arabluatex

ArabTEX for LuaLATEX

Robert Alessi
mailto:alessi@robertalessi.net

2023/05/21 – v1.21-dev

Contents

List of Tables

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
```

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

MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

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

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

1 Introduction

In comparison to Prof. Lagally's outstanding ArabTEX,[2] ArabLuaTEX is at present nothing more than a modest piece of software. Hopefully—if I may say so—it will eventually provide all of its valuable qualities to the LuaLATEX users.

arabtex dates back to 1992. As far as I know, it was then the first and only way to typeset Arabic texts with TEX and LATEX. To achieve that, arabtex provided—and still does—an Arabic font in *Nashī* style and a macro package that defined its own input notation which was, as the author stated, "both machine, and human, readable, and suited for electronic transmission and e-mail communication".[3] Even if the same can be said about Unicode, ArabTEX ASCII input notation still surpasses Unicode input, in my opinion, when it comes to typesetting complex documents, such as scientific documents or critical editions where footnotes and other kind of annotations can be particulary abundant. It must also be said that most text editors have trouble in displaying Arabic script connected with preceding or following LATEX commands: it often happens that commands seem misplaced, not to mention punctuation marks, or opening or closing braces, brackets or parentheses that are unexpectedly displayed in the wrong direction. Of course, some text editors provide ways to get around such difficulties by inserting invisible Unicode characters, such as LEFT-TO-RIGHT or RIGHT-TO-LEFT MARKS (U+200E, U+200F), RTL/LTR "embed" characters (U+202B, U+202A) and RLO/LRO "bidi-override" characters (U+202E, U+202D).[4] Nonetheless, it remains that inserting all the time these invisible characters in complex documents rapidly becomes confusing and cumbersome.

The great advantage of ArabTEX notation is that it is immune from all these difficulties, let alone its being clear and straightforward. One also must remember that computers are designed to process code. ArabTEX notation is a way of encoding Arabic language,

[2] See http://ctan.org/pkg/arabtex

[3] Klaus Lagally, *ArabTEX: Typesetting Arabic and Hebrew* (version 4.00) [User Manual Version 4.00] (Nov. 3, 2004), http://mirrors.ctan.org/language/arabic/arabtex/doc/html/arabtex.htm, 2.

[4] Gáspár Sinai's Yudit probably has the best Unicode support. See http://www.yudit.org.

just as TEX "mathematics mode" is a way of processing code to display mathematics. As such, not only does it allow greater control over typographical features, but it also can be processed in several different ways: so without going into details, depending on one's wishes, 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 XƎLATEX has arabxetex. Both of them are much more advanced than arabluatex, as they can process a number of different languages,[5] 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 `\emph`, `\textbf` 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 form what is not, it is processed by other Lua functions which rely on different sets of correpondence 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:

[5] 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[6] like so:—

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

Rem. *a* By default Amiri places the *kasrah* in combination with the *tašdīd* below the consonant, like so: ـِّـ. That is correct, as at least in the oldest manuscripts ـٍّـ may stand for ـٍّـ as well as ـِّـ. 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:—[8]

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

Other Arabic fonts may behave differently.

Rem. *b* `\newfontfamily` can be used to have either Indian of 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 overriden at any point of the document (see below sect. 2.3.1 on page 8):

`voc` `Default`

In this mode, which is the one selected by default, every short vowel written generates its corresponding diacritical mark: *ḍammah* (ـُـ), *fatḥah* (ـَـ) and *kasrah* (ـِـ). If a vowel is followed by `N`, viz. ⟨*uN*, *aN*, *iN*⟩, then the corresponding *tanwīn* (ـٌـ, ـًا, ـًة, ـًى or ـًـ) is generated. Finally, ⟨*u*, *a*, *i*⟩ at the commencement of a word indicate a "connective *ʾalif*" (*ʾalifu ʾl-waṣli*), but `voc` mode does not show the *waṣlah* above the *ʾalif*; instead, the accompanying vowel may be expressed at the beginning of a sentence (اُ اَ اِ).

`fullvoc`

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

`novoc`

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

`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):
dmg/dmg+ *Deutsche Morgenländische Gesellschaft* `dmg` is selected by default;
loc *Library of Congress*;
arabica *Arabica*.
More standards will be included in future releases of arabluatex.

`export` `export=true|false` `Default: false`

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

[6] Khaled Hosny, *Amiri* (Dec. 13, 2017), http://www.amirifont.org/.

[7] 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. 14 C–D.

[8] See the documentation of amiri, ibid., 6.

2.2.1 Classic contrasted with modern typesetting of Arabic

New feature v.1.2

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

- 'Classic' *maddah*: when *ʾalif* and *hamzah* accompanied by a simple vowel or *tanwīn* is preceded by an *ʾalif* of prolongation (ـَا), then a mere *hamzah* is written on the line, and a *maddah* is placed over the *ʾalif*, like so:—

  `samA'uN` سَمَآءٌ *samāʾun*, `jA'a` جَآءَ *ǧāʾa*, `yatasA'alUna` يَتَسَآءَلُونَ *yatasāʾalūna*[9] (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 شَيْـًٔا, ظِمْـًٔا and the like, the *hamzah* alone is not written over the letter *yāʾ* with no diacritical points below as in شَيْـًٔا, ظِمْـًٔا, 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`
New feature v1.4.4
`\SetArbEasy*`

Such refinements as 'classic' *maddah* may be discarded by the `\SetArbEasy` command, either globally in the preamble or locally at any point of the document. The difference between `\SetArbEasy` and its 'starred' version `\SetArbEasy*` is that the former keeps the *sukūn* that is generated by the `fullvoc` mode, while the latter further takes it away.

`\SetArbDflt`
New feature v1.6
`\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' version of this command `\SetArbDflt*` either in the preamble or at any point of the document.[10] Examples follow:—

(a) `\SetArbDflt`:
  i. `voc` وَمَاتَ استِسقَآءَ قَبلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَآءِ
  ii. `fullvoc` وَمَاتَ ٱسْتِسْقَآءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ ٱلسَّمَآءِ
  iii. `trans` *wa-māta ʾstisqāʾan qabla ʾan yutimma kitāba-hu fī nuǧūmi ʾs-samāʾi*

(b) `\SetArbDflt*`:
  i. `voc` وَمَاتَ استِسقَآءَ قَبلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَآءِ
  ii. `fullvoc` وَمَاتَ ٱسْتِسْقَآءَ قَبْلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ ٱلسَّمَآءِ
  iii. `trans` *wa-māta ʾstisqāʾan qabla ʾay yutimma kitāba-hu fī nuǧūmi ʾs-samāʾi*

(c) `\SetArbEasy`:
  i. `voc` وَمَاتَ استِسقَاءَ قَبلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ السَمَاءِ
  ii. `fullvoc` وَمَاتَ ٱسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ ٱلسَّمَاءِ
  iii. `trans` *wa-māta ʾstisqāʾan qabla ʾan yutimma kitāba-hu fī nuǧūmi ʾs-samāʾi*

(d) `\SetArbEasy*`:
  i. `voc` وَمَاتَ استِسقَاءَ قَبلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ السَمَاءِ

[9] Note that in old mss. such forms as سَمَأَ, جَأَ 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.

[10] For an example, see sect. 5.1 on page 28.

ii. `fullvoc` وَمَاتَ ٱسۡتِسۡقَاءً قَبلَ أَن يُتِمَّ كِتَابَهُۥ فِي نُجُومِ ٱلسَّمَاءِ

iii. `trans` *wa-māta ʾstisqāʾ^{an} qabla ʾan yutimma kitāba-hu fī nuǧūm^{i} ʾs-samāʾ^{i}*

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

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 \textcite[i. 1 A]{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).
```

From Wright:[a]— Arabic, like Hebrew and Syriac, is written and read from right to left. The letters of the alphabet (حُرُوفُ الهِجَآءِ, حُرُوفُ التَّهَجِّي, اَلحُرُوفُ الهِجَآئِيَّةُ, or حُرُوفُ المُعجَمِ) are twenty-eight in number and are all consonants, though three of them are also used as vowels (see § 3).

[a] Wright, see n. 7, i. 1 A.

The following example comes from Wright:[11]—

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

XVI. فَوَاعِلُ*.

1. فَاعَلٌ; as خَاتَمٌ *a signet-ring*, ...

`arab` (*env.*) Running paragraphs of Arabic text should rather be placed inside an *Arabic environment*

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

like so:—

```
\begin{arab}
  'at_A .sadIquN 'il_A ju.hA ya.tlubu min-hu .himAra-hu
  li-yarkaba-hu fI safraTiN qa.sIraTiN fa-qAla la-hu:
```

[11] Wright, see n. 7, i. 213 C.

```
\enquote{sawfa 'u`Idu-hu 'ilay-ka fI 'l-masA'-i
wa-'adfa`u la-ka 'ujraTaN.} fa-qAla ju.hA:
\enquote{'anA 'AsifuN jiddaN 'annI lA 'asta.tI`u 'an
'u.haqqiqa la-ka ra.gbata-ka fa-'l-.himAr-u laysa hunA
'l-yawm-a.}  wa-qabla 'an yutimma ju.hA kalAma-hu bada'a
'l-.himAr-u yanhaqu fI 'i.s.tabli-hi. fa-qAla la-hu
.sadIqu-hu: \enquote{'innI 'asma`u .himAra-ka yA ju.hA
yanhaqu.} fa-qAla la-hu ju.hA: \enquote{.garIbuN
'amru-ka yA .sadIqI 'a-tu.saddiqu 'l-.himAr-a
wa-tuka_d_diba-nI?}
\end{arab}
```

أَتَى صَدِيقٌ إِلَى جُحَا يَطلُبُ مِنهُ حِمَارَهُ لِيَركَبَهُ فِي سَفرَةٍ قَصِيرَةٍ فَقَالَ لَهُ: "سَوفَ أُعِيدُهُ إِلَيكَ فِي المَسَاءِ وَأَدفَعُ لَكَ أُجرَةً." فَقَالَ جُحَا: "أَنَا آسِفٌ جِدًّا أَنِّي لَا أَستَطِيعُ أَن أُحَقِّقَ لَكَ رَغبَتَكَ فَالحِمَارُ لَيسَ هُنَا اليَومَ." وَقَبلَ أَن يُتِمَّ جُحَا كَلَامَهُ بَدَأَ الحِمَارُ يَنهَقُ فِي إِصطَبلِهِ. فَقَالَ لَهُ صَدِيقُهُ: "إِنِّي أَسمَعُ حِمَارَكَ يَا جُحَا يَنهَقُ." فَقَالَ لَهُ جُحَا: "غَرِيبٌ أَمرُكَ يَا صَدِيقِي أَتُصَدِّقُ الحِمَارَ وَتُكَذِّبُنِي؟"

2.3.1 Local options

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

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

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

3 Standard ArabTEX input

3.1 Consonants

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

Letter	Transliteration[12]			ArabTEX notation
	`dmg+`	`loc`	`arabica`	
ا أَ أُ إِ[13]	*ʾu, ʾa, ʾi*	*'u, 'a, 'i*	*ʾu, ʾa, ʾi*	`'u` or `'a` or `'i`
ب	*b*	*b*	*b*	`b`
ت	*t*	*t*	*t*	`t`
ث	*t̲*	*th*	*t̲*	`_t`

Table 1: Standard ArabTEX (consonants)

[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.

Letter	Transliteration dmg+	loc	arabica	ArabTEX notation
ج	*ǧ*	*j*	*ǧ*	^g or j
ح	*ḥ*	*ḥ*	*ḥ*	.h
خ	*ḫ*	*kh*	*ḫ*	_h or x
د	*d*	*d*	*d*	d
ذ	*ḏ*	*dh*	*ḏ*	_d
ر	*r*	*r*	*r*	r
ز	*z*	*z*	*z*	z
س	*s*	*s*	*s*	s
ش	*š*	*sh*	*š*	^s
ص	*ṣ*	*ṣ*	*ṣ*	.s
ض	*ḍ*	*ḍ*	*ḍ*	.d
ط	*ṭ*	*ṭ*	*ṭ*	.t
ظ	*ẓ*	*ẓ*	*ẓ*	.z
ع	ʿ	ʿ	ʿ	`
غ	*ġ*	*gh*	*ġ*	.g
ف	*f*	*f*	*f*	f
ق	*q*	*q*	*q*	q
ك	*k*	*k*	*k*	k
ل	*l*	*l*	*l*	l
م	*m*	*m*	*m*	m
ن	*n*	*n*	*n*	n
ه	*h*	*h*	*h*	h
و	*w*	*w*	*w*	w
ي	*y*	*y*	*y*	y[14]
ة	*ah*	*ah*	*a*	T

Table 1: Standard ArabTEX (consonants)

Rem. *a* Please note that in all cases of elision, the *ʾalifu ʾl-waṣli* is expressed only by the vowel that accompanies the omitted *hamzah*: ⟨*u*, *a*, *i*⟩ as in `wa-inhazama` وَٱنْهَزَمَ *wa-ʾnhazama*. For more details on the definite article and the *ʾalifu ʾl-waṣli* see sect. 4.2 on page 16.
That said, ا 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 *ʾalif* may also be encoded as `.A` whether it be followed by a vowel or not, like so: `wa-.An` وَان *wa-.n* (where the dot symbolizes the absence of vowel), `wa-.Aan` وَآن *wa-an*, `wa-.Ain` وَإن *wa-in*.

Rem. *b* The letter ي with two points below, اَلِيَآءُ المُثَنَّاةُ مِن تَحتِهَا, may also be written without diacritical points as ى. When it is used as a consonant, it is encoded `aY`, where `a` recalls the *fatḥah* placed above the preceding letter in vocalized Arabic, like so: `qaY'uN` قَىٔ *qayʾun*, `^saY'uN` شَىٔ *šayʾun*, `^saY'aN` *šayʾan* شَئًا.
The same result may be achieved by encoding this letter as `.y`, like so: `qa.y'uN` قَىٔ *qayʾun*, `^sa.y'uN` شَىٔ *šayʾun*, `^sa.y'aN` *šayʾan* شَئًا.

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] For the letter ى with no diacritical points below, see Rem. *b* below.

Letter	Transliteration[15]			ArabTEX notation
	dmg+	loc	arabica[16]	
پ	*p*	*p*	*p*	p
چ	*č*	*ch*	*č*	^c
ژ	*ž*	*zh*	*ž*	^z
ڤ[17]	*v*	*v*	*v*	v
گ	*g*	*g*	*g*	g
ڭ[18]	*ñ*	*ñ*	*ñ*	^n

Table 2: Standard ArabTEX (additional characters)

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

3.3 Vowels

3.3.1 Long vowels

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

Letter	Transliteration[19]			ArabTEX notation
	dmg+	loc	arabica	
ا	*ā*	*ā*	*ā*	A
و	*ū*	*ū*	*ū*	U
ي	*ī*	*ī*	*ī*	I[20]
ى[21]	*ạ́*	*á*	*ā*	_A or Y
ـَـٰ	*ā*	*ā*	*ā*	_a
ـُ	*ū*	*ū*	*ū*	_u
ـِ	*ī*	*ī*	*ī*	_i

Table 3: Standard ArabTEX (long vowels)

Rem. *a* The long vowels *ā*, *ū*, *ī*, otherwise called *ḥurūf*^{u} *'l-madd*^{i}, *the letters of prolongation*, involve the placing of the short vowels *a*, *u*, *i* before the letters ا, و, ي respectively. arabluatex does that automatically in case any from voc, fullvoc or trans modes is selected e.g. قَالَ *qāla*, قِيلَ *qīla*, يَقُولُ *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.

[17] This character is not found in Carl Brockelmann et al., "Die Transliteration der arabischen Schrift in ihrer Anwendung auf die Hauptliteratursprachen der islamischen Welt", in *Denkschrift dem 19. internationalen Orientalistenkongreß in Rom vorgelegt von der Transkriptionkommission der Deutschen Morgenländischen Gesellschaft*, in collab. with Ph. S. van Ronkel and Otto Spies (Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockaus, 1935), http://www.naher-osten.uni-muenchen.de/studium_lehre/werkzeugkasten/dmgtransliteration.pdf, 2. It is taken from the *Information and Documentation - Romanization of the Arabic Alphabet for Arabic, Ottoman-Turkish, Persian, Kurdish, Urdu and Pushto* (July 2011), http://www.din.de standard.

[18] See 17.

[19] See below sect. 8 on page 37.

[20] For the letter ي with no diacritical points, see *Rem. c.* below.

[21] = *al-ʾalif*^{u} *'l-maqṣūrat*^{u}.

Rem. *b* Defective writings, such as ـٰ, *al-ʾalifᵘ ʾl-maḥḏūfatᵘ*, or defective writings of *ū* and *ī* are encoded `_a` `_u` and `_i` respectively, e.g. `_d_alika` ذٰلِكَ, `al-mal_a'ikaT-u 'l-ra.hm_an-u` اَلْمَلٰٓئِكَةُ الرَّحْمٰنُ, `.hu_dayfaT-u bn-u 'l-yamAn_i` حُذَيْفَةُ بْنُ ٱلْيَمَٰان for *Ḥuḏayfatᵘ bnᵘ ʾl-Yamānī*, etc.

Rem. *c* The letter ي with two points below, اَلْيَآءُ الْمُثَنَّاةُ مِن تَحتِهَا, 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: `liY` لِي *lī*, `yam^siY` يَمْشِى *yamšī*.

3.3.2 Short vowels

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

Letter	Transliteration[22]			ArabTEX notation
	`dmg+`	`loc`	`arabica`	
ـَ	*a*	*a*	*a*	`a`
ـُ	*u*	*u*	*u*	`u`
ـِ	*i*	*i*	*i*	`i`
ـً	*an*	*an*	*an*	`aN`
ـٌ	*un*	*un*	*un*	`uN`
ـٍ	*in*	*in*	*in*	`iN`

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.[23]

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

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

- at the commencement of a word, to indicate that a connective *ʾalif* is needed, with the exception of the article (see below sect. 4.4 on page 19);
- when arabluatex needs to perform a contextual analysis to determine the carrier of the *hamzah*;
- in the various transliteration modes, as vowels are always expressed in romanized Arabic.

