

The arabluatex package

v1.9 – 2017/07/05

Robert Alessi

alessi@robertalessi.net*

CNRS UMR 8167 Paris (France)

Contents

License and disclaimer	2	4.8.2 The <i>abjad</i>	24
1 Introduction	3	4.9 Additional characters	25
1.1 arabluatex is for Lua [®] TeX	4	4.10 Arabic emphasis	26
2 The basics of arabluatex	4	5 Arabic poetry	26
2.1 Activating arabluatex	4	5.1 Example	29
2.2 Options	5	6 Special applications	30
2.2.1 Classic contrasted with modern typesetting of Arabic	6	7 Transliteration	31
2.3 Typing Arabic	7	7.1 Additional note on dmg convention .	32
2.3.1 Local options	8	7.2 Examples	33
3 Standard Arab[®]TeX input	8	8 Buckwalter input scheme	34
3.1 Consonants	8	9 Unicode Arabic input	37
3.2 Additional characters	10	10 L[®]TeX Commands in Arabic environments	37
3.3 Vowels	10	10.1 New commands	39
3.3.1 Long vowels	10	10.2 Environments	40
3.3.2 Short vowels	11	10.2.1 Lists	41
4 arabluatex in action	12	10.3 csquotes	42
4.1 The vowels and diphthongs	12	10.4 Two-argument commands	43
4.2 Other orthographic signs	14	10.5 quran	44
4.3 Special orthographies	19	11 Future work	45
4.4 Quoting	20	12 Implementation	45
4.4.1 Quoting the <i>hamzah</i>	22	References	54
4.5 The “pipe” character ()	22	Change History	55
4.6 Putting back on broken contextual analysis rules	23	Index	56
4.7 Stretching characters: the <i>taṭwīl</i> .	24		
4.8 Digits	24		
4.8.1 Numerical figures	24		

*Institutional email: robert.alessi@cnrs.fr

List of Tables

1	ArabTeX consonants	9	5	“Quoted” <i>hamzah</i>	22
2	ArabTeX additional characters . . .	10	6	Additional Arabic codings	25
3	ArabTeX long vowels	10	7	Buckwalter scheme	35
4	ArabTeX short vowels	11			

Abstract

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

License and disclaimer

License applicable to the software

`arabluatex` — Processing ArabTeX notation under Lua^ATeX.

Copyright © 2016–2017 Robert Alessi

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>

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`


gpl3+

- `arabluatex_trans.lua`
- `arabluatex.el`

License applicable to this document

Copyright © 2016–2017 Robert Alessi

CC BY-SA 4.0

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

1 Introduction

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

`arabtex` dates back to 1992. As far as I know, it was then the first and only way to typeset Arabic texts with T_EX and L^AT_EX. To achieve that, `arabtex` provided—and still does—an Arabic font in *Nashī* style and a macro package that defined its own input notation which was, as the author stated, “both machine, and human, readable, and suited for electronic transmission and e-mail communication”.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 particularly abundant. It must also be said that most text editors have trouble in displaying Arabic script connected with preceding or following L^AT_EX 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 T_EX “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 Lua^AT_EX

It goes without saying that `arabluatex` requires Lua^AT_EX. T_EX and L^AT_EX have `arabxetex`, and X_YL^AT_EX 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 ArabT_EX input on the fly into Unicode Arabic script, whereas the latter passes ArabT_EX input on to a set of Lua functions. At first, L^AT_EX 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 from what is not, it is processed by other Lua functions which rely on different sets of correspondence tables to do the actual conversion in accordance with one’s wishes. Finally, Lua returns to T_EX the converted strings—which may in turn contain some other ArabT_EX 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 Lua^AT_EX; 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.

⁴To date, both packages support Arabic, Maghribi, Urdu, Pashto, Sindhi, Kashmiri, Uighuric and Old Malay; in addition to these, `arabxetex` 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 font⁵ like so:—

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

REM. By default Amiri places the *kasrah* in combination with the *tašdīd* below the consonant, like so: `ك`. That is correct, as at least in the oldest manuscripts `ك` may stand for `ك` as well as `ك`. 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 overridden at any point of the document (see below section 2.3.1 on page 8):

voc default
In this mode, which is the one selected by default, every short vowel written generates its corresponding diacritical mark: *dammah* (◌ُ), *fathah* (◌َ) and *kasrah* (◌ِ). If a vowel is followed by N, viz. $\langle uN, aN, iN \rangle$, then the corresponding *tanwīn* (◌ِ, ◌ُ, ◌ِ, ◌ِ or ◌ِ) is generated. Finally, $\langle u, a, i \rangle$ 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 (◌ِ◌ِ◌ِ).

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 20).

trans This mode transliterates the ArabT_EX input into one of the accepted standards. At present, three standards are supported (see below section 7 on page 31 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`.

⁵Hosny (2015).

⁶See 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, arabxatex typesets Arabic in a classic, traditional style the most prominent features of which are the following:

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

samA'uN سَمَاءُ ^{un} samā, jA'a جَاءَ ^a jā'a, yatasA'alUna يَتَسَاءَلُونَ ^a yatasā'alūna⁷
(see on page 15 for further details).

- The euphonic *tašdīd* is generated (see on page 16).
- In *fullvoc* mode, the *sukūn* is expressed.

New feature
v1.4.4

\SetArbEasy

Such refinements as ‘classic’ *maddah* may be discarded by the \SetArbEasy command, either globally in the preamble or locally at any point of the document. The difference between \SetArbEasy and its ‘starred’ version \SetArbEasy* is that the former keeps the *sukūn* that is generated by the *fullvoc* mode, while the latter further takes it away. 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 16 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:—

New feature
v1.6

\SetArbDflt*

(a) \SetArbDflt:

- voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- trans wa-māta 'stisqā^{an} qabla 'an yutimma kitāba-hu fī nuġūmⁱ 's-samāⁱ

(b) \SetArbDflt*:

- voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- trans wa-māta 'stisqā^{an} qabla 'ay yutimma kitāba-hu fī nuġūmⁱ 's-samāⁱ

(c) \SetArbEasy:

- voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- trans wa-māta 'stisqā^{an} qabla 'an yutimma kitāba-hu fī nuġūmⁱ 's-samāⁱ

(d) \SetArbEasy*:

- voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- trans wa-māta 'stisqā^{an} qabla 'an yutimma kitāba-hu fī nuġūmⁱ 's-samāⁱ

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

⁷Note that in old mss. such forms as سَمَاءُ, جَاءَ are also found; see Wright (1896, i. 24 D).

⁸For an example, see section 5.1 on page 29.

2.3 Typing Arabic

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

```
1 From \textcite[i. 1 A]{Wright}:--- Arabic, like Hebrew and
2 Syriac, is written and read from right to left. The letters
3 of the alphabet (\arb{.hurUf-u 'l-hijA'-i}, \arb{.hurUf-u
4 'l-tahajjI}, \arb{al-.hurUf-u 'l-hijA'iyyaT-u}, or
5 \arb{.hurUf-u 'l-mu`jam-i}) are twenty-eight in number and
6 are all consonants, though three of them are also used as
7 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 الْهِجَاءِ, حُرُوفُ الْهِجَائِيَّةِ, or حُرُوفُ الْمُعْجَمِ) 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):—

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

XVI. فَوَاعِلُ*.

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

`arab` Running paragraphs of Arabic text should rather be placed inside an *Arabic environment*

