The arabluatex package
v1.8.3 – 2017/04/27

Robert Alessi
alessi@robertalessi.net*

CNRS UMR 8167 Paris (France)

Contents

*Institutional email: robert.alessi@cnrs.fr

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. It will eventually interact with some other packages yet to come to produce from `.tex` source files, in addition to printed books, `TEI xml` compliant critical editions and/or lexicons that can be searched, analyzed and correlated in various ways.

License and disclamer

License applicable to the software

arabluatex — Processing ArabTEX notation under LuaLaTeX.

Please send error reports and suggestions for improvements to Robert Alessi:

- email: <alessi@robertalessi.net>
- website: http://www.robertalessi.net/arabluatex
- development: http://git.robertalessi.net/arabluatex
- comments, feature requests, bug reports: https://notabug.org/ralessi/arabluatex/issues

gpl3+

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

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

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

This release of arabluatex consists of the following source files:

- `arabluatex.ins`
- `arabluatex.dtx`
- `arabluatex.lua`
- `arabluatex_voc.lua`
- `arabluatex_fullvoc.lua`
- `arabluatex_novoc.lua`

- arabluatex_trans.lua
- arabluatex.el

License applicable to this document

1 Introduction

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

arabtex dates back to 1992. As far as I know, it was then the first and only way to typeset Arabic texts with TEX and LATEX. To achieve that, arabtex provided—and still does—an Arabic font in *Nasḫī* style and a macro package that defined its own input notation which was, as the author stated, "both machine, and human, readable, and suited for electronic transmission and e-mail communication".[2] Even if the same can 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).[3] Nonetheless, it remains that inserting all the time these invisible characters in complex documents rapidly becomes confusing and cumbersome.

The great advantage of ArabTEX notation is that it is immune from all these difficulties, let alone its being clear and straightforward. One also must remember that computers are designed to process code. ArabTEX notation is a way of encoding Arabic language, just as TEX "mathematics mode" is a way of processing code to display mathematics. As such, not only does it allow greater control over typographical features, but it also can be processed in several different ways: so without going into details, depending on one's wishes, ArabTEX input can be full

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

[2] Lagally (2004, p. 2).

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

vocalized Arabic (*scriptio plena*), vocalized Arabic or non-vocalized Arabic (*scriptio defectiva*); it further can be transliterated into whichever romanization standard the user may choose.

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

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

1.1 arabluatex is for LuaLaTeX

It goes without saying that arabluatex requires LuaLaTeX. TeX and LaTeX have arabtex, and XeLaTeX has arabxetex. Both of them are much more advanced than arabluatex, as they can process a number of different languages,[4] 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

As usual put in your preamble:

```
\usepackage{arabluatex}
```

The only requirement of arabluatex is LuaLaTeX; it will complain if you try to compile your document with another engine. That aside, arabluatex does not load packages such as polyglossia or luabidi. It can work with polyglossia though, but does not require it.

[4] 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.

Font setup If you wish to use your own Arabic font, you can define it before loading arabluatex. Assuming that fontspec is loaded, put this in your preamble just above the line that loads arabluatex:

```
\newfontfamily\arabicfont{⟨fontname⟩}[Script=Arabic]
```

where ⟨*fontname*⟩ is the standard name of the Arabic font you wish to use.

By default, if no Arabic font is selected, arabluatex will issue a warning message and attempt to load the Amiri font[5] like so:—

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

> Rem. 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 (1896, i.14.C–D). The placement of the *kasrah* above the consonant may be obtained by selecting the ss05 feature of the Amiri font, like so:—[6]
>
> ```
> \newfontfamily\arabicfont{Amiri}[Script=Arabic,RawFeature={+ss05}]
> ```
>
> Other Arabic fonts may behave differently.

2.2 Options

arabluatex may be loaded with four mutually exclusive global options, each of which may be overriden at any point of the document (see below section 2.3.1 on page 8):

`default`

voc

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, ـًـ, ـٍـ, ـًا, ـًى or ـًة) 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 section 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 section 7 on page 30 for more details):
dmg *Deutsche Morgenländische Gesellschaft*, which is selected by default;
loc *Library of Congress*;
arabica *Arabica*.
More standards will be included in future releases of arabluatex.

[5] Hosny (2015).
[6] See the documentation of amiri, Hosny (2015, p. 5).

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*[7] (see on page 14 for further details).

- The euphonic *tašdīd* is generated (see on page 15).
- In `fullvoc` mode, the *sukūn* is expressed.

`\SetArbEasy`

New feature v1.4.4

`\SetArbEasy*`

`\SetArbDflt`

New feature v1.6

`\SetArbDflt*`

Such refinements as 'classic' *maddah* may be discarded by the `\SetArbEasy` command, either globally in the preamble or locally at any point of the document. The difference between `\SetArbEasy` and its 'starred' version `\SetArbEasy*` is that the former keeps the *sukūn* that is generated by the `fullvoc` mode, while the latter further takes it away. Default 'classic' rules may be set back at any point of the document with the `\SetArbDflt` command. Assimilation rules laid on item b on page 15 may also be applied by the 'starred' version of this command `\SetArbDflt*` either in the preamble or at any point of the document.[8] Examples follow:—

(a) `\SetArbDflt`:

i. `voc` وَمَاتَ استِسقَآءً قَبلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَآءِ

ii. `fullvoc` وَمَاتَ ٱسْتِسْقَآءً 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 استِسقَآءً قَبلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَآءِ

ii. `fullvoc` وَمَاتَ ٱسْتِسْقَآءً قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ ٱلسَّمَآءِ

iii. `trans` *wa-māta ʾstisqāʾan qabla ʾay yutimma kitāba-hu fī nuǧūmⁱ ʾs-samāʾi*

(c) `\SetArbEasy`:

i. `voc` وَمَاتَ استِسقَاءً قَبلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ السَمَاءِ

ii. `fullvoc` وَمَاتَ ٱسْتِسْقَاءً 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 استِسقَاءً قَبلَ أَن يُتِمَّ كِتَابَهُ فِي نُجُومِ السَمَاءِ

ii. `fullvoc` وَمَاتَ ٱسْتِسْقَاءً قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ ٱلسَّمَاءِ

iii. `trans` *wa-māta ʾstisqāʾan qabla ʾan yutimma kitāba-hu fī nuǧūmⁱ ʾs-samāʾi*

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

[7] Note that in old mss. such forms as سَمَآأَ, جَآأَ are also found; see Wright (1896, i. 24 D).

[8] For an example, see section 5.1 on page 28.

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 (1896, i. 1 A):— Arabic, like Hebrew and Syriac, is written and read from right to left. The letters of the alphabet (حُرُوفُ الهِجَاءِ, حُرُوفُ التَّهَجِّي, اَلحُرُوفُ الهِجَائِيَّةُ, or حُرُوفُ المُعجَمِ) are twenty-eight in number and are all consonants, though three of them are also used as vowels (see § 3).

