unicode Arabic text in paragraphs;
- The `\txarab (env.)` environment for inserting running paragraphs of Arabic text, like so:—

\begin{txarab}
  \textit{Unicode Arabic text}
\end{txarab}

11 \LaTeX\ Commands in Arabic environments

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

\abjad{45} \textit{kitAbu-hu fi 'l-`AdAti}\hfill \textbf{45}

`\abjad{45}` in the argument is odd in Arabic script, but using such features as `\textbf` or `\textit` is a matter of personal taste.

The same applies to footnotes:—

\begin{arab}
\footnote{al-muqAtilaT-u: .tA'ifaT-u mina 'l-furs-i.}.\footnote{al-muqAtilaT-i: \textbf{al-muqAtilaT-i.}}, wa-kAnat \textit{`ummI min `u.zamA'-i buyUt-i}
\end{arab}

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

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

`\RL{⟨arg⟩}` does the same as `\LR{⟨arg⟩}`, but typesets its argument from right to left.

---

\footnote{This is odd in Arabic script, but using such features as `\textbf` or `\textit` is a matter of personal taste.}
\footnote{\framebox has been adapted from `\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]
\begin{exequote}[arb][trans]{\uc{z}ayd-uN ibnu}{zayduN ibn-u \uc{`a}mr-iNU}
\end{exequote}
\end{arab}

\textit{Zayd is the son of `Amr": the second noun is not in apposition to the first, but forms part of the predicate…}

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

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

Another solution is to put in the preamble, below the line that loads \texttt{arabluatex}, the \texttt{\FixArbFtnmk} command. However, for more control over the layout of footnotes marks, it is advisable to use the \texttt{scrextend} package.66

\LRmarginpar does for marginal notes the same as \LRfootnote does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that \texttt{\marginpar} also works in Arabic environments, but it acts as any other single-argument command inserted in Arabic environments. The general principle laid on the 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}
\section*[\arb\{barzawayhi li-buzurjumihra bn-i' l-buxtikAni\}]
q\'{a}la barzawayhi bn-u 'azhar-a, ra's-u 'a.tibbA'-i f\texttt{Ar}&\texttt{is}-a...
\end{arab}

\textit{...بَرْزُواٰهُ لِبَرْزُّوُهُ بُنٌ أَزَهْرٍ، رَأَسٌ أَلْبَاءٍ، فَأَرَسَ…}

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

In some particular cases, it may be useful to define new commands to be inserted in Arabic environments. From the general principle laid on page 44, it follows that any command that is found inside an Arabic environment is assumed to have Arabic text in its argument which arablatex will process as such before passing it on to the command itself for any further processing. As a result of this feature, such a command as:

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

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

\textbf{\texttt{\textbackslash MkArbBreak}}

The \texttt{\textbackslash MkArbBreak\{csv list of commands\}} command can be used in the preamble to give any command—either new or already existing—the precedence over arablatex 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{\textbackslash MkArbBreak\{onecmd, anothercmd, yetanothercmd, ...\}}
```

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

```
1 \texttt{\textbackslash MkArbBreak\{fvred\}}
2 \texttt{\newcommand{\fvred}[1]{\arbcolor[red]{\arb[fullvoc]{#1}}}}
3 \texttt{\begin{arab}\[voc\]}
4 _\texttt{\textbackslash fvred\{\texttt{(ma.tli`-i `l-^sams-i\})\} wa-lA binA’-a la-hum}
5 \texttt{yu’amminu-hum mina `l-^sams-i.} \\
6 \texttt{\end{arab}}
```

```
ِنيَنرَقلﺍوُﺫَقَلَتنٱَّمُث
ﺇ
ىَل
ُﺃ
ٍةَّم
ُﺃ
ُيمُهَلَﺀآَنِباَلَو﴾ِسْمَّشلٱِعــِــلْطَم﴿يِفىَرخ
ِسمَّشلﺍَنِممُهُنِّم
```

It must be noted that the arguments, either optional or mandatory, of commands declared with \texttt{\textbackslash MkArbBreak} are not to be processed by arablatex. Therefore, as in the previous example, any of their argument to be rendered in Arabic must be inserted again in \arb. These commands themselves may have up to two optional and/or mandatory arguments followed by one optional argument, like so:—

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

(g) \texttt{\textbackslash command\{opt1\}\{opt2\}\{arg1\}\{arg2\}\{opt3\} (uppermost combination)}

\textbf{\texttt{\textbackslash MkArbBreak\*}}

As said above, \texttt{\textbackslash MkArbBreak\*\{csv list of commands\}}—is also provided. It goes a step further, as it directs arablatex to close the current Arabic environment before any of the ‘declared’ commands, then resume it just after.