```
1 \begin{arab}
2 [...]
3 \end{arab}
```

like so:—

```

1 \begin{arab}
2   'at_A .sadIquN 'il_A ju.hA ya.tlubu min-hu .himAra-hu
3   li-yarkaba-hu fI safraTiN qa.sIraTiN fa-qAla la-hu:
4   \enquote{sawfa 'u`Idu-hu 'ilay-ka fI 'l-masA'-i
5   wa-'adfa`u la-ka 'ujraTaN.} fa-qAla ju.hA:
6   \enquote{'anA 'AsifuN jiddaN 'annI lA 'asta.tI`u 'an
7   'u.haqqiqa la-ka ra.gbata-ka fa-'l-.himAr-u laysa hunA
8   'l-yawm-a.} wa-qabla 'an yutimma ju.hA kalAma-hu bada'a
9   'l-.himAr-u yanhaqu fI 'i.s.tabli-hi. fa-qAla la-hu
10  .sadIqu-hu: \enquote{'innI 'asma`u .himAra-ka yA ju.hA
11  yanhaqu.} fa-qAla la-hu ju.hA: \enquote{.garIbuN
12  'amru-ka yA .sadIqI 'a-tu.saddiqu 'l-.himAr-a
13  wa-tuka_d_diba-nI?}
14 \end{arab}

```

أَتَى صَدِيقٌ إِلَى جُحَّا يَطْلُبُ مِنْهُ حِمَارَهُ لِيَرْكَبَهُ فِي سَفَرَةٍ قَصِيرَةٍ فَقَالَ لَهُ: "سَوْفَ أُعِيدُهُ إِلَيْكَ فِي الْمَسَاءِ
وَأَدْفَعُ لَكَ أُجْرَةً." فَقَالَ جُحَّا: "أَنَا آسَفٌ جِدًّا أَنِّي لَا أَسْتَطِيعُ أَنْ أُحَقِّقَ لَكَ رَغْبَتَكَ فَالْحِمَارُ لَيْسَ هُنَا
الْيَوْمَ." وَقَبْلَ أَنْ يَتِمَّ جُحَّا كَلَامَهُ بَدَأَ الْحِمَارُ يَنْهَقُ فِي إِصْطِبَلِهِ. فَقَالَ لَهُ صَدِيقُهُ: "إِنِّي أَسْمَعُ حِمَارَكَ يَا جُحَّا
يَنْهَقُ." فَقَالَ لَهُ جُحَّا: "غَرِيبُ أَمْرِكَ يَا صَدِيقِي أَتَصَدِّقُ الْحِمَارَ وَتَكْذِبُنِي؟"

2.3.1 Local options

As seen above in section 2.2 on page 5, arabuatex 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 overridden 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:—

<code>voc</code>	— <code>\arb[voc]{\langle Arabic text \rangle};</code>
<code>fullvoc</code>	— <code>\arb[fullvoc]{\langle Arabic text \rangle};</code>
<code>novoc</code>	— <code>\arb[novoc]{\langle Arabic text \rangle};</code>
<code>trans</code>	— <code>\arb[trans]{\langle Arabic text \rangle}.</code>

The same optional arguments may be passed to the environment `arab`: one may have `\begin{arab}[\langle mode \rangle] ... \end{arab}`, where `\langle mode \rangle` 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			ArabT _E X notation
	dmg	loc	arabica	
أ ¹⁰	'u 'a 'i	'u, 'a, 'i	'u, 'a, 'i	'u or 'a or 'i
ب	<i>b</i>	<i>b</i>	<i>b</i>	b
ت	<i>t</i>	<i>t</i>	<i>t</i>	t
ث	<i>ṭ</i>	<i>th</i>	<i>ṭ</i>	_t
ج	<i>ǧ</i>	<i>j</i>	<i>ǧ</i>	^g or j
ح	<i>ḥ</i>	<i>ḥ</i>	<i>ḥ</i>	.h
خ	<i>ḫ</i>	<i>kh</i>	<i>ḫ</i>	_h or x
د	<i>d</i>	<i>d</i>	<i>d</i>	d
ذ	<i>ḍ</i>	<i>dh</i>	<i>ḍ</i>	_d
ر	<i>r</i>	<i>r</i>	<i>r</i>	r
ز	<i>z</i>	<i>z</i>	<i>z</i>	z
س	<i>s</i>	<i>s</i>	<i>s</i>	s
ش	<i>š</i>	<i>sh</i>	<i>š</i>	^s
ص	<i>ṣ</i>	<i>ṣ</i>	<i>ṣ</i>	.s
ض	<i>ḍ</i>	<i>ḍ</i>	<i>ḍ</i>	.d
ط	<i>ṭ</i>	<i>ṭ</i>	<i>ṭ</i>	.t
ظ	<i>ẓ</i>	<i>ẓ</i>	<i>ẓ</i>	.z
ع	<i>ʿ</i>	<i>ʿ</i>	<i>ʿ</i>	`
غ	<i>ǧ</i>	<i>gh</i>	<i>ǧ</i>	.g
ف	<i>f</i>	<i>f</i>	<i>f</i>	f
ق	<i>q</i>	<i>q</i>	<i>q</i>	q
ك	<i>k</i>	<i>k</i>	<i>k</i>	k
ل	<i>l</i>	<i>l</i>	<i>l</i>	l
م	<i>m</i>	<i>m</i>	<i>m</i>	m
ن	<i>n</i>	<i>n</i>	<i>n</i>	n
ه	<i>h</i>	<i>h</i>	<i>h</i>	h
و	<i>w</i>	<i>w</i>	<i>w</i>	w
ي	<i>y</i>	<i>y</i>	<i>y</i>	y ¹¹
ة	<i>ah</i>	<i>ah</i>	<i>a</i>	T

Table 1: Standard ArabT_EX (consonants)

REM. *a*. Please note that in all cases of elision, the *ʿalifu ʿl-waṣli* is expressed only by the vowel that accompanies the omitted *hamzah*: ⟨*u, a, i*⟩ as in **wa-inhazama** وانْهَزَمَ *wa-ʿnhazama*. For more details on the definite article and the *ʿalifu ʿl-waṣli* see section 4.2 on page 16.

⁹See below section 7 on page 31.

¹⁰See below, Rem. *a*. For *ʿalif* as a consonant, see Wright (1896, i. 16 D). The *hamzah* itself is encoded <'> and may be followed by either ⟨*u, a*⟩ or ⟨*i*⟩. See below section 4.2 on page 14.

¹¹For the letter ع with no diacritical point below, see Rem. *b*. below.

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

REM. *b.* The letter ي with two points below, أَلْيَاءُ الْمُتَّاءُ مِنْ تَحْتِهَا, may also be written without diacritical points as ى. When it is used as a consonant, it is encoded **aY**, where **a** recalls the *fathah* placed above the preceding letter in vocalized Arabic, like so: **qaY'uN** قَائِنٌ *qay'un*, **ˆsaY'uN** سَائِنٌ *say'un*, **ˆsaY'aN** سَائِنٌ *say'an*.

The same result may be achieved by encoding this letter as **.y**, like so: **qa.y'uN** قَائِنٌ *qay'un*, **ˆsa.y'uN** سَائِنٌ *say'un*, **ˆsa.y'aN** سَائِنٌ *say'an*.

3.2 Additional characters

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

New feature
v1.8.5

Letter	Transliteration ¹²			ArabTeX notation
	dmg	loc	arabica ¹³	
پ	<i>p</i>	<i>p</i>	<i>p</i>	p
چ	<i>č</i>	<i>ch</i>	<i>č</i>	ˆc
ژ	<i>ž</i>	<i>zh</i>	<i>ž</i>	ˆz
ف ¹⁴	<i>v</i>	<i>v</i>	<i>v</i>	v
گ	<i>g</i>	<i>g</i>	<i>g</i>	g
ن ¹⁵	<i>ñ</i>	<i>ñ</i>	<i>ñ</i>	ˆn

Table 2: Standard ArabTeX (additional characters)

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

3.3 Vowels

3.3.1 Long vowels

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

Letter	Transliteration ¹⁶			ArabTeX notation
	dmg	loc	arabica	
ا	<i>ā</i>	<i>ā</i>	<i>ā</i>	A

¹²See below section 7 on page 31.

¹³The characters that are listed in this table are not included in this standard. However, as **arabica** is based on **dmg**, the **dmg** equivalents have been used here.

¹⁴This character is not found in Brockelmann et al. (1935, p. 2). It is taken from the DIN 31 635 (2011) standard.

¹⁵See footnote 14.

¹⁶See below section 7 on page 31.

Letter	Transliteration			ArabTeX notation
	dmg	loc	arabica	
و	\bar{u}	\bar{u}	\bar{u}	U
ي	\bar{i}	\bar{i}	\bar{i}	I ¹⁷
ى ¹⁸	\bar{a}	\bar{a}	\bar{a}	_A or Y
ا	\bar{a}	\bar{a}	\bar{a}	_a
ـ	\bar{u}	\bar{u}	\bar{u}	_u
ـ	\bar{i}	\bar{i}	\bar{i}	_i

Table 3: Standard ArabTeX (long vowels)

REM. *a.* The long vowels \bar{a} , \bar{u} , \bar{i} , otherwise called *hurūf^u 'l-maddⁱ*, 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 ا, *al-'alif^u 'l-mahdūfat^u*, or defective writings of \bar{u} and \bar{i} are encoded _a_u and _i respectively, e.g. _d_alika ذلك, al-mal_a'ikaT-u 'l-ra.hm_an-u الملكُ الرَّحْمَنُ, .hu_dayfaT-u bn-u 'l-yamAn_i الْيَمَانُ حَدِيْفَةُ بَنُ الْيَمَانِ for *Hudayfat^u bn^u 'l-Yamānī*, etc.

REM. *c.* The letter ي with two points below, أَلْيَاءُ الْمُنَاةُ مِنْ تَحْتِهَا, may also be written without diacritical points as ى. When it is used as a long vowel, it is encoded iY, where i recalls the *kasrah* placed below the preceding letter in vocalized Arabic, like so: liY لِي, yam~siY يَمَشِي *yamšī*.

3.3.2 Short vowels

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

Letter	Transliteration ¹⁹			ArabTeX notation
	dmg	loc	arabica	
ا	<i>a</i>	<i>a</i>	<i>a</i>	a
و	<i>u</i>	<i>u</i>	<i>u</i>	u
ي	<i>i</i>	<i>i</i>	<i>i</i>	i
ان	<i>an</i>	<i>an</i>	<i>an</i>	aN
un	<i>un</i>	<i>un</i>	<i>un</i>	uN
in	<i>in</i>	<i>in</i>	<i>in</i>	iN

Table 4: Standard ArabTeX (short vowels)

¹⁷For the letter ى with no diacritical points, see *Rem. c.* below.

¹⁸= *al-'alif^u 'l-maqṣūrat^u*.

¹⁹See below section 7 on page 31.

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.²⁰

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 *fathah*, *dammah* or *kasrah* except in the following cases:—

- at the commencement of a word, to indicate that a connective *ʿalif* is needed, with the exception of the article (see below section 4.4 on page 20);
- 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 $\langle a, u, i \rangle$:

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

bi-hi بِهِ *bi-hi*, 'aqi.tuN أَقِطْ *'aqiṭ^un*.

la-hu لَهُ *la-hu*, .hujjaTuN حُجَّةٌ *huġġat^un*.

Long vowels They are written $\langle U, A, I \rangle$:

qAla قَالَ *qāla*, bI`a بَيْعٌ *bī`a*, .tUruN طُورٌ *tūr^un*, .tInuN طِينٌ *tīn^un*,
murU'aTuN مُرُوَّةٌ *murū`at^un*.

ʿalif maqṣūrah It is written $\langle _A \rangle$ or $\langle Y \rangle$:

al-fat_A الْفَتَى *al-fatā*, al-maqh_A الْمَقْهَى *al-maqhā*, 'il_A إِلَى *ilā*.

ʿalif otiosum Said *ʿalif^u ʿl-wiqāyatⁱ*, “the guarding *ʿalif*”, after و at the end of a word, both when preceded by *dammah* and by *fathah* is written $\langle UA \rangle$ or $\langle aW, aWA \rangle$:

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

²⁰See below section 4.4 on page 20.

‘alif maḥdūfah and defective ū, ī They are written ⟨_a, _i _u⟩:

al-l_ah-u اللهُ *al-lāh^u*, 'il_ahNإله *ilāh^{un}*.
 al-ra.hm_an-uالرَّحْمَنُ *ar-raḥmān^u*, l_akinلَكِنْ *lākīn*, h_ahunAهَهْنَأُ *hāhunā*,
 .hunayn-u bn-u 'is.h_aq-aإِسْحَاقُ بْنُ إِسْحَاقَ *Hunayn^u bn^u 'Ishāq^a*, rabb_i
 رَبِّ *rabbī*, al-`A.s_iالْعَاصِ *al-Āṣī*.