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

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

```
  \begin{enumerate}[XVI.]
  \item \indent\arb{fawA`ilu}*.
    \begin{enumerate}[1.]
    \item \indent\arb{fA`aluN}; as \arb{_hAtamuN} \emph{a
        signet-ring}, ...
    \end{enumerate}
  \end{enumerate}
```

XVI. فَوَاعِلُ*.

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

`arab` 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:
  \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 وَتُكَذِّبَنِي؟"

2.3.1 Local options

As seen above in section 2.2 on page 5, arabluatex may be loaded with four mutually exclusive global options: `voc` (which is the default option), `fullvoc`, `novoc` and `trans`. Whatever choice has been made globally, it may be overriden at any point of the document, as the `\arb` command may take any of the `voc`, `fullvoc`, `novoc` or `trans` modes as optional arguments, 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[9]			ArabTEX notation
	`dmg`	`loc`	`arabica`	
ا[10]	*a*	*a*	*a*	`a`
ب	*b*	*b*	*b*	`b`
ت	*t*	*t*	*t*	`t`
ث	*t̲*	*th*	*t̲*	`_t`
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	*d̲*	*dh*	*d̲*	`_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`
ة	*ah*	*ah*	*a*	`T`

Table 1: Standard ArabTEX (consonants)

3.2 Vowels

3.2.1 Long vowels

Table 2 gives the ArabTEX equivalents for the Arabic long vowels.

[9] See below section 7 on page 30.

[10] For *ʾalif* as a consonant, see Wright (1896, i. 16 D). The *hamzah* itself is encoded `<'>`. See below section 4.2 on page 13.

Letter	Transliteration[11] dmg	loc	arabica	ArabTEX notation
ا	$\bar{a}$	$\bar{a}$	$\bar{a}$	A
و	$\bar{u}$	$\bar{u}$	$\bar{u}$	U
ي	$\bar{\imath}$	$\bar{\imath}$	$\bar{\imath}$	I
ى[12]	$\acute{a}$	$\acute{a}$	$\bar{a}$	_A or Y
ـٰـ	$\bar{a}$	$\bar{a}$	$\bar{a}$	_a
ـُـ	$\bar{u}$	$\bar{u}$	$\bar{u}$	_u
ـِـ	$\bar{\imath}$	$\bar{\imath}$	$\bar{\imath}$	_i

Table 2: Standard ArabTEX (long vowels)

Rem. *a.* The long vowels $\bar{a}$, $\bar{u}$, $\bar{\imath}$, 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*.

Rem. *b.* Defective writings, such as ـٰـ, *al-'alif*^{u} *'l-maḥḏūfat*^{u}, or defective writings of $\bar{u}$ and $\bar{\imath}$ 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 *Ḥuḏayfat*^{u} *bn*^{u} *'l-Yamānī*, etc.

3.2.2 Short vowels

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

Letter	Transliteration[13] dmg	loc	arabica	ArabTEX notation
ـَـ	*a*	*a*	*a*	a
ـُـ	*u*	*u*	*u*	u
ـِـ	*i*	*i*	*i*	i
ـً	*an*	*an*	*an*	aN
ـٌ	*un*	*un*	*un*	uN
ـٍ	*in*	*in*	*in*	iN

Table 3: Standard ArabTEX (short vowels)

[11] See below section 7 on page 30.
[12] = *al-'alif*^{u} *'l-maqṣūrat*^{u}.
[13] See below section 7 on page 30.

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.[14]

> 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 section 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*⟩:

`_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*.

[14] See below section 4.4 on page 19.

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

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

al-ra.hm_an-u اَلرَّحْمٰنُ *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 ـَوة instead of ـَاة: see Wright (1896, i. 12 A). 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 maḫḏūfah* and are discarded in trans mode:

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

And so also: al-rib_aIT-u اَلرِّبٰوِيَةُ *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 (1896, i. 12 C). In no way this affects the sound of the *tanwīn*, so it has to be discarded in trans mode:

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

When the *tanwīn* falls away (Wright 1896, 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 *šayʾ^{un}*, ^say'aN شَيْئًا *šayʾ^{an}*, ^say'iN شَيْءٍ *šayʾ^{in}*.

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*^{in}.

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 ⟨ ʼ⟩, its carrier being determined by contextual analysis. In case one wishes to bypass this mechanism, he can use the "quoting" feature that is described below in section 4.4 on page 19.

Initial hamzah: `'asaduN` أَسَدٌ *ʾasad*^{un}, `'u_htuN` أُختٌ *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 *ʾūlā*, `'_UlU` أُولُو *ʾūlū*, `'_UlA'ika` أُولَائِكَ *ʾūlāʾika*.

hamzah followed by the long vowel ي is encoded `'_I`: `'_ImAnuN` إِيمَانٌ *ʾīmān*^{un}[15].

Middle hamzah: `xA.ti'-Ina` خَاطِئِينَ *ḫāṭiʾīna*, `ru'UsuN` رُؤُوسٌ *ruʾūs*^{un}, `xa.tI'aTuN` خَطِيئَةٌ *ḫaṭīʾat*^{un}, `su'ila` سُئِلَ *suʾila*, `'as'ilaTuN` أَسئِلَةٌ *ʾasʾilat*^{un}, `mas'alaTuN` مَسأَلَةٌ *masʾalat*^{un}, `'as'alu` أَسأَلُ *ʾasʾalu*, `yatasA'alUna` يَتَسَآءَلُونَ *yatasāʾalūna*, `murU'aTuN` مُرُوءَةٌ *murūʾat*^{un}, `ta'xIruN` تَأْخِيرٌ *taʾḫīr*^{un}, `ta'ax xara` تَأَخَّرَ *taʾaḫḫara*, `ji'tu-ka` جِئتُكَ *ǧiʾtu-ka*, `qA'iluN` قَائِلٌ *qāʾil*^{un}, `.hIna'i_diN` حِينَئِذٍ *ḥīnaʾiḏ*^{in}.

From Wright (1896, i. 14 B):— All consonants, whatsoever, not even *ʾalif hèmzatum* excepted, admit of being doubled and take *tašdīd*. Hence we speak and write `ra''AsuN` رَأَّاسٌ *raʾʾās*^{un}, `sa''AluN` سَأَّالٌ *saʾʾāl*^{un}, `na''AjuN` نَأَّاجٌ *naʾʾāǧ*^{un}.