```
\texttt{\textbackslash MkArbBreak\*}
```

It must be noted that \texttt{\textbackslash MkArbBreak\*} must be used with the utmost care and \emph{should never be used} if \texttt{\textbackslash 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{quote}
\begin{arab}
\begin{<environment>}[<options>]
<Arabic text>
\end{<environment>}
\end{arab}
\end{quote}

In the following example, the \texttt{quoting} package is used:

\begin{quote}
\setquotestyle{arabic}
\begin{arab}[fullvoc]
\begin{<environment>}[<options>]
<kAnat u{abU} u{1-hu_dayli} 'ahd_A 'il_A u{muwaysiN}>
dajAjaTa\u{N}. wa-k\u{a}nat dajAju-hu 'llatI 'ahdA-h\u{a} d\u{a}na m\u{a} k\u{a}na
yu\texttt{utta}_xa_du li-'u{muwaysiN}\u{N}. wa-l\_a\texttt{k\u{a}n}a-hu bi-k\u{a}r\u{m}a-hi
wa-bi-hu\texttt{nis} xulu\texttt{qi}-hi 'a.z\u{h}ara '1-t\u{a}'ajuba min s\u{i}nani-h\u{a}
w\u{a}-t\u{b}i la.h\u{a}ni-h\u{a} \u{b}u-w\u{a}-k\u{a}na \u{b}u-'u{abU} u{1-hu_dayli}>
yu\texttt{r}u\texttt{fu}-bi-'1-'u{msA}k\u{i} '1-'u{sadIdi}. fa-q\u{a}la: \texttt{\enquote}{wa-kayfa}
ra\texttt{a}yya ya u{abA} u{imAna} tilka '1-daj\u{a}Ta\u{N}?} q\u{a}la:
\texttt{\enquote}{k\u{a}N\u{a}t \u{a}jaba\u{N} mina '1-'u{jabit}} fa-yaq\u{u}\u{u}lu:
\begin{quoting}[begintext=\textquotedblleft, endtext=\textquotedblright]
w\u{a}-t\u{d}r\u{I} m\u{a} j\u{in}su-h\u{a}? wa-t\u{d}r\u{I} m\u{a} s\u{i}nu-h\u{a}? fa-'im\u{n}a
'1-daj\u{a}Ta \u{a}\texttt{im\u{a}}-m\u{a} ta.t\u{b}u bi-'1-j\u{i}nsi wa-'1-s\u{i}nni.
wa-t\u{d}r\u{I} bi-'ayyi \u{z}ay'i\u{h} k\u{u}n\u{a} k\u{u}\texttt{ussa}m\u{u}n-h\u{a}? wa-f\u{u}l \u{a}yyi
mak\u{a}n\u{u}n k\u{u}n\u{a} n\u{a}'lifu-h\u{a}?
\end{quoting}
\end{arab}
\end{quote}

\begin{quote}
\setquotestyle{arabic}
\begin{arab}[fullvoc]
k\u{a}na u{abU} u{1-hu_dayli} 'ahd_A 'il_A u{muwaysiN}\u{N}.
daj\u{a}j\u{a}Ta\u{N}. va-k\u{a}nat daj\u{a}ju-hu 'llatI 'ahdA-h\u{a} d\u{a}na m\u{a} k\u{a}na
yu\texttt{utta}_xa_du li-'u{muwaysiN}\u{N}. wa-l\_a\texttt{k\u{a}n}a-hu bi-k\u{a}r\u{m}a-hi
wa-bi-hu\texttt{nis} xulu\texttt{qi}-hi 'a.z\u{h}ara '1-t\u{a}'ajuba min s\u{i}nani-h\u{a}
w\u{a}-t\u{b}i la.h\u{a}ni-h\u{a} \u{b}u-w\u{a}-k\u{a}na \u{b}u-'u{abU} u{1-hu_dayli}>
yu\texttt{r}u\texttt{fu}-bi-'1-'u{msA}k\u{i} '1-'u{sadIdi}. fa-q\u{a}la: \texttt{\enquote}{wa-kayfa}
ra\texttt{a}yya ya u{abA} u{imAna} tilka '1-daj\u{a}Ta\u{N}?} q\u{a}la:
\texttt{\enquote}{k\u{a}N\u{a}t \u{a}jaba\u{N} mina '1-'u{jabit}} fa-yaq\u{u}\u{u}lu:
\begin{quoting}[begintext=\textquotedblleft, endtext=\textquotedblright]
w\u{a}-t\u{d}r\u{I} m\u{a} j\u{in}su-h\u{a}? wa-t\u{d}r\u{I} m\u{a} s\u{i}nu-h\u{a}? fa-'im\u{n}a
'1-daj\u{a}Ta \u{a}\texttt{im\u{a}}-m\u{a} ta.t\u{b}u bi-'1-j\u{i}nsi wa-'1-s\u{i}nni.
wa-t\u{d}r\u{I} bi-'ayyi \u{z}ay'i\u{h} k\u{u}n\u{a} k\u{u}\texttt{ussa}m\u{u}n-h\u{a}? wa-f\u{u}l \u{a}yyi
mak\u{a}n\u{u}n k\u{u}n\u{a} n\u{a}'lifu-h\u{a}?
\end{quoting}
\end{arab}
\end{quote}

11.3.1 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:—

`\usepackage{enumitem}`

`\newlist{enumabjad}{enumerate}{10}`

`\setlist[enumabjad]{nosep, label={\abjad{\arabic*}}}`

As a third example, abjad-numbered lists can be typeset in conjunction with the `enumitem` package,\(^67\) like so:—

\(^67\) See the documentation of `enumitem` for more details: 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.