4 arabluatex in action

4.1 The vowels and diphthongs

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

[22] See below sect. 8 on page 37.
[23] See below sect. 4.4 on page 19.

_halaqa (or xalaqa) خَلَقَ *ḫalaqa*, ^samsuN شَمْسٌ *šams*^{un}, karImuN كَرِيمٌ *Karīm*^{un}.

bi-hi بِهِ *bi-hi*, 'aqi.tuN أَقِطٌ *ʾaqiṭ*^{un}.

la-hu لَهُ *la-hu*, .hujjaTuN حُجَّةٌ *ḥuǧǧat*^{un}.

Long vowels They are written ⟨*U, A, I*⟩:

qAla قَالَ *qāla*, bI`a بِيعَ *bīʿa*, .tUruN طُورٌ *ṭūr*^{un}, .tInuN طِينٌ *ṭīn*^{un}, murU'aTuN مُرُوءَةٌ *murūʾat*^{un}.

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

al-fat_A اَلْفَتَى *al-fatā*, al-maqh_A اَلْمَقْهَى *al-maqhā*, 'il_A إِلَى *ʾilā*.

ʾalif otiosum Said *ʾalif*^{u} *ʾl-wiqāyat*^{i}, "the guarding *ʾalif*", after و at the end of a word, both when preceded by *ḍammah* and by *fatḥah* is written ⟨*UA*⟩ or ⟨*aW, aWA*⟩:

na.sarUA نَصَرُوا *naṣarū*, katabUA كَتَبُوا *katabū*, ya.gzUA يَغْزُوا *yaġzū*, ramaW رَمَوْا *ramaw*, banaWA بَنَوْا, *banaw*.

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

al-l_ah-u اَللهُ *al-lāh*^{u}, 'il_ahuN إِلٰهٌ *ʾilāh*^{un}.

al-ra.hm_an-u اَلرَّحمٰنُ *ar-raḥmān*^{u}, l_akin لٰكِن *lākin*, h_ahunA هٰهُنَا *hāhunā*, .hunayn-u bn-u 'is.h_aq-a حُنَيْنُ بْنُ إِسْحٰقَ *Ḥunayn*^{u} *bn*^{u} *ʾIsḥāq*^{a}, rabb_i رَبِّ *rabbī*, al-`A.s_i اَلْعَاصِ *al-ʿĀṣī*.

Silent و/ي Some words ending with ـَاة are usually written ـَوة or ـوة instead of ـَاة: see Wright.[24] arabluatex preserves that particular writing; the same applies to words ending in ـَية for ـَاة. Long vowels ⟨*U, I*⟩ shall receive no *sukūn* after a *ʾalif maḥḏūfah* and are discarded in trans mode:

.hay_aUTuN حَيْوٰةٌ *ḥayāt*^{un}, .sal_aUTuN صَلٰوةٌ *ṣalāt*^{un}, mi^sk_aUTuN مِشْكٰوةٌ *miš-kāt*^{un}, tawr_aITuN تَوْرٰيةٌ *tawrāt*^{un}.

And so also: al-rib_aIT-u اَلرِّبٰيةُ *ar-ribāt*^{u}.

***ʿAmr*^{un}, and the silent و** To that name a silent و is added to distinguish it from *ʿUmar*^{u}: see Wright.[25] In no way this affects the sound of the *tanwīn*, so it has to be discarded in trans mode:

[24] Wright, see n. 7, i. 12 A.
[25] Ibid., i. 12 C.

`amruNU عَمْرٌو *ʿamr*$^{un}$, `amraNU عَمْرًا *ʿamr*^{an}, `amriNU عَمْرٍو *ʿamr*^{in}.

When the *tanwīn* falls away (Wright, *A Grammar of the Arabic Language*, i. 249 B): `amr-uU bn-u mu.hammadiN عَمْرُو بْنُ مُحَمَّدٍ *ʿAmr*$^{u}$ *bn*$^{u}$ *Muḥammad*$^{in}$, mu.hammad-u bn-u `amr-iU bn-i _hAlidiN مُحَمَّدُ بْنُ عَمْرِو بْنِ خَالِدٍ *Muḥammad*^{u} *bn*^{u} *ʿAmr*^{i} *bn*^{i} *Ḫālid*^{in}.

And so also: al-rib_aUA اَلرِّبٰوا *ar-ribā*, ribaNU رِبًوا *rib*^{an}.

tanwīn The marks of doubled short vowels, ـٌ, ـً, ـٍ, are written ⟨*uN*, *aN*, *iN*⟩ respectively. arabluatex deals with special cases, such as ـً taking an ا after all consonants except ة, and *tanwīn* preceding ى as in هُدًى, which is written ⟨*aN_A*⟩ or ⟨*aNY*⟩:

mAluN مَالٌ *māl*^{un}, bAbaN بَابًا *bāb*^{an}, madInaTaN مَدِينَةً *madīnat*^{an}, bintiN بِنتٍ *bint*^{in} maqhaN_A مَقهًى *maqhạ*^{n}, fataNY فَتًى *fatạ*^{n}.

arabluatex is aware of special orthographies: ^say'uN شَيْءٌ *šayʾun*, ^say'aN شَيْئًا *šayʾan*, ^say'iN شَيْءٍ *šayʾin*.

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

al-qA.dI اَلقَاضِي *al-qāḍī*, qA.diyaN قَاضِيًا *qāḍiy*^{an}, qA.diNI قَاضٍ *qāḍi*^{n}.

4.2 Other orthographic signs

tāʾ marbūṭah It is written ⟨*T*⟩:

madInaTuN مَدِينَةٌ *madīnat*^{un}, madInaTaN مَدِينَةً *madīnat*^{an}, madInaTiN مَدِينَةٍ *madīnat*^{in}.

hamzah It is written ⟨ʾ⟩, 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*: 'asaduN أَسَدٌ *ʾasad*^{un}, 'u_htuN أُخْتٌ *ʾuḫt*^{un}, 'iqlIduN إِقلِيدٌ *ʾiqlīd*^{un}, 'anna أَنَّ *ʾanna*, 'inna إِنَّ *ʾinna*.

hamzah followed by the long vowel و is encoded '_U: '_Ul_A أُولَى *ʾūlạ*, '_UlU أُولُو *ʾūlū*, '_UlA'ika أُولَآئِكَ *ʾūlāʾika*.

hamzah followed by the long vowel ي is encoded '_I: '_ImAnuN إِيمَانٌ *ʾīmān*^{un}[26].

[26] For another way of encoding the initial *hamzah* followed by a long vowel, see the *taḫfīf*^{u} *ʾl-hamzat*^{i} on the next page.

Middle *hamzah*: xA.ti'-Ina خَاطِئِينَ *ḫāṭiʾ^{ī}na*, ru'UsuN رُءُوسٌ, *ruʾūs^{un}*, xa.tI'aTuN خَطِيئَةٌ *ḫaṭīʾat^{un}*, su'ila سُئِلَ *suʾila*, 'as'ilaTuN أَسئِلَةٌ *ʾasʾilat^{un}*, mas'alaTuN مَسئَلَةٌ *masʾalat^{un}*, 'as'alu أَسئَلُ *ʾasʾalu*, yatasA'alUna يَتَسَآءَلُونَ *yatasāʾalūna*, murU'aTuN مُرُوءَةٌ *murūʾat^{un}*, taw'amuN تَوْءَمٌ *tawʾam^{un}*, ta'xIruN تَأْخِيرٌ *taʾḫīr^{un}*, ta'axxara تَأَخَّرَ *taʾaḫḫara*, ji'tu-ka جِئْتُكَ *ǧiʾtu-ka*, qA'iluN قَآئِلٌ *qāʾil^{un}*, .hIna'i_diN *ḥīnaʾiḏ^{in}* حِينَئِذٍ, hay'aTuN هَيْئَةٌ *hayʾat^{un}*, hay'AtuN هَيْئَاتٌ *hayʾāt^{un}*.

From Wright:[27]— All consonants, whatsoever, not even *ʾalif hèmzatum* excepted, admit of being doubled and take *tašdīd*. Hence we speak and write ra''AsuN رَأَّسٌ *raʾʾās^{un}*, sa''AluN سَأَّالٌ *saʾʾāl^{un}*, na''AjuN نَأَّاجٌ *naʾʾāǧ^{un}*.

Final *hamzah*: xa.ta'uN خَطَأٌ *ḫaṭaʾ^{un}*, xa.ta'aN خَطَأً *ḫaṭaʾ^{an}*, xa.ta'iN خَطَإٍ *ḫaṭaʾ^{in}*, 'aqra'u أَقرَأُ *ʾaqraʾu*, taqra'Ina تَقرَئِينَ *taqraʾīna*, taqra'Una تَقرَؤُونَ *taqraʾūna*, yaqra'na يَقرَأْنَ *yaqraʾna*, yaxba'Ani يَخبَآنِ *yaḫbaʾāni*, xaba'A خَبَآ *ḫabaʾā*, xubi'a خُبِئَ *ḫubiʾa*, xubi'UA خُبِئُوا *ḫubiʾū*, jA'a جَآءَ *ǧāʾa*, ridA'uN رِدَآءٌ *ridāʾ^{un}*, ridA'aN رِدَآءً *ridāʾ^{an}*, jI'a جِيءَ *ǧīʾa*, radI'iN رَدِيءٍ *radīʾ^{in}*, sU'uN سُوءٌ *sūʾ^{un}*, .daw'uN ضَوءٌ *ḍawʾ^{un}*, qay'iN قَيءٍ *qayʾ^{in}*, ^sifA'I شِفَآئِي *šifāʾī*, man^sa'I مَنشَئِي *manšaʾī*, nisA'uN نِسَآءٌ *nisāʾ^{un}*, nisA'u-hu نِسَآؤُهُ *nisāʾu-hu*, nisA'i-hi نِسَآئِهِ *nisāʾi-hi*, nisA'I نِسَآئِي *nisāʾī*.

^say'uN شَيءٌ *šayʾ^{un}*, ^say'aN شَيئًا *šayʾ^{an}*, ^say'iN شَيءٍ *šayʾ^{in}*, al-^say'-u اَلشَّيءُ *aš-šayʾu*, 'a^syA'-u أَشيَآءُ *ʾašyāʾu*, 'a^syA'-a أَشيَآءَ *ʾašyāʾa*, .zim'aN ظِمئًا *ẓimʾ^{an}*, radI'aN رَدِيئًا *radīʾ^{an}*.

taḫfīf^{u} ʾl-hamzat^{i}: if the *hamzah* has *ǧazmah* and is preceded by *ʾalif hamzatum*, it must be changed into the letter of prolongation that is homogeneous with the preceding vowel; hence: 'a'mana آمَنَ *ʾāmana*, 'u'minu أُومِنُ *ʾūminu*, 'i'mAnuN إِيمَانٌ *ʾīmān^{un}*. For other possible ways of encoding such sequences, see on the preceding page (*hamzah* followed by و and ي) and the *maddah* on the next page.

Imperatives of verbs that have the *hamzah* as the first radical are other cases of *taḫfīf^{u} ʾl-hamzat^{i}*: i'sir اِسِرْ *īsir*, i'_dan اِيذَنْ *īḏan*, u'mul اُومُلْ *ūmul*. arabluatex also provides ways of encoding those words when the initial *ʾalif* comes into *waṣl*, so as to make the *ʾalif waṣl* fall away when preceded by وَ or فَ: wa-'sir وَأْسِرْ *wa-ʾsir*, fa-'_dan فَأْذَنْ *fa-ʾḏan*, fa-'ti فَأْتِ *fa-ʾti*, wa-'tamirUA وَأْتَمِرُوا *wa-ʾtamirū*; or be retained outside the imperative, as in fa-i'tazarat فَاِئْتَزَرَتْ *fa-ʾʾtazarat*, ba`da i'tilAfiN بَعْدَ اِئْتِلَافٍ *baʿda ʾʾtilāf^{in}*.

[27] Wright, see n. 7, i. 14 B.

The strange spelling of *mi'at^{un}*: `mi'aTuN` مِائَةٌ *mi'at^{un}*, `mi'atAni` مِائَتَانِ *mi'atāni*, `mi'atayni` مِائَتَيْنِ *mi'atayni*, `mi'Una` مِئُونَ *mi'ūna*, `mi'AtuN` مِئَاتٌ *mi'ātun*, `mi'aN_A` مِأًى *mi'a^{n}*. Of course, the 'pipe' character can be used to prevent this rule from being applied (see sect. 4.5 on page 21): `mi'a|TuN` مِئَةٌ *mi'at^{un}*.

maddah At the beginning of a syllabe, *'alif* with *hamzah* and *fatḥah* (أَ) followed by *'alifu 'l-maddi* (*'alif* of prolongation) or *'alif* with *hamzah* and *ǧazmah* (أْ) are both represented in writing *'alif* with *maddah*: آ (see Wright, *A Grammar of the Arabic Language*, i. 25 A–B).

> Hence one should keep to this distinction and encode `'a'kulu` آكُلُ *'ākulu* and `'AkiluN` آكِلٌ *'ākil^{un}* respectively.

arabluatex otherwise determines *al-'alif^{u} 'l-mamdūdat^{u}* by context analysis.

`'is'AduN` إِسْآدٌ *'is'ād^{un}*, `'AkilUna` آكِلُونَ *'ākilūna*, `'a'mannA` آمَنَّا *'āmannā*, `al-qur'An-u` اَلْقُرْآنُ *al-qur'ān^{u}*.

`jA'a` جَآءَ *ǧā'a*, `yatasA'alUna` يَتَسَآءَلُونَ *yatasā'alūna*, `ridA'uN` رِدْآءٌ *ridā'^{un}*, `xaba'A` خَبَآ *ḫaba'ā*, `yaxba'Ani` يَخْبَآنِ *yaḫba'āni*.

šaddah *tašdīd* is either *necessary* or *euphonic*.

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:

`` `allaqa `` عَلَّقَ *'allaqa*, `mAdduN` مَادٌّ *mādd^{un}*, `'ammara` أَمَّرَ *ammara*, `murruN` مُرٌّ *murr^{un}*.

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 ت, ث, د, ذ, ر, ز, س, ش, ص, ض, ط, ظ, ل, ن, after the article اَلْ:—

> 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` اَلشَّمْسُ *aš-šams^{u}*, or `a^s-^sams-u` اَلشَّمْسُ *aš-šams^{u}*.

`al-tamr-u` اَلتَّمْرُ *at-tamr^{u}*, `al-ra.hm_an-u` اَلرَّحْمٰنُ *ar-raḥmān^{u}*, `al-.zulm-u` اَلظُّلْمُ *aẓ-ẓulm^{u}*, `al-lu.gaT-u` اَللُّغَةُ *al-luġat^{u}*.

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

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` مِن رَّبِّهِ, *mir rabbi-hi*, `min layliN` مِن لَّيْلٍ *mil layl*^{in}, `'an yaqtula` أَن يَقْتُلَ *ʾay yaqtula*.

With *tanwīn*: `kitAbuN mubInuN` كِتَابٌ مُّبِينٌ *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 *ʾidġām*, will be included in arabluatex gradually.

(c) With the letter ت after the dentals ث, د, ذ, ض, ط, ظ in certain parts of the verb: this kind of assimilation, e.g. لَبِتُّ for لَبِثْتُ *labiṯṯu*, 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 *ʾalif*^{u} *ʾl-waṣl*^{i} At the beginning of a sentence, اَ is never written, as اَلْحَمْدُ لِلّٰهِ; instead, to indicate that the *ʾalif* is a connective *ʾalif* (*ʾalif*^{u} *ʾl-waṣl*^{i}), the *hamzah* is omitted and only its accompanying vowel is expressed:

`al-.hamd-u li-l-l_ah-i` اَلْحَمْدُ لِلّٰهِ *al-ḥamd*^{u} *li-l-lāh*^{i}.

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 *ʾalifu ʾl-waṣli* is expressed by the vowel that accompanies the omitted *hamzah*: ⟨*u*, *a*, *i*⟩.

Article: `bAb-u 'l-madrasaT-i` بَابُ اَلْمَدْرَسَةِ *bāb*^{u} *ʾl-madrasat*^{i}, `al-maqAlaT-u 'l-'_Ul_A` اَلْمَقَالَةُ اَلْأُولَى *al-maqālat*^{u} *ʾl-ʾūlā*, `al-lu.gaT-u 'l-`arabiyyaT-u` اَللُّغَةُ اَلْعَرَبِيَّةُ *al-luġat*$^{u}$ *ʾl-ʿarabiyyat*$^{u}$, `fI .sinA`aT-i 'l-.tibb-i` فِي صِنَاعَةِ اَلطِّبِّ *fī ṣināʿat*^{i} *ʾṭ-ṭibb*^{i}, `'il_A 'l-intiqA.d-i` إِلَى اَلِانْتِقَاضِ *ʾilạ ʾl-intiqāḍ*^{i}, `fI 'l-ibtidA'-i` فِي اَلِابْتِدَاءِ *fi ʾl-ibtidāʾ*^{i}, `'abU 'l-wazIr-i` أَبُو اَلْوَزِيرِ *ʾabu ʾl-wazīr*^{i}, `fa-lammA ra'aW 'l-najm-a` فَلَمَّا رَأَوُا اَلنَّجْمَ *fa-lammā raʾawu ʾn-naǧm*^{a}.

Particles:—

(a) *li-*: *ʾalif*^{u} *ʾl-waṣl*^{i} is omitted in the article اَلْ when it is preceded by the preposition لِ: `li-l-rajul-i` لِلرَّجُلِ *li-r-raǧul*^{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` لِلَّيْلَةِ *li-l-laylat*^{i}.

(b) *la-*: the same applies to the affirmative particle لَ: `la-l-.haqq-u` لَلْحَقُّ *la-l-ḥaqq^{u}*.