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

Final hamzah: `xa.ta'uN` خَطَأٌ *ḫaṭaʾun*, `xa.ta'aN` خَطَأً *ḫaṭaʾan*, `xa.ta'iN` خَطَإٍ *ḫaṭaʾin*, `'aqra'u` أَقرَأُ *ʾaqraʾu*, `taqra'Ina` تَقرَئِينَ *taqraʾīna*, `taqra'Una` تَقرَؤُونَ *taqraʾūna*, `yaqra'na` يَقْرَأْنَ *yaqraʾna*, `yaxba'Ani` يَخبَآنِ *yaḫbaʾāni*, `xaba'A` خَبَآ *ḫabaʾā*, `xubi'a` خُبِئَ *ḫubiʾa*, `xubi'UA` خُبِئُوا *ḫubiʾū*, `jA'a` جَآءَ *ǧāʾa*, `ridA'uN` رِدَآءٌ *ridāʾun*, `ridA'aN` رِدَآءً *ridāʾan*, `jI'a` جِيءَ *ǧīʾa*, `radI'iN` رَدِيءٍ *radīʾin*, `sU'uN` سُوءٌ *sūʾun*, `.daw'uN` ضَوءٌ *ḍawʾun*, `qay'iN` قَيءٍ *qayʾin*, `^sifA'I` شِفَآئِي *šifāʾī*, `man^sa'I` مَنشَإِي *manšaʾī*.

`^say'uN` شَيءٌ *šayʾun*, `^say'aN` شَيئًا *šayʾan*, `^say'iN` شَيءٍ *šayʾin*, `al-^say'-u` اَلشَّيءُ *aš-šayʾu*, `'a^syA'-u` أَشيَآءُ *ʾašyāʾu*, `'a^syA'-a` أَشيَآءَ *ʾašyāʾa*, `.zim'aN` ظِمئًا *ẓimʾan*.

taḫfīf ’l-hamzat: 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 *ʾīmānun*. For other possible ways of encoding such sequences, see on page 13 (*hamzah* followed by و and ي) and the *maddah* on the current page.

maddah At the beginning of a syllabe, *ʾalif* with *hamzah* and *fatḥah* (أَ) followed by *ʾalifu ’l-maddi* (*ʾalif* of prolongation) or *ʾalif* with *hamzah* and *ǧazmah* (أْ) are both represented in writing *ʾalif* with *maddah*: آ (see Wright 1896, i. 25 A–B).

> Hence one should keep to this distinction and encode `'a'kulu` آكُلُ *ʾākulu* and `'AkiluN` آكِلٌ *ʾākilun* respectively.

arabluatex otherwise determines *al-ʾalif ’l-mamdūdat* by context analysis.

`'is'AduN` إِسآدٌ *ʾisʾādun*, `'AkilUna` آكِلُونَ *ʾākilūna*, `'a'mannA` آمَنَّا *ʾāmannā*, `al-qur'An-u` اَلقُرآنُ *al-qurʾānu*.

`jA'a` جَآءَ *ǧāʾa*, `yatasA'alUna` يَتَسَآءَلُونَ *yatasāʾalūna*, `ridA'uN` رِدَآءٌ *ridāʾun*, `xaba'A` خَبَآ *ḫabaʾā*, `yaxba'Ani` يَخبَآنِ *yaḫbaʾāni*.

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

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

`allaqa عَلَّقَ *ʿallaqa*, 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 (Wright 1896, 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 اَلشَّمسُ *aš-šams*^{u}, or a^s-^sams-u اَلشَّمسُ *aš-šams*^{u}.

al-tamr-u اَلتَّمرُ *at-tamr*^{u}, al-ra.hm_an-u اَلرَّحمٰنُ *ar-raḥmān*^{u}, al-.zulm-u اَلظُّلمُ *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 لَّيلٍ *mil layl*^{in}, 'an yaqtula أَن يَّقتُلَ *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 section 2.2.1 on page 6 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. لَبِثُّ for لَبِثتُ *labiṯtu*, will be discarded here, as it is largely condemned by the grammarians (see Wright 1896, 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-llāh*^{i}.

As said above 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 arabluatex 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 ٱلْمَدْرَسَةِ *bābᵘ ʾl-madrasatⁱ*, `al-maqA laT-u 'l-'_Ul_A` ٱلْمَقَالَةُ ٱلْأُولَى *al-maqālatᵘ ʾl-ʾūlā*, `al-lu.gaT-u 'l-`ara biyyaT-u` ٱللُّغَةُ ٱلْعَرَبِيَّةُ *al-luġatᵘ ʾl-ʿarabiyyatᵘ*, `fI .sinA`aT-i 'l-.tibb-i` فِي صِنَاعَةِ ٱلطِّبِّ *fī ṣināʿatⁱ ʾṭ-ṭibbⁱ*, `'il_A 'l-intiqA.d-i` إِلَى ٱلْٱنْتِقَاضِ *ʾilạ ʾl-intiqāḍⁱ*, `fI 'l-ibtidA'-i` فِي ٱلْٱبْتِدَآءِ *fi ʾl-ibtidāʾⁱ*, `'abU 'l-wazIr-i` أَبُو ٱلْوَزِيرِ *ʾabu ʾl-wazīrⁱ*, `fa-lammA ra'aW 'l-najm-a` فَلَمَّا رَأَوُا ٱلنَّجْمَ *fa-lammā raʾawu ʾn-nağmᵃ*.

Particles:—

(a) *li-*: *ʾalifᵘ ʾl-waṣlⁱ* is omitted in the article ٱلْ when it is preceded by the preposition لِ: `li-l-rajul-i` لِلرَّجُلِ *li-r-rağulⁱ*.
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-llaylatⁱ*.

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

(c) With the other particles, *ʾalifᵘ ʾl-waṣlⁱ* is expressed: `fI 'l-madIna T-i` فِي ٱلْمَدِينَةِ *fi ʾl-madīnatⁱ*, `wa-'l-rajul-u` وَٱلرَّجُلُ *wa-ʾr-rağulᵘ*, `bi-'l-qalam-i` بِٱلْقَلَمِ *bi-ʾl-qalamⁱ*, `bi-'l-ru`b-i` بِٱلرُّعْبِ *bi-ʾr-ruʿbⁱ*.

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*, `qad-i in.sarafa` قَدِ ٱنْصَرَفَ *qadi ʾnṣarafa*, `al-iqtidAr-u` ٱلِاقْتِدَارُ *al-iqtidārᵘ*, `'il_A 'l-inti qA.d-i` إِلَى ٱلْٱنْتِقَاضِ *ʾilạ ʾl-intiqāḍⁱ*, `law istaqbala` لَوِ ٱسْتَقْبَلَ *lawi ʾstaqbala*.

Other cases: `'awi ismu-hu` أَوِ ٱسْمُهُ *ʾawi ʾsmu-hu*, `zayduN ibn-u `amriNU` زَيْدٌ ٱبْنُ عَمْرٍو *Zaydᵘⁿⁱ ʾbnᵘ ʿAmrⁱⁿ*,[16] `umar-u ibn-u 'l-_ha.t.tAb-i` عُمَرُ ٱبْنُ ٱلْخَطَّابِ *ʿUmarᵘ ʾbnᵘ ʾl-Ḫaṭṭābⁱ*,[17] `imru'-u 'l-qays-i` اِمْرُؤُ ٱلْقَيْسِ *Imruʾᵘ ʾl-Qaysⁱ*, `la-aymun-u 'l-l_ah-i` لَأَيْمُنُ ٱللهِ *la-ʾymunᵘ ʾl-lāhⁱ*.