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

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

The recommended way of inserting quotation marks in running Arabic text is to use csquotes. With the help of the \DeclareQuoteStyle command, one can define an Arabic style, like so:

\begin{verbatim}
\usepackage{csquotes}
\DeclareQuoteStyle{arabic}{\textquotedblright}{\textquotedblleft}
\end{verbatim}

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

\begin{verbatim}
\setquotestyle{arabic}
\begin{arab}
\textcolor{red}{\uc{m}uha_d_dabu \uc{r}{`l-d}Ini \uc{a}bdu \uc{r}{`l-r}{a}hImi bnu \uc{r}{`a}liyyiN} huwa ^say_hu-nA \uc{m}uha_d_dabu \uc{r}{`l-d}Ini \uc{a}bU \uc{r}{`a}bU\text{AmidiN} wa-\text{yu`rafu bi-\text{a}_\text{hwari}.}
\end{verbatim}

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: \footnote{\texttt{arabluatex} provides its own \texttt{arbcolor} command which is able to render syllabes or diacritics in colors. See sect. 7 on page 33.}

\begin{verbatim}
\begin{arab}
\textcolor{red}{\uc{m}uha_d_dabu \uc{r}{`l-d}Ini \uc{a}bdu \uc{r}{`l-r}{a}hImi bnu \uc{r}{`a}liyyiN} huwa ^say_hu-nA \uc{m}uha_d_dabu \uc{r}{`l-d}Ini \uc{a}bU \uc{r}{`a}bU\text{AmidiN} wa-\text{yu`rafu bi-\text{a}_\text{hwari}.}
\end{verbatim}
reledmac The two-argument command \edtext{\{lemma\}}{\{commands\}} is supported inside \begin{arab} ... \end{arab}.\(^{69}\) As an example, one may get arablatex and reledmac to work together like so:—

\begin{verbatim}
\begin{numbering}
\begin{pstart}
\begin{begin(arab)}
wa-ya.siru ta.hta \text{'l-jild-i}
\end{begin(arab)}
\end{pstart}
\end{numbering}
\end{verbatim}

11.5 quran

arablatex 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 arablatex takes care of formatting the Arabic: therefore, it is recommended to load the quran package with the nopar option, after arablatex itself has been loaded, like so:—

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

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

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

12 Exporting Unicode Arabic to an external file

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

\(^{69}\)\texttt{pstart} and \texttt{pend} are also supported inside the \texttt{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, arablatex must be loaded with the export global option enabled,\(^{70}\) like so:—

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

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 (env.) 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:—

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

arablatex 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`\(^{71}\) or `\txtrans`\(^{72}\) accordingly.

(b) Additionally, Arabic paragraphs may receive `\arbpardir`, which arablatex 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.\(^{73}\)

(c) Proper names are inserted as arguments of `\prname*`.\(^{74}\)

**\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.

\(^{72}\)`\texttt{\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 \texttt{bayt} is replaced with \texttt{bayt*}.75

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:—