(c) With the other particles, *ʾalif^{u} ʾl-waṣl^{i}* is expressed: `fI 'l-madInaT-i` فِي ٱلْمَدِينَةِ *fi ʾl-madīnat^{i}*, `wa-'l-rajul-u` وَٱلرَّجُلُ *wa-ʾr-raǧul^{u}*, `bi-'l-qalam-i` بِٱلْقَلَمِ *bi-ʾl-qalam^{i}*, `bi-'l-ru`b-i` بِٱلرُّعْبِ *bi-ʾr-ruʿb^{i}*.

Perfect active, imperative, nomen actionis: `qAla isma`` قَالَ ٱسْمَعْ *qāla ʾsmaʿ*, `qAla uqtul` قَالَ ٱقْتُلْ *qāla ʾqtul*, `huwa inhazama` هُوَ ٱنْهَزَمَ *huwa ʾnhazama*, `wa-ustu`mila` وَٱسْتُعْمِلَ *wa-ʾstuʿmila*, `qadi in.sarafa` قَدِ ٱنْصَرَفَ *qadi ʾnṣarafa*, `al-iqtidAr-u` ٱلِاقْتِدَارُ *al-iqtidār$^{u}$*, `'il_A 'l-intiqA.d-i` إِلَى ٱلِانْتِقَاضِ *ʾilạ ʾl-intiqāḍ$^{i}$*, `law istaqbala` لَوِ ٱسْتَقْبَلَ *lawi ʾstaqbala*.

Other cases: `'awi ismu-hu` أَوِ ٱسْمُهُ *ʾawi ʾsmu-hu*, `zayduN ibn-u `amriNU` زَيْدٌ ٱبْنُ عَمْرٍو *Zayd^{uni} ʾbn^{u} ʿAmr^{in}*,[28] ``umar-u ibn-u 'l-_ha.t.tAb-i` عُمَرُ ٱبْنُ ٱلْخَطَّابِ *ʿUmar$^{u}$ ʾbn$^{u}$ ʾl-Ḫaṭṭāb$^{i}$*,[29] `imru'-u 'l-qays-i` اِمْرُؤُ ٱلْقَيْسِ *Imruʾ$^{u}$ ʾl-Qays$^{i}$*, `la-aymun-u 'l-l_ah-i` لَأَيْمُنُ ٱللهِ *la-ʾymun^{u} ʾl-lāh^{i}*.

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

`fI 'l-nAs-i` فِي ٱلنَّاسِ *fi ʾn-nās^{i}*, `'abU 'l-wazIr-i` أَبُو ٱلْوَزِيرِ *ʾabu ʾl-wazīr^{i}*, `fI 'l-ibtidA'-i` فِي ٱلِابْتِدَاءِ *fi ʾl-ibtidāʾ^{i}*, `_dU 'l-i`lAl-i` ذُو ٱلِاعْلَالِ *ḏu ʾl-iʿlāl$^{i}$*, `maqh_A 'l-'amIr-i` مَقْهَى الأَمِيرِ *maqhạ ʾl-ʾamīr^{i}*.

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

`fI `aynay 'l-malik-i` فِي عَيْنَيِ ٱلْمَلِكِ *fī ʿaynayi ʾl-malik$^{i}$*, `ix^say 'l-qawm-a` اِخْشَيِ ٱلْقَوْمَ *iḫšayi ʾl-qawm$^{a}$*, `mu.s.tafaw 'l-l_ah-i` مُصْطَفَوُ ٱللهِ *muṣṭafawu ʾl-lāh^{i}*.

`ramaW 'l-.hijAraT-a` رَمَوُا ٱلْحِجَارَةَ *ramawu ʾl-ḥiǧārat^{a}*, `fa-lammA ra'aW 'l-najm-a` فَلَمَّا رَأَوُا ٱلنَّجْمَ *fa-lammā raʾawu ʾn-naǧm^{a}*.

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

[28] "*Zayd* is the son of *ʿAmr*": the second noun is not in apposition to the first, but forms part of the predicate. Hence زَيْدٌ ابْنُ عَمْرٍو and not زَيْدُ بْنُ عَمْرٍو, "Zayd, son of ʿAmr".

[29] "*ʿUmar* is the son of *al-Ḫaṭṭāb*" (see n. 28).

'antumu 'l-kA_dib-Una أَنْتُمُ ٱلْكَاذِبُونَ *ʾantumu ʾl-kāḏib^{ūna}*, ra'aytumu 'l-rajul-a رَأَيْتُمُ ٱلرَّجُلَ *raʾaytumu ʾr-raǧul^{a}*, mani 'l-ka_d_dAb-u مَنِ ٱلْكَذَّابُ *mani ʾl-kaḏḏāb^{u}*, qatalati 'l-rUm-u قَتَلَتِ ٱلرُّومُ *qatalati ʾr-Rūm^{u}*.

However, the Arabic script does not show the *kasrah* or the *ḍammah* 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-nabI مُحَمَّدٌ ٱلنَّبِي *Muḥammad^{uni} ʾn-nabī*, salAmuN ud_hulUA سَلَامٌ ٱدْخُلُوا *salām^{unu} ʾdḫulū*, qa.sIdata-hu fI qatl-i \uc{'a}bI \uc{m}uslimiN 'llatI yaqUlu fI-hA قَصِيدَتَهُ فِي قَتْلِ أَبِي مُسْلِمٍ ٱلَّتِي يَقُولُ فِيهَا *qaṣīdata-hu fī qatl^{i} ʾAbī Muslim^{ini} ʾllatī yaqūlu fī-hā*.

4.3 Special orthographies

The name of God The name of God, ٱللّٰهُ, is compounded of the article ٱلْ, and إِلَاهُ (noted إِلٰهُ with the defective *ʾalif*) so that it becomes ٱلْإِلَاهُ; then the *hamzah* is suppressed, its vowel being transferred to the ل before it, so that there remains ٱلْلَاهُ (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 *tašdīd* 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-l_ah-u or 'l-l_ah-u:—

al-l_ah-u ٱللّٰهُ *al-lāh^{u}*, yA|[30] al-l_ah-u يَا ٱللّٰهُ *yā al-lāh^{u}*, 'a-fa|[31]-al-l_ah-i la-ta.g`alanna أَفَٱللّٰهِ لَتَجْعَلَنَّ *ʾa-fa-al-lāh^{i} la-taǧʿalanna*, bi-'l-l_ah-i بِٱللّٰهِ *bi-ʾl-lāh^{i}*, wa-'l-l_ah-i وَٱللّٰهِ *wa-ʾl-lāh^{i}*, bi-sm-i 'l-l_ah-i بِسْمِ ٱللّٰهِ *bi-sm^{i} ʾl-lāh^{i}*, al-.hamd-u li-l-l_ah-i ٱلْحَمْدُ لِلّٰهِ *al-ḥamd^{u} li-l-lāh^{i}*, li-l-l_ah-i 'l-qA'il-u لِلّٰهِ ٱلْقَائِلُ *li-l-lāh^{i} ʾl-qāʾil^{u}*.

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

'a_hAfu mina 'l-malik-i 'lla_dI ya.zlimu 'l-nAs-a أَخَافُ مِنَ ٱلْمَلِكِ ٱلَّذِي يَظْلِمُ ٱلنَّاسَ *ʾaḫāfu mina ʾl-malik^{i} ʾllaḏī yaẓlimu ʾn-nās^{a}*, `udtu 'l-^say_h-a 'lla_dI huwa marI.duN عُدْتُ ٱلشَّيْخَ ٱلَّذِي هُوَ مَرِيضٌ *ʿudtu ʾš-šayḥ^{a} ʾllaḏī huwa marīḍ^{un}*, mA 'anA bi-'lla_dI qA'iluN la-ka ^say'aN مَا أَنَا بِٱلَّذِي قَائِلٌ لَكَ شَيْئًا *mā ʾanā bi-ʾllaḏī qāʾil^{un} la-ka šayʾ^{an}*.

[30] Note the "pipe" character '|' here after yA and below after fa before footnote mark 31: it is needed by the dmg transliteration mode as in this mode any vowel at the commencement of a word preceded by a word that ends with a vowel, either short or long, is absorbed by this vowel viz. *ʿalạ ʾṭ-ṭarīq^{i}*. See sect. 4.5 on page 21 on the "pipe" and sect. 8 on page 37 on dmg mode.

[31] See 30.

`'ari-nA 'lla_dayni 'a.dallA-nA mina 'l-jinn-i wa-'l-'ins-i` أَرِنَا ٱلَّذَيْنِ أَضَلَّانَا مِنَ ٱلْجِنِّ وَٱلْإِنْسِ *ʾari-na ʾllaḍayni ʾaḍallā-nā mina ʾl-ǧinn^{i} wa-ʾl-ʾins^{i}*.

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

`fa-'innA na_dkuru 'l-.sawt-ayni 'l-la_dayni rawaynA-humA `an ja.h.zaT-a` فَإِنَّا نَذْكُرُ ٱلصَّوْتَيْنِ ٱلَّذَيْنِ رَوَيْنَاهُمَا عَنْ جَحْظَةَ *fa-ʾinnā naḍkuru ʾṣ-ṣawt^{ayni} ʾl-laḍayni rawaynā-humā ʿan Ǧaḥẓat^{a}*.

And also: `al-la_dAni` اَللَّذَانِ *al-laḍāni*, `al-la_dayni` اَللَّذَيْنِ *al-laḍayni*, `al-latAni` اَللَّتَانِ *al-latāni*, `al-latayni` اَللَّتَيْنِ *al-latayni*, `al-lAtI` اَللَّاتِي *al-lātī*, `al-lA'|Ati`[32] اَللَّآءَاتِ *al-lāʾāti*, `al-lA'I` اَللَّآئِي *al-lāʾī*, and so forth.

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: علّم علم الهيئة; is it عُلِّمَ, *he was taught the science of astronomy*, or عَلَّمَ, *he taught the science of astronomy*? In order to disambiguate this clause, it may be sensible to put a *ḍammah* above the first ع: عُلِّم علم الهيئة, which is achieved by "quoting" the vowel `u`, like so: `` `"ullima ``, or, with no other vowel than the required `u`: `` `"ullm ``.

This is how the "quoting" mechanism works: metaphorically speaking, it acts as a *toggle switch*. If something, in a given mode, is supposed to be visible, "quoting" hides it; conversely, if it is supposed not to, it makes it visible.

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

`novoc` In this mode, "quoting" essentially means make visible something that ought not to be so.

(a) Quoting a vowel, either short or long, makes the *ḍammah*, *fatḥah* or *kasrah* appear above the appropriate consonant:—

`` `"ullima `ilm-a 'l-hay'aT-i `` عُلِّم علم الهيئة *ʿullima ʿilm^{a} ʾl-hayʾat^{i}*, `ya.gz"UA` يغزُوا *yaġzū*.

(b) The same applies when "quoting" the *tanwīn*:—

`wa-'innA sawfa tudriku-nA 'l-manAyA muqadd"araT"aN` وإنّا سوف تدركنا المنايا مقدَّرةً, *wa-ʾinnā sawfa tudriku-na ʾl-manāyā muqaddarat^{an}*.

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

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

[33] See Klaus Lagally, *ArabTEX: Typesetting Arabic and Hebrew* (version 4.00) [User Manual Version 4.00] (Nov. 3, 2004), http://mirrors.ctan.org/language/arabic/arabtex/doc/html/arabtex.htm, 22

qAla isma`" قال اسمعْ *qāla ʾsmaʿ*, jA'at" hinduN جاءَتْ هِندٌ *ǧāʾat Hind$^{un}$*, ^sabIhuN bi-man q"u.ti`at" qadamA-hu شبيه بمن قُطعتْ قدماه *šabīh^{un} bi-man quṭiʿat qadamā-hu*.

(d) At the commencement of a word, the straight double quote is interpreted as *ʾalif^{u} ʾl-waṣl^{i}*:—

wa-"ust"u`mila واستُعمل *wa-ʾstuʿmila*, huwa "inhazama هو انهزم *huwa ʾnhazama*, al-"intiqA.d-u الانتقاض *al-intiqāḍ^{u}*.

voc In accordance with the general rule, in this mode, "quoting" makes the vowels and the *tanwīn* disappear, should this feature be required for some reason:—

(a) Short and long vowels:—

q"Ala q"A'iluN قالَ قائِلٌ *qāla qāʾil^{un}*, ibn-u 'abI 'u.saybi`aT-"a اِبنُ أَبِي أُصَيبِعَة *Ibn^{u} ʾAbī ʾUṣaybiʿat^{a}*.

(b) *tanwīn*:—

madInaT"aN مَدِينَة *madīnat^{an}*, bAb"aN بَابا *bāb^{an}*, hud"aN_A هُدى *huḍ^{an}*, ^say'"iN شَيء *šay^{ʾin}*.

One may more usefully "quote" the initial vowels to write the *waṣlah* above the *ʾalif* or insert a straight double quote after a consonant not followed by a vowel to make the *sukūn* appear:—

(a) *ʾalif^{u} ʾl-waṣl^{i}*:—

fI "istiq.sA'-iN فِي اِستِقصَآءٍ *fi ʾstiqṣāʾ^{in}*, wa-"istiq.sA'-uN وَاِستِقصَآءٌ *wa-ʾstiqṣāʾ^{un}*, qAla "uhrub fa-lan tuqtala قَالَ اُهرُب فَلَن تُقتَلَ *qāla ʾhrub fa-lan tuqtala*.

(b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala قَالَ اُقتُلْ فَلَن تُقتَلَ *qāla ʾqtul fa-lan tuqtala*, mA jA'at" mini imra'aTiN مَا جاءَتْ مِن امرَأَةٍ *mā ǧāʾat mini ʾmraʾat^{in}*, kam" qad" ma.dat" min" laylaTiN كَمْ قَدْ مَضَتْ مِنْ لَيلَةٍ *kam qad maḍat min laylat^{in}*.

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 اَلجَمْرُ اَلصَّيْفِيُّ اَلَّذِي كَانَ بِقرَانُونَ *al-ǧamr^{u} ʾṣ-ṣayfiyy^{u} ʾllaḏī kāna bi-Qrānnūn^{a}*.

4.4.1 Quoting the *hamzah*

As said above in sect. 4.2 on page 13, the *hamzah* is always written ⟨ ʾ ⟩, its carrier being determined by contextual analysis. "Quoting" that straight single quote character like so: ⟨ "ʾ ⟩ 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.

Letter	Transliteration[34] dmg+	loc	arabica	ArabTEX notation
ء	ʾ	’	ʾ	|"'
آ	ʾā	’ā	ʾā	A"'
أ	ʾ	’	ʾ	a"'
أ	ʾ	’	ʾ	u"'
ؤ	ʾ	’	ʾ	w"'
إ	ʾ	’	ʾ	i"'
ئ	ʾ	’	ʾ	y"'

Table 5: “Quoted” *hamzah*

As one can see from table 5, the carrier of the *hamzah* is inferred from the letter that precedes the straight double quote ⟨"⟩. Of course, any “quoted” *hamzah* may take a short vowel, which is to be written *after* the ArabTEX equivalent for the *hamzah* itself, namely ⟨'⟩. For example, ؤَ is encoded ⟨w"'a⟩, while ؤْ 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.

'a`dA'ukum أَعْدَاؤُكُمْ *ʾaʿdāʾukum*, 'a`dA|"'ukum أَعْدَاءُكُمْ *ʾaʿdāʾukum*, 'a`dA'ikum أَعْدَائِكُمْ *ʾaʿdāʾikum*, 'a`dA|"'ikum أَعْدَاءِكُمْ *ʾaʿdāʾikum*.

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: |"' ء.

Aside from that usage, the “pipe” character is used to prevent almost any of the contextual analysis rules that are described above from being applied. Two examples have already been given to demonstrate how that particular mechanism works in 30 on page 18 and in 32 on page 19. One more example follows:—

bi-qrAn|nUn-a بِقرَانُنُونَ *bi-Qrānnūn*^{a}, “in Crannon” (Thessaly, Greece).[35]

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

4.6 Putting back on broken contextual analysis rules

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:—

```
This is wrong:
\begin{arab}[fullvoc]
  fa-lammA ra'aW\LRfootnote{A footnote which interferes with
    the contextual analysis.} 'l-na^gma...
```

[34] See below sect. 8 on page 37.

[35] See more context on the previous page.

```
\end{arab}
```

This is wrong:

فَلَمَّا رَأَوْا[a] ٱلنَّجْمَ...

[a] 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 *ʾalif*[u] *ʾl-waṣl*[i], as رَأَوُا ٱلنَّجْمَ.

\arbnull The `\arbnull` command is provided so as to put back on contextual analysis rules in such situations. It takes as argument the word that must be brought back for any given rule to be applied as it ought to. Depending on the contexts that have to be restored, `\arbnull` may be found just after or before Arabic words.

In any case, *no space must be left* after or before the Arabic word that `\arbnull` is applied to.

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

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

  qAla\LRfootnote{A footnote which interferes with the
    contextual analysis.} \arbnull{qAla}uhrub fa-lan tuqtala.