ʾalifᵘ ʾl-waṣlⁱ preceded by a long vowel The long vowel preceding the connective *ʾalif* is shortened in pronunciation (Wright 1896, i. 21 B–D). This does not appear in the Arabic script, but arabluatex takes it into account in some transliteration standards:—

`fI 'l-nAs-i` فِي ٱلنَّاسِ *fi ʾn-nāsⁱ*, `'abU 'l-wazIr-i` أَبُو ٱلْوَزِيرِ *ʾabu ʾl-wazīrⁱ*, `fI 'l-ibtidA'-i` فِي ٱلِابْتِدَاءِ *fi ʾl-ibtidāʾⁱ*, `_dU 'l-i`lAl-i` ذُو ٱلِاعْلَالِ *ḏu ʾl-iʿlālⁱ*, `maqh_A 'l-'amIr-i` مَقهَى الأَمِيرِ *maqhạ ʾl-ʾamīrⁱ*.

ʾalifᵘ ʾl-waṣlⁱ preceded by a diphthong The diphthong is resolved into two simple vowels (Wright 1896, 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-malikⁱ*, `ix^say 'l-qaw m-a` اِخْشَيِ ٱلْقَوْمَ *iḫšayi ʾl-qawmᵃ*, `mu.s.tafaw 'l-l_ah-i` مُصْطَفَوُ ٱللهِ *muṣṭafawu ʾl-lāhⁱ*.

`ramaW 'l-.hijAraT-a` رَمَوُا ٱلْحِجَارَةَ *ramawu ʾl-ḥiğāratᵃ*, `fa-lammA ra'aW 'l-najm-a` فَلَمَّا رَأَوُا ٱلنَّجْمَ *fa-lammā raʾawu ʾn-naǧmᵃ*.

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

`'antumu 'l-kA_dib-Una` أَنْتُمُ ٱلْكَاذِبُونَ *ʾantumu ʾl-kāḏibᵘⁿᵃ*, `ra'aytumu 'l-rajul-a` رَأَيْتُمُ ٱلرَّجُلَ *raʾaytumu ʾr-raǧulᵃ*, `mani 'l-ka_d_dAb-u` مَنِ ٱلْكَذَّابُ *mani ʾl-kaḏḏābᵘ*, `qatalati 'l-rUm-u` قَتَلَتِ ٱلرُّومُ *qatalati ʾr-Rūmᵘ*.

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:—

[16] "*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".

[17] "*ʿUmar* is the son of *al-Ḫaṭṭāb*" (see footnote 16).

`mu.hammaduN 'l-nabI` مُحَمَّدٌ ٱلنَّبِي *Muḥammad*^{uni} *ʾn-nabī*, `salAmuN ud_hulUA` سَلَامٌ ٱدْخُلُوا *salām*^{unu} *ʾdẖulū*, `qa.sIdata-hu fI qatl-i \cap{'a}bI \cap{m}uslimiN 'llatI yaqUlu fI-hA` قَصِيدَتَهُ فِي قَتْلِ أَبِي مُسْلِمٍ ٱلَّتِي يَقُولُ فِيهَا *qaṣīdata-hu fī qatl*^{i} *ʾAbī Muslim*^{imi} *ʾllatī yaqūlu fī-hā*.

4.3 Special orthographies

The name of God The name of God, اَللّٰهُ, is compounded of the article اَلْ, 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 اَلْلٰهُ (I refer to 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` اَللّٰهُ *al-lāh*^{u}, `yA|`[18] `al-l_ah-u` يَا اَللّٰهُ *yā al-lāh*^{u}, `'a-fa|`[19]`-al-l_ah-i la-ta.g`alanna` أَفَاَللّٰهِ لَتَفْعَلَنَّ *ʾa-fa-al-lāh*$^{i}$ *la-taġʿalanna*, `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` اَلْحَمْدُ لِلّٰهِ *al-ḥamd*$^{u}$ *li-llāh*$^{i}$, `li-l-l_ah-i 'l-qA'il-u` لِلّٰهِ ٱلْقَآئِلُ *li-llāh*^{i} *ʾl-qāʾil*^{u}.

The conjunctive اَلَّذِي Although it is compounded of the article اَلْ, 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-šayḫ*^{a} *ʾllaḏī huwa marīḍ*^{un}, `mA 'anA bi-'lla_dI qA'iluN la-ka ^say'aN` مَا أَنَا بِٱلَّذِي قَآئِلٌ لَكَ شَيْئًا *mā ʾanā bi-ʾllaḏī qāʾil*^{un} *la-ka šayʾan*.

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

[18] Note the "pipe" character '|' here after `yA` and below after `fa` before footnote mark 19: 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 section 4.5 on page 21 on the "pipe" and section 7 on page 30 on `dmg` mode.

[19] See footnote 18.

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 ٱللَّذَيْنِ رَوَيْنَاهُمَا عَنْ جَحْظَةَ *fa-ʾinnā naḏkuru ʾṣ-ṣawt^{ayni} ʾl-laḏayni rawaynā-humā ʿan Ǧaḥẓat^{a}*.

And also: `al-la_dAni` اَللَّذَانِ *al-laḏāni*, `al-la_dayni` اَللَّذَيْنِ *al-laḏayni*, `al-latAni` اَللَّتَانِ *al-latāni*, `al-latayni` اَللَّتَيْنِ *al-latayni*, `al-lAtI` اَللَّاتِي *al-lātī*, `al-lA'|Ati`[20] 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.[21] 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: عُلّم 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 `` عُلّم 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:—

[20] Note here the "pipe" character '|': as already stated on page 14, 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 section 4.5 on page 21.

[21] See Lagally (2004, p. 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 وٱستُعمل *wa-ʾstuʿmila*, huwa "inhazama هو ٱنهزم *huwa ʾnhazama*, al-"intiqA.d-u الٱنتقاض *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 قآئِلٌ *qāla qāʾil*^{un}, ibn-u 'abI 'u.saybi`aT-"a ٱبنُ أَبِي أُصَيبِعَة *Ibn*^{u} *ʾAbī ʾUṣaybiʿat*^{a}.

(b) *tanwīn*:—

madInaT"aN مَدِينَة *madīnat*^{an}, bAb"aN بَابا *bāb*^{an}, hud"aN_A هُدى *huḍ*^{n}, ^say'"iN شَيئ *šayʾ*^{in}.

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 ٱستِقصَآءٍ *fi ʾstiqṣāʾ*^{in}, wa-"istiq.sA'-uN وَٱستِقصَآءٌ *wa-ʾstiqṣāʾ*^{un}, qAla "uhrub fa-lan tuqtala قَالَ ٱهرُب فَلَن تُقتَلَ *qāla ʾhrub fa-lan tuqtala*.