```
\begin{arabexport}
\end{arabexport}
```

Where \texttt{⟨jobname⟩} is the name of the original .tex source file, and \texttt{⟨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 pandoc76. 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 \texttt{\txarb}, \texttt{\txtrans} or \texttt{\prname*}, which are not known to pandoc, must be redefined explicitly in the preamble to prevent the converter from gobbling their arguments, like so:—

```
\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}
```

13 Future work

A short, uncommented, list of what is planned in the versions of arabluatex to come follows:

(a) Short-term:

---

75See above 41 on page 26 for more information.
76See 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.

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

Declare the global options, and define them:

\RequirePackage{iftex}
arabluatex requires LuaLaTeX of course. Issue a warning if the document is processed with another engine.
\RequireLuaTeX

\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{voc}
\def\al@mode@fullvoc{fullvoc}
\def\al@mode@novoc{novoc}
\def\al@mode@trans{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{\ifdefined\arabicfont\relax\else
\PackageInfo{arabluatex}{\string\arabicfont\ is not defined.\MessageBreak
arabluatex will try to load Amiri}\fi}%
```

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

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

\texttt{\setLR} The same applies to \texttt{\setLR}:

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

\texttt{\LR} This command typesets its argument from left to right. As \texttt{\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\reset@font#1\egroup}}}
```

\texttt{\SetArbNumbers} The \texttt{\SetArbNumbers} command can be used to give any command—either new or already existing—the precedence over \texttt{arabluatex} inside Arabic environments. It is actually coded in Lua.

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

\texttt{\RL} This one typesets its argument from right to left. Same remark as above regarding the need of redefinition.

```latex
\AtBeginDocument{\ifdef{\RL}%
{\RenewDocumentCommand{\RL}{m}{\bgroup\textdir TRT\reset@font#1\egroup}}
{\NewDocumentCommand{\RL}{m}{\bgroup\textdir TRT\reset@font#1\egroup}}}
```

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

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

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

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

\texttt{\aemph*} The ‘starred’ version of this command alway puts the stroke over its argument. As of v1.19, \texttt{arabluatex} uses \texttt{lua-ul} to render the strokes, thus allowing line breaks and manual hyphenation for transliterated Arabic.

\texttt{\aoline} \texttt{\aoline} and \texttt{\auline} derive from \texttt{newunderlinetype} provided by the \texttt{lua-ul} package \texttt{\aoline*} whereas \texttt{\aoline*}, which uses \texttt{\overline} in math-mode, is better suited for so-called \texttt{\aoline} 'abǧad' numbers.
\newunderlinetype\@aoverLine{\leaders\vrule height 3ex depth -2.9ex}
\def\aoline{\@ifstar\@aoline\@@aoline}
\def\@aoline#1{\ensuremath{\overline{\mbox{#1}}}}
\newunderlinetype\@aunderLine{\leaders\vrule height -.65ex depth .75ex}
\def\auline#1{{\@aunderLine#1}}
\AtBeginDocument{\ifdef{\aemph}{\RenewDocumentCommand{\aemph}{s m}{\IfBooleanTF{#1}{\luadirect{\text.sprint(arabluatex.aemph(\luastringN{#2},"over"))}}{\luadirect{\text.sprint(arabluatex.aemph(\luastringN{#2},"dflt"))}}}}}{\NewDocumentCommand{\aemph}{s m}{\IfBooleanTF{#1}{\luadirect{\text.sprint(arabluatex.aemph(\luastringN{#2},"over"))}}{\luadirect{\text.sprint(arabluatex.aemph(\luastringN{#2},"dflt"))}}}}}
\arbcolor\arbcolor[(color)]{(Arabic text)} takes the Arabic text to be colored as argument.
\SetInputScheme arabluatex is designed for processing Arab\TeX input notation. \SetInputScheme may be used in the preamble or at any point of the document should the user wish to use a different notation such as the ‘Buckwalter scheme’.
\SetArbEasy By default, arabluatex applies complex rules to generate euphonic tašdīd, ‘alif mamdūdah and sukūn depending on the modes which are selected, either voc, fullvoc or trans. Such refinements can be discarded with \SetArbEasy, either globally in the preamble or at any point of the document. Note that \SetArbEasy keeps the sukūn that is generated, while the starred version \SetArbEasy* takes it away. Default complex rules can be set back at any point of the document with \SetArbDflt.
\SetArbDflt* As of v1.6, arabluatex does not applies any more the assimilation rules that are laid on (b) on page 15; a new starred version \SetArbDflt* is now available to the user should he wish to apply them.
\SetTranslitFont By default, the font that is used for transliterated text is the main font of the document. Any other font may also be selected with the font-selecting commands of the fontspec package.
\SetTranslitFont
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:

```latex
89 \def\al@trans@style{\itshape}
90 \NewDocumentCommand{\SetTranslitStyle}{m}{\def\al@trans@style{#1}}
```

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

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

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

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

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

```latex
95 \NewDocumentCommand{\ar@arbup@dflt}{m}{\textsuperscript{#1}}%
96 \NewDocumentCommand{\ar@arbup}{m}{\ar@arbup@dflt{#1}}
97 \NewDocumentCommand{\arbup}{m}{\ar@arbup{#1}}
98 \NewDocumentCommand{\ArbUpDflt}{}{\let\ar@arbup=\ar@arbup@dflt}
99 \NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\ar@arbup}{m}{##1}}
100 \NewDocumentCommand{\SetArbUp}{m}{\RenewDocumentCommand{\ar@arbup}{m}{#1}}
```

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.

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

`\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.

```latex
104 \let\Uc\uc
```

`\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.

```latex
105 \NewDocumentCommand{\prname}{s m}{\bgroup\SetTranslitStyle{\relax}}
106 \IfBooleanTF{#1}{\txtrans{#2}}{\arb[trans]{\uc{\uc{#2}}}}\egroup
```

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

```latex
109 \NewDocumentCommand{\prname*}{s m}{\bgroup\SetTranslitStyle{\relax}}
110 \IfBooleanTF{#1}{\txtrans{#2}}{\arb[trans]{\uc{\uc{#2}}}}\egroup
```
\txarb \txarb sets the direction to right-to-left and selects the Arabic font. It is used internally by several Lua functions, but available to the user should he wish to insert utf8 Arabic text in his document.

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

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

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

\arb The \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.
\arbmark \arbmark[{r1|l1}] \{\textit{shorthand}\} \ takes \ one \ argument \ from \ a \ list \ of \ defined \ elements. \ The mark to be inserted is determined by contextual analysis or by an optional argument, either r1 or l1. This command is coded in Lua.

\newDocumentCommand{\arbmark}{O{} m}{\bgroup\SetInputScheme{arabtex}\luadirect{tex.sprint(arabluatex.processarbmarks(\luastringN{#2}, \luastringN{#1}))}\egroup}

\newarbmark \newarbmark \ lets \ the \ user \ define \ additional \ Arabic \ marks. \ As \ \arbmark, \ this \ command \ is coded \ in \ Lua. \ It \ takes \ three \ arguments: \ the \ abbreviated \ form \ to \ be \ used \ as \ argument \ of \ \arbmark, \ the \ rendition \ in \ Arabic \ script \ and \ the \ rendition \ in \ romanized \ Arabic.

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

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

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

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

Each verse consists of two hemistichs; therefore the \bayt command takes two arguments, the first receives the \sadr and the second the \'ağuz. That two subsequent hemistichs should be connected with one another is technically named \tadwīr. In some of these cases, the hemistichs may be connected by a prominent horizontal flexible stroke which is drawn by the \alverse\stroke command.

\StretchBayt \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}.

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

\begin{verbatim}
\newif\ifal\warp\bayt
\al\warp\baytttrue
\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\bayttfalse
    \else\PackageError{arabluatex}{\string\stretchbayt\space must be \texttt{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}}
\providebool{ekd@state}
\NewDocumentCommand{\bayt}{t+ s m o m}{% 
  \IfBooleanTF{#1}{% 
    \ifekd@state
      \leavevmode
      \stepcounter{ekd@lab}\
      \zlabel{ekd:\theekd@lab}\
      \luadirect{ekdosis.storeabspg(
        \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}
      \add@apparatus
    \fi
  }{% 
    \IfBooleanTF{#2}{% 
      \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}{% 
      \relax
    }{% 
      \relax
    }

\end{verbatim}

61
\arind\{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: index, root and form.

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

\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
\csname index\endcsname{#2}