  \uc{z}ayduN\arbnull{ibnu}\LRfootnote{A footnote which
    interferes with the contextual analysis.}
  \arbnull{zayduN}ibn-u \uc{`a}mriNU.\LRfootnote{See
    \vref{fn:zayd-is-son}.}
\end{arab}

\begin{arab}[trans]
  \uc{z}ayduN\arbnull{ibnu}\LRfootnote{A footnote which
    interferes with the contextual analysis.}
  \arbnull{zayduN}ibn-u \uc{`a}mriNU.\LRfootnote{See
    \vref{fn:zayd-is-son}.}
\end{arab}
```

فَلَمَّا رَأَوُا[a] ٱلنَّجْمَ...
قَالَ[b] ٱهْرُبْ فَلَنْ تُقْتَلَ.
زَيْدٌ[c] ٱبْنُ عَمْرٍو.[d]

Zayd[une] *ʾbn*[u] *ʿAmr*[in].[f]

[a] A footnote which interferes with the contextual analysis.
[b] A footnote which interferes with the contextual analysis.
[c] A footnote which interferes with the contextual analysis.
[d] See 28 on page 17.
[e] A footnote which interferes with the contextual analysis.
[f] See 28 on page 17.

4.7 Stretching characters: the *taṭwīl*

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:–

`.hunayn-u bn-u 'is.h--_aq-a` حُنَيْنُ بْنُ إِسحٰقَ *Ḥunayn*^{u} *bn*^{u} *ʾIsḥāq*^{a}

4.8 Digits

4.8.1 Numerical figures

The *Indian numbers*, *ar-raqam*^{u} *ʾl-hindiyy*^{u}, are ten in number, and they are compounded in exactly the same way as our numerals:—

`1874` ١٨٧٤, `123-456,789` ١٢٣-٤٥٦,٧٨٩, `fI sanaT-i 1024` فِي سَنَةِ ١٠٢٤

`\SetArbNumbers` `\SetArbNumbers{Indian|Arabic}` `Default: Indian`

New feature v1.21 As described above, arabluatex prints Indian numbers by default. `\SetArbNumbers{Arabic}` can be used at any point of the document to have Arabic numbers printed. Furthermore, `\SetArbNumbers{Arabic}` gives control over the way numbers are to be printed by means of the `anum` font feature, like so:—

```
\usepackage{arabluatex}
\SetArbNumbers{Arabic}
% use '+anum' for Arabic numbers or '-anum' for Indian numbers:
\newfontfamily\arabicfont{Amiri}[Script=Arabic, RawFeature={+anum}]
```

`\SetArbNumbers` must be used *outside* Arabic environments. Once used, this commands operates on subsequent Arabic environments.

4.8.2 The *abjad*

The numbers may also be expressed with letters from right to left arranged in accordance with the order of the Hebrew and Aramaic alphabets (see Wright, *A Grammar of the Arabic Language*, i. 28 B–C). The *ʾabǧad* numbers are usually distinguished from the surrounding words by a stroke placed over them.

`\abjad` *New feature* v.1.1 *ʾabǧad* numbers are inserted with the `\abjad{`⟨*number*⟩`}` command in any of the `voc`, `fullvoc` and `novoc` modes, where ⟨*number*⟩ may be any number between 1 and 1999, like so:—

`\abjad{45} kitAbu-hu fI 'l-`AdAt-i مة كِتَابُهُ فِي العَادَاتِ *45 kitābu-hu fi ʾl-ʿādāt*^{i}.

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

Rem. *b* `\abjad` may also be found outside Arabic environments. In that case, arabluatex does not print the stroke as a distinctive mark over the number for it is not surrounded by other Arabic words. In case one nonetheless wishes to print the stroke, he can either use the `\aoline*` command that is described below in sect. 4.10.1 on page 25 or insert the *ʾabǧad* number in `\arb[novoc]{}`:—

`The \arb[trans]{'abjad} number for 1874 is \abjad{1874}` The *ʾabǧad* number for 1874 is غضعد.

`The \arb[trans]{'abjad} number for 1874 is \aoline*{\abjad{1874}}` The *ʾabǧad* number for 1874 is غضعد.

`The \arb[trans]{'abjad} number for 1874 is \arb[novoc]{\abjad{1874}}` The *ʾabǧad* number for 1874 is غضعد.

New feature v1.12 `\abjad` may also be used to convert values of counters into *ʾabǧad* numbers, like so:—

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

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

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-ḥurūfᵘ ʾl-muhmalatᵘ*, are sometimes further distinguished from the pointed by various contrivances, as explained in Wright.[36] One may find these letters written in a smaller size below the line, or with a dot or another mark below. As representing all the possible contrivances leads to much complexity and also needs to be agreed among scholars, new ways of encoding them will be proposed and gradually included as arabluatex will mature.

For the time being, the following is included:—

Letter	Transliteration[37]			ArabTEX notation
	`dmg+`	`loc`	`arabica`	
ب	*ḇ*	*b*	*b*	`.b`
د	*ḏ*	*d*	*d*	`^d`
ف	*f̣*	*f*	*f*	`.f`
ق	*q̣*	*q*	*q*	`.q`
ک	*k*	*k*	*k*	`.k`
ں	*ṉ*	*n*	*n*	`.n`
ه	*ah*	*ah*	*a*	`H`[38]
ى	*y*	*y*	*y*	`.y`[39]
﴾	*(*	*(*	*(*	`((`
﴿	*)*	*)*	*)*	`))`

Table 6: Additional Arabic codings

`'afAman.tUs` Gal.(M) `.fmn.n.ts` (sic) Gal.(E1), أفامنطوس Gal.(M) ڤنطس (sic) Gal.(E1), *ʾafāmanṭūs* Gal.(M) *f̣mnṉṭs* (sic) Gal.(E1).

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.

[36] Wright, see n. 7, i. 4 B–C.

[37] See below sect. 8 on page 37.

[38] 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`, `madInaHaN`, اَلْمَدِينَهَ, مَدِينَهً.

[39] See above Rem. *b* on page 9.

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

\abjad{45}: kitAbu-hu \aemph{fI 'l-`AdAt-i} مَة: كِتَابُهُ فِي العَادَاتِ *45: kitābu-hu fi 'l-ʿĀdāt[i]*.

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

New feature v1.9.2 \aemph*

Rem. *b* \aemph* is also provided should one wish to always have the horizontal stroke printed over the emphasized words, like so: \abjad{45}: kitAbu-hu \aemph*{fI 'l-`AdAt-i} مَة: كِتَابُهُ فِي العَادَاتِ *45: kitābu-hu fi 'l-ʿĀdāt[i]*.

4.10.1 Underlining words or numbers

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

(a) \aoline, which is equivalent to \aemph* described above.

New feature v1.19 (b) \aoline*, which is the same as \aoline, but better suited for *ʾabǧad* numbers.[40]

(c) \auline, which can be used to underline Arabic words.

5 Arabic poetry

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

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

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

The description of the optional arguments follows:—

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

width width=⟨*length*⟩ Default: 0.3\linewidth

The default width of each hemistich that the verse consists of. It may be expressed in any accepted unit of measurement, such as 4cm or 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.

gutter gutter=⟨*width*⟩ 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.

metre metre=⟨*name*⟩ Default: none

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

delim delim=true|false Default: false

[40] See the example provided above sect. 4.8.2 on page 23.

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

`\SetHemistichDelim`

`utf` `utf=true|false` Default: `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.

`color` `color=`⟨*color name*⟩ Default: `not set`

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

`export` `export=true|false` Default: `false`

New feature v.1.13 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.

`\bayt` Inside the `arabverse` environment, each line is typeset by the `\bayt` command which takes two mandatory arguments and may accept one optional argument.[41] Additionally, every `\bayt` command *must* be followed with `\\` like so:—

```
\bayt{⟨ṣadr⟩}[⟨tadwīr⟩] {⟨ʿaǧuz⟩}\\
```

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

(a) If the two hemistichs be connected with one another by a prominent horizontal flexible stroke, the *taṭwīl* should be used, like so: `[--]` (see sect. 4.7 on page 23). Of course, the ending word of the *ṣadr* and the word at the commencement of the *ʿaǧuz* must have the *taṭwīl* too so that the proper shapes of the letters be selected. Consider for example the following:—

```
\begin{arabverse}[mode=fullvoc, width=.3\linewidth]
  \bayt{lA 'ar_A man `ahidtu fI-hA fa-'abkI 'l---}[--]{---yawma
    dalhaN wa-mA yaruddu 'l-bukA'u}\\
\end{arabverse}
```

لَا أَرَى مَنْ عَهِدْتُ فِيهَا فَأَبْكِي ٱلْـــــيَوْمَ دَلْهًا وَمَا يَرُدُّ ٱلْبُكَآءُ

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 *ʾalifᵘ ʾl-waṣlⁱ*,[42] while the following two select the figure of the letter *lām* connected with a following letter. In the *ʿaǧuz*, the last two hyphens select the letter *yāʾ* 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:—

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

[42] See sect. 4.2 on page 16.

```
\begin{arabverse}[mode=trans, width=.4\linewidth]
  \bayt{lA 'ar_A man `ahidtu fI-hA fa-'abkI 'l---}[--]{---yawma
    dalhaN wa-mA yaruddu 'l-bukA'u}\\
\end{arabverse}
```

lā ʾarā man ʿahidtu fī-hā fa-ʾabki ʾl- -yawma dalh^{an} wa-mā yaruddu ʾl-bukāʾu

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

```
\begin{arabverse}[mode=fullvoc, width=.25\linewidth,
  gutter=1cm]
  \bayt{.gayra 'annI qad 'asta`Inu `al_A 'l-ha--}[--mmi ]{'i_dA
    _haffa bi-'l-_tawiyyi 'l-na^gA'u}\\
  \bayt{bi-zaf--UfiN ka-'anna-hA hiq|--laTuN}[ 'ummu ]{ri'AliN
    dawwiyyaTuN saqfA'u}\\
\end{arabverse}
```

غَيْرَ أَنِّي قَدْ أَسْتَعِينُ عَلَى ٱلْهَـمِّ إِذَا خَفَّ بِٱلثَّوِيِّ ٱلنَّجَآءُ
يَزَفُوفٍ كَأَنَّهَا هِقْلَةٌ أُمُّ رِئَالٍ دَوِّيَّةٌ سَقْفَآءُ

In the first line, the word اَلْهَمِّ should be split into اَلْهَمْـمِ as the first part of it belongs to the *ṣadr* and the second to the *ʿağuz*. 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 أُمُّ should be split into أُمْـمُ, so that the only way to avoid splitting it into two parts is to write it all inside the *tadwīr*. In that case, as the word is to be placed in the middle, it has been surrounded by spaces.

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

`\StretchBayt` `\StretchBayt[true|false]` `Default: true`

New feature v1.20 `\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[43]) 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.[44]

`\bayt+` *New feature* v1.21

Critical Notes If the ekdosis package be loaded,[45] the `\bayt` command also accepts a `+` 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.[46]

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 ʾl-Qaysi's *Muʿallaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the *ʾidġām* that is fit to this declamatory poetry:—[47]

```
\begin{arab}[fullvoc]
  qAla imru'u 'l-\uc{q}aysi fI mu`allaqati-hi:
\end{arab}

\begin{arabverse}[mode=fullvoc, metre={(al-.darbu 'l-_tAnI mina
    'l-`arU.di 'l-'_Ul_A mina 'l-.tawIli)}]
  \SetArbDflt*
  \begin{linenumbers*}
    \bayt{qifA nabki min _dikr_A .habIbiN wa-manzili}{bi-siq.ti
      'l-liw_A bayna \uc{'l-d}a_hUli fa-\uc{.h}awmali}\\
    \bayt{fa-\uc{t}U.di.ha fa-'l-\uc{m}iqrATi lam ya`fu
      rasmu-hA}{limA nasa^gat-hA min ^ganUbiN wa-^sam'ali}\\
    \bayt{tar_A ba`ara 'l-'ar'Ami fI `ara.sAti-hA}{wa-qI`Ani-hA
      ka-'anna-hu .habbu fulfuli}\\
    \bayt{ka-'annI .gadATa 'l-bayni yawma ta.hammalUA}{lad_A
      samurAti 'l-.hayyi nAqifu .han.zali}\\
    \bayt{wuqUfaN bi-hA .sa.hbI `alayya ma.tiyya-hum}{yaqUlUna
      lA tahlik 'asaN_A wa-ta^gammali}\\
    \bayt{wa-'inna ^sifA'I `abraTuN muharAqaTuN}{fa-hal `inda
      rasmiN dArisiN min mu`awwali}\\
  \end{linenumbers*}
\end{arabverse}
```

`\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.

[45] Robert Alessi, *The Ekdosis package: Typesetting TEI-xml compliant Critical Editions* (version 1.4) (Nov. 21, 2021), http://www.ekdosis.org.

[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 *Muʿallaqāt* that I know of feature the *ʾidġām* in the Arabic text, although it is often strongly marked in declamation.

قَالَ ٱمْرُؤُ ٱلْقَيْسِ فِي مُعَلَّقَتِهِ:

قِفَا نَبْكِ مِنْ ذِكْرَى حَبِيبٍ وَّمَنْزِلِ بِسِقْطِ ٱللِّوَى بَيْنَ ٱلدَّخُولِ فَحَوْمَلِ
فَتُوضِحَ فَٱلْمِقْرَاةِ لَمْ يَعْفُ رَسْمُهَا لِمَا نَسَجَتْهَا مِنْ جَنُوبٍ وَّشَمْأَلِ
تَرَى بَعَرَ ٱلْأَرْآمِ فِي عَرَصَاتِهَا وَقِيعَانِهَا كَأَنَّهُ حَبُّ فُلْفُلِ
كَأَنِّي غَدَاةَ ٱلْبَيْنِ يَوْمَ تَحَمَّلُوا لَدَى سَمُرَاتِ ٱلْحَيِّ نَاقِفُ حَنْظَلِ
وُقُوفًا بِهَا صَحْبِي عَلَيَّ مَطِيَّهُمْ يَقُولُونَ لَا تَهْلِكْ أَسًى وَّتَجَمَّلِ
وَإِنَّ شِفَآئِي عَبْرَةٌ مُّهَرَاقَةٌ فَهَلْ عِنْدَ رَسْمٍ دَارِسٍ مِّن مُّعَوَّلِ

(اَلضَّرْبُ ٱلثَّانِي مِنَ ٱلْعَرُوضِ ٱلْأُولَى مِنَ ٱلطَّوِيلِ)

qāla ’mruʾu ’l-Qaysi fī muʿallaqati-hi:

qifā nabki min ḏikrā ḥabīb^{iw} wa-manzili bi-siqṭi ’l-liwā bayna ’d-Daḫūli fa-Ḥawmali
fa-Tūḍiḥa fa-’l-Miqrāti lam yaʿfu rasmu-hā limā nasağat-hā min ǧanūb^{iw} wa-šamʾali
tarā baʿara ’l-ʾarʾāmi fī ʿaraṣāti-hā wa-qīʿāni-hā ka-ʾanna-hu ḥabbu fulfuli
ka-ʾannī ġadāta ’l-bayni yawma taḥammalū ladā samurāti ’l-ḥayyi nāqifu ḥanẓali
wuqūf^{an} bi-hā ṣaḥbī ʿalayya maṭiyya-hum yaqūlūna lā tahlik ʾasā^{w} wa-taǧammali
wa-ʾinna šifāʾī ʿabrat^{um} muharāqat^{un} fa-hal ʿinda rasm^{in} dāris^{im} mim muʿawwali

(aḍ-ḍarbu ’l-ṯānī mina ’l-ʿarūḍi ’l-ʾūlā mina ’ṭ-ṭawīli)

`\StretchBayt[false]`:—
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.

قَالَ ٱمْرُؤُ ٱلْقَيْسِ فِي مُعَلَّقَتِهِ:

قِفَا نَبْكِ مِنْ ذِكْرَى حَبِيبٍ وَّمَنْزِلِ بِسِقْطِ ٱللِّوَى بَيْنَ ٱلدَّخُولِ فَحَوْمَلِ
فَتُوضِحَ فَٱلْمِقْرَاةِ لَمْ يَعْفُ رَسْمُهَا لِمَا نَسَجَتْهَا مِنْ جَنُوبٍ وَّشَمْأَلِ
تَرَى بَعَرَ ٱلْأَرْآمِ فِي عَرَصَاتِهَا وَقِيعَانِهَا كَأَنَّهُ حَبُّ فُلْفُلِ
كَأَنِّي غَدَاةَ ٱلْبَيْنِ يَوْمَ تَحَمَّلُوا لَدَى سَمُرَاتِ ٱلْحَيِّ نَاقِفُ حَنْظَلِ
وُقُوفًا بِهَا صَحْبِي عَلَيَّ مَطِيَّهُمْ يَقُولُونَ لَا تَهْلِكْ أَسًى وَّتَجَمَّلِ
وَإِنَّ شِفَآئِي عَبْرَةٌ مُّهَرَاقَةٌ فَهَلْ عِنْدَ رَسْمٍ دَارِسٍ مِّن مُّعَوَّلِ

(اَلضَّرْبُ ٱلثَّانِي مِنَ ٱلْعَرُوضِ ٱلْأُولَى مِنَ ٱلطَّوِيلِ)

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

qāla ’mruʾu ’l-Qaysi fī muʿallaqati-hi:

qifā nabki min ḏikrā ḥabīb^{iw} wa-manzili bi-siqṭi ’l-liwā bayna ’d-Daḫūli fa-Ḥawmali
fa-Tūḍiḥa fa-’l-Miqrāti lam yaʿfu rasmu-hā limā nasağat-hā min ǧanūb^{iw} wa-šamʾali
tarā baʿara ’l-ʾarʾāmi fī ʿaraṣāti-hā wa-qīʿāni-hā ka-ʾanna-hu ḥabbu fulfuli
ka-ʾannī ġadāta ’l-bayni yawma taḥammalū ladā samurāti ’l-ḥayyi nāqifu ḥanẓali
wuqūf^{an} bi-hā ṣaḥbī ʿalayya maṭiyya-hum yaqūlūna lā tahlik ʾasā^{w} wa-taǧammali
wa-ʾinna šifāʾī ʿabrat^{um} muharāqat^{un} fa-hal ʿinda rasm^{in} dāris^{im} mim muʿawwali

(aḍ-ḍarbu ’l-ṯānī mina ’l-ʿarūḍi ’l-ʾūlā mina ’ṭ-ṭawīli)

6 Special applications

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

`Bu Ba Bi BuN BaN BiN` ـٍـَـُ *u a i un an in*, `BBu BBa BBi` ـّـُـّـَـّـِـّ *u a i*, `B--aN` ـًـ *-an*, `B"` ـْ.

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. arabluatex 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: `{}`

`\abraces` 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{arab}
  \abraces{wa-qAla} 'inna 'abI kAna mina 'l-muqAtilaTi
  wa-kAna--<--t> 'ummI min `u.zamA'i buyUti 'l-zamAzimaTi.
\end{arab}
```

{وَقَالَ} إِنَّ أَبِي كَانَ مِنَ المُقَاتِلَةِ وَكَانَت> أُمِّي مِن عُظَمَآءِ بُيُوتِ الزَّمَازِمَةِ.