(b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala قَالَ ٱقتُلْ 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" may 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 ٱلصَّيفِيُّ ٱلَّذِي كَانَ بِقرَانُونَ *al-ǧamr*^{u} *ʾṣ-ṣayfiyy*^{u} *ʾllaḍī kāna bi-Qrānnūn*^{a}.

4.4.1 Quoting the hamzah

As said above in section 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: ⟨ `"'` ⟩ allows to determine the carrier of the *hamzah* freely, without any consideration for the context. Table 4 gives the equivalents for all the possible carriers the *hamzah* may take.

Letter	Transliteration[22]			ArabTEX notation
	`dmg`	`loc`	`arabica`	
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 4: "Quoted" *hamzah*

As one can see from table 4, 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, وَ is encoded ⟨`w"'a`⟩, while وْ 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 section 4.4.1, 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 footnote 18 on page 18 and in footnote 20 on page 19. One more example follows:—

`bi-qrAn|nUn-a` بِقِرَانُونَ *bi-Qrānnūn*[a], "in Crannon" (Thessaly, Greece).[23]

[22] See below section 7 on page 30.
[23] See more context on page 20.

As one can see, the “pipe” character between the two ⟨*n*⟩ prevents the necessary *tašdīd* rule (page 14) 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 that interferes with the
    contextual analysis.} 'l-na^gma...
\end{arab}
```

This is wrong:

فَلَمَّا رَأَوْا[a] ٱلنَّجْمَ...

[a] A footnote that 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 رَأَُوا 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 that
    interferes with the contextual analysis.} 'l-na^gma...

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

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

```
\begin{arab}[trans]
  \cap{z}ayduN\arbnull{ibnu}\LRfootnote{A footnote that interferes
    with the contextual analysis.} \arbnull{zayduN}ibn-u
  \cap{`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[unie] *ʾbn*[u] *ʿAmr*[in].[f]

[a] A footnote that interferes with the contextual analysis.
[b] A footnote that interferes with the contextual analysis.
[c] A footnote that interferes with the contextual analysis.
[d] See footnote 16 on page 17.
[e] A footnote that interferes with the contextual analysis.
[f] See footnote 16 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 may 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
```

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 1896, i. 28 B–C). The *ʾabǧad* numbers are usually distinguished from the surrounding words by a stroke placed over them.

`\abjad` *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:—

New feature v.1.1

`\abjad{45}` `kitAbu-hu fI 'l-`AdAt-i` كِتَابُهُ فِي العَادَاتِ مَة *45 kitābu-hu fi 'l-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 use the `\aemph` command that is described below in section 4.10 on the next page:—

> `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 \aemph{\abjad{1874}}` The *ʾabǧad* number for 1874 is غضعد.

4.9 Additional characters

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

For the time being, the following is included:—

Letter	Transliteration[24]			ArabTEX notation
	`dmg`	`loc`	`arabica`	
ب	*ḇ*	*b*	*b*	`.b`
د	*d́*	*d*	*d*	`^d`
ف	*f̣*	*f*	*f*	`.f`
ق	*q̇*	*q*	*q*	`.q`
ک	*k*	*k*	*k*	`.k`
ں	*n̤*	*n*	*n*	`.n`
﴾	*(*	*(*	*(*	`((`
﴿	*)*	*)*	*)*	`))`

Table 5: Additional Arabic codings

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

[24] See below section 7 on page 30.

4.10 Arabic emphasis

As already seen in section 4.8.2 on page 23, 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.

`\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 ʾl-ʿĀdāt^{i}.*

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` The `arabverse` environment may take up to six 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 section 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`
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 section 9 on page 36 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. Additionally, every `\bayt` command *must* be followed with `\\` like so:—

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

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

(a) If the two hemistichs be connected with one another by a prominent horizontal flexible stroke, the *taṭwīl* should be used, like so: `[--]` (see section 4.7 on page 23). Of course, the ending word of the *ṣadr* and the word at the commencement of the *ʿağuz* 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*,[25] while the following two select the figure of the letter *lām* connected with a following letter. In the *ʿağuz*, 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:—

```
\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 dalhᵃⁿ wa-mā yaruddu ʾl-bukāʾu

[25] See section 4.2 on page 15

(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
بِزَفُوفٍ كَأَنَّهَا هِقْلَةٌ أُمُّ رِئَالٍ دَوِّيَّةٌ سَقْفَآءُ

In the first line, the word اَلْهَمِّ should be split into اَلْهَمْـ ـمِ 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 أُمْـ ـمُ, 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.

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 `\footnote mark{}` (please mind the braces) 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.[26]

[26] The footnote package may also be used for the same effect. However, it must be loaded *after* arabluatex.

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

5.1 Example

Here follow the first lines of Imruʾu ’l-Qaysi’s *Muʿallaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the *ʾidġām* that is fit to this declamatory poetry:—[27]

```
\begin{arab}[fullvoc]
  qAla \cap{i}mru'u 'l-\cap{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-saq.ti
      'l-liw_A bayna \cap{'l-d}a_hUli fa-\cap{.h}awmali}\\
    \bayt{fa-\cap{t}U.di.ha fa-'l-\cap{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
      dAsiriN min mu`awwali}\\
  \end{linenumbers*}
\end{arabverse}
```

قَالَ ٱمْرُؤُ ٱلْقَيْسِ فِي مُعَلَّقَتِه:

1 قِفَا نَبْكِ مِنْ ذِكْرَى حَبِيبٍ وَمَنْزِلِ بِسَقْطِ ٱللِّوَى بَيْنَ ٱلدَّخُولِ فَحَوْمَلِ
2 فَتُوضِحَ فَٱلْمِقْرَاةِ لَمْ يَعْفُ رَسْمُهَا لِمَا نَسَجَتْهَا مِنْ جَنُوبٍ وَشَمْأَلِ
3 تَرَى بَعَرَ ٱلْأَرْآمِ فِي عَرَصَاتِهَا وَقِيعَانِهَا كَأَنَّهُ حَبُّ فُلْفُلِ
4 كَأَنِّي غَدَاةَ ٱلْبَيْنِ يَوْمَ تَحَمَّلُوا لَدَى سَمُرَاتِ ٱلْحَيِّ نَاقِفُ حَنْظَلِ
5 وُقُوفًا بِهَا صَحْبِي عَلَيَّ مَطِيَّهُمْ يَقُولُونَ لَا تَهْلِكْ أَسًى وَتَجَمَّلِ

[27] Please note that for the time being only the assimilation rules that are laid on item b on page 15 are applied. See section 2.2.1 on page 6 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.