\egroup
}\ifdefined\spewnotes\spewnotes\else\fi
}
\abjad\{number\} expresses its argument in Arabic letters in accordance with the \textit{abjad} arrangement of the alphabet. \textit{number} must be between 1 and 1999. It is now coded in Lua so that \textit{polyglossia} is no longer needed. See \texttt{arabluatex.lua} for more information.

\AtBeginDocument{\ifdefined\abjad\RenewDocumentCommand{\abjad}{m}{\ifbool{al@rlmode}{\aoline*{\luadirect{\text{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}}{\luadirect{\text{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}}\else\NewDocumentCommand{\abjad}{m}{\ifbool{al@rlmode}{\aoline*{\luadirect{\text{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}}{\luadirect{\text{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}}\fi\fi\fi}
\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.

\arbnull The \arbnull 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 \textdir.

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

\LRfootnote \LRfootnote and \RLfootnote are supposed to be used in Arabic environments for insertions of non Arabic text. \LRfootnote typesets its argument left-to-right...

\RLfootnote while \RLfootnote typesets its argument left-to-right.

\FixArbFtnmk In the preamble, just below \usepackage{arabluatex}, \FixArbFtnmk may be of some help in case the footnote numbers at the bottom of the page are printed in the wrong direction. This quick fix uses and loads \scrextend if it is not already loaded.

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

\ArbOutFile \ArbOutFile\{⟨newline⟩\}\{⟨string⟩\} silently exports ⟨string⟩ to the external selected file. It may take ⟨newline⟩ as an optional argument in which case a carriage return is appended to ⟨string⟩.
\texttt{\verb|ArbOutFile*| \verb|ArbOutFile*|\{\texttt{newline}\}\{\texttt{string}\}} does the same as \texttt{\verb|ArbOutFile|} but also inserts \texttt{\{string\}} in the current .tex source file.

\begin{verbatim}
406 \NewDocumentCommand{\ArbOutFile}{s O{no} +m}{% 407   \If@pkg@export% 408     \IfBooleanTF{#1}{% 409       \IfBooleanTF{#3}{% 410         \exp@writefile{\luaroots{\luastringN{#3}}}{#2}}{% 411         \luadirect{\arblatex.tooutfile(\luastringN{#3}, "#2")}}% 412       }{% 413       \else\IfBooleanTF{#1}{#3}{}\fi}%
\end{verbatim}

\texttt{arabexport} \textit{(env.)} The \texttt{arabexport} environment processes and prints its argument unchanged to the current .pdf file. Additionally, if \texttt{arabluatex} is loaded with the \texttt{export} option, this argument is exported to the external selected .tex file with Unicode in place of the original \texttt{arabtex} or \texttt{buckwalter} strings.

\begin{verbatim}
412 \NewDocumentEnvironment{arabexport}{+b}{% 413   \If@pkg@export% 414   \par#1\fi 415   \luadirect{arabluatex.doexport("yes")}% 416   \luadirect{\text.sprint(arabluatex.arbtoutf(\luastringN{#1}))}% 417   \luadirect{arabluatex.doexport("no")}%
\end{verbatim}

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

\begin{verbatim}
421 \NewDocumentEnvironment{arab@v@export}{O{} +b}{% 422   \setkeys[al]{verse}[width, gutter, color, utf, delim, mode, 423   metre]{#1}% 424   \If@pkg@export 425   \ifal@verse@export 426     \ifekd@state 427       \begin{ekdverse}[width=\al@verse@twidth]
428       #2
429       \end{ekdverse}
430     \else
431       #2
432     \fi
433   \else
434     \ifekd@state
435       \begin{ekdverse}[width=\al@verse@twidth]
436       #2
437       \end{ekdverse}
438     \else
439       \begin{ekdverse}[width=\al@verse@twidth]
440       #2
441       \end{ekdverse}
442     \fi
443   \fi
444 \end{verbatim}
\arbpardir is automatically inserted by arabluatex at the beginning of Arabic paragraphs converted to Unicode so that they are printed in the right direction.

Errors and Warnings

Errors and Warnings

That is it. Say goodbye before leaving.

Patches

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 LuaTeX primitives \bodydir and \pagedir will eventually allow us to get rid of this:

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.
\abjad: New and more flexible \abjad command. ..................... 63

v1.2.
\SetArbEasy: New
\SetArbEasy/\SetArbDflt for ‘modern’ or ‘classic’ Arabic styles. . 56
\SetHemistichDelim: New