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 *Qurʾān* or in classical poetry.

`\arbmark` *New feature* v1.11 `\arbmark[`⟨*rl|lr*⟩`]{`⟨*shorthand*⟩`}` can be used to insert such characters either in Unicode or in romanized Arabic environments. It takes as argument a shorthand defined beforehand in a default list which consists of the following at the time of writing:—

Codepoint	Shorthand	Glyph	Transliteration
FDFD	bismillah	بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ	*bi-ʾsmi ʾLlāhi ʾr-raḥmāni ʾr-raḥīmi*
FDF5	salam	صلعم	*ṣallạ ʾLlāhu ʿalay-hi wa-sallama*
FDFA	slm	ﷺ	*ṣallạ ʾLlāhu ʿalay-hi wa-sallama*
FDFB	jalla	ﷻ	*ǧalla ǧalāla-hu*

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 tranliterated equivalent.

\newarbmark
New feature v1.11

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

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

As regards the right-to-left codepoint, it may be either typed in Unicode or selected as Unicode codepoint. To that end, the 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 ArabTeX 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_tUA
  'l-nisA'a karhaN\arbmark{sly} wa-lA ta`.dulU-hunna li-ta_dhabUA
  bi-ba`.di mA 'a'taytumU-hunna 'illA 'an ya'tIna bi-fA.hi^saTiN
  mubayyinaTiN\arbmark{jim} wa-`A^sirU-hunna
  bi-'l-ma`rUfi\arbmark{jim} fa-'in karihtumU-hunna fa-`as_A_a
  'an takrahUA ^say'aN wa-ya^g`ala 'l-l_ahu fI-hi _hayraN
  ka_tIraN ((19))
\end{arab}
```

سُورَةُ النِّسَآءِ، ١٩:

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُواْ لَا يَحِلُّ لَكُمۡ أَن تَرِثُواْ ٱلنِّسَآءَ كَرۡهٗاۖ وَلَا تَعۡضُلُوهُنَّ لِتَذۡهَبُواْ بِبَعۡضِ مَآ ءَاتَيۡتُمُوهُنَّ إِلَّآ أَن يَأۡتِينَ بِفَٰحِشَةٖ مُّبَيِّنَةٖۚ وَعَاشِرُوهُنَّ بِٱلۡمَعۡرُوفِۚ فَإِن كَرِهۡتُمُوهُنَّ فَعَسَىٰٓ أَن تَكۡرَهُواْ شَيۡـٔٗا وَيَجۡعَلَ ٱللَّهُ فِيهِ خَيۡرٗا كَثِيرٗا ﴿١٩﴾

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 ArabTEX 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 ʾIsḥāq (إِسْحٰق):[48]—

```
\begin{arab}[fullvoc]
  huwa 'abU zaydiN .hunaynu bnu 'is&\underLine{&.h_a}qa
  'l-`a\underLine{bA}diyyu bi-fat.hi 'l-`ayni wa-ta_hfIfi 'l-bA'i.

  huwa 'abU zaydiN .hunaynu bnu 'is&\highLight{&.h_a}qa
  'l-`a\highLight{bA}diyyu bi-fat.hi 'l-`ayni wa-ta_hfIfi 'l-bA'i.
\end{arab}
```

هُوَ أَبُو زَيْدٍ حُنَيْنُ بْنُ إِسْحٰقَ ٱلْعِبَادِيُّ بِفَتْحِ ٱلْعَيْنِ وَتَخْفِيفِ ٱلْبَآءِ.

هُوَ أَبُو زَيْدٍ حُنَيْنُ بْنُ إِسْحٰقَ ٱلْعَبَادِيُّ بِفَتْحِ ٱلْعَيْنِ وَتَخْفِيفِ ٱلْبَآءِ.

6.1 The Qur’ān

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

\ayah
New feature v1.15

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

\ayah{123} ۝١٢٣ *(123)*.

An example follows:—

```
\SetArbDflt*
\newarbmark{alifsp}{^^^^0627}{\arb[trans]{'alif} }
\newarbmark{lamsp}{^^^^0644^^^^0653}{\arb[trans]{lAm} }
\newarbmark{mim}{^^^^0645^^^^0653}{\arb[trans]{mIm}}
\begin{arab}[fullvoc]
  min ((sUraTi \uc{'l-b}aqaraTi)):
\end{arab}
\begin{arab}[fullvoc]
  \arbmark{alifsp}\arbmark{lamsp}\arbmark{mim}~\ayah{1}
  _d_alika 'l-kit_abu lA rayba fI-hi hudaN_A
  li-l-muttaqIna~\ayah{2} 'lla_dIna yu'minUna bi-'l-.gaybi
  wa-yuqImUna 'l-.sal_aUTa wa-mimmA razaqn_a-hum
  yunfiqUna~\ayah{3}
\end{arab}
```

[48] \underLine and \highLight are taken from the lua-ul package which is loaded by arabluatex. See Marcel Krüger, *The Lua-ul package: Underlining for LuaLaTeX* (version 0.0.1) (Mar. 12, 2020), http://www.ctan.org/pkg/lua-ul.

مِنْ ﴿سُورَةِ ٱلْبَقَرَةِ﴾:
الٓمٓ ﴿١﴾ ذَٰلِكَ ٱلْكِتَٰبُ لَا رَيْبَ فِيهِ هُدًى لِّلْمُتَّقِينَ ﴿٢﴾ ٱلَّذِينَ يُؤْمِنُونَ بِٱلْغَيْبِ وَيُقِيمُونَ ٱلصَّلَوٰةَ وَمِمَّا رَزَقْنَٰهُمْ يُنفِقُونَ ﴿٣﴾

min (sūrati 'l-Baqarati):
'alif lām mīm (1) ḏālika 'l-kitābu lā rayba fī-hi huḍa[l] li-l-muttaqīna (2) 'lladīna yu'minūna bi-'l-ġaybi wa-yuqīmūna 's-ṣalāta wa-mimmā razaqnā-hum yunfiqūna (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:[49]—

(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 takes the text to be colored into ⟨*color*⟩ as an argument:—

`\arbcolor[⟨color⟩]{⟨Arabic text⟩}`

```
\begin{arab}
  \arbcolor[red]{al-bAbu 'l-_hAmisu} fI .tabaqAti 'l-'a.tibbA'i
  'lla_dIna kAnUA mun_du zamAni \uc{^gAlInUsa} wa-qarIbaN
  min-hu. \arbcolor[red]{\uc{^gAlInUsu}}: wa-l-na.da` 'awwalaN
  kalAmaN kulliyyaN fI 'a_hbAri \uc{^gAlInUsa} wa-mA kAna
  `alay-hi...
\end{arab}
\begin{arab}[trans]
  \arbcolor[red]{al-bAbu 'l-_hAmisu} fI .tabaqAti 'l-'a.tibbA'i
  'lla_dIna kAnUA mun_du zamAni \uc{^gAlInUsa} wa-qarIbaN
  min-hu. \arbcolor[red]{\uc{^gAlInUsu}}: wa-l-na.da` 'awwalaN
  kalAmaN kulliyyaN fI 'a_hbAri \uc{^gAlInUsa} wa-mA kAna
  `alay-hi...
\end{arab}
```

اَلبَابُ الخَامِسُ فِي طَبَقَاتِ الأَطِبَّاءِ الَّذِينَ كَانُوا مُنذُ زَمَانِ جَالِينُوسَ وَقَرِيبًا مِنهُ. جَالِينُوسُ: وَلنَضَع أَوَّلًا كَلَامًا كُلِّيًّا فِي أَخبَارِ جَالِينُوسَ وَمَا كَانَ عَلَيهِ...

[49] 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.

al-bābu ’l-ḫāmisu fī ṭabaqāti ’l-ʾaṭibbāʾi ’lladīna kānū munḏu zamāni Ǧālīnūsa wa-qarīb^{an} *min-hu. Ǧālīnūsu: wa-l-naḍaʿ ʾawwal*^{an} *kalām*^{an} *kulliyy*^{an} *fī ʾaḥbāri Ǧālīnūsa wa-mā kāna ʿalay-hi...*

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

‘voc’ mode:

```
i^stara\arbcolor[brown]{y}tu-hu bi-_tama\arbcolor[red]{niN}
'a`\arbcolor[blue]{^ga}ba-ka
```

اِشْتَرَيْتُهُ بِثَمَنٍ أَعْجَبَكَ *ištaraytu-hu bi-ṯaman*^{in} *ʾaʿǧaba-ka*.

‘fullvoc’ mode:

```
i^stara\arbcolor[brown]{y}tu-hu bi-_tama\arbcolor[red]{niN}
'a`\arbcolor[blue]{^ga}ba-ka
```

اِشْتَرَيْتُهُ بِثَمَنٍ أَعْجَبَكَ *ištaraytu-hu bi-ṯaman*^{in} *ʾaʿǧaba-ka*.

7.1 Tricks of the trade

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

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

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

Diacritic	Transliteration[50]			ArabTEX notation
	`dmg+`	`loc`	`arabica`	
◌َ	*a*	*a*	*a*	`.a`
◌ُ	*u*	*u*	*u*	`.u`
◌ِ	*i*	*i*	*i*	`.i`
◌ْ				`o`

Table 8: ArabTEX diacritics for `\arbcolor`

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

‘voc’ mode:

```
i^staraytu-hu bi-_taman\arbcolor[red]{iN} 'a`^g\arbcolor[red]{.a}ba-
ka
```

اِشْتَرَيْتُهُ بِثَمَنٍ أَعْجَبَكَ *ištaraytu-hu bi-ṯaman*^{in} *ʾaʿǧaba-ka*.

[50] See below sect. 8 on page 37.

`i^staraytu-hu bi-_tama\arbcolor[red]{n}iN 'a`\arbcolor[red]{^g}.aba-ka` اِشتَرَيتُهُ بِثَمَنٍ أَعجَبَكَ *ištaraytu-hu bi-ṯaman*^{in} *ʾaʿğaba-ka*.

`i^staraytu-hu bi-_tama\arbcolor[red]{n}\arbcolor[blue]{iN} 'a`\arbcolor[red]{^g}\arbcolor[blue]{.a}ba-ka` اِشتَرَيتُهُ بِثَمَنٍ أَعجَبَكَ *ištaraytu-hu bi-ṯaman*^{in} *ʾaʿğaba-ka*.

'fullvoc' mode:

`i^staray"\arbcolor[red]{o}tu-hu bi-_taman"\arbcolor[red]{iN} 'a``g"\arbcolor[red]{.a}ba-ka` اِشْتَرَيتُهُ بِثَمَنٍ أَعْجَبَكَ *ištaraytu-hu bi-ṯaman*^{in} *ʾaʿğaba-ka*.

`i^stara\arbcolor[red]{y"}otu-hu bi-_tama\arbcolor[red]{n"}iN 'a`\arbcolor[red]{^g"}.aba-ka` اِشْتَرَيتُهُ بِثَمَنٍ أَعْجَبَكَ *ištaraytu-hu bi-ṯaman*^{in} *ʾaʿğaba-ka*.

`i^stara\arbcolor[red]{y"}\arbcolor[blue]{o}tu-hu bi-_tama\arbcolor[red]{n"}\arbcolor[blue]{iN} 'a`\arbcolor[red] {^g"}\arb color[blue]{.a}ba-ka` اِشْتَرَيتُهُ بِثَمَنٍ أَعْجَبَكَ *ištaraytu-hu bi-ṯaman*^{in} *ʾaʿğaba-ka*.

As can be seen, `fullvoc` required the letters `y`, `n` and `^g` before `\arbcolor` to be 'quoted'. Otherwise, unwanted *sukūns* would have been generated because of the absence of a vowel after those consonants.

tanwīn `\arbnull` must be used with *fatḥatān* (ً) so as to put back on contextual analysis rules:—

`mu`allim\arbcolor[red]{\arbnull{m}aN}` مُعَلِّمًا *muʿallim*^{an},

`istisqA'\arbcolor[red]{\arbnull{A'}aN}` اِستِسقَآءً *istisqāʾ*^{an},

`^say'\arbcolor[red]{\arbnull{ay'}aN}` شَيئًا *šayʾ*^{an},

`^gAmi`aT|\arbcolor[red]{\arbnull{T}aN}` جَامِعَةً *ğāmiʿat*^{an}.

Rem. Note that in the last example (*ğāmiʿat*^{an}), the 'pipe' character has been inserted before `\arbcolor`. Otherwise, the `dmg` mode of the transliteration rules would have interpreted the *tāʾ marbūṭah* as *final* (e.g. *h* instead of the expected *t*).[51]

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

```
1 \newarbmark{Y}{^^^^0649}{}
2 \arb{hud\arbcolor[red]{aN\arbnull{_A}}\arbmark{Y}}
3 \arb[trans]{hud\arbcolor[red]{aN\arbnull{_A}}\arbmark{Y}}
```

هُدًى *hudaⁿ*

[51] See also on page 40 "Discarding the *ʾiʿrāb*" for more information.

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

`wa-\arbcolor[red]{\arbnull{wa}i}stisqA'uN` وَٱسْتِسْقَآءٌ *wa-ʾstisqāʾun*[52].

`fI "al".i-\arbcolor[red]{\arbnull{'l-}i}btidA'i`
فِي ٱلِٱبْتِدَآءِ *fi ʾli-ʾbtidāʾi*.

`\arbcolor[red]{'a'\arbnull{k}}kulu` آكُلُ *ʾākulu*,
`\arbcolor[red]{'A\arbnull{k}}kiluN` آكِلٌ *ʾākilun*.

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

```
\newarbmark{alifmaddahred}{^^^^0627\arbcolor[red]{^^^^0653}}%
{\arb[trans]{\arbcolor[red]{'a'\arbnull{k}}}}
\arb{\arbmark{alifmaddahred}kulu}
\arb[trans]{\arbmark{alifmaddahred}kulu}.
```

آكُلُ *ʾākulu*.

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 ٱلْمُعَلِّمُونَ, *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:—

```
\newarbmark{lamshaddah}{^^^^0651}{l}
\arb[fullvoc]{al-mu`al"\arbcolor[red]{\arbmark{lamshaddah}}.imUna}
\arb[trans]{al-mu`al"\arbcolor[red]{\arbmark{lamshaddah}}.imUna}.
```

ٱلْمُعَلِّمُونَ *al-muʿallimūna*.

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 `fI 'l-nAsi` فِي ٱلنَّاسِ *fi ʾn-nāsi*, the initial *ʾalifu ʾl-waṣli* can be rendered in red like so: `\arbcolor[red]{\arbnull{al-}a}` ٱ. 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:—

```
\newarbmark{lnn}{^^^^0644^^^^0646^^^^0651}{n-n}
\arb[fullvoc]{fI\arbnull{al-}
  \arbcolor[red]{\arbnull{al-}a}\arbmark{lnn}Asi}
\arb[trans]{fI\arbnull{al-}
  \arbcolor[red]{\arbnull{al-}a}\arbmark{lnn}Asi}.
```

فِي ٱلنَّاسِ *fi ʾn-nāsi*.

[52] To the knowledge of the writer, the *waṣlah* alone is not part of the Arabic Unicode block.

hamzah The 'quoting' technique provides an easy way to determine the carrier of the *hamzah*, as shown in table 5 on page 21:—

```
yatasA\arbnull{'a}\arbcolor[red]{|"'}.alUna يَتَسَآءَلُونَ yatasāʾalūna,
^say\arbcolor[red]{|"'}\arbnull{'}aN شَيْـًٔا šayʾan, ^say\arbcolor[red]
{|"'}iN شَيْءٍ šayʾin, \arbcolor[red]{a"'}.as\arbcolor[red]{y"'}.ilaTuN
أَسْـِٔلَةٌ ʾasʾilatun.
```

8 Transliteration

It may be more appropriate to speak of "romanization" than "transliteration" of Arabic. As seen above in sect. 2.2 on page 5 on pages 5–8, the "transliteration mode" may be selected globally or locally.

This mode transliterates the ArabTEX input into one of the accepted standards. As said above on page 5, three standards are supported at present:

dmg *Deutsche Morgenländische Gesellschaft*, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.[53] `dmg` transliteration convention is selected by default;

dmg+ This is the same as above, with the difference that the hamza is always printed, even in words preceded by the definite article, as in *al-ʾasadu*.

loc *Library of Congress*: this standard is part of a large set of standards for romanization of non-roman scripts adopted by the American Library Association and the Library of Congress;[54]

New feature v1.8 **arabica** *Journal of Arabic and Islamic Studies/Revue d'études arabes et islamiques*: this standard is most widely used by scholars in the field of Arabic studies.[55]

More standards will be included in future releases of arabluatex.

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

```
\usepackage{arabluatex}
\SetTranslitConvention{loc}
```

`\SetTranslitStyle` **Style** Any transliterated Arabic text is printed in italics by default. This also can be changed either globally in the preamble or locally at any point of the document by the `\SetTranslitStyle{`⟨*style*⟩`}` command, where ⟨*style*⟩ may be any font shape selection command, e.g. `\upshape`, `\itshape`, `\slshape`, and so forth.

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

[53] See Brockelmann et al., "Die Transliteration der arabischen Schrift".
[54] See http://www.loc.gov/catdir/cpso/roman.html for the source document concerning Arabic language.
[55] See http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf.

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

\uc **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 lower case in all positions. Moreover, if the inital letter, apart from the article, cannot be uppercased, viz. ʾ or ʿ, the letter next to it will be uppercased:—

\uc{.hunayn-u} bn-u \uc{'is.h_aq-a} حُنَيْنُ بْنُ إِسْحٰقَ *Ḥunayn*^{u} *bn*^{u} *ʾIsḥāq*^{a}, \uc{`u_tm_an-u} عُثْمٰنُ *ʿUṯmān*$^{u}$, .daraba \uc{zayd-u} bn-u \uc{_h_alidiN} \uc{sa`d-a} bn-a \uc{`awf-i} bn-i \uc{`abd-i} \uc{'l-l_ah-i} ضَرَبَ زَيْدُ بْنُ خٰلِدٍ سَعْدَ بْنَ عَوْفِ بْنِ عَبْدِ ٱللّٰهِ *ḍaraba Zayd*^{u} *bn*^{u} *Ḫālid*^{in} *Saʿd*^{a} *bn*^{a} *ʿAwf*^{i} *bn*^{i} *ʿAbd*^{i} *ʾLlāh*^{i}.

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 مُحَمَّدٌ ٱلنَّبِي must be *Muḥammad*^{uni} *ʾn-nabī*, as nouns having the *tanwīn* take a *kasrah* in pronunciation before *ʾalifu ʾl-waṣli*. In that case, encoding مُحَمَّدٌ like so: \uc{mu.hammaduN} is wrong, because the closing brace would prevent arabluatex from detecting the sequence ⟨*-uN*⟩ immediately followed by ⟨*ʾl-*⟩. 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 arabluatex, 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} أَبُو بَكْرٍ مُحَمَّدُ بْنُ زَكَرِيَّآءَ الرَّازِيُّ *ʾAbū Bakr*^{in} *Muḥammad*^{u} *bn*^{u} *Zakariyyāʾ*^{a} *ʾr-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}} *ʾAbū Zayd*^{in} *Ḥunaynu bnu ʾIsḥāqa ʾl-ʿIbādiyyu*.

\prname *New feature* v1.10 **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:[56]—

```
From \textcite[i. 23 C]{Wright}:--- If the name following
\arb[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
```

[56] Just as \uc, \prname is also able to process proper names consisting of several subsequent words.