6 وَإِنَّ شِفَآئِي عَبْرَةٌ مُّهَرَاقَةٌ فَهَلْ عِنْدَ رَسْمٍ دَاسِرٍ مِّن مُّعَوَّلِ
(اَلضَّرْبُ ٱلثَّانِي مِنَ ٱلْعَرُوضِ ٱلْأُولَى مِنَ ٱلطَّوِيلِ)

qāla Imruʾu ’l-Qaysi fī muʿallaqati-hi:

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

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

6 Special applications

Linguistics The same horizontal stroke as the *taṭwīl* (see section 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 ـُـَـِـٌـًـٍ *u a i ^{un} ^{an} ^{in}*, BBu BBa BBi ـُّـَّـِّ *u a i*, B--aN ـًـ $^{-an}$, B" ـْـ.

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 وَكَانَ<ـت> أُمِّي مِن عُظَمَآءِ بُيُوتِ الزَّمَازِمَةِ.

7 Transliteration

It may be more appropriate to speak of "romanization" than "transliteration" of Arabic. As seen above in section 2.2 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.[28] `dmg` transliteration convention is selected by default;

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;[29]

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.[30]

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`, `loc` or `arabica`. This command is also accepted in the preamble should one wish to set the transliteration mode globally, eg.:—

```
\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, eg. `\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 may be used for rendering transliterated text:—

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

[28] See Brockelmann et al. (1935).

[29] See http://www.loc.gov/catdir/cpso/roman.html for the source document concerning Arabic language.

[30] See http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf.

\cap **Proper names** Proper names or book titles that must have their first letters uppercased may be passed as arguments to the `\cap{⟨word⟩}` command. `\cap` 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:—

> `\cap{.hunayn-u} bn-u \cap{'is.h_aq-a}` حُنَيْنُ بْنُ إِسْحٰقَ *Ḥunayn*^{u} *bn*^{u} *ʾIsḥāq*^{a}, `\cap{`u_tm_an-u}` عُثْمٰنُ *ʿUṯmān*$^{u}$, `.daraba \cap{zayd-u} bn-u \cap{_h_alidiN} \cap{sa`d-a} bn-a \cap{`awf-i} bn-i \cap{`abd-i} \cap{'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} *ʾl-Lāh*^{i}.

However, `\cap` 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 17, 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: `\cap{mu.hammaduN}` is wrong, because the closing brace would prevent arabluatex from detecting the sequence ⟨*-uN*⟩ immediately followed by ⟨*ʾ-l*⟩. Fortunately, this can be circumvented in a straightforward way by inserting only part of the noun in the argument of `\cap` vz. up to the first letter that is to be uppercased, like so: `\cap{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:—

> `\cap{'abU} \cap{bakriN} \cap{mu\-.ham\-madu} bnu \cap{za\-ka\-riy\-yA'a} \cap{'l-rAzI}` أَبُو بَكْرٍ مُحَمَّدُ بْنُ زَكَرِيَّآءَ الرَّازِي *ʾAbū Bakr*^{in} *Muḥammad*^{u} *bn*^{u} *Zakariyyāʾ*^{a} *ʾr-Rāzī*.

7.1 Additional note on dmg convention

New feature v1.3

According to Brockelmann et al. (1935, p. 6), 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*⟩} may be used to customize the way *ʾiʿrāb* is displayed. To take one example, here is how Arabic *ʾiʿrāb* may be rendered as subscript text:—

```
\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 مِئذَنَتُهُ: *raʾaytu ǧāmiʿ*$_{an}$ *muhaddamat*$_{an}$ *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. *munṣarif*^{un} or *ġayr*^{u} *munṣarif*^{in}. The former is automatically parsed by arabluatex, whereas the latter has to be delimited with an hyphen, like so:—

munṣarif: mu`allimuN مُعَلِّمٌ *muʿallim*$^{un}$, kA'inuN كَآئِنٌ *kāʾin*$^{un}$, kA'inAtuN كَآئِنَاتٌ *kāʾin*$^{ātun}$, \cap{`amraNU} عَمرًوا *ʿAmr*^{an}, fataN_A فَتًى *fatạ*^{n}, qA.diNI قَاضٍ *qāḍi*^{n}.

ġayr munṣarif: al-mu`allim-u اَلمُعَلِّمُ *al-muʿallim*^{u}, kitAb-Ani كِتَابَانِ *kitāb*^{āni}, ra^sa'-Ani رَشَآنِ *raša*^{ʾāni}, sAriq-Una سَارِقُونَ *sāriq*^{ūna}, qA.d-Una قَاضُونَ *qāḍ*^{ūna}, al-.zulm-Atu اَلظُّلمَاتُ *aẓ-ẓulm*^{ātu}.

Rem. *a*. As the *tanwīn* is passed over in pronunciation when it is followed by the letters ر, ل, م, و, ي (see item b on page 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 أَيدُهَا *kāna ġaniyy*^{an} *lākinna-hu labisa ǧubbat*^{an} *mumazzaq*^{an} *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-nās*^{a} *kāffat*^{an} *min muḥāṭabati-hi ʾaḥad*^{un} *bi-sayyidi-nā*.

7.2 Examples

Here follows in transliteration the story of *Ǧuḥā* and his donkey (جُحَا وَحِمَارُهُ). See the code on page 8:—

'dmg' standard: *ʾatā 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 ʾāsifᵘⁿ 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 uʿīdu-hu ilay-ka fī al-masāʾi wa-adfaʿu la-ka ujratan." fa-qāla Juḥā: "anā āsifun jiddan annī lā astaṭīʿu an uḥaqqiqa la-ka raghbata-ka fa-al-ḥimāru laysa hunā al-yawma." wa-qabla an yutimma Juḥā kalāma-hu badaʾa al-ḥimāru yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: "innī asmaʿu 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?"*

8 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.[31] The Buckwalter scheme is actually processed in two steps, as it is first converted into ArabTEX. Then, once this is accomplished, the ArabTEX scheme is processed through the above described functions. In this way, the Buckwalter input scheme can make the most of the arabluatex special features that are presented in section 2.2 on page 5.