\SetHemistichDelim command for changing the default delimiter between hemistichs .................. 61

v1.3.
\arbup: ʿrāb is now written as superscript text in dmg mode by default. ..................... 57

v1.4.
\SetInputScheme: \SetInputScheme can be used to process other input schemes such as ‘Buckwalter’ .... 56
\SetTranslitFont: For selecting a specific font for transliterated texts 56
\SetArbEasy*: this starred version discards the sukūn in addition to what is already discarded by \SetArbEasy. ..................... 56

v1.4.3.
\abrac: New \abrac command which expresses its argument between braces. ..................... 64

v1.4.4.
\SetArbEasy*: this starred version discards the sukūn in addition to what is already discarded by \SetArbEasy. ..................... 56
\MkArbBreak: New \MkArbBreak command for inserting user-defined macros in Arabic environments . 55
\MkArbBreak*: ‘starred’ version which closes Arabic environments before processing declared commands. . 55

v1.5.
General: Compatibility with the quran package .......................... 51
Environments may be nested inside the arab environment .......... 47
\txarab: New \txarab environment for typesetting running paragraphs in Unicode Arabic ................. 58

v1.6.
\arabverse: New environment \arabverse for typesetting Arabic poetry .......................... 59
\bayt: New macro \bayt for typesetting each verse inside the \arabverse environment 61
\SetArbDflt*: This starred version applies the assimilation rules in addition to what \SetArbDflt already does. ..................... 56
\uc: \uc supersedes \cap .......................... 38

v1.7.
\arbnul: New \arbnul command for putting back on any contextual analysis rule broken by other commands. ..................... 64

v1.8.
General: arabica transliteration standard is now supported .......... 37
\MkArbBreak: New \MkArbBreak command for inserting user-defined macros in Arabic environments . 55
\MkArbBreak*: ‘starred’ version which closes Arabic environments before processing declared commands. . 55

v1.8.5.
General: Six additional Persian characters are now available ....... 9
\arbnul: New \arbnul command for putting back on any contextual analysis rule broken by other commands. ..................... 64

v1.9.
\MkArbBreak: New \MkArbBreak command for inserting user-defined macros in Arabic environments . 55
\abj: can now process \L\LaTeX counters .......................... 24
\arbcolor: Standard color command for Arabic environments .......... 56
\arbc: ‘starred’ version which closes Arabic environments before processing declared commands. . 55

v1.10.
\uc supersedes \cap .......................... 38
\prname: New command for typesetting Arabic proper names in transliteration .......................... 57

v1.11.
\arbmark: New command for inserting additional marks in Arabic environments .......................... 59
\newarbmark: Allows defining additional sets of Arabic marks .......... 59

v1.12.
\abjad can now process \L\LaTeX counters .......................... 24
\arbcolor: Standard color command for Arabic environments .......... 56
\arbc: ‘starred’ version which closes Arabic environments before processing declared commands. . 55

v1.13.
arbexport: Processes and print its argument in the current file and exports it in full Unicode in the external selected .tex file. .......................... 65
\arabverse: New options color and export to \arabverse environment. .......................... 59
\arbmark: New optional argument: either rl or lr .......................... 59
\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

v1.15.
\ayah: Prints End of Ayah sign .......... 63
v1.16.
\aemph: Now uses ulem ................. 55
v1.18.
\arind: New command \arind for building indexes ................. 62
v1.19.
\aemph: Now uses lua-ul ............... 55
\auline: Non context-sensitive command to underline Arabic words is provided ................................. 55
\StretchBayt: Optionally removes stretching from lines of poetry ........................................... 61
\SetArbNumbers: selects Indian or Arabic numbers .................................................. 55

v1.20.
\altrfont: new command \altrfont to store font and shape information (trans mode) .................. 57
\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.