Silent ي/و Some words ending with كة are usually written حوة or نوة instead of كة: see Wright (1896, i. 12 A). arablutex preserves that particular writing; the same applies to words ending in ية for كة. Long vowels ⟨U, I⟩ shall receive no *sukūn* after a ‘alif maḥdūfah and are discarded in *trans* mode:

.hay_aUTuNحَيوة *hayāt^{un}*, .sal_aUTuNصَلوة *ṣalāt^{un}*, mi^sk_aUTuNمَشْكُوة *miš-kāt^{un}*, tawr_aITuNتَوْرِيَّة *tawrāt^{un}*.
 And so also: al-rib_aIT-uالرَّيْبِيَّة *ar-ribāt^u*.

‘Amr^{un}, and the silent و To that name a silent و is added to distinguish it from ‘Umar^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:

`amruNUعَمْرُو *amr^{un}*, `amraNUعَمْرُوا *amr^{an}*, `amriNUعَمْرُو *amrⁱⁿ*.
 When the *tanwīn* falls away (Wright 1896, i. 249 B): `amr-uU bn-u mu.ḥammadiNعَمْرُو بْنُ مُحَمَّدٍ *Amr^u bn^u Muḥammadⁱⁿ*, mu.ḥammad-u bn-u `amr-iU bn-i_hAlidiNعَمْرُو بْنُ خَالِدٍ *Muḥammad^u bn^u Amrⁱ bnⁱ Ḥālidⁱⁿ*.
 And so also: al-rib_aUAالرَّيْبَا *ar-ribā*, ribaNUرَيْبَا *rib^{an}*.

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

mAluNمَالٌ *māl^{un}*, bAbaNبَابًا *bāb^{an}*, madInaTaNمَدِينَةٌ *madīnat^{an}*, bintiNبِنْتٌ *bintⁱⁿ* maqhaN_Aمَقْهَى *maqhaⁿ*, fataNYفَتَى *fataⁿ*.
 arablutex is aware of special orthographies: ^say'uNشَيْءٌ *šay^{un}*,
 ^say'aNشَيْئًا *šay^{an}*, ^say'iNشَيْءٍ *šayⁱⁿ*.

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⟩:—

a1-qA.dI الْقَاضِي *al-qādī*, qA.diyaN قَاضِيًا *qāḍiy^{an}*, qA.diNI قَاضٍ *qāḍiⁿ*.

4.2 Other orthographic signs

tā' marbūṭah It is written ⟨T⟩:

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

hamzah It is written ⟨'⟩, its carrier being determined by contextual analysis. In case one wishes to bypass this mechanism, he can use the “quoting” feature that is described below in section 4.4 on page 20.

Initial hamzah: 'asaduN أَسَدٌ *'asad^{un}*, 'u_htuN أُخْتُ *'uht^{un}*, 'iqlIduN إِقْلِيدٌ *'iqlīd^{un}*, 'anna أَنْ *'anna*, 'inna إِنْ *'inna*.

hamzah followed by the long vowel و is encoded '_U: '_U1_A أُولَى *'ulā*, '_U1U أُولُو *'ulū*, '_U1A'ika أُولَآئِكَ *'ulā'ika*.

hamzah followed by the long vowel ي is encoded '_I: '_ImAnuN إِيْمَانٌ *'īmā-n^{un}*²¹.

Middle hamzah: xA.ti'-Ina خَاطِئِينَ *ḫāṭi'īna*, ru'UsuN رُؤُوسٌ *ru'ūs^{un}*, xa.tI'aTuN خَاطِئَةً *ḫāṭi'at^{un}*, su'ila سُئِلَ *su'ila*, 'as'ilaTuN أَسْئَلَهُ *'as'ilat^{un}*, mas'alaTuN مَسْأَلَةً *mas'alat^{un}*, 'as'alu أَسْأَلَ *'as'alu*, yataSA'alUna يَتَسَاءَلُونَ *yatasā'alūna*, murU'aTuN مُرُوءَةٌ *murū'at^{un}*, ta'xIruN تَأْخِيرٌ *ta'ḫīr^{un}*, ta'ax xara تَأَخَّرَ *ta'ahḫara*, ji'tu-ka جِئْتُكَ *ḡi'tu-ka*, qA'iluN قَائِلٌ *qā'il^{un}*, .hIna'i_diN هِنْدٌ *ḥīna'idⁱⁿ*.

From Wright (1896, i. 14 B):— All consonants, whatsoever, not even *ʿalifhəmzatum* excepted, admit of being doubled and take *tašdīd*. Hence we speak and write ra' 'AsuN رَأْسٌ *ra'ās^{un}*, sa' 'AluN سَأَلَ *sa'āl^{un}*, na' 'AjuN نَاجٌ *na'āḡ^{un}*.

²¹ For another way of encoding the initial *hamzah* followed by a long vowel, see the *taḥfīf^u 'l-hamzat* on the next page.

Final hamzah: xa.ta'uN خَطَأُ *ḥaṭaʿun*, xa.ta'aN خَطَأٌ *ḥaṭaʿan*, xa.ta'iN خَطِئَ *ḥaṭaʿin*, 'aqra'u أَقْرَأُ *'aqra'u*, taqra'Inا تَقْرَأُ *taqra'ina*, taqra'Una تَقْرَأُونَ *taqra'una*, yaqra'na يَقْرَأُ *yaqra'na*, yaxba'Anي يَخْبَأُ *yahbaʿani*, xaba'A خَبَأَ *ḥabaʿa*, xubi'a خُبِئَ *ḥubiʿa*, xubi'UA خُبُوا *ḥubiʿu*, jA'a جَاءَ *ǧāʿa*, ridA'uN رَدَأَ *ridāʿun*, ridA'aN رَدَأَ *ridāʿan*, jI'a جِئَ *ǧīʿa*, radI'iN رَدِئَ *radīʿin*, sU'uN سُوءُ *sūʿun*, .daw'uN دَوَّءُ *dawʿun*, qay'iN قَيَّ *qayʿin*, ~sifA'I شَفَائِي *šifāʿi*, man~sa'I مَنْشَأِي *manšaʿi*, nisA'uN نِسَاءُ *ni-sāʿun*, nisA'u-hu نِسَاءُهُ *nisāʿu-hu*, nisA'i-hi نِسَائِهِ *nisāʿi-hi*, nisA'I نِسَائِي *nisāʿi*.

~say'uN سَيَّ *šayʿun*, ~say'aN سَيَّأُ *šayʿan*, ~say'iN سَيَّئَ *šayʿin*, al-~say'-u أَلْسَيَّ *aš-šayʿu*, 'a~syA'-u أَشْيَاءُ *ašyāʿu*, 'a~syA'-a أَشْيَاءُ *ašyāʿa*, .zim'aN ظَمَأُ *ẓimʿan*.

tabḥfif^u 'l-hamzatⁱ: if the *hamzah* has *ǧazmah* and is preceded by *'alif hamzatum*, it must be changed into the letter of prolongation that is homogeneous with the preceding vowel; hence: 'a'mana آمَنَ *āmana*, 'u'minu أَوْمِنُ *ūminu*, 'i'mAnuN إِيمَانُ *īmān^{un}*. For other possible ways of encoding such sequences, see on page 14 (*hamzah* followed by و and ي) and the *maddah* on the current page.

maddah At the beginning of a syllable, *'alif* with *hamzah* and *fathah* (أ) followed by *'alifu 'l-maddi* (*'alif* of prolongation) or *'alif* with *hamzah* and *ǧazmah* (إ) are both represented in writing *'alif* with *maddah*: آ (see Wright 1896, i. 25 A–B).

Hence one should keep to this distinction and encode 'a'kulu أَكُلُ *ākulu* and 'AkiluN أَكِلُ *ākil^{un}* respectively.

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

'is'AduN إِسَادُ *isʿād^{un}*, 'AkilUna أَكِلُونُ *ākilūna*, 'a'manna آمَنَّا *āmannā*, al-qur'An-u الْقُرْآنُ *al-qurʿān^u*.

jA'a جَاءَ *ǧāʿa*, yatasA'alUna يَتَسَاءَلُونَ *yatasāʿalūna*, ridA'uN رَدَأَ *ridāʿun*, xaba'A خَبَأَ *ḥabaʿa*, yaxba'Anي يَخْبَأُ *yahbaʿani*.

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:

ʿallaqa عَلَقَ ʿallaqa, mAdduN مَادُّ mādd^{un}, ʿammara أَمَّرَ ammara, murruN مَرَّرَ 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 ال:—

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-ʿsams-u الشَّمْسُ aš-šams^u, or aʿs-ʿsams-u الشَّمْسُ aš-šams^u.

al-tamr-u التَّمْرُ at-tamr^u, al-ra.hm_an-u الرَّحْمَنُ ar-rahmān^u, al-zulm-u الظُّلْمُ az-ẓulm^u, al-lu.gaT-u اللَّغَةُ al-luġat^u.

- (b) With the letters ر, ل, م, و, ي after ن with *ġazmah*, 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 مِنْ رَبِّهِ *mir rabbi-hi*, min laylīN مِنْ لَيْلٍ *mil laylⁱⁿ*, ʿan yaqtula أَنْ يَقْتُلَ *ʿay yaqtula*.

With *tanwīn*: kitAbuN mubInuN كِتَابٌ مُبِينٌ 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 *ʿidġām*, will be included in arabluatex gradually.

- (c) With the letter ت after the dentals ث, د, ذ, ر, ز, س, ش, ص, ض, ط, ظ in certain parts of the verb: this kind of assimilation, e.g. لَبِثْتُ for لَبِثْتُ *labittu*, will be discarded here, as it is largely condemned by the grammarians (see Wright 1896, i. 16 B–C).

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

al-.hamd-u li-l-l_ah-i اَلْحَمْدُ لِلّٰهِ *al-ḥamd^u li-l-lāhⁱ*.

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 arabu-latex 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 *'alifu 'l-waṣli* is expressed by the vowel that accompanies the omitted *hamzah*: *<u, a, i>*.

Article: bAb-u 'l-madrasaT-i الْمَدْرَسَةُ *bāb^u 'l-madrasatⁱ*, al-maqA
laT-u 'l-'_U1_A الْمَقَالَةُ *al-maqālat^u 'l-'ūlā*, al-lu.gaT-u 'l-'ara
biyyaT-u اللُّغَةُ الْعَرَبِيَّةُ *al-luḡat^u 'l-'arabiyyat^u*, fI .sinA`aT-i 'l-.tibb-
i إِلَى الْإِنْتِقَاضِ *ilā 'l-intiqā.d-i* فِي صِنَاعَةِ الطِّبِّ *fī ṣinā`atⁱ 'l-ṭibbⁱ*, 'il_A 'l-intiqA.d-i
'l-intiqādⁱ, fI 'l-ibtidA'-i فِي الْإِبْتِدَاءِ *fī 'l-ibtidā`i*, 'abU 'l-wazIr-i أَبُو
الْوَزِيرِ *abu 'l-wazīrⁱ*, fa-lamma ra'aW 'l-najm-a رَأَوْا النَّجْمَ *fa-lammā*
ra'awu 'n-nağm^a.

Particles:—

- (a) *li-*: *'alif^u 'l-waṣlⁱ* is omitted in the article *أَلْ* 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-l-laylatⁱ*.

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

- (c) With the other particles, *'alif^u 'l-waṣlⁱ* is expressed: fI 'l-madInaT-i فِي الْمَدِينَةِ *fī 'l-madīnatⁱ*, wa-'l-rajul-u وَالرَّجُلُ *wa-'r-rağul^u*, bi-'l-qalam-i بِالْقَلَمِ *bi-'l-qalamⁱ*, bi-'l-ru`b-i بِالرُّعْبِ *bi-'r-ru`bⁱ*.

Perfect active, imperative, nomen actionis: qAla isma` قَالَ أَسْمَعُ *qāla 'sma`*, qAla uqtul قَالَ أَقْتُلُ *qāla 'qtul*, huwa inhazama هُوَ أَنْهَزَمَ *huwa 'nhazama*, wa-ustu`mila وَأَسْتَعْمِلَ *wa-'stu`mila*, qad-i in.sarafa قَدْ أَنْصَرَفَ *qadi 'nṣarafa*, al-iqtidAr-u الْأَقْتِدَارُ *al-iqtidār^u*, 'il_A 'l-intiqA.d-i إِلَى الْإِنْتِقَاضِ *ilā 'l-intiqādⁱ*, law istaqbala لَوْ اسْتَقْبَلَ *lawi 'staqbala*.

Other cases: 'awi ismu-hu أو اسمه 'awi 'smu-hu, zayduN ibn-u `amriNU
 زَيْدُ بْنُ عَمْرٍو Zayd^{uni} 'bn^u 'Amrⁱⁿ,²² `umar-u ibn-u 'l-ha.t.tAb-
 i اِمْرُؤُا اَنْطَابِ Umar^u 'bn^u 'l-Haṭṭābⁱ,²³ imru'-u 'l-qays-i اِمْرُؤُا
 اَلْقَيْسِ Imru^u 'l-Qaysⁱ, la-aymun-u 'l-l_ah-i لاَ اَيْمُنُ اللهُ la-'ymun^u 'l-lāhⁱ.

'alif^u 'l-waṣlⁱ preceded by a long vowel The long vowel preceding the connective 'alif 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 النَّاسِ فِي fī 'n-nāsⁱ, 'abU 'l-wazIr-i أَبُو الْوَزِيرِ 'abu 'l-wazīrⁱ,