```
5    man.sUriN}, \enquote{\prname{`ammAr} the (grand)son of
6    \prname{man.sUr}}.
```

From Wright:[a] — If the name following اِبْنٌ be that of the mother or the grandfather, the اٌ is retained; as عِيسَى اٱبْنُ مَرْيَمَ, "Jesus the son of Mary"; عَمَّارُ اٱبْنُ مَنْصُورٍ, "ʿAmmār the (grand)son of Manṣūr".

[a] Wright, see n. 7, i. 23 C.

The following example shows how \prname can be used in conjunction with the nameauth package to have Arabic proper names printed first in full then in partial forms:[57] —

```
1  \begin{nameauth}
2    \< Hunayn & \prname{'abU zayd} & \prname{.hunayn}, \prname{{i}bn
3      'is.h_aq al-`ibAdiyy} & > %
4    \< Razi & \prname{'abU bakr mu.hammad ibn zakariyyA'} &
5    \prname{al-rAziyy} & > %
6  \end{nameauth}
7
8  On first occurrence, proper names are printed as \Hunayn, \Razi.
9  Then as \Hunayn, \Razi.
```

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

\prname*

Rem. arabluatex 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.[58]

8.1 Additional note on dmg convention

New feature v1.3

According to Brockelmann et al.,[59] Arabic *ʾiʿrāb* may be rendered into dmg in three different ways:

(a) In full: *ʿAmrun*;
(b) As superscript text: *ʿAmr*^{un};
(c) Discarded: *ʿAmr*.

\arbup

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

\NoArbUp
\ArbUpDflt

\NoArbUp may be used either in the preamble or at any point of the document in case one wishes to apply rule (a). The default rule (b) can be set back with \ArbUpDflt at any point of the document.

\SetArbUp

Finally, \SetArbUp{⟨*formatting directives*⟩} can be used to customize the way *ʾiʿrāb* is displayed. To take one example, here is how Arabic *ʾiʿrāb* may be rendered as subscript text:—

[57] See the documentation of nameauth for more details: https://ctan.org/pkg/nameauth

[58] See below sect. 12 on page 51 for more details.

[59] Carl Brockelmann et al., "Die Transliteration der arabischen Schrift in ihrer Anwendung auf die Hauptliteratursprachen der islamischen Welt", in *Denkschrift dem 19. internationalen Orientalistenkongreß in Rom vorgelegt von der Transkriptionkommission der Deutschen Morgenländischen Gesellschaft*, in collab. with Ph. S. van Ronkel and Otto Spies (Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockaus, 1935), http://www.naher-osten.uni-muenchen.de/studium_lehre/werkzeugkasten/dmgtransliteration.pdf, 6.

```
\SetArbUp{\textsubscript{#1}}
Arabic |dmg| transliteration for \arb{ra'aytu ^gAmi`aN
  muhaddamaTaN mi'_danatu-hu}: \arb[trans]{ra'aytu
  ^gAmi`aN muhaddamaTaN mi'_danatu-hu.}
```

Arabic dmg transliteration for رَأَيتُ جَامِعًا مُهَدَّمَةً مِئْذَنَتُهُ: *raʾaytu ǧāmiʿan muhaddamatan miʾḏanatu-hu.*

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.

***ʾiʿrāb* boundaries** Every declinable noun (*muʿrab*) may be declined either with or without *tanwīn*, viz. *munṣarifun* or *ġayru munṣarifin*. The former is automatically parsed by arabluatex, whereas the latter has to be delimited with an hyphen, like so:—

munṣarif: `mu`allimuN` مُعَلِّمٌ *muʿallimun*, `kA'inuN` كَائِنٌ *kāʾinun*, `kA'inAtuN` كَائِنَاتٌ *kāʾinātun*, `\uc{`amraNU}` عَمْرُوا *ʿAmran*, `fataN_A` فَتًى *fataⁿ*, `qA.diNI` قَاضٍ *qāḍiⁿ*.

ġayr munṣarif: `al-mu`allim-u` اَلْمُعَلِّمُ *al-muʿallimu*, `kitAb-Ani` كِتَابَانِ *kitābāni*, `ra^sa'-Ani` رَشَآنِ *rašaʾāni*, `sAriq-Una` سَارِقُونَ *sāriqūna*, `qA.d-Una` قَاضُونَ *qāḍūna*, `al-.zulm-Atu` اَلظُّلْمَاتُ *aẓ-ẓulmātu*.

Rem. *a* As the *tanwīn* is passed over in pronunciation when it is followed by the letters ر, ل, م, و, ي (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:—

`kAna .ganiyyaN l_akinna-hu labisa ^gubbaTaN mumazzaqaN 'aydu-hA` كَانَ غَنِيًّا لٰكِنَّهُ لَبِسَ جُبَّةً مُمَزَّقًا أَيْدُهَا *kāna ġaniyyan lākinna-hu labisa ǧubbatan mumazzaqan ʾaydu-hā.*

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:—

`mana`a 'l-nAs-a kAffaT-aN min mu_hA.tabati-hi 'a.had-uN bi-sayyidi-nA` مَنَعَ النَّاسَ كَافَّةً مِن مُخَاطَبَتِهِ أَحَدٌ بِسَيِّدِنَا *manaʿa ʾn-nāsa kāffatan min muḥāṭabati-hi ʾaḥadun bi-sayyidi-nā.*

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

(a) Nouns followed by an adjective in apposition: `madInaT kabIraT` *madīnah kabīrah*, `al-madInaT al-kabIraT` *al-madīnah al-kabīrah*.

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

Rem. 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: `\uc{r}isAlaT fI tartIb qirA'aT| kutub \uc{^g}AlInUs` *Risālah fī tartīb qirāʾat kutub Ǧālīnūs.*

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: فَعِّلَ. In transliteration, the uncertain vowels go between slashes and are separated by commas: `fa`uaila` فَعِّلَ *faʿ/u,a,i/la*.

8.2 Examples

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

***'dmg+'* standard:** *ʾaṭā ṣadīqᵘⁿ ʾilā Ǧuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratⁱⁿ qaṣīratⁱⁿ fa-qāla la-hu: "sawfa ʾuʿīdu-hu ʾilay-ka fi 'l-masāʾⁱ wa-ʾadfaʿu la-ka ʾuǧratᵃⁿ." fa-qāla Ǧuḥā: "ʾanā āsifᵘⁿ ǧiddᵃⁿ ʾannī lā ʾastaṭīʿu ʾan ʾuḥaqqiqa la-ka raġbata-ka fa-'l-ḥimārᵘ laysa huna 'l-yawmᵃ." wa-qabla ʾan yutimma Ǧuḥā kalāma-hu badaʾa 'l-ḥimārᵘ yanhaqu fī ʾiṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: "ʾinnī ʾasmaʿu ḥimāra-ka yā Ǧuḥā yanhaqu." fa-qāla la-hu Ǧuḥā: "ġarībᵘⁿ ʾamru-ka yā ṣadīqī ʾa-tuṣaddiqu 'l-ḥimārᵃ wa-tukaḏḏiba-nī?"*

***'loc'* standard:** *atá ṣadīqun ilá Juḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratin qaṣīratin fa-qāla la-hu: "sawfa u'īdu-hu ilay-ka fī al-masā'i wa-adfa'u la-ka ujratan." fa-qāla Juḥā: "anā āsifun jiddan annī lā astaṭī'u an uḥaqqiqa la-ka raghbata-ka fa-al-ḥimāru laysa hunā al-yawma." wa-qabla an yutimma Juḥā kalāma-hu bada'a al-ḥimāru yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: "innī asma'u ḥimāra-ka yā Juḥā yanhaqu." fa-qāla la-hu Juḥā: "gharībun amru-ka yā ṣadīqī a-tuṣaddiqu al-ḥimāra wa-tukadhdhiba-nī?"*

***'arabica'* standard:** *atā ṣadīqun ilā Ǧuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratin qaṣīratin fa-qāla la-hu: "sawfa uʿīdu-hu ilay-ka fī l-masāʾi wa-adfaʿu la-ka uǧratan." fa-qāla Ǧuḥā: "anā āsifun ǧiddan annī lā astaṭīʿu an uḥaqqiqa la-ka raġbata-ka fa-l-ḥimāru laysa hunā l-yawma." wa-qabla an yutimma Ǧuḥā kalāma-hu badaʾa l-ḥimāru yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: "innī asmaʿu ḥimāra-ka yā Ǧuḥā yanhaqu." fa-qāla la-hu Ǧuḥā: "ġarībun amru-ka yā ṣadīqī a-tuṣaddiqu l-ḥimāra wa-tukaḏḏiba-nī?"*

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:—

```
1 \usepackage{arabluatex}
2 \SetInputScheme{buckwalter}
```

[60] See http://www.qamus.org/transliteration.htm

'base', 'xml' and 'safe' schemes arabluatex can use any of the so-called Buckwalter 'base', 'xml' or 'safe' schemes as they are described in Habash.[61][62] However, the following limitation apply to the 'base' and 'xml' schemes: the braces { and }, which are used to encode أ and ئ, 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.

Letter	Transliteration[63]			Buckwalter notation	
	dmg+	loc	arabica	base/xml	safe
ا	*a*	*a*	*a*	A	A
ب	*b*	*b*	*b*	b	b
ت	*t*	*t*	*t*	t	t
ث	*ṯ*	*th*	*ṯ*	v	v
ج	*ǧ*	*j*	*ǧ*	j	j
ح	*ḥ*	*ḥ*	*ḥ*	H	H
خ	*ḫ*	*kh*	*ḫ*	x	x
د	*d*	*d*	*d*	d	d
ذ	*ḏ*	*dh*	*ḏ*	*	V
ر	*r*	*r*	*r*	r	r
ز	*z*	*z*	*z*	z	z
س	*s*	*s*	*s*	s	s
ش	*š*	*sh*	*š*	$	c
ص	*ṣ*	*ṣ*	*ṣ*	S	S
ض	*ḍ*	*ḍ*	*ḍ*	D	D
ط	*ṭ*	*ṭ*	*ṭ*	T	T
ظ	*ẓ*	*ẓ*	*ẓ*	Z	Z
ع	ʿ	ʿ	ʿ	E	E
غ	*ġ*	*gh*	*ġ*	g	g
ف	*f*	*f*	*f*	f	f
ق	*q*	*q*	*q*	q	q
ك	*k*	*k*	*k*	k	k
ل	*l*	*l*	*l*	l	l
م	*m*	*m*	*m*	m	m
ن	*n*	*n*	*n*	n	n
ه	*h*	*h*	*h*	h	h
و	*w*	*w*	*w*	w	w
ي	*y*	*y*	*y*	y	y
ى	*ā*	*á*	*ā*	Y	Y
ة	*ah*	*ah*	*a*	p	p
ء	ʾ	’	ʾ	'	C
آ	ʾ*ā*	’*ā*	ʾ*ā*	|	M
أ	ʾ	’	ʾ	>	O
ؤ	ʾ	’	ʾ	&	W

Table 9: Buckwalter input scheme

[61] Nizar Y. Habash, *Introduction to Arabic Natural Language Processing* (Synthesis Lectures on Human Language Technologies, 10; Toronto: Morgan & Claypool Publishers, 2010), 25–6.

[62] I am grateful to Graeme Andrews who suggested that the 'safe' scheme be included in arabluatex.

[63] See sect. 8 on page 37.

Letter	Transliteration			Buckwalter notation	
	dmg+	loc	arabica	base/xml	safe
إ	ʾ	’	ʾ	<	I
ئ	ʾ	’	ʾ	]	Q
ـّـ	—	—	—	~	~
ـٱـ	’	’	—	[	L
ـَـ	*a*	*a*	*a*	a	a
ـُـ	*u*	*u*	*u*	u	u
ـِـ	*i*	*i*	*i*	i	i
ـًـ	*an*	*an*	*an*	F	F
ـٌـ	*un*	*un*	*un*	N	N
ـٍـ	*in*	*in*	*in*	K	K
ـٰـ	—	—	—	o	o
ـٰ	*ā*	*ā*	*ā*	`	e
ـ (*taṭwīl*)	—	—	—	_	_

Table 9: Buckwalter input scheme

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` اَلعَالَمُ *al-ʿālam^{u}*, `Al-camsu` اَلشَّمسُ *aš-šams^{u}*, `bi-SinaAEapi Al-T~ib~i`, بِصِنَاعَةِ الطِّبِّ *bi-ṣināʿat^{i} ’ṭ-ṭibb^{i}*.

`wa-Al-l~ehi` وَاللهِ *wa-’l-lāh^{i}*, `Al-Hamdu li-l~ehi` اَلحَمدُ لِلهِ *al-ḥamd^{u} li-llāh^{i}*.

Similary, it is not advisable to use `|` and `[` (‘base’ and ‘`xml`’ schemes) or `M` and `L` (‘safe’ scheme) to encode the *ʾalif^{u} ’l-mamdūdat^{i}* and the *ʾalif^{u} ’l-waṣl^{i}* 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-LntiqaADi` gives إِلَى الاَنتِقَاضِ as expected, but only `<ilY Al-intiqADi` can be transliterated as *ʾilạ ’l-intiqāḍi* with the correct vowel ⟨*i*⟩ in place of the *ʾalif^{u} ’l-waṣl^{i}*.

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 ArabTEX notation, it also accepts Unicode Arabic input. It should be noted that arabluatex does in no way interfere with Unicode Arabic input: none of the `voc`, `fullvoc`, `novoc` or `trans` options will have any effect on plain Unicode Arabic for the time being.

That said, there are two ways of inserting Unicode Arabic:

`\txarb` (a) The `\txarb{`⟨*Unicode Arabic*⟩`}` command for inserting Unicode Arabic text in paragraphs;

`txarab` (*env.*) (b) The `txarab` environment for inserting running paragraphs of Arabic text, like so:—

```
\begin{txarab}
  <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 `\emph{`⟨*text*⟩`}`, `\textbf{`⟨*text*⟩`}` and the like, is assumed to have Arabic text in its mandatory argument:—

```
\abjad{45} kitAbu-hu \emph{fI 'l-\uc{`AdAt-i}}
```
٤٥ مه كِتَابُهُ فِي العَادَاتِ
kitābu-hu fi 'l-ʿĀdāt[i].[64]

```
\arb{\abjad{45} \rlframebox[1in][s]{kitAbu-hu fI 'l-`AdAti}}
```
٤٥ مه كِتَابُهُ فِي العَادَاتِ [65]

The same applies to footnotes:—

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

إِنَّ أَبِي كَانَ مِنَ المُقَاتِلَةِ[a]، وَكَانَت أُمِّي مِن عُظَمَآءِ بُيُوتِ الزَّمَازِمَةِ[b].

[a] المُقَاتِلَة: المُقَاتِلِينَ.
[b] الزَّمَازِمَة: طَائِفَةٌ مِنَ الفُرسِ.

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

[64] This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.
[65] `\rlframefox` has been adapted from `\framebox` for insertions of right-to-left text.

Even in an Arabic environment, this command may be useful.

\LRfootnote
\RLfootnote

`\LRfootnote{`⟨*text*⟩`}` and `\RLfootnote{`⟨*text*⟩`}` typeset left-to-right and right-to-left footnotes respectively in Arabic environments. Unlike `\footnote{`⟨*text*⟩`}`, the arguments of both `\LRfootnote` and `\RLfootnote` are not expected to be Arabic text. For example, `\LRfootnote` can be used to insert English footnotes in running Arabic text:—

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

زَيْدٌ[a] ٱبْنُ عَمْرٍو

[a] "*Zayd* is the son of *ʿAmr*": 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. arabluatex is not responsible for that, but should it happen, it may be necessary to redefine in the preamble the LaTeX macro `\thefootnote` like so:—

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

\FixArbFtnmk

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

\LRmarginpar

The `\LRmarginpar[`⟨*left*⟩`]{`⟨*right*⟩`}` command does for marginal notes the same as `\LRfootnote` does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that `\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
\setLR

`\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:—

```
\setRL
\section*{\arb{barzawayhi li-buzurjumihra bn-i 'l-buxtikAni}}
\setLR
\begin{arab}
  qAla barzawayhi bn-u 'azhar-a, ra's-u 'a.tibbA'-i fAris-a...
\end{arab}
```

بَرزَوَيهِ لِبُزُرجُمهرَ بنِ البُختَكَانِ

قَالَ بَرزَوَيهِ بنُ أَزهَرَ، رَأسُ أَطِبَّآءِ فَارِسَ...