`\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}
```

[31] 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 (2010, pp. 25–26).[32] However, the following limitation apply to the 'base' and 'xml' schemes: the braces { and }, which are used to encode آ and ئ, must be replaced with square brackets viz. [and] respectively.

It is therefore recommended to use the Buckwalter 'safe' scheme.

Table 6 gives the Buckwalter equivalents that are currently used by arabluatex. The additional characters that are defined in table 5 on page 24 are also available.

Letter	Transliteration[33]			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	*t̲*	*th*	*t̲*	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	*d̲*	*dh*	*d̲*	*	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

[32] I am grateful to Graeme Andrews who suggested that the 'safe' scheme be included in arabluatex.
[33] See section 7 on page 30.

Letter	Transliteration			Buckwalter notation	
	dmg	loc	arabica	base/xml	safe
ء	ʾ	’	ʾ	'	C
آ	ʾā	’ā	ʾā	|	M
أ	ʾ	’	ʾ	>	O
ؤ	ʾ	’	ʾ	&	W
إ	ʾ	’	ʾ	<	I
ئ	ʾ	’	ʾ	]	Q
ـّـ	—	—	—	~	~
ـاً	’	’	—	[	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
ـۃ	—	—	—	o	o
ـٰـ	*ā*	*ā*	*ā*	`	e
ـ (*taṭwīl*)	—	—	—	_	_

Table 6: Buckwalter scheme

Transliteration The Buckwalter notation can also be transliterated into any accepted romanization standard of Arabic. See above section 7 on page 30 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} *ʾṭ-ṭibb*^{i}.
>
> `wa-Al-l~ehi` وَاللهِ *wa-ʾl-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} *ʾl-mamdūdat*^{i} and the *ʾalif*^{u} *ʾl-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 الٱنتِقَاضِ as expected, but only `<ilY Al-intiqADi` can be transliterated as *ʾilạ ʾl-intiqāḍi* with the correct vowel ⟨*i*⟩ in place of the *ʾalif*^{u} *ʾl-waṣl*^{i}.

9 Unicode Arabic input

New feature v1.5

As said above in section 8 on page 33 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` (b) The `txarab` environment for inserting running paragraphs of Arabic text, like so:—

```
\begin{txarab}
  <Unicode Arabic text>
\end{txarab}
```

10 LaTeX Commands in Arabic environments

General principle LaTeX commands are accepted in Arabic environments. The general principle which applies is that single-argument commands (`\command{`⟨*arg*⟩`}`) such as `\emph{`⟨*text*⟩`}`, `\textbf{`⟨*text*⟩`}` and the like, are assumed to have Arabic text as their arguments:—

`\abjad{45}` `kitAbu-hu` `\emph{fI 'l-\cap{`AdAt-i}}` 45 مَة كِتَابُهُ فِي العَادَاتِ
kitābu-hu fi ʾl-ʿĀdāt^{i}.[34]

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.}.
```

[34] This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.

```
\end{arab}
```

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

[a] اَلمُقَاتِلَة: اَلمُقَاتِلِينَ.
[b] اَلزَّمَازِمَة: طَآئِفَةُ مِنَ الفُرسِ.

Some commands, however, do not expect running text in their arguments, or one may wish to insert English text eg. 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. Even in an Arabic environment, this command may be useful.

For example, to distinguish words with a different color, one may proceed like so:—

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

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

`\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` may be used to insert English footnotes in running Arabic text:—

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

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.[35]

`\LRmarginpar` The `\LRmarginpar` command does for marginal notes the same as `\LRfootnote` does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that `\marginpar` also works in Arabic environments, but it acts as any other single-argument command inserted in Arabic environments. The general principle laid on page 36 applies.

`\setRL` `\setLR` `\setRL` and `\setLR` may 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:—

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

بَرزَوَيهِ لِبُزُرجُمهرَ بنِ البُختِكَانِ

قَالَ بَرزَوَيهِ بنُ أَزهَرَ، رَأسُ أَطِبَّاءِ فَارِسَ...

10.1 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:—

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

```
\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 \cap{'abU} \cap{'l-hu_dayli} 'ahd_A 'il_A \cap{muwaysiN}
    dajAjaTaN. wa-kAnat dajAjatu-hu 'llatI 'ahdA-hA dUna mA kAna
    yuttaxa_du li-\cap{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 <\cap{'abU} \cap{'l-hu_dayli}>
    yu`rafu
    bi-'l-'imsAki 'l-^sadIdi. fa-qAla: \enquote{wa-kayfa ra'ayta yA
      \cap{'abA} \cap{`imrAna} tilka 'l-dajAjaTa?} qAla:
    \enquote{kAnat `ajabaN mina 'l-`ajabi!}  fa-yaqUlu:
    \begin{quoting}[begintext=», endtext=«]
      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 \cap{'abU} \cap{'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 ضَحْكًا نَعْرِفُهُ نَحْنُ وَلَا يَعْرِفُهُ أَبُو ٱلْهُذَيْلِ.

10.1.1 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 a package that provides additional refinements such as paralist.

To take a first example, should one wish to typeset a list of manuscripts, the `description` environment may 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-\cap{bArIs} 2860
   `arabiyyuN.
  \item[s] max.tU.tu 'l-maktabaTi 'l-'ahliyyaTi bi-\cap{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

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 `StandardLayout=true` option, like so:—

```
\usepackage[french]{babel}
\frenchbsetup{StandardLayout=true}
```

This option will prevent babel-french from interfering with the layout of the document. Then the package paralist may be used to make the lists 'compact' as babel-french do.

10.2 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}
{\rmfamily\textquotedblright}{\rmfamily\textquotedblleft}
{\rmfamily\textquoteright}{\rmfamily\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.

10.3 reledmac

The two-arguments command `\edtext{`⟨*lemma*⟩`}{`⟨*commands*⟩`}` is supported inside `\begin{arab}` ... `\end{arab}`. 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
```

10.4 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 section 9 on page 36 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 *sūrah al-Fātiḥah*:—

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

بِسمِ اللَّهِ الرَّحمٰنِ الرَّحيمِ ﴿١﴾ الحَمدُ لِلَّهِ رَبِّ العالَمينَ ﴿٢﴾ الرَّحمٰنِ 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﴾

11 Future work

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

(a) Short-term:
   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 footnote 4 on page 4.
   ii. Formulate propositions for extending the ArabTEX notation and the transliteration tables. Include them in arabluatex. See section 4.9 on page 24.

12 Implementation

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

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{arabluatex}%
[2017/04/27 v1.8.3 An ArabTeX-like interface for LuaLaTeX]
\RequirePackage{ifluatex}
```

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

```
\ifluatex\else
 \PackageError{arabluatex}{lualatex needed}{%
  Package `arabluatex' needs LuaTeX.\MessageBreak
  So you should use `lualatex' to process your document.\MessageBreak
  See documentation of `arabluatex' for further information.}%
  \expandafter\expandafter\expandafter\csname endinput\endcsname
\fi
```

Declare the global options, and define them:

```
\DeclareOption{voc}{\def\al@mode{voc}}
\DeclareOption{fullvoc}{\def\al@mode{fullvoc}}
\DeclareOption{novoc}{\def\al@mode{novoc}}
\DeclareOption{trans}{\def\al@mode{trans}}
\ExecuteOptions{voc}
\ProcessOptions\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{etoolbox}
\RequirePackage{arabluatex-patch}
```

```
\RequirePackage{fontspec}
\RequirePackage{amsmath}
\RequirePackage{luacode}
\RequirePackage{xparse}
\RequirePackage{environ}
\RequirePackage{adjustbox}
\RequirePackage{xkeyval}
```

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

```
\newbool{al@rlmode}
```

Here begins the real work: load `arabluatex.lua`:

```
\luadirect{dofile(kpse.find_file("arabluatex.lua"))}
```

This is needed by the current versions of polyglossia and luabidi. luabidi provides a `\Footnote` command. Use it as well if it is loaded.