 fI 'l-ibtidA'-i اَلْاِبْتِدَاءِ فِي fī 'l-ibtidā'ⁱ, _dU 'l-i-lA1-i ذُو الْاَعْلَالِ du
 'l-i'lālⁱ, maqh_A 'l-'amIr-i اَلْمَقْهَى الْاَمِيرِ maqha 'l-'amīrⁱ.

'alif^u 'l-waṣlⁱ preceded by a diphthong The diphthong is resolved into two simple vowels (Wright 1896, i. 21 D–22 A) viz. *ay* → *āi* and *aw* → *āū*. arabluatex detects the cases in which this rule applies:—

fI `aynay 'l-malik-i اَلْمَلِكِ فِي عَيْنِي fī 'aynayi 'l-malikⁱ, ix^say 'l-qaw
 m-a اِخْشَى الْقَوْمِ iḥṣayⁱ 'l-qawm^a, mu.s.tafaw 'l-l_ah-i مُصْطَفَوُ اللهِ muṣṭa-
 fawu 'l-lāhⁱ.
 ramaW 'l-.hiJAraT-a رَمَوْا الْحِجَارَةَ ramawu 'l-ḥiḡārat^a, fa-lamma ra'aW
 'l-najm-a فَلَمَّا رَأَوْا النَّجْمَ fa-lammā ra'awu 'n-naġm^a.

'alif^u 'l-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 'alif 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 اَنْتُمْ اَلْكَادِبُونَ 'antumU 'l-kādīb^{ūna}, ra'aytumu
 'l-rajul-a رَاَيْتُمُ الرَّجُلَ ra'aytumu 'r-raġul^a, mani 'l-ka_d_dAb-u مَنِ
 اَلْكَدَّابُ mani 'l-kaddāb^u, qatalati 'l-rUm-u قَتَلَتِ الرُّومُ qatalati 'r-Rūm^u.

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

²² “Zayd is the son of ‘Amr”: the second noun is not in apposition to the first, but forms part of the predicate. Hence زَيْدُ بْنُ عَمْرٍو and not عَمْرٍو بْنُ زَيْدٍ, “Zayd, son of ‘Amr”.

²³ “‘Umar is the son of al-Haṭṭāb” (see footnote 22).

mu.hammaduN 'l-nabI مُحَمَّدٌ النَّبِيُّ *Muḥammad^{uni} 'n-nabī*, salAmuN ud_hulUA
 سَلَامٌ أَدْخُلُوا *salām^{unu} 'dhulū*, qa.sIdata-hu fI qatl-i \cap{'a}bI
 \cap{m}uslimiN 'llatI yaqUlu fI-hA الَّذِي يَقُولُ أَنِّي مُسْلِمٌ
 قَصِيدَتُهُ فِي قَتْلِ أَبِي مُسْلِمٍ *qasīdata-hu fī qatlⁱ 'Abī Muslimⁱⁿⁱ 'llatī yaqūlu fī-hā*.

4.3 Special orthographies

The name of God The name of God, اَللّٰهُ, is compounded of the article اَلْ, and اِلَٰه (noted اِله with the defective *alif*) so that it becomes اِلَٰه; then the *hamzah* is suppressed, its vowel being transferred to the ل before it, so that there remains اَللّٰهُ (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 16), 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|²⁴ al-l_ah-u يَ اَللّٰهُ *yā al-lāh^u*, 'a-fa|²⁵-al-l_ah-i la-ta.g`alanna أَفَاللّٰهُ لَتَغَلَنَّ *'a-fa-al-lāhⁱ la-tag'alanna*, bi-'l-l_ah-i بِاللّٰهِ *bi-'l-lāhⁱ*, wa-'l-l_ah-i وَاللّٰهِ *wa-'l-lāhⁱ*, bi-sm-i 'l-l_ah-i بِسْمِ اللّٰهِ *bi-smⁱ 'l-lāhⁱ*, al-.hamd-u li-l-l_ah-i لِلّٰهِ اَلْحَمْدُ *al-ḥamd^u li-l-lāhⁱ*, li-l-l_ah-i 'l-qA'il-u لِلّٰهِ الْقَائِلُ *li-l-lāhⁱ 'l-qā'il^u*.

The conjunctive الَّذِي 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 *alif* are encoded 'lla_dI and 'llatI:—

'a_hAfu mina 'l-malik-i 'lla_dI ya.zlimu 'l-nAs-a أَخَافُ مِنَ الْمَلِكِ *'aḥāfu mina 'l-malikⁱ 'lladī yaẓlimu 'n-nās^a*, `udtu 'l-عُدْتُ الشَّيْخَ الَّذِي هُوَ مَرِيضٌ *'udtu 'š-šayḥ^a 'lladī huwa marīḍ^{un}*, mA 'anA bi-'lla_dI qA'iluN la-ka مَا بِيْ اَنَا بِالَّذِي قَائِلٌ لَكَ شَيْئًا *mā 'anā bi-'lladī qā'il^{un} la-ka šay^{an}*.
 'ari-nA 'lla_dayni 'a.dallA-nA mina 'l-jinn-i wa-'l-'ins-i أَرِنَا الَّذِي أَضَلَّانَا مِنَ الْجِنِّ وَالْإِنْسِ *'ari-na 'lladayni 'aḍallā-nā mina 'l-ḡinnⁱ wa-'l-'insⁱ*.

²⁴Note the “pipe” character ‘|’ here after yA and below after fa before footnote mark 25: it is needed by the *dmg* transliteration mode as in this mode any vowel at the commencement of a word preceded by a word that ends with a vowel, either short or long, is absorbed by this vowel viz. *‘alā ‘t-tariqⁱ*. See section 4.5 on page 22 on the “pipe” and section 7 on page 31 on *dmg* mode.

²⁵See footnote 24.

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 حَظَّةً رَوَيْنَاهُمَا عَنِ الَّذِينَ رَوَيْنَاهُمَا عَنْ فَانًا نَذْرًا fa-'innā nadkuru 'ṣ-ṣawt^{ayni}
'l-la_{dayni} rawaynā-humā `an Ġahẓat^a.

And also: al-la_dAni اللَّذَانِ al-ladāni, al-la_dayni اللَّذَيْنِ al-ladayni,
al-latAni اللَّتَانِ al-latāni, al-latayni اللَّتَيْنِ al-latayni, al-lAtI اللَّاتِي al-
lātī, al-lA' |Ati²⁶ اللَّاءَاتِ al-lā'āti, al-lA' I اللَّائِي al-lā'ī, and so forth.

4.4 Quoting

It is here referred to “quoting” after the arabtex package.²⁷ 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: عِلْمٌ علم الهيئة; is it عِلِّمٌ, he was taught the science of astronomy, or عِلِّمٌ, he taught the science of astronomy? In order to disambiguate this clause, it may be sensible to put a *ḍammah* above the first ع, عِلْمٌ 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*, *fathah* or *kasrah* appear above the appropriate consonant:—

`ullima `ilm-a 'l-hay'aT-i علم الهيئة ullima 'ilm^a 'l-hay'atⁱ,
ya.gz"UA يَغْزُوا yağzū.

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

wa-'innA sawfa tudriku-nA 'l-manAyA muqadd"araT"aN وإِنَّا سَوْفَ
تُدْرِكُنَا الْمَنَآيَا مُقَدَّرَةً wa-'innā sawfa tudriku-na 'l-manāyā muqaddarat^{an}.

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

²⁶Note here the “pipe” character '|’: as already stated on page 15, the sequence 'A usually encodes *ʿalif* with *hamzah* followed by *ʿalif* of prolongation, which is represented in writing *ʿalif* with *maddah*: ٱ. The “pipe” character prevents this rule from being applied. See section 4.5 on page 22.

²⁷See Lagally (2004, p. 22)

qAla isma`" قال اسمع *qāla 'sma*, jA'at" hinduN جَاءَتْ هِنْدُ *ǧā'at*
Hind^{un}, ḥsabIhuN bi-man q"u.ti`at" qadamA-hu شَبِيهَ بِنِ قُطِعَتْ *qadamA-hu*
šabīh^{un} bi-man quṭi'at qadamā-hu.

- (d) At the commencement of a word, the straight double quote is interpreted as 'alif^u 'l-waṣl':—

wa-"ust"u`mila وَأَسْتَعْمَل *wa-'stu'mila*, huwa "inhazama هو أَنْهَزَم *huwa*
'nhazama, al-"intiqA.d-u الِإِنْتِقَاضُ *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 قَالَ قَائِلٌ *qāla qā'il^{un}*, ibn-u 'abI 'u.saybi`aT-"a
 ابْنُ أَبِي أُصْبَيْعَةَ *Ibn^u 'Abī 'Uṣaybi'at^a*.

- (b) *tanwīn*:—

madInaT"aN مَدِينَةٌ *madīnat^{an}*, bAb"aN بَابًا *bāb^{an}*, hud"aN_A هُدًى *hudāⁿ*,
 شَيْءٌ *šayⁱⁿ*.

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

- (a) 'alif^u 'l-waṣl':—

fI "istiq.sA'-iN فِي اسْتِقْصَاءٍ *fi 'stiqṣā'ⁱⁿ*, wa-"istiq.sA'-uN وَأَسْتِقْصَاءُ
wa-'stiqṣā'^{un}, qAla "uhrub fa-lan tuqtala قَالَهُ هَرُبَ فَلَنْ تُقْتَلَ *qāla*
'hrub fa-lan tuqtala.

- (b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala قَالَهُ قَاتِلُ فَلَنْ تُقْتَلَ *qāla 'qtul fa-lan tuq-*
tala, mA jA'at" mini imra'aTiN مَا جَاءَتْ مِنْ امْرَأَةٍ *mā ǧā'at mini*
 'mra'atⁱⁿ, kam" qad" ma.dat" min" laylaTiN كَمْ قَدْ مَضَتْ مِنْ لَيْلَةٍ *kam*
qad maḍat min laylatⁱⁿ.

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 أَلْجَرُ الصِّفِيُّ
 الَّذِي كَانَ يَقْرَأُونَ *al-ǧamr^u 'ṣ-ṣayfiyy^u 'llaḍī kāna bi-Qrānnūn^a*.

4.4.1 Quoting the hamzah

As said above in section 4.2 on page 14, the *hamzah* is always written ⟨'⟩, its carrier being determined by contextual analysis. “Quoting” that straight single quote character like so: ⟨"'"⟩ allows to determine the carrier of the *hamzah* freely, without any consideration for the context. Table 5 gives the equivalents for all the possible carriers the *hamzah* may take.

Letter	Transliteration ²⁸			ArabTeX notation
	dmg	loc	arabica	
ء	،	،	،	"'"
آ	ā	ā	ā	A"'"
أ	،	،	،	a"'"
إ	،	،	،	u"'"
ؤ	،	،	،	w"'"
أ	،	،	،	i"'"
ي	،	،	،	y"'"

Table 5: “Quoted” *hamzah*

As one can see from table 5, the carrier of the *hamzah* is inferred from the letter that precedes the straight double quote ⟨"⟩. Of course, any “quoted” *hamzah* may take a short vowel, which is to be written *after* the ArabTeX equivalent for the *hamzah* itself, namely ⟨'⟩. For example, ؤ is encoded ⟨w"'"a⟩, while ؤ is encoded ⟨w"'"'⟩. In the latter example, the second straight double quote encodes the *sukūn* in voc mode in accordance with the rule laid above on pages 20–21.

'a`dA'ukum أَعْدَاؤُكُمْ 'a`dā'ukum, 'a`dA|"'"'ukum أَعْدَاءُكُمْ 'a`dā'ukum, 'a`dA'ikum أَعْدَائِكُمْ 'a`dā'ikum, 'a`dA|"'"'ikum أَعْدَاءُكُمْ 'a`dā'ikum.

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 24 on page 19 and in footnote 26 on page 20. One more example follows:—

bi-qrAn|nUn-a بِقِرَآنُونْ^a bi-Qrānnūn^a, “in Crannon” (Thessaly, Greece).²⁹

²⁸See below section 7 on page 31.

²⁹See more context on page 21.

As one can see, the “pipe” character between the two $\langle n \rangle$ prevents the necessary *tašdīd* rule (page 16) from being applied.

4.6 Putting back on broken contextual analysis rules

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