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

11.1 New commands

New feature v1.9 In some particular cases, it may be useful to define new commands to be inserted in Arabic environments. From the general principle laid on page 44, it follows that any command that is found inside an Arabic environment is assumed to have Arabic text in its argument which arabluatex will process as such before passing it on to the command itself for any further processing. As a result of this feature, such a command as:

```
\newcommand{\fvarabic}[1]{\arb[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.

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

```
\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:—

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

ثُمَّ ٱتَّبَعَ ذُو القَرنَينِ إِلَى أُمَّةٍ أُخرَى فِي ﴿مَطْلِعِ ٱلشَّمْسِ﴾ وَلَا بِنَآءَ لَهُم يُؤَمِّنُهُم مِنَ الشَّمسِ.

It must be noted that the arguments, either optional or mandatory, of commands declared with `\MkArbBreak` are not to be processed by arabluatex. Therefore, as in the previous example, any of their argument to be rendered in Arabic must be inserted again *New feature* v1.12 in `\arb`. These commands themselves may have up to two optional and/or mandatory arguments followed by one optional argument, like so:—

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

`\MkArbBreak*` *New feature* v1.12 As said above, `\MkArbBreak` prevents arabluatex from processing the arguments of 'declared' commands as Arabic text. This technique proves sufficient in most cases. However, a 'starred' version of this command—`\MkArbBreak*{`⟨*csv list of commands*⟩`}`—is also provided. It goes a step further, as it directs arabluatex to *close* the current Arabic environment before any of the 'declared' commands, then *resume* it just after.

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

11.2 Environments

New feature v1.5 Environments such as `\begin{quote} ... \end{quote}` may be nested inside the `arab` environment. Up to one optional argument may be passed to each nested environment, like so:—

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

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

```
\setquotestyle{arabic}
\begin{arab}[fullvoc]
  kAna \uc{'abU} \uc{'l-hu_dayli} 'ahd_A 'il_A \uc{muwaysiN}
  dajAjaTaN. wa-kAnat dajAjatu-hu 'llatI 'ahdA-hA dUna mA kAna
  yuttaxa_du li-\uc{muwaysiN}. wa-l_akinna-hu bi-karami-hi
  wa-bi-.husni xuluqi-hi 'a.zhara 'l-ta`ajjuba min simani-hA
  wa-.tIbi la.hmi-hA. wa-kAna <\uc{'abU} \uc{'l-hu_dayli}>
  yu`rafu bi-'l-'imsAki 'l-^sadIdi. fa-qAla: \enquote{wa-kayfa
    ra'ayta yA \uc{'abA} \uc{`imrAna} tilka 'l-dajAjaTa?} qAla:
  \enquote{kAnat `ajabaN mina 'l-`ajabi!}  fa-yaqUlu:
  \begin{quoting}[begintext=\textquotedblright,
    endtext=\textquotedblleft]
    wa-tadrI mA jinsu-hA? wa-tadrI mA sinnu-hA?  fa-'inna
    'l-dajAjaTa 'inna-mA ta.tIbu bi-'l-jinsi wa-'l-sinni.
    wa-tadrI bi-'ayyi ^say'iN kunnA nusamminu-hA? wa-fI 'ayyi
    makAniN kunnA na`lifu-hA?
  \end{quoting}
  fa-lA yazAlu fI h_a_dA wa-'l-'A_haru ya.d.haku .da.hkaN
  na`rifu-hu na.hnu wa-lA ya`rifu-hu \uc{'abU} \uc{'l-hu_dayli}.
\end{arab}
```

كَانَ أَبُو ٱلْهُذَيْلِ أَهْدَى إِلَى مُوَيْسٍ دَجَاجَةً. وَكَانَتْ دَجَاجَتُهُ ٱلَّتِي أَهْدَاهَا دُونَ مَا كَانَ يُتَّخَذُ لِمُوَيْسٍ. وَلٰكِنَّهُ بِكَرَمِهِ وَبِحُسْنِ خُلُقِهِ أَظْهَرَ ٱلتَّعَجُّبَ مِنْ سِمَنِهَا وَطِيبِ لَحْمِهَا. وَكَانَ ‹أَبُو ٱلْهُذَيْلِ› يُعْرَفُ بِٱلْإِمْسَاكِ ٱلشَّدِيدِ. فَقَالَ: ”وَكَيْفَ رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ ٱلدَّجَاجَةَ؟“ قَالَ: ”كَانَتْ عَجَبًا مِنَ ٱلْعَجَبِ!“ فَيَقُولُ:

” وَتَدْرِي مَا جِنْسُهَا؟ وَتَدْرِي مَا سِنُّهَا؟ فَإِنَّ ٱلدَّجَاجَةَ إِنَّمَا تَطِيبُ بِٱلْجِنْسِ وَٱلسِّنِّ. وَتَدْرِي بِأَيِّ شَيْءٍ كُنَّا نُسَمِّنُهَا؟ وَفِي أَيِّ مَكَانٍ كُنَّا نَعْلِفُهَا؟“

فَلَا يَزَالُ فِي هٰذَا وَٱلْآخَرُ يَضْحَكُ ضَحِكًا نَعْرِفُهُ نَحْنُ وَلَا يَعْرِفُهُ أَبُو ٱلْهُذَيْلِ.

11.2.1 Lists

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

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

```
\setRL\paragraph{\arb[novoc]{rumUzi 'l-kitAbi}}\setLR
\begin{arab}[novoc]
  \begin{description}
  \item[b] max.tU.tu 'l-maktabaTi 'l-'ahliyyaTi bi-\uc{bArIs} 2860
    `arabiyyuN.
  \item[s] max.tU.tu 'l-maktabaTi 'l-'ahliyyaTi bi-\uc{bArIs} 2859
    `arabiyyuN.
  \item[m] max.tU.tu majlisi \arb[novoc]{^sUrAY malY} .tahrAna 521.
  \end{description}
\end{arab}
```

رموز الكتاب
ب مخطوط المكتبة الأهليّة بباريس ٢٨٦٠ عربيّ.
س مخطوط المكتبة الأهليّة بباريس ٢٨٥٩ عربيّ.
م مخطوط مجلس شورای ملی طهران ٥٢١.

As a second example, the contents of a treatise may be typeset with the standard list environments, like so:—

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

اَلقَانُونُ فِي الطِّبِّ
- اَلفَنُّ الأَوَّلُ فِي حَدِّ الطِّبِّ وَمَوضُوعَاتِهِ مِنَ الأُمُورِ الطَّبِيعِيَّةِ وَيَشتَمِلُ عَلَى سِتَّةِ تَعَالِيمٍ
  - اَلتَّعلِيمُ الأَوَّلُ [وَهُوَ فَصلَانِ]
    - اَلفَصلُ الأَوَّلُ

As a third example, abjad-numbered lists can be typeset in conjunction with the enumitem package,[67] like so:—

```
% preamble:---
\usepackage{enumitem}
\newlist{enumabjad}{enumerate}{10}
\setlist[enumabjad]{nosep, label={\abjad{\arabic*}}}
\usepackage{multicol}
```

[67] See the documentation of enumitem for more details: https://ctan.org/pkg/enumitem

```
From \textcite[i. 29 B--C]{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.
\RLmulticolcolumns
\begin{multicols}{3}
  \begin{arab}[fullvoc]
    \begin{enumabjad}
    \item fa`ala
    \item fa``ala
    \item fA`ala
    \item 'af`ala
    \item tafa``ala
    \item tafA`ala
    \item infa`ala
    \item ifta`ala
    \item if`alla
    \item istaf`ala
    \item if`Alla
    \item if`aw`ala
    \item if`awwala
    \item if`anlala
    \item if`anl_A
    \end{enumabjad}
  \end{arab}
\end{multicols}
```

From Wright:[a] — 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.

أ فَعَلَ	وَ تَفَاعَلَ	يأَ اِفْعَالَّ
بَ فَعَّلَ	زَ اِنْفَعَلَ	يبَ اِفْعَوْعَلَ
جَ فَاعَلَ	حَ اِفْتَعَلَ	يجَ اِفْعَوَّلَ
دَ أَفْعَلَ	طَ اِفْعَلَّ	يدَ اِفْعَنْلَلَ
هَ تَفَعَّلَ	يَ اِسْتَفْعَلَ	يهَ اِفْعَنْلَى

[a] Wright, see n. 7, i. 29 B–C.

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

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

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

11.3 csquotes

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

```
\usepackage{csquotes}
\DeclareQuoteStyle{arabic}
{\textquotedblright}{\textquotedblleft}
{\textquoteright}{\textquoteleft}
```

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

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

فَقَالَ لَهُ جُحَا: "غَرِيبٌ أَمرُكَ يَا صَدِيقِي أَتُصَدِّقُ الحِمَارَ وَتُكَذِّبَنِي؟"

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

11.4 Two-argument special commands

textcolor The two-argument command `\textcolor{`⟨*color*⟩`}{`⟨*Arabic text*⟩`}` is supported inside `\begin{arab}` ... `\end{arab}`. One simple example follows:[68]—

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

مُهَذَّبُ الدِّينِ عَبدُ الرَّحِيمِ بنُ عَلِيٍّ هُوَ شَيخُنَا الإِمَامُ الصَّدرُ الكَبِيرُ العَالِمُ الفَاضِلُ مُهَذَّبُ الدِّينِ أَبُو مُحَمَّدٍ عَبدُ الرَّحِيمِ بنُ عَلِيِّ بنِ حَامِدٍ وَيُعرَفُ بِالدَّخوَرِ.

[68] arabluatex provides its own `\arbcolor` command which is able to render syllabes or diacritics in colors. See sect. 7 on page 33.

Muḥaḍḍabu ’d-Dīni ʿAbdu ’r-Raḥīmi bnu ʿAliyy[in] *huwa šayḫu-na ’l-ʾimāmu ’ṣ-ṣadru ’l-kabīru ’l-ʿālimu ’l-fāḍilu Muḥaḍḍabu ’d-Dīni ʾAbū Muḥammad*[in] *ʿAbdu ’r-Raḥīmi bnu ʿAliyyi bni Ḥāmid*[in] *wa-yuʿrafu bi-’d-Daḫwari.*

reledmac The two-argument command `\edtext{`⟨*lemma*⟩`}{`⟨*commands*⟩`}` is supported inside `\begin{arab}` ... `\end{arab}`.[69] As an example, one may get arabluatex and reledmac to work together like so:—

```
\beginnumbering
\pstart
\begin{arab}
  wa-ya.sIru ta.hta 'l-jild-i
  \edtext{\arb{.sadId-uN}}{\Afootnote{M: \arb{.sadId-aN} E1}}
\end{arab}
\pend
\endnumbering
```

11.5 quran

arabluatex is compatible with the quran package so that both can be used in conjunction with one another for typesetting the *Qurʾān*. As quran draws the text of the *Qurʾān* from a Unicode encoded database, its commands have to be passed as arguments to the `\txarb` 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 arabluatex takes care of formatting the Arabic: therefore, it is recommended to load the quran package with the `nopar` option, after arabluatex itself has been loaded, like so:—

```
\usepackage{arabluatex}
\usepackage[nopar]{quran}
```

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

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

بِسمِ اللَّهِ الرَّحمـٰنِ الرَّحيمِ ﴿١﴾ الحَمدُ لِلَّهِ رَبِّ العالَمينَ ﴿٢﴾ الرَّحمـٰنِ الرَّحيمِ ﴿٣﴾ مالِكِ يَومِ الدّينِ ﴿٤﴾ إِيّاكَ نَعبُدُ وَإِيّاكَ نَستَعينُ ﴿٥﴾ اهدِنَا الصِّراطَ المُستَقيمَ ﴿٦﴾ صِراطَ الَّذينَ أَنعَمتَ عَلَيهِم غَيرِ المَغضوبِ عَلَيهِم وَلَا الضّالّينَ ﴿٧﴾

12 Exporting Unicode Arabic to an external file

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

[69] `\pstart` and `\pend` are also supported inside the `arab` environment.

or buckwalter strings will have been replaced with Unicode equivalents, either in Arabic script or in any accepted standard of transliteration. Exporting ASCII strings to Unicode while preserving the exact selected global or local options is a fairly complex operation which may require LuaLaTeX to be run several times as will be explained below.

12.1 Commands and environments

export **export global option** First, arabluatex must be loaded with the export global option enabled,[70] like so:—

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

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

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

arabexport (*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:—

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

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

(a) Unicode Arabic text, either in Arabic script or in transliteration, is inserted as argument of \txarb[71] or \txtrans[72] accordingly.

\arbpardir (b) Additionally, Arabic paragraphs may receive \arbpardir, which arabluatex uses to determine the direction of Arabic paragraphs to be set by default, or either \setRL or \setLR depending on what may have been set locally.[73]

\prname* (c) Proper names are inserted as arguments of \prname*.[74]

\ArbOutFile \ArbOutFile* **Appending words or commands to the external file only** \ArbOutFile[⟨*newline*⟩]{⟨*argument*⟩} silently exports its argument to the external file. It may take the string newline as an optional argument, in which case a carriage return is appended to the contents of the argument. \ArbOutFile*[⟨*newline*⟩]{⟨*argument*⟩} does the same as \ArbOutFile, but also inserts its argument into the current .tex source file.

[70] See above on page 5 for more information.
[71] See above sect. 10 on page 43.
[72] \txtrans is used internally by several Lua functions to format transliterated Arabic. Therefore, it is not documented.
[73] See above on page 45.
[74] See above on page 39.

Exporting Arabic poetry Lines of Arabic poetry are exported as described above on page 26 when the `export` option that is specific to the `arabverse` environment is set to `true`. As a result of this particular feature, `arabverse` environments must be left outside `\begin{arabexport}` ... `\end{arabexport}`.

Please note that inside `arabverse` environments `\bayt` is replaced with `\bayt*`.[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:—

```
Package arabluatex Warning: There are still 'arabtex' strings
to be converted. Please open <jobname><suffix>.tex and compile
it one more time.
```

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

12.3 Further processing of Unicode converted files

Unicode files can be further processed by document converters such as John McFarlane's `pandoc`[76]. To take here one simple example, here is how `file_out.tex` can be converted from LuaLaTeX into Open Document format (`.odt`):—

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

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

```
% preamble:
\usepackage{arabluatex} % note that 'export' has been removed
\renewcommand{\txarb}[1]{#1}
\renewcommand{\txtrans}[1]{\emph{#1}}
\renewcommand{\arbup}[1]{\textsuperscript{#1}}
% now that \prname{} has been replaced with \prname*{} it should
% be safe to say:
\renewcommand{\prname}[2]{#2}
% &c
```

13 Future work

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

(a) Short-term:

[75] See above 41 on page 26 for more information.

[76] See http://pandoc.org/

i. `TEI xml` support: arabluatex will interoperate with `TEI xml` through new global and local options that will output Arabic in a `TEI xml` compliant file in addition to the usual PDF output: see on page 4.

(b) Medium-term:

i. More languages: the list of supported languages will eventually be the same as arabtex: see 5 on page 4.

ii. Formulate propositions for extending the ArabTEX notation and the transliteration tables. Include them in arabluatex. See sect. 4.9 on page 24.

14 Implementation

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

```
\RequirePackage{iftex}
```

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

```
\RequireLuaTeX
```

Declare the global options, and define them:

```
\RequirePackage{xkeyval}
\DeclareOptionX{voc}{\def\al@mode{voc}}
\DeclareOptionX{fullvoc}{\def\al@mode{fullvoc}}
\DeclareOptionX{novoc}{\def\al@mode{novoc}}
\DeclareOptionX{trans}{\def\al@mode{trans}}
\define@boolkey{arabluatex.sty}[@pkg@]{export}[true]{%
  \if@pkg@export%
  \AtBeginDocument{\luadirect{arabluatex.openstream()}%
    \MkArbBreak{@al@ob,@al@cb,@al@cb@sp}}
  \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:

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

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

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

`\setLR` The same applies to `\setLR`:

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

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

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

`\SetArbNumbers`

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

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

```
\AtBeginDocument{\ifdef{\RL}%
  {\RenewDocumentCommand{\RL}{m}{\bgroup\textdir TRT\rmfamily#1\egroup}}
  {\NewDocumentCommand{\RL}{m}{\bgroup\textdir TRT#1\rmfamily\egroup}}}
```

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

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

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

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

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

`\aoline` `\aoline` and `\auline` derive from `\newunderlinetype` provided by the lua-ul package
`\aoline*` whereas `\aoline*`, which uses `\overline` in math-mode, is better suited for so-called
`\auline` *ʾabǧad* numbers.

```
\newunderlinetype\@aoverLine{\leaders\vrule height 3ex depth -2.9ex}
\def\aoline{\@ifstar\@aoline\@@aoline}
\def\@aoline#1{\ensuremath{\overline{\mbox{#1}}}}
\def\@@aoline#1{{\@aoverLine#1}}
\newunderlinetype\@aunderLine{\leaders\vrule height -.65ex depth .75ex}
\def\auline#1{{\@aunderLine#1}}
\AtBeginDocument{\ifdef{\aemph}%
  {\RenewDocumentCommand{\aemph}{s m}{%
      \IfBooleanTF{#1}{%
        \luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
          "over"))}}
      {\luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
          "dflt"))}}}}
  {\NewDocumentCommand{\aemph}{s m}{%
      \IfBooleanTF{#1}{%
        \luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
          "over"))}}
      {\luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
          "dflt"))}}}}}
```

`\arbcolor` `\arbcolor[`⟨*color*⟩`]{`⟨*Arabic text*⟩`}` takes the Arabic text to be colored as argument.

```
\NewDocumentCommand{\arbcolor}{o m}{%
  \IfNoValueTF{#1}{#2}{\textcolor{#1}{#2}}}
```

`\SetInputScheme` arabluatex is designed for processing ArabTEX 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'.

```
\def\al@input@scheme{arabtex}
\NewDocumentCommand{\SetInputScheme}{m}{\def\al@input@scheme{#1}}
```

`\SetArbEasy` `\SetArbEasy*` `\SetArbDflt` By default, arabluatex applies complex rules to generate euphonic *tašdīd*, *ʾalif mamdūdah* and *sukūn* depending on the modes which are selected, either `voc`, `fullvoc` or `trans`. Such refinements can be discarded with `\SetArbEasy`, either globally in the preamble or at any point of the document. Note that `\SetArbEasy` keeps the *sukūn* that is generated, while the starred version `\SetArbEasy*` takes it away. Default complex rules can be set back at any point of the document with `\SetArbDflt`.

`\SetArbDflt*` As of v1.6, arabluatex does not applies any more the assimilation rules that are laid on (b) on page 15; a new starred version `\SetArbDflt*` is now available to the user should he wish to apply them.