```
\luadirect{tex.enableprimitives("luatex",tex.extraprimitives("omega"))}
```

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
\PackageWarning{arabluatex}{\string\arabicfont\ is not defined.^^JI
  will try to load Amiri}%
\newfontfamily\arabicfont[Script=Arabic]{Amiri}\fi}%
```

`\setRL` This neutralizes what is defined by the same command in luabidi:

```
\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\rmfamily#1\egroup}}
  {\NewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\rmfamily#1\egroup}}}
```

`\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}}}
```

`\aemph` Arabic emphasis. Needs to be redefined as well.

```
\AtBeginDocument{\ifdef{\aemph}%
  {\RenewDocumentCommand{\aemph}{m}{$\overline{\text{#1}}$}}
  {\NewDocumentCommand{\aemph}{m}{$\overline{\text{#1}}$}}}
```

\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 item 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}}
```

\SetTranslitConvention

\SetTranslitConvention{⟨*convention*⟩} may 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
\NoArbUp
\ArbUpDflt
\SetArbUp

By default, \arbup is set to \textsuperscript. This is how the *tanwīn* that takes place at the end of a word should be displayed in `dmg` mode. \NoArbUp may be used either in the preamble or at any point of the document in case one wishes to have the *tanwīn* on the line. The default rule can be set back with \ArbUpDflt at

any point of the document. Finally \SetArbUp may 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{\thinspace#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}}
```

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

```
\DeclareDocumentCommand{\cap}{m}%
  {\luadirect{tex.sprint(cap(\luastringN{#1}))}}
```

\Cap \cap may 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. \Cap does the same as \cap except that *it is never discarded*. For that reason, \Cap *should never be used outside the* trans *mode*. arabluatex uses \Cap internally so as to prevent \cap 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\Cap\cap
```

\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.

```
\DeclareDocumentCommand{\txarb}{+m}{\bgroup\textdir
  TRT\arabicfont#1\egroup}
\DeclareDocumentCommand{\txtrans}{+m}{\bgroup\textdir
  TLT\al@trans@font#1\egroup}
```

txarab 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}
```

\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.

```
\DeclareDocumentCommand{\arb}{O{\al@mode} +m}%
{\edef\@tempa{#1}%
  \ifx\@tempa\al@mode@voc%
  \bgroup\textdir TRT\arabicfont%
```

```
\luadirect{tex.sprint(processvoc(\luastringN{#2},
  \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
\else%
\ifx\@tempa\al@mode@fullvoc%
\bgroup\textdir TRT\arabicfont%
\luadirect{tex.sprint(processfullvoc(\luastringN{#2},
  \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
\else%
\ifx\@tempa\al@mode@novoc%
\bgroup\textdir TRT\arabicfont%
\luadirect{tex.sprint(processnovoc(\luastringN{#2},
  \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
\else%
\ifx\@tempa\al@mode@trans%
\bgroup\textdir TLT\al@trans@style%
\luadirect{tex.sprint(processtrans(\luastringN{#2},
  \luastringO{\al@trans@convention},
  \luastringO{\al@arb@rules},
  \luastringO{\al@input@scheme}))}\egroup%
\else%
\fi\fi\fi\fi}
```

arab The arab environment does for paragraphs the same as \arb does for short insertions of Arabic text.

```
\NewEnviron{arab}[1][\al@mode]%
{\par\edef\@tempa{#1}%
  \ifx\@tempa\al@mode@voc%
  \booltrue{al@rlmode}%
  \bgroup\pardir TRT\textdir TRT\arabicfont%
  \luadirect{tex.sprint(processvoc(\luastringO{\BODY},
    \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(processfullvoc(\luastringO{\BODY},
    \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(processnovoc(\luastringO{\BODY},
    \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
  \else%
  \ifx\@tempa\al@mode@trans%
  \bgroup\pardir TLT\textdir TLT\al@trans@style%
  \luadirect{tex.sprint(processtrans(\luastringO{\BODY},
    \luastringO{\al@trans@convention},
    \luastringO{\al@arb@rules},
    \luastringO{\al@input@scheme}))}\egroup%
```

```
  \else \fi\fi\fi\fi}[\par]
```

arabverse The `arabverse` environment may receive different options: `mode`, `width`, `gutter`, `metre`, `utf` and `delim`; all of them are defined here just before the `arabverse` environment:

```
\newlength{\al@bayt@width}
\newlength{\al@gutter@width}
\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}{\arb{#1}}
\define@boolkey[al]{verse}{utf}[true]{}
\define@boolkey[al]{verse}{delim}[true]{}
\define@choicekey[al]{verse}{mode}{fullvoc, voc, novoc,
  trans}{\def\al@mode{#1}}
\presetkeys[al]{verse}{metre=\LR{\vskip -\baselineskip}, utf=false,
  delim=false}{}
```

Then follows the environment itself:

```
\NewDocumentEnvironment{arabverse}{O{}}%
{\par\centering\noindent\bgroup\setkeys[al]{verse}[metre]{#1}%
  \ifx\al@mode\al@mode@trans%
  \ifal@verse@utf\setRL\else\setLR\fi%
  \else\setRL\fi}%
{\hfill\setkeys[al]{verse}[width,gutter,utf,mode]{#1}\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.

\SetHemistichDelim A hemistich delimiter also may be defined. 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.

```
\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}
\NewDocumentCommand{\bayt}{m o m}{%
  \ifdefined\savenotes\savenotes\else\fi%
  \edef\al@tatweel{--}%
  \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#1}}%
  \IfNoValueTF{#2}{%
    \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
    \else%
    \hspace{\al@gutter@width}%
```

```
    \fi
  }{%
    \edef\@tempa{#2}%
    \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%
    \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#2}}%
    \else%
    \makebox[\al@gutter@width][s]{\arb@utf{#2}}%
    \fi\fi}%
  \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
  \ifdefined\spewnotes\spewnotes\else\fi%
}
```

`\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}%
  {\luadirect{tex.sprint(abjadify(#1))}}%
  \else%
  \NewDocumentCommand{\abjad}{m}%
  {\luadirect{tex.sprint(abjadify(#1))}}
\fi}
```

`\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(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}{m}{\marginpar{\textdir TLT #1}}
```

`\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\LR{\footnote{#1}}\egroup}
\DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pardir
  TRT\LR{\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}}}}
```

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}}
```

References

Brockelmann, Carl et al. (1935). "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. URL: http://www.naher-osten.uni-muenchen.de/studium_lehre/werkzeugkasten/dmgtransliteration.pdf.

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

Hosny, Khaled (2015). *Amiri*. URL: http://www.amirifont.org/.

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

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

Wright, W. LL.D (1896). *A Grammar of the Arabic Language*. Rev. by W. Robertson Smith and M. J. de Goeje. With a forew. by Pierre Cachia. 3rd ed. 2 vols. Beirut: Librairie du Liban.

Change History

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.