```

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

```

This is wrong:

فَلَمَّا رَأَوْا^a النَّجْمَ...

^a A footnote that interferes with the contextual analysis.

According to the rule stated on page 18, the diphthong in *ra'aw* must be resolved into two simple vowels before the *'alif*^a *'l-waṣl*ⁱ, as رَأَوْا النَّجْمَ.

\arbnul

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

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

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

```

1 \begin{arab}[fullvoc]
2   fa-lamma ra'aW\arbnul{'l-na^gma}\LRfootnote{A footnote that
3     interferes with the contextual analysis.} 'l-na^gma...
4
5   qAla\LRfootnote{A footnote that interferes with the contextual
6     analysis.} \arbnul{qAla}uhrub fa-lan tuqtala.
7
8   \cap{z}ayduN\arbnul{ibnu}\LRfootnote{A footnote that interferes
9     with the contextual analysis.} \arbnul{zayduN}ibn-u
10  \cap{'a}mriNU.\LRfootnote{See \vref{fn:zayd-is-son}.}
11 \end{arab}

```

```

12 \begin{arab}[trans]
13   \cap{z}aydu\arbnnull{ibnu}\LRfootnote{A footnote that interferes
14     with the contextual analysis.} \arbnnull{zaydu}ibn-u
15   \cap{`a}mriNU.\LRfootnote{See \vref{fn:zayd-is-son}.}
16 \end{arab}

```

فَلَمَّا رَأَوْا^a النَّجْمَ...
 قَالَ^b أَهْرَبْ فَلَنْ تُقْتَلَ.
 زَيْدٌ^c ابْنُ عَمْرٍو^d.

Zayd^{unie} 'bn^u 'Amrⁱⁿ.^f

^aA footnote that interferes with the contextual analysis.

^bA footnote that interferes with the contextual analysis.

^cA footnote that interferes with the contextual analysis.

^dSee footnote 22 on page 18.

^eA footnote that interferes with the contextual analysis.

^fSee footnote 22 on page 18.

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 إِسْحَاقَ *Hunayn^u bn^u 'Ishāq^a*

4.8 Digits

4.8.1 Numerical figures

The *Indian numbers*, *ar-raḡam^u 'l-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 ١٠٢٤ فِي سَنَةِ

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 *abjad* numbers are usually distinguished from the surrounding words by a stroke placed over them.

`\abjad` *abjad* numbers are inserted with the `\abjad{<number>}` command in any of the `voc`, `fullvoc` and `novoc` modes, where <number> may be any number between 1 and 1999, like so:—

New feature
v.1.1

`\abjad{45}` kitAbu-hu fI 'l-`AdAt-i مَكَّابَةُ فِي الْعَادَاتِ 45 *kitābu-hu fi*
'l-`ādāt.

REM. *a.* As can be seen in the above given example, `arabluatex` expresses the *abjad* 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 *abjad* number for 1874 is غَضَد.

The `\arb[trans]{\abjad}` number for 1874 is `\aemph{\abjad{1874}}` The *abjad* number for 1874 is غَضَد.

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 ³⁰			ArabT _E X notation
	dmg	loc	arabica	
ب	<i>ḅ</i>	<i>b</i>	<i>b</i>	.b
د	<i>ḍ</i>	<i>d</i>	<i>d</i>	˘d
ف	<i>f̣</i>	<i>f</i>	<i>f</i>	.f
ق	<i>q̣</i>	<i>q</i>	<i>q</i>	.q
ك	<i>ḳ</i>	<i>k</i>	<i>k</i>	.k
ن	<i>ṇ</i>	<i>n</i>	<i>n</i>	.n
⸀	(((((
⸁)))))

Table 6: Additional Arabic codings

'afAman.tUs Gal.(M) .fmn.n.ts (sic) Gal.(E1), أفامنطوس Gal.(M) فمנטس (sic) Gal.(E1), *'afāmanṭūs* Gal.(M) *fmnṇṭs* (sic) Gal.(E1).

³⁰See below section 7 on page 31.

4.10 Arabic emphasis

As already seen in section 4.8.2 on page 24, the *ʿabjad* 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-ʿAdāt-i}` 45: كِتَابُهُ فِي الْعَادَاتِ
kitābu-hu fi l-ʿĀdāt.

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:—

```
1 \begin{arabverse}[key1=value1, key2=value2, ...]
2 <verses>
3 \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 37 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>] {<ağuz>}\\
```

That two subsequent hemistichs should be connected with one another is technically named *tadwīr*. Should that happen, either the *ṣadr* or the *ağuz* 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 24). Of course, the ending word of the *ṣadr* and the word at the commencement of the *ağuz* must have the *taṭwīl* too so that the proper shapes of the letters be selected. Consider for example the following:—

```
1 \begin{arabverse}[mode=fullvoc, width=.3\linewidth]
2 \bayt{1A 'ar_A man `ahidtu fī-hA fa-'abkI 'l---}{---yawma
3 dalhaN wa-mA yaruddu 'l-bukA'u}\\
4 \end{arabverse}
```

لَا أَرَى مِنْ عَهْدُتُ فِيهَا فَأُبْكِي الْيَوْمَ دَهْلًا وَمَا يَرُدُّ الْبُكَاءُ

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 *ʿalif^u* *ʿl-waṣlⁱ*,³¹ while the following two select the figure of the letter *lām* connected with a following letter. In the *ağuz*, the last two hyphens select the letter *yā* connected with a preceding letter, while the first one is simply discarded in this mode, but still may appear as it should, if the `trans` mode be selected:—

```
1 \begin{arabverse}[mode=trans, width=.4\linewidth]
2 \bayt{1A 'ar_A man `ahidtu fī-hA fa-'abkI 'l---}{---yawma
3 dalhaN wa-mA yaruddu 'l-bukA'u}\\
4 \end{arabverse}
```

lā 'arā man 'ahidtu fī-hā fa-'abki 'l- -yawma dālḥ^{am} wa-mā yaruddu 'l-bukā'u

³¹See section 4.2 on page 16

- (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:—

```

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

```

غَيْرَ أَيِّ قَدْ اسْتَعِينُ عَلَى الْهَمِّ إِذَا خَفَّ بِالتَّوَيِّ النَّجَاءُ
 يَرْفُوفٍ كَانَهَا هِقْلَةً أُمُّ رِثَالٍ دَوِيَّةً سَقْفَاءُ

In the first line, the word *أَلْهَمَّ* should be split into *أَلْهَمَّ* as the first part of it belongs to the *šadr* 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 *أُمُّ*, 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.³²

³²The `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 Imru'u 'l-Qaysi's *Mu'allaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the *idgām* that is fit to this declamatory poetry:—³³

```

1 \begin{arab}[fullvoc]
2   qAla \cap{i}mru'u 'l-\cap{q}aysi fI mu`allaqati-hi:
3 \end{arab}
4
5 \begin{arabverse}[mode=fullvoc, metre={(al-.darbu 'l-_tAnI mina
6   'l-`arU.di 'l-'_Ul_A mina 'l-.tawIli)}]
7 \SetArbDflt*
8 \begin{linenumbers*}
9   \bayt{qifA nabki min _dikr_A .habIbiN wa-manzili}{bi-saq.ti
10    'l-liw_A bayna \cap{'l-d}a_hUli fa-\cap{.h}awmali}\\
11   \bayt{fa-\cap{t}U.di.ha fa-'l-\cap{m}iqrATi lam ya`fu
12    rasmu-hA}{limA nasa^gat-hA min ^ganUbiN wa-^sam'ali}\\
13   \bayt{tar_A ba`ara 'l-'ar'Ami fI `ara.sAti-hA}{wa-qI`Ani-hA
14    ka-'anna-hu .habbu fulfuli}\\
15   \bayt{ka-'annI .gadATa 'l-bayni yawma ta.hammalUA}{lad_A
16    samurAti 'l-.hayyi nAqifu .han.zali}\\
17   \bayt{wuqUfaN bi-hA .sa.hbI `alayya ma.tiyya-hum}{yaqUlUna lA
18    tahlik 'asa_NA wa-ta^gammali}\\
19   \bayt{wa-'inna ^sifA'I `abraTuN muharAqaTuN}{fa-hal `inda rasmiN
20    dAsiriN min mu`awwali}\\
21 \end{linenumbers*}
22 \end{arabverse}

```

قَالَ أَمْرُؤُ الْقَيْسِ فِي مَعْلَقَتِهِ:

1	بَسَطَ الْوَيْ بَيْنَ الدَّخُولِ حَوْمَلِ	فَقَا نَبِكْ مِنْ ذِكْرِي حَبِيبٍ وَمَنْزَلِ
2	لَمَّا نَسَجْتَهَا مِنْ جَنُوبٍ وَشَمَالِ	فَتَوَضَّعَ فَاَلْمَقْرَأَةُ لَمْ يَعْفُ رَسْمَهَا
3	وَقِيَعَانَهَا كَأَنَّهُ حَبٌّ فَلْفُلِ	تَرَى بَعَرَ الْأَرَامِ فِي عَرَصَاتِهَا
4	لَدَى سِمَرَاتِ الْحَيِّ نَاقِفُ حَنْظَلِ	كَأَنِّي غَدَاةَ الْبَيْنِ يَوْمَ تَحْمَلُوا
5	يَقُولُونَ لَا تَهْلِكْ أَسَى وَتَجَلِ	وَقُوفًا بِهَا صَحْبِي عَلَيَّ مَطْلَبِهِمْ

³³Please note that for the time being only the assimilation rules that are laid on item **b** on page 16 are applied. See section 2.2.1 on page 6 for more information. None of the editions of the *Mu'allaqāt* that I know of feature the *idgām* in the Arabic text, although it is often strongly marked in declamation.

6 وَأَنَّ شِفَاتِي عِبْرَةٌ مَهْرَاقَةٌ فَهَلْ عِنْدَ رَسْمٍ دَاسِرٍ مِّنْ مُّوَلِّ
(الضَّرْبُ الثَّانِي مِنَ الْعُرُوضِ الْأُولَى مِنَ الطَّوِيلِ)

qāla Imru'u 'l-Qaysi fī mu'allaqati-hi:

1	<i>qifā nabki min dīkrā ḥabīb^{iw} wa-manzili</i>	<i>bi-saqṭi 'l-liwā bayna 'd-Daḥūli fa-Ḥawmali</i>
2	<i>fa-Tūḍiḥa fa-'l-Migrāti lam ya'fu rasmu-hā</i>	<i>limā nasaḡat-hā min ḡanūb^{iw} wa-šam'ali</i>
3	<i>tarā ba'ara 'l-'ar'āmi fī 'arašāti-hā</i>	<i>wa-qī'āni-hā ka-'anna-hu ḥabbu fulfuli</i>
4	<i>ka-'annī ḡadāta 'l-bayni yawma taḥammalū</i>	<i>ladā samurāti 'l-ḥayyi nāḡifu ḥanzali</i>
5	<i>wuqūf^{un} bi-hā ṣaḥbī 'alayya maṭiyya-hum</i>	<i>yaqūlūna lā tahlik 'aṣa^w wa-taḡammali</i>
6	<i>wa-'inna šifā'i 'abrat^{um} muharāḡat^{un}</i>	<i>fa-hal 'inda rasmⁱⁿ dāsir^{im} mim mu'awwali</i>

(aḡ-ḡarbu 't-tānī mina 'l-arūḡi 'l-'ulā mina 't-tawīli)

6 Special applications

Linguistics The same horizontal stroke as the *taṭwīl* (see section 4.7 on page 24) may be encoded $\langle B \rangle$; $\langle BB \rangle$ will receive the *tašdīd*. This is useful to make linguistic annotations and comments on vowels:—

Bu Ba Bi BuN BaN BiN $\overset{\text{u}}{\text{u}} \overset{\text{a}}{\text{a}} \overset{\text{i}}{\text{i}} \overset{\text{un}}{\text{un}} \overset{\text{an}}{\text{an}} \overset{\text{in}}{\text{in}}$, BBu BBa BBi $\overset{\text{u}}{\text{u}} \overset{\text{a}}{\text{a}} \overset{\text{i}}{\text{i}}$, B--aN
 ُ -^{an}, B" ̣.

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: $\langle \rangle$
- square brackets: $\llbracket \rrbracket$
- angle brackets: $\langle \rangle$
- braces: $\{ \}$

\abracas 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 `\abracas` command:—

```
1 \begin{arab}
2   \abracas{wa-qAla} 'inna 'abI kAna mina 'l-muqAtilaTi
3   wa-kAna--<--t> 'ummI min `u.zamA'i buyUti 'l-zamAzimaTi.
4 \end{arab}
```

{وَقَالَ} إِنَّ أَبِي كَانَ مِنَ الْمُقَاتِلَةِ وَكَانَتْ أُمِّي مِنْ عُظَمَاءِ بَيُوتِ الزَّمَاةِ.

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 1935.³⁴ **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;³⁵

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.³⁶

More standards will be included in future releases of arabluatex.

New feature
v1.8

`\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.:—

```
1 \usepackage{arabluatex}
2 \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{}` 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:—

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

³⁴See Brockelmann et al. (1935).