```
\def\al@arb@rules{dflt}
\NewDocumentCommand{\SetArbEasy}{s}{%
  \IfBooleanTF{#1}
  {\def\al@arb@rules{easynosukun}}
  {\def\al@arb@rules{easy}}}
\NewDocumentCommand{\SetArbDflt}{s}{%
  \IfBooleanTF{#1}
  {\def\al@arb@rules{idgham}}
  {\def\al@arb@rules{dflt}}}
```

`\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.

```
\def\al@trans@font{\rmfamily}%
\NewDocumentCommand{\SetTranslitFont}{m}{\def\al@trans@font{#1}}
```

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

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

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

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

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

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

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

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

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

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

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

```
\let\Uc\uc
```

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

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

```
\NewDocumentCommand{\prname}{s m}{%
  \bgroup\SetTranslitStyle{\relax}%
  \IfBooleanTF{#1}{\txtrans{#2}}{\arb[trans]{\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.

```
\NewDocumentCommand{\txarb}{+m}{%
  \ifvmode\leavevmode\fi%
  \bgroup\textdir TRT\arabicfont#1\egroup}
\NewDocumentCommand{\txtrans}{+m}{%
  \bgroup\textdir TLT\altrfont#1\egroup}
```

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

```
\NewDocumentEnvironment{txarab}{}{%
  \par%
  \booltrue{al@rlmode}%
  \pardir TRT\textdir TRT\arabicfont}{\par}
```

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

```
\NewDocumentEnvironment{txarabtr}{}{%
  \par%
  \pardir TLT\textdir TLT%
  \altrfont}{\par}
```

\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.

```
\NewDocumentCommand{\arb}{O{\al@mode} +m}%
{\edef\@tempa{#1}%
  \ifx\@tempa\al@mode@voc%
  \ifvmode\leavevmode\fi%
  \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
  \luadirect{tex.sprint(arabluatex.processvoc(\luastringN{#2},
    \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
  \else%
  \ifx\@tempa\al@mode@fullvoc%
  \ifvmode\leavevmode\fi%
  \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
  \luadirect{tex.sprint(arabluatex.processfullvoc(\luastringN{#2},
    \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
  \else%
  \ifx\@tempa\al@mode@novoc%
  \ifvmode\leavevmode\fi%
  \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
  \luadirect{tex.sprint(arabluatex.processnovoc(\luastringN{#2},
    \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
  \else%
  \ifx\@tempa\al@mode@trans%
  \bgroup\textdir TLT%
  \luadirect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
    \luastringO{\al@trans@convention},
    \luastringO{\al@arb@rules},
    \luastringO{\al@input@scheme}))}\egroup%
  \else%
```

```
\fi\fi\fi\fi}
```

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

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

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

```
\NewDocumentCommand{\newarbmark}{m m m}{%
  \luadirect{arabluatex.newarbmark(\luastringN{#1}, \luastringN{#2},
    \luastringN{#3})}}
```

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

```
\NewDocumentEnvironment{arab}{!O{\al@mode} +b}%
{\par\edef\@tempa{#1}%
  \ifx\@tempa\al@mode@voc%
  \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}{\par}
```

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

```
\newlength{\al@bayt@width}
\newlength{\al@gutter@width}
```

```
\newlength{\al@verse@twidth}
\setlength{\al@bayt@width}{.3\textwidth}
\setlength{\al@gutter@width}{.15\al@bayt@width}
\define@key[al]{verse}{width}{\setlength{\al@bayt@width}{#1}}
\define@key[al]{verse}{gutter}{\setlength{\al@gutter@width}{#1}}
\define@key[al]{verse}{metre}{\def\al@verse@metre@value{\arb{#1}}}
\define@key[al]{verse}{color}[]{\color{#1}}
\define@boolkey[al]{verse}{utf}[true]{}
\define@boolkey[al]{verse}{delim}[true]{}
\define@boolkey[al]{verse}{export}[true]{}
\define@choicekey[al]{verse}{mode}{fullvoc, voc, novoc,
  trans}{\def\al@mode{#1}}
\presetkeys[al]{verse}{utf=false, delim=false}{}
```

Then follows the environment itself:

```
\NewDocumentEnvironment{arabverse}{!O{}}%
{\bgroup\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%
        \IfSubStr[1]{#1}{utf}%
        {\luadirect{arabluatex.tooutfile(\luastringN{[#1]})}}%
        {\luadirect{arabluatex.tooutfile(\luastringN{[#1, utf]})}}%
        % \fi
      \else
      \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]
  }%
  {\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}
```

\bayt Each verse consists of two hemistichs; therefore the `\bayt` command takes two arguments, the first receives the *ṣadr* and the second the *ʿaǧuz*. That two subsequent hemistichs should be connected with one another is technically named *tadwīr*. In some of these cases, the hemistichs may be connected by a prominent horizontal flexible stroke which is drawn by the `\al@verse@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 `true` or `false`, and may be used either in the preamble or at any point of the document. By default, it is set to `true`.

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

```
\newif\ifal@warp@bayt
\al@warp@bayttrue
\NewDocumentCommand{\StretchBayt}{O{true}}{
  \edef\oarg@true{true}
  \edef\oarg@false{false}
  \edef\@tempa{#1}
  \ifx\@tempa\oarg@true\al@warp@bayttrue
  \else
  \ifx\@tempa\oarg@false\al@warp@baytfalse
  \else
  \PackageError{arabluatex}{\string\StretchBayt\space must be
    either 'true' or 'false'}{}
  \fi
  \fi
}
\NewDocumentCommand{\arb@utf}{m}{%
  \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
\def\al@hemistich@delim{*}
\NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
\def\al@verse@stroke{\leavevmode\xleaders\hbox{\arb{--}}\hfill\kern0pt}
\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
  }{\relax}%
  \IfBooleanTF{#2}{\relax}{\relax}%
  \ifdefined\savenotes\savenotes\else\fi%
  \edef\al@tatweel{--}%
  \ifal@warp@bayt%
    \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
  \else%
    \makebox[\al@bayt@width][s]{\arb@utf{#3}}%
  \fi%
  \IfNoValueTF{#4}{%
```

```
    \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
    \else%
    \hspace{\al@gutter@width}%
    \fi
  }{%
    \edef\@tempa{#4}%
    \ifx\@tempa\al@tatweel%
    \ifx\al@mode\al@mode@trans%
    \hspace{\al@gutter@width}%
    \else%
    \makebox[\al@gutter@width][s]{\al@verse@stroke}%
    \fi%
    \else%
    \ifx\al@mode\al@mode@trans%
    \ifal@warp@bayt%
      \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#4}}%
    \else%
      \makebox[\al@gutter@width][s]{\arb@utf{#4}}%
    \fi%
    \else%
    \makebox[\al@gutter@width][s]{\arb@utf{#4}}%
    \fi\fi}%
  \ifal@warp@bayt%
    \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#5}}%
  \else%
    \makebox[\al@bayt@width][s]{\arb@utf{#5}}%
  \fi%
  \ifdefined\spewnotes\spewnotes\else\fi%
}
```

\arind `\arind{`⟨*root*⟩`}` is a command specialized in the contruction of indexes. As a mandadory 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
}

\def\al@index@mode{\al@mode}
\NewDocumentCommand{\SetIndexMode}{m}{
  \def\al@index@mode{#1}
}

\define@cmdkeys[al]{index}[alind@]{index,root,form,pipe}
\NewDocumentCommand{\arind}{o m}{%
  \IfNoValueTF{#1}{%
    \ifdefined\al@default@index%
      \csname index\endcsname[\al@default@index]{#2}%
    \else%
      \csname index\endcsname{#2}%
    \fi%
  }{%
    \bgroup
```

```
    \setkeys[al]{index}{#1}%
    \def\al@one{%
      \ifdefined\alind@root
        \ifnum\alind@root < 10%
          !\LR{0\alind@root}%
        \else
          !\LR{\alind@root}%
        \fi
      \else
      !\LR{01}%
      \fi}%
    \def\al@two{%
      \ifdefined\alind@form @\arb[\al@index@mode]{\alind@form}\else\fi}%
    \ifdefined\alind@index%
      \ifdefined\alind@pipe
        \csname index\endcsname[\alind@index]{#2\al@one\al@two|\alind@pipe}%
      \else
        \csname index\endcsname[\alind@index]{#2\al@one\al@two}%
      \fi
    \else%
      \ifdefined\al@default@index%
        \ifdefined\alind@pipe
          \csname index\endcsname[\al@default@index]{%
            #2\al@one\al@two|\alind@pipe}%
        \else
          \csname index\endcsname[\al@default@index]{%
            #2\al@one\al@two}%
        \fi
      \else%
        \ifdefined\alind@pipe
          \csname index\endcsname{#2\al@one\al@two|\alind@pipe}%
        \else
          \csname index\endcsname{#2\al@one\al@two}%
        \fi
      \fi%
    \fi%
    \egroup}}
```

`\abjad` `\abjad{`⟨*number*⟩`}` expresses its argument in Arabic letters in accordance with the *ʾabǧad* arrangement of the alphabet. ⟨*number*⟩ must be between 1 and 1999. It is now coded in Lua so that polyglossia is no longer needed. See `arabluatex.lua` for more information.

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

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

\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.

```
\NewDocumentCommand{\arbnull}{m}{\relax}
```

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

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

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

```
\DeclareDocumentCommand{\LRmarginpar}{o m}{%
  \IfNoValueTF{#1}
  {\marginpar{\textdir TLT #2}}
  {\marginpar[\textdir TLT #1]{\textdir TLT #2}}}
```

\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.

```
\DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pardir
  TLT\textdir TLT\footnote{#1}\egroup}
\DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pardir
  TRT\textdir TRT\footnote{#1}\egroup}
```

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

```
\NewDocumentCommand{\FixArbFtnmk}{}{%
  \@ifpackageloaded{scrextend}%
  {\AtBeginDocument{%
      \deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%
  {\RequirePackage{scrextend}
    \AtBeginDocument{%
      \deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}}
```

Exporting Unicode Arabic to external file

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

```
\NewDocumentCommand{\SetArbOutSuffix}{m}{
  \luadirect{arabluatex.utffilesuffix(\luastringN{#1})}}
```

\ArbOutFile `\ArbOutFile[`⟨*newline*⟩`]{`⟨*string*⟩`}` silently exports ⟨*string*⟩ to the external selected file. It may take `newline` as an optional argument in which case a carriage return is appended to `string`.

`\ArbOutFile*` `\ArbOutFile*[`⟨*newline*⟩`]{`⟨*string*⟩`}` does the same as `\ArbOutFile` but also inserts ⟨*string*⟩ in the current `.tex` source file.

```
\NewDocumentCommand{\ArbOutFile}{s O{no} +m}{%
  \if@pkg@export%
  \IfBooleanTF{#1}{%
    #3\luadirect{arabluatex.tooutfile(\luastringN{#3}, "#2")}}{%
    \luadirect{arabluatex.tooutfile(\luastringN{#3}, "#2")}}%
  \else\IfBooleanTF{#1}{#3}{}\fi}
```

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

```
\NewDocumentEnvironment{arabexport}{+b}{%
  \if@pkg@export%
  \par
  #1
  \luadirect{arabluatex.doexport("yes")}
  \luadirect{tex.sprint(arabluatex.arbtoutf(\luastringN{#1}))}
  \luadirect{arabluatex.doexport("no")}
  \else\par#1\fi
  }{\par}
```

`arab@v@export` (*env.*) The `arab@v@export` environment does for `arabverse` the same as `arabexport`. It is used internally by `arabverse`.

```
\NewDocumentEnvironment{arab@v@export}{O{} +b}{%
  \setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
  metre]{#1}
  \if@pkg@export
    \ifal@verse@export
      \ifekd@state
        \begin{ekdverse}[width=\al@verse@twidth]
          #2
        \end{ekdverse}
      \else
        #2
      \fi
      \luadirect{arabluatex.doexport("arabverse")}%
      \luadirect{tex.sprint(arabluatex.arbtoutf(\luastringN{#2}))}%
      \luadirect{arabluatex.doexport("no")}%
    \else
      \ifekd@state
        \begin{ekdverse}[width=\al@verse@twidth]
          #2
        \end{ekdverse}
      \else
        #2
      \fi
    \fi
  \else
    \ifekd@state
      \begin{ekdverse}[width=\al@verse@twidth]
        #2
      \end{ekdverse}
    \else
```

```
      #2
    \fi
  \fi
}{}
```

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

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

Errors and Warnings

```
\newcommand{\al@warning}[1]{\PackageWarning{arabluatex}{#1}}
\newcommand{\al@error}[2]{\PackageError{arabluatex}{#1}{#2}}
\newcommand{\al@wrong@nesting}{\al@error{%
    (RL/LR)\string\footnote\space is not allowed\MessageBreak inside
    \string\RL{} and \string\RL{} commands}{%
    Get rid of the surrounding \string\RL{} or \string\LR{} command.}}
\newcommand{\al@wrong@mark}{\al@warning{%
    Unknown Arabic mark in \string\arbmark{}. Replaced
    with\MessageBreak <??>. Please check your code}}
```

That is it. Say goodbye before leaving.

Patches

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

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

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

```
\def\list#1#2{%
  \ifnum \@listdepth >5\relax
    \@toodeep
  \else
    \global\advance\@listdepth\@ne
  \fi
  \rightmargin\z@
  \listparindent\z@
  \itemindent\z@
  \csname @list\romannumeral\the\@listdepth\endcsname
  \def\@itemlabel{#1}%
  \let\makelabel\@mklab
  \@nmbrlistfalse
  #2\relax
  \@trivlist
  \parskip\parsep
  \parindent\listparindent
  \advance\linewidth -\rightmargin
  \advance\linewidth -\leftmargin
```

patch begins:

```
  \ifbool{al@rlmode}{\advance\@totalleftmargin \rightmargin}%
  {\advance\@totalleftmargin \leftmargin}
```

patch ends.

```
  \parshape \@ne \@totalleftmargin \linewidth
  \ignorespaces}
\def\@item[#1]{%
  \if@noparitem
    \@donoparitem
  \else
    \if@inlabel
      \indent \par
    \fi
    \ifhmode
      \unskip\unskip \par
    \fi
    \if@newlist
      \if@nobreak
        \@nbitem
      \else
        \addpenalty\@beginparpenalty
        \addvspace\@topsep
        \addvspace{-\parskip}%
      \fi
    \else
      \addpenalty\@itempenalty
      \addvspace\itemsep
    \fi
    \global\@inlabeltrue
  \fi
  \everypar{%
    \@minipagefalse
    \global\@newlistfalse
    \if@inlabel
      \global\@inlabelfalse
      {\setbox\z@\lastbox
       \ifvoid\z@
         \kern-\itemindent
       \fi}%
      \box\@labels
      \penalty\z@
    \fi
    \if@nobreak
      \@nobreakfalse
      \clubpenalty \@M
    \else
      \clubpenalty \@clubpenalty
      \everypar{}%
    \fi}%
  \if@noitemarg
    \@noitemargfalse
    \if@nmbrlist
      \refstepcounter\@listctr
    \fi
  \fi
```

patch begins:

```
  \ifbool{al@rlmode}{\sRLbox\@tempboxa{\makelabel{#1}}}{%
  \sbox\@tempboxa{\makelabel{#1}}}%
  \ifbool{al@rlmode}{\global\setbox\@labels\hbox dir TRT}%
  {\global\setbox\@labels\hbox}{%
```

patch ends.

```
    \unhbox\@labels
    \hskip \itemindent
    \hskip -\labelwidth
    \hskip -\labelsep
    \ifdim \wd\@tempboxa >\labelwidth
      \box\@tempboxa
    \else
      \hbox to\labelwidth {\unhbox\@tempboxa}%
    \fi
    \hskip \labelsep}%
  \ignorespaces}
```

This is adapted from Vafa Khalighi's bidi package. Thanks to him.

```
\long\def\sRLbox#1#2{\setbox#1\hbox dir TRT{%
  \color@setgroup#2\color@endgroup}}
```

15 References

Alessi, Robert, *The Ekdosis package: Typesetting TEI-xml compliant Critical Editions* (version 1.4) (Nov. 21, 2021), http://www.ekdosis.org.

Brockelmann, Carl, et al., "Die Transliteration der arabischen Schrift in ihrer Anwendung auf die Hauptliteratursprachen der islamischen Welt", in *Denkschrift dem 19. internationalen Orientalistenkongreß in Rom vorgelegt von der Transkriptionkommission der Deutschen Morgenländischen Gesellschaft*, in collab. with Ph. S. van Ronkel and Otto Spies (Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockaus, 1935), http://www.naher-osten.uni-muenchen.de/studium_lehre/werkzeugkasten/dmgtransliteration.pdf.

Habash, Nizar Y., *Introduction to Arabic Natural Language Processing* (Synthesis Lectures on Human Language Technologies, 10; Toronto: Morgan & Claypool Publishers, 2010).

Hosny, Khaled, *Amiri* (Dec. 13, 2017), http://www.amirifont.org/.

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

Krüger, Marcel, *The Lua-ul package: Underlining for LuaLaTeX* (version 0.0.1) (Mar. 12, 2020), http://www.ctan.org/pkg/lua-ul.

Lagally, Klaus, *ArabTEX: Typesetting Arabic and Hebrew* (version 4.00) [User Manual Version 4.00] (Nov. 3, 2004), http://mirrors.ctan.org/language/arabic/arabtex/doc/html/arabtex.htm.

Lane, Edward William, *An Arabic-English lexicon*, 8 vols. (London – Edinburgh: Williams and Norgate, 1863–1893).

Wright, W. LL.D, *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).

16 Change History

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.

T

U

V

W

X

Z