³⁵See <http://www.loc.gov/catdir/cpsd/roman.html> for the source document concerning Arabic language.

³⁶See 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 initial letter, apart from the article, cannot be uppercased, viz. ‘ or ‘, the letter next to it will be uppercased:—

`\cap{.hunayn-u} bn-u \cap{'is.h_aq-a}` حُنَيْنُ بْنُ إِسْحَقَ *Hunayn^u bn^u*
`\cap{'u_tm_an-u}` عُثْمَانُ *Utmān^u*, .daraba `\cap{zayd-u} bn-`
`\cap{_h_alidiN} \cap{sa`d-a} bn-a \cap{`awf-i} bn-i \cap{``
`\cap{abd-i} \cap{'l-l_ah-i}` زَيْدُ بْنُ خَلْدٍ سَعْدُ بْنُ عَوْفِ بْنِ عَبْدِ اللَّهِ *daraba*
Zayd^u bn^u Ḥālidⁱⁿ Sa`d^a bn^a Awfⁱ bnⁱ Abdⁱ 'l-Lāhⁱ.

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 18, the transliteration of مُحَمَّدٌ النَّبِيُّ must be *Muḥammad^{uni} 'n-nabī*, as nouns having the *tanwīn* take a *kasrah* in pronunciation before *'alifu 'l-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 `⟨'-l⟩`. Fortunately, this can be circumvented in a straightforward way by inserting only part of the noun in the argument of `\cap` viz. up to the first letter that is to be uppercased, like so: `\cap{m}u.hammaduN`.

Hyphenation In case transliterated Arabic words break the T_EX 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 زَكْرِيَّا الرَّازِي *Abū Bakrⁱⁿ Mu-*
ḥammad^u bn^u Zakariyyā^a 'r-Rāzī.

7.1 Additional note on dṃg convention

According to Brockelmann et al. (1935, p. 6), Arabic *ʾiṣrāb* may be rendered into *dṃg* in three different ways:

- (a) In full: *Amrrun*;
- (b) As superscript text: *Amr^{un}*;
- (c) Discarded: *Amr*.

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

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

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

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

Arabic dmg transliteration for رَأَيْتُ جَامِعًا مَهْدَمَةً مِثْلَتَهُ: *ra'aytu ḡami`an muhaddamat_{an} mi'danatu-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.

ʿirāb boundaries Every declinable noun (*muʿrab*) may be declined either with or without *tanwīn*, viz. *munṣarif^{un}* or *gayr^u munṣarifⁱⁿ*. The former is automatically parsed by *arabluatex*, whereas the latter has to be delimited with an hyphen, like so:—

munṣarif: *mu`allimuN* مُعَلِّمٌ *mu'allim^{un}*, *kA'inuN* كَاتِبٌ *kā'in^{un}*, *kA'inAtuN* كَاتِبَاتٌ *kā'in^{ātun}*, `\cap{`amraNU}` عَمَرُوا *Amr^{an}*, *fataN_A* فَتَى *fataⁿ*, *qA.diNI* قَاضٍ *qāḍiⁿ*.

ḡayr munṣarif: *al-mu`allim-u* الْمُعَلِّمُ *al-mu'allim^u*, *kitAb-Ani* كِتَابَانِ *kitāb^{āni}*, *ra`sa'-Ani* رِشَانٍ *raša'^{āni}*, *sAriq-Una* سَارِقُونَ *sāriq^{āna}*, *qA.d-Una* قَاضُونَ *qāḍ^{āna}*, *al-.zulm-Atu* الظُّلُمَاتُ *aẓ-ẓulm^{ātu}*.

REM. a. As the *tanwīn* is passed over in pronunciation when it is followed by the letters ر, ل, ي, و, م (see item b on page 16), it may be desirable to further distinguish it by putting it above the line, but not to do the same for *gayr munṣarif* terminations. This can be achieved by simply omitting the hyphen before any *gayr munṣarif* termination:—

kAna .ganiyyaN l_akinna-hu labisa ḡubbaTaN mumazzaqaN 'aydu-hA كَانَ غَنِيًّا لَكِنَّهُ لَيْسَ جَبَّةً مَزَقًا *kāna ḡaniyy^{an} lākinna-hu labisa ḡubbat^{an} mumazzaq^{an} 'aydu-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 مَنَّ النَّاسُ كَافَّةً مِنْ أَهَادٍ بِي سَيِّدِي *mana'a 'n-nās^a kāffat^{an} min muḥāṭabati-hi 'aḥad^{un} bi-sayyidi-nā*.

7.2 Examples

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

‘dmg’ standard: *‘atā ṣadīq^{un} ilā Ġuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratⁱⁿ qaṣīratⁱⁿ fa-qāla la-hu: “sawfa u’idu-hu ilay-ka fī l-masā’i wa-adfa’u la-ka uḡrat^{an}.” fa-qāla Ġuḥā: “anā āsīf^{un} ḡidd^{an} annī lā astatī’u an uḥaqqiqa la-ka raġbata-ka fa-l-ḥimār^u laysa huna l-yawm^a.” wa-qabla an yutimma Ġuḥā kalāma-hu bada’a l-ḥimār^u yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: “innī asma’u ḥimāra-ka yā Ġuḥā yanhaqu.” fa-qāla la-hu Ġuḥā: “ġarīb^{un} amru-ka yā ṣadīqī a-tuṣaddiqu l-ḥimār^a wa-tukaddība-nī?”*

‘loc’ standard: *atā ṣadīqun ilā Juḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratⁱⁿ qaṣīratⁱⁿ fa-qāla la-hu: “sawfa u’idu-hu ilay-ka fī al-masā’i wa-adfa’u la-ka ujratan.” fa-qāla Juḥā: “anā āsīfun jiddan annī lā astatī’u an uḥaqqiqa la-ka raġbata-ka fa-al-ḥimāru laysa hunā al-yawma.” wa-qabla an yutimma Juḥā kalāma-hu bada’a al-ḥimāru yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: “innī asma’u ḥimāra-ka yā Juḥā yanhaqu.” fa-qāla la-hu Juḥā: “ġharībun amru-ka yā ṣadīqī a-tuṣaddiqu al-ḥimāra wa-tukadhdhiba-nī?”*

‘arabica’ standard: *atā ṣadīqun ilā Ġuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratⁱⁿ qaṣīratⁱⁿ fa-qāla la-hu: “sawfa u’idu-hu ilay-ka fī l-masā’i wa-adfa’u la-ka uḡratan.” fa-qāla Ġuḥā: “anā āsīfun ḡiddan annī lā astatī’u an uḥaqqiqa la-ka raġbata-ka fa-l-ḥimāru laysa hunā l-yawma.” wa-qabla an yutimma Ġuḥā kalāma-hu bada’a l-ḥimāru yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: “innī asma’u ḥimāra-ka yā Ġuḥā yanhaqu.” fa-qāla la-hu Ġuḥā: “ġarībun amru-ka yā ṣadīqī a-tuṣaddiqu l-ḥimāra wa-tukaddība-nī?”*

8 Buckwalter input scheme

Even though arabluatex is primarily designed to process the ArabTeX notation, it can also process the Buckwalter input scheme to a large extent.³⁷ The Buckwalter scheme is actually processed in two steps, as it is first converted into ArabTeX. Then, once this is accomplished, the ArabTeX scheme is processed through the above described functions. In this way, the Buckwalter input scheme can make the most of the arabluatex special features that are presented in section 2.2 on page 5.

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

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

³⁷See <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).³⁸ However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode ا and ئ, must be replaced with square brackets viz. [and] respectively.

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

Table 7 gives the Buckwalter equivalents that are currently used by `arabluatex`. The additional characters that are defined in table 6 on page 25 are also available.

Letter	Transliteration ³⁹			Buckwalter notation	
	dmg	loc	arabica	base/xml	safe
ا	<i>a</i>	<i>a</i>	<i>a</i>	A	A
ب	<i>b</i>	<i>b</i>	<i>b</i>	b	b
ت	<i>t</i>	<i>t</i>	<i>t</i>	t	t
ث	<i>ṭ</i>	<i>th</i>	<i>ṭ</i>	v	v
ج	<i>ǧ</i>	<i>j</i>	<i>ǧ</i>	j	j
ح	<i>h</i>	<i>h</i>	<i>h</i>	H	H
خ	<i>ḫ</i>	<i>kh</i>	<i>ḫ</i>	x	x
د	<i>d</i>	<i>d</i>	<i>d</i>	d	d
ذ	<i>ḍ</i>	<i>dh</i>	<i>ḍ</i>	*	V
ر	<i>r</i>	<i>r</i>	<i>r</i>	r	r
ز	<i>z</i>	<i>z</i>	<i>z</i>	z	z
س	<i>s</i>	<i>s</i>	<i>s</i>	s	s
ش	<i>š</i>	<i>sh</i>	<i>š</i>	\$	c
ص	<i>ṣ</i>	<i>ṣ</i>	<i>ṣ</i>	S	S
ض	<i>ḍ</i>	<i>ḍ</i>	<i>ḍ</i>	D	D
ط	<i>ṭ</i>	<i>ṭ</i>	<i>ṭ</i>	T	T
ظ	<i>ẓ</i>	<i>ẓ</i>	<i>ẓ</i>	Z	Z
ع	<i>ʿ</i>	<i>ʿ</i>	<i>ʿ</i>	E	E
غ	<i>ǧ</i>	<i>gh</i>	<i>ǧ</i>	g	g
ف	<i>f</i>	<i>f</i>	<i>f</i>	f	f
ق	<i>q</i>	<i>q</i>	<i>q</i>	q	q
ك	<i>k</i>	<i>k</i>	<i>k</i>	k	k
ل	<i>l</i>	<i>l</i>	<i>l</i>	l	l
م	<i>m</i>	<i>m</i>	<i>m</i>	m	m
ن	<i>n</i>	<i>n</i>	<i>n</i>	n	n
ه	<i>h</i>	<i>h</i>	<i>h</i>	h	h
و	<i>w</i>	<i>w</i>	<i>w</i>	w	w
ي	<i>y</i>	<i>y</i>	<i>y</i>	y	y
ى	<i>ā</i>	<i>á</i>	<i>ā</i>	Y	Y
ة	<i>ah</i>	<i>ah</i>	<i>a</i>	p	p

³⁸I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in `arabluatex`.

³⁹See section 7 on page 31.

Letter	Transliteration			Buckwalter notation	
	dmg	loc	arabica	base/xml	safe
ع	’	’	’	’	C
آ	’ā	’ā	’ā		M
أ	’	’	’	>	O
ؤ	’	’	’	&	W
إ	’	’	’	<	I
ئ	’	’	’]	Q
ـ	—	—	—	~	~
ـ	,	,	—	[L
ا	<i>a</i>	<i>a</i>	<i>a</i>	a	a
ـ	<i>u</i>	<i>u</i>	<i>u</i>	u	u
ـ	<i>i</i>	<i>i</i>	<i>i</i>	i	i
ان	<i>an</i>	<i>an</i>	<i>an</i>	F	F
ـ	<i>un</i>	<i>un</i>	<i>un</i>	N	N
ـ	<i>in</i>	<i>in</i>	<i>in</i>	K	K
ـ	—	—	—	o	o
ا	ā	ā	ā	`	e
ـ (taṭwīl)	—	—	—	—	—

Table 7: Buckwalter scheme

Transliteration The Buckwalter notation can also be transliterated into any accepted romanization standard of Arabic. See above section 7 on page 31 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 الْعَالَمُ *al-ālam^u*, Al-camsu الشَّمْسُ *aš-šams^u*, bi-SinaAEapi
Al-T~ib~i, بِصِنَاعَةِ الطِّبِّ *bi-ṣinā‘atⁱ ’t-tibbⁱ*.
wa-Al-l~ehi وَاللَّهِ *wa-l-lāhⁱ*, Al-Hamdu li-l~ehi اَلْحَمْدُ لِلَّهِ *al-ḥamd^u*
li-llāhⁱ.

Similarly, it is not advisable to use | and [(‘base’ and ‘xml’ schemes) or M and L (‘safe’ scheme) to encode the *‘alif^u ’l-mamdūdātⁱ* and the *‘alif^u ’l-waṣlⁱ* 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 اِلَى الْاِنْتِقَاضِ as expected, but only <ilY Al-intiqADi can be transliterated as *ila 'l-intiqāḍi* with the correct vowel ⟨i⟩ in place of the *'alif^u 'l-waṣlⁱ*.

9 Unicode Arabic input

As said above in section 8 on page 34 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:—

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

10 L^AT_EX Commands in Arabic environments

General principle L^AT_EX 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}}` مَهْ كَاتِبُهُ فِي الْعَادَاتِ 45
kitābu-hu fi 'l-Ādātⁱ.⁴⁰

The same applies to footnotes:—

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

⁴⁰This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.

8 \end{arab}

إِنَّ أَبِي كَانَ مِنَ الْمُقَاتِلَةِ^٥، وَكَانَتْ أُمِّي مِنْ عُظَمَاءِ بُيُوتِ الزَّمَاذِمَةِ^٦.

^٥الْمُقَاتِلَةُ: الْمُقَاتِلِينَ.
^٦الزَّمَاذِمَةُ: طَائِفَةٌ مِنَ الْفُرْسِ.

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

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

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

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

زَيْدٌ^٥ ابْنُ عَمْرٍو

^٥ “*Zayd* is the son of ‘*Amr*’: the second noun is not in apposition to the first, but forms part of the predicate...

When footnotes are typeset from right to left, it may happen that the numbers of the footnotes that are at the bottom of the page be typeset in the wrong direction. For example, instead of an expected number 18, one may get 81. `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.⁴¹

`\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 37 applies.

`\setRL` `\setLR` `\setLR` 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}

```

برزويه لبزجمهر بن البختكان
قال برزويه بن أزهر، رأس أطباء فارس...

10.1 New commands

New feature
v1.9

In some particular cases, it may be useful to define new commands to be inserted in Arabic environments. From the general principle laid on page 37, it follows that any command that is found inside an Arabic environment is assumed to have Arabic text in its argument which `arabluatex` will process as such before passing it on to the command itself for any further processing. As a result of this feature, such a command as:

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

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

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

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

⁴¹See <http://ctan.org/pkg/koma-script>; read the documentation of KOMA-script for details about the `\deffootnotemark` and `\deffootnote` commands.

For example, here follows a way to define a new command `\fvred` to distinguish words with a different color and always print them in fully vocalized Arabic:—⁴²

```

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

```

ثُمَّ اتَّخَذَ ذُو الْقَرْنَيْنِ إِلَى أُمَّةٍ أُخْرَىٰ فِي **مَطْلَعِ الشَّمْسِ** وَلَا بِنَاءَ لَهُمْ يَوْمَئِذٍ مِنَ الشَّمْسِ.

10.2 Environments

New feature
v1.5

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

```

1 \begin{arab}
2 \begin{<environment>}[<options>]
3 <Arabic text>
4 \end{<environment>}
5 \end{arab}

```

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

```

1 \setquotestyle{arabic}
2 \begin{arab}[fullvoc]
3 kAna \cap{'abU} \cap{'l-hu_dayli} 'ahd_A 'il_A \cap{muwaysiN}
4 dajAjaTaN. wa-kAnat dajAjatu-hu 'llatI 'ahdA-hA dUna mA kAna
5 yuttaxa_du li-\cap{muwaysiN}. wa-l_akinna-hu bi-karami-hi
6 wa-bi-.husni xuluqi-hi 'a.zhara 'l-ta`ajjuba min simani-hA
7 wa-.tIbi la.hmi-hA. wa-kAna <\cap{'abU} \cap{'l-hu_dayli}>
8 yu`rafu
9 bi-'l-'imsAKi 'l-^sadIdi. fa-qAla: \enquote{wa-kayfa ra'ayta yA
10 \cap{'abA} \cap{'imrAna} tilka 'l-dajAjaTa?} qAla:
11 \enquote{kAnat `ajabaN mina 'l-`ajabi!} fa-yaqUlu:
12 \begin{quoting}[begintext=», endtext=«]
13 wa-tadrI mA jinsu-hA? wa-tadrI mA sinnu-hA? fa-'inna
14 'l-dajAjaTa 'inna-mA ta.tIbu bi-'l-jinsi wa-'l-sinni. wa-tadrI
15 bi-'ayyi `say'iN kunna nusamminu-hA? wa-fI 'ayyi makAniN kunna
16 na`lifu-hA?

```

⁴²See below section 10.4 on page 43 for more information on two-argument commands.


```

16 \end{quoting}
17 fa-lA yazAlu fI h_a_dA wa-'l-'A_haru ya.d.haku .da.hkaN
na`rifu-hu
18 na.hnu wa-lA ya`rifu-hu \cap{'abU} \cap{'l-hu_dayli}.
19 \end{arab}

```

كَانَ أَبُو الْهَذِيلِ أَهْدَى إِلَى مُوسَى دَجَاجَةً. وَكَانَتْ دَجَاجَتُهُ الَّتِي أَهْدَاهَا دُونَ مَا كَانَ يَتَّخِذُ لِمُوسَى. وَلَكِنَّهُ
بِكَرَمِهِ وَيُحْسِنُ خُلُقَهُ أَظْهَرَ التَّعَجُّبَ مِنْ سَمْنِهَا وَطَيِّبِ لَحْمِهَا. وَكَانَ <أَبُو الْهَذِيلِ> يُعْرِفُ بِالْإِمْسَاكِ الشَّدِيدِ.
فَقَالَ: ”وَكَيْفَ رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ الدَّجَاجَةَ؟“ قَالَ: ”كَانَتْ عَجَبًا مِنَ الْعَجَبِ!“ فَيَقُولُ:

« وَتَدْرِي مَا جَنَسُهَا؟ وَتَدْرِي مَا سِنُّهَا؟ فَإِنَّ الدَّجَاجَةَ إِنَّمَا تَطِيبُ بِالْجَنَسِ وَالسِّنِّ. وَتَدْرِي بِأَيِّ شَيْءٍ كُنَّا نَسَمِّيُهَا؟ وَفِي أَيِّ
مَكَانٍ كُنَّا نَعْلِفُهَا؟ »

فَلَا يَزَالُ فِي هَذَا وَالْآخَرِ يَضْحَكُ ضَحْكًا نَعْرِفُهُ نَحْنُ وَلَا يَعْرِفُهُ أَبُو الْهَذِيلِ.

10.2.1 Lists

Lists environments are also accepted inside the `arab` environment. One may either use any of the three standard list environments, viz. `itemize`, `enumerate` and `description` or use 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:—

```

1 \setRL\paragraph{\arb[novoc]{rumUzi 'l-kitAbi}}\setLR
2 \begin{arab}[novoc]
3 \begin{description}
4 \item[b] max.tU.tu 'l-maktabaTi 'l-'ahliyyaTi bi-\cap{bArIs} 2860
5 `arabiyyuN.
6 \item[s] max.tU.tu 'l-maktabaTi 'l-'ahliyyaTi bi-\cap{bArIs} 2859
7 `arabiyyuN.
8 \item[m] max.tU.tu majlisi \arb[novoc]{~sUrAY malY} .tahrAna 521.
9 \end{description}
10 \end{arab}

```

رموز الكتاب

ب مخطوط المكتبة الأهلية بباريس ٢٨٦٠ عربي.

س مخطوط المكتبة الأهلية بباريس ٢٨٥٩ عربي.

م مخطوط مجلس شوراى ملي طهران ٥٢١.

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

```

1 \setRL\centerline{\arb{\textbf{al-qAnUnu fI 'l-.tibbi}}}\setLR
2 \begin{arab}
3   \begin{itemize}
4     \item \textbf{al-fannu 'l-'awwalu} fI .haddi 'l-.tibbi
5       wa-maw.dU`Ati-hi mina 'l-'umUri 'l-.tabI`iyyaTi wa-ya`stamilu
6       `al_A sittaTi ta`AlImiN
7     \begin{itemize}
8       \item \textbf{al-ta`lImu 'l-'awwalu} [wa-huwa fa.slAni]
9       \begin{itemize}
10        \item \textbf{al-fa.slu 'l-'awwalu}
11        \end{itemize}
12      \end{itemize}
13    \end{itemize}
14  \end{arab}

```

الْقَانُونُ فِي الطِّبِّ
 - الْفَنُّ الْأَوَّلُ فِي حَدِّ الطِّبِّ وَمَوْضُوعَاتِهِ مِنَ الْأُمُورِ الطَّبِيعِيَّةِ وَشَتَّى عَلَى سِتَّةِ تَعَالِيمٍ
 - التَّعْلِيمُ الْأَوَّلُ [وَهُوَ فَصْلَانِ]
 - الْفَصْلُ الْأَوَّلُ

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

```

1 \usepackage[french]{babel}
2 \frenchsetup{StandardLists=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.

10.3 csquotes

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

```

1 \usepackage{csquotes}
2 \DeclareQuoteStyle{arabic}
3 {\rmfamily\textquotedblright}{\rmfamily\textquotedblleft}
4 {\rmfamily\textquoteright}{\rmfamily\textquoteleft}

```

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

```

1 \setquotestyle{arabic}
2 \begin{arab}
3   fa-qAla la-hu ju.hA: \enquote{\garIb-uN 'amru-ka yA .sadiqI
4     'a-tu.saddiqu 'l-.himAr-a wa-tuka_d_diba-nI?}
5 \end{arab}
6 \setquotestyle{english}

```

فَقَالَ لَهُ جُحَا: “غَرِيبُ أَمْرُكَ يَا صَدِيقِي أَتَصَدِّقُ الْجَمَارَ وَتَكْذِبُنِي؟”

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.

10.4 Two-argument commands

Color The two-argument command `\textcolor{<color>}{<Arabic text>}` is supported inside `\begin{arab} ... \end{arab}`. One simple example follows:—

```

1 \begin{arab}
2   \textcolor{red}{\cap{m}uha_d_dabu \cap{'l-d}Ini \cap{'a}bdu
3     \cap{'l-r}a.hImi bnu \cap{'a}liyyiN} huwa ^say_hu-nA 'l-'imAmu
4     'l-.sadru 'l-kabIru 'l-'Alimu 'l-fA.dilu \cap{m}uha_d_dabu
5     \cap{'l-d}Ini \cap{'a}bU \cap{m}u.hammadIN \cap{'a}bdu
6     \cap{'l-r}a.hImi bnu \cap{'a}liyyi bni \cap{.h}AmidiN wa-yu`rafu
7     bi-\cap{'l-d}a_hwari.
8 \end{arab}
9 \begin{arab}[trans]
10  \textcolor{red}{\cap{m}uha_d_dabu \cap{'l-d}Ini \cap{'a}bdu
11    \cap{'l-r}a.hImi bnu \cap{'a}liyyiN} huwa ^say_hu-nA 'l-'imAmu
12    'l-.sadru 'l-kabIru 'l-'Alimu 'l-fA.dilu \cap{m}uha_d_dabu
13    \cap{'l-d}Ini \cap{'a}bU \cap{m}u.hammadIN \cap{'a}bdu
14    \cap{'l-r}a.hImi bnu \cap{'a}liyyi bni \cap{.h}AmidiN wa-yu`rafu
15    bi-\cap{'l-d}a_hwari.
16 \end{arab}

```

مُهَذَّبُ الدِّينِ عَبْدُ الرَّحِيمِ بْنُ عَلِيٍّ هُوَ شَيْخُنَا الْإِمَامُ الصَّدْرُ الْكَبِيرُ الْعَالِمُ الْفَاضِلُ مُهَذَّبُ الدِّينِ أَبُو مُحَمَّدٍ عَبْدُ الرَّحِيمِ بْنُ عَلِيٍّ بْنِ حَامِدٍ وَيُعرفُ بِالذَّخْوَرِ.

Muḥaddabu 'd-Dīni 'Abdu 'r-Raḥīmi bnu 'Aliyyⁱⁿ huwa šayḥu-na 'l-'imāmu 'ṣ-ṣadru 'l-kabīru 'l-'ālimu 'l-fāḍilu Muḥaddabu 'd-Dīni 'Abū Muḥammadⁱⁿ 'Abdu 'r-Raḥīmi bnu 'Aliyyi bni Ḥāmidⁱⁿ wa-yu'rafu bi-'d-Daḥwari.

reledmac The two-argument 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:—

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

10.5 quran

`arabluatex` is compatible with the `quran` package so that both can be used in conjunction with one another for typesetting the *Qur'ān*. As `quran` draws the text of the *Qur'ān* 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 *Qur'ānic* text (see above section 9 on page 37 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:—

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

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

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

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ ﴿١﴾ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ ﴿٢﴾ الرَّحْمَنِ الرَّحِيمِ ﴿٣﴾ مَالِكِ يَوْمِ الدِّينِ ﴿٤﴾
إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ ﴿٥﴾ اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ ﴿٦﴾ صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ
عَلَيْهِمْ وَلَا الضَّالِّينَ ﴿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 25.

12 Implementation

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

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{arabluatex}%
3 [2017/07/05 v1.9 An ArabTeX-like interface for LuaLaTeX]
4 \RequirePackage{ifluatex}
```

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

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

Declare the global options, and define them:

```
12 \DeclareOption{voc}{\def\al@mode{voc}}
13 \DeclareOption{fullvoc}{\def\al@mode{fullvoc}}
14 \DeclareOption{novoc}{\def\al@mode{novoc}}
15 \DeclareOption{trans}{\def\al@mode{trans}}
16 \ExecuteOptions{voc}
17 \ProcessOptions\relax
18 \def\al@mode@voc{voc}
19 \def\al@mode@fullvoc{fullvoc}
20 \def\al@mode@novoc{novoc}
21 \def\al@mode@trans{trans}
```

Packages that are required by `arabluatex`:

```
22 \RequirePackage{etoolbox}
23 \RequirePackage{arabluatex-patch}
```

```

24 \RequirePackage{fontspec}
25 \RequirePackage{amsmath}
26 \RequirePackage{luacode}
27 \RequirePackage{xparse}
28 \RequirePackage{environ}
29 \RequirePackage{adjustbox}
30 \RequirePackage{xkeyval}

```

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

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

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

```
32 \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.

```
33 \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`:

```

34 \AtBeginDocument{\ifdefined\arabicfont\relax\else
35 \PackageWarning{arabluatex}{\string\arabicfont\ is not defined.^^JI
36 will try to load Amiri}%
37 \newfontfamily\arabicfont[Script=Arabic]{Amiri}\fi}%

```

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

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

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

```
39 \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:

```

40 \AtBeginDocument{\ifdef{\LR}%
41 {\RenewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\rmfamily#1\egroup}}
42 {\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.

```

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

```

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

```

46 \DeclareDocumentCommand{\MkArbBreak}{m}{%
47 \luadirect{mkarbbreak(\luastringN{#1})}}

```

<code>\aemph</code>	Arabic emphasis. Needs to be redefined as well. <pre> 48 \AtBeginDocument{\ifdef{\aemph}% 49 {\RenewDocumentCommand{\aemph}{m}{\overline{\text{#1}}}} 50 {\NewDocumentCommand{\aemph}{m}{\overline{\text{#1}}}}} </pre>
<code>\SetInputScheme</code>	arabluatex is designed for processing Arab _T _E X input notation. <code>\SetInputScheme</code> 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’. <pre> 51 \def\al@input@scheme{arabtex} 52 \NewDocumentCommand{\SetInputScheme}{m}{\def\al@input@scheme{#1}} </pre>
<code>\SetArbEasy</code> <code>\SetArbEasy*</code> <code>\SetArbDflt</code>	By default, arabluatex applies complex rules to generate euphonic <i>tašdīd</i> , <i>ʿalif mamdūdah</i> and <i>sukūn</i> depending on the modes which are selected, either <code>voc</code> , <code>fullvoc</code> or <code>trans</code> . Such refinements can be discarded with <code>\SetArbEasy</code> , either globally in the preamble or at any point of the document. Note that <code>\SetArbEasy</code> keeps the <i>sukūn</i> that is generated, while the starred version <code>\SetArbEasy*</code> takes it away. Default complex rules can be set back at any point of the document with <code>\SetArbDflt</code> .
<code>\SetArbDflt*</code>	As of v1.6, arabluatex does not applies any more the assimilation rules that are laid on item b on page 16 ; a new starred version <code>\SetArbDflt*</code> is now available to the user should he wish to apply them. <pre> 53 \def\al@arb@rules{dflt} 54 \NewDocumentCommand{\SetArbEasy}{s}{% 55 \IfBooleanTF{#1} 56 {\def\al@arb@rules{easynosukun}} 57 {\def\al@arb@rules{easy}}} 58 \NewDocumentCommand{\SetArbDflt}{s}{% 59 \IfBooleanTF{#1} 60 {\def\al@arb@rules{idgham}} 61 {\def\al@arb@rules{dflt}}} </pre>
<code>\SetTranslitFont</code>	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 <code>fontspec</code> package. <pre> 62 \def\al@trans@font{\rmfamily}% 63 \NewDocumentCommand{\SetTranslitFont}{m}{\def\al@trans@font{#1}} </pre>
<code>\SetTranslitStyle</code>	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: <pre> 64 \def\al@trans@style{\itshape}% 65 \NewDocumentCommand{\SetTranslitStyle}{m}{\def\al@trans@style{#1}} </pre>
<code>\SetTranslitConvention</code>	<code>\SetTranslitConvention{<i>convention</i>}</code> may be used to change the transliteration convention, which is <code>dmg</code> by default: <pre> 66 \def\al@trans@convention{dmg} 67 \NewDocumentCommand{\SetTranslitConvention}{m}{\def\al@trans@convention{#1}} </pre>

`\arbup` 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 dmḡ 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>}`.

```

68 \NewDocumentCommand{\al@arbup@dflt}{m}{\textsuperscript{\thinspace#1}}%
69 \NewDocumentCommand{\al@arbup}{m}{\al@arbup@dflt{#1}}
70 \NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
71 \NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt}
72 \NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\al@arbup}{m}{##1}}
73 \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.

```

74 \DeclareDocumentCommand{\cap}{m}%
75   {\luairect{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.

```

76 \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.

```

77 \DeclareDocumentCommand{\txarb}{+m}{\bgroup\textdir
78   TRT\arabicfont#1\egroup}
79 \DeclareDocumentCommand{\txtrans}{+m}{\bgroup\textdir
80   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.

```

81 \NewDocumentEnvironment{txarab}{}{%
82   \par%
83   \booltrue{al@rlmode}%
84   \pdir 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.


```

85 \DeclareDocumentCommand{\arb}{0{\al@mode} +m}%
86 {\edef\@tempa{#1}%
87   \ifx\@tempa\al@mode@voc%
88     \bgroup\textdir TRT\arabicfont%
89     \luadirect{tex.sprint(processvoc(\luastringN{#2},
90       \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
91   \else%
92     \ifx\@tempa\al@mode@fullvoc%
93       \bgroup\textdir TRT\arabicfont%
94       \luadirect{tex.sprint(processfullvoc(\luastringN{#2},
95         \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
96     \else%
97       \ifx\@tempa\al@mode@novoc%
98         \bgroup\textdir TRT\arabicfont%
99         \luadirect{tex.sprint(processnovoc(\luastringN{#2},
100           \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
101       \else%
102         \ifx\@tempa\al@mode@trans%
103           \bgroup\textdir TLT\al@trans@style%
104           \luadirect{tex.sprint(processtrans(\luastringN{#2},
105             \luastring0{\al@trans@convention},
106             \luastring0{\al@arb@rules},
107             \luastring0{\al@input@scheme}}))\egroup%
108         \else%
109           \fi\fi\fi\fi}

```

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

```

110 \NewEnviron{arab}[1][\al@mode]%
111 {\par\edef\@tempa{#1}%
112   \ifx\@tempa\al@mode@voc%
113     \booltrue{al@rlmode}%
114     \bgroup\pardir TRT\textdir TRT\arabicfont%
115     \luadirect{tex.sprint(processvoc(\luastring0{\BODY},
116       \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
117   \else%
118     \ifx\@tempa\al@mode@fullvoc%
119       \booltrue{al@rlmode}%
120       \bgroup\pardir TRT\textdir TRT\arabicfont%
121       \luadirect{tex.sprint(processfullvoc(\luastring0{\BODY},
122         \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
123     \else%
124       \ifx\@tempa\al@mode@novoc%
125         \booltrue{al@rlmode}%
126         \bgroup\pardir TRT\textdir TRT\arabicfont%
127         \luadirect{tex.sprint(processnovoc(\luastring0{\BODY},
128           \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
129       \else%
130         \ifx\@tempa\al@mode@trans%
131           \bgroup\pardir TLT\textdir TLT\al@trans@style%

```

```

132 \luadirect{tex.sprint(processtrans(\luastring0{\BODY},
133 \luastring0{\al@trans@convention},
134 \luastring0{\al@arb@rules},
135 \luastring0{\al@input@scheme}))}\egroup%
136 \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:

```

137 \newlength{\al@bayt@width}
138 \newlength{\al@gutter@width}
139 \setlength{\al@bayt@width}{.3\textwidth}
140 \setlength{\al@gutter@width}{.15\al@bayt@width}
141 \define@key[al]{verse}{width}{\setlength{\al@bayt@width}{#1}}
142 \define@key[al]{verse}{gutter}{\setlength{\al@gutter@width}{#1}}
143 \define@key[al]{verse}{metre}{\arb{#1}}
144 \define@boolkey[al]{verse}{utf}[true]{}
145 \define@boolkey[al]{verse}{delim}[true]{}
146 \define@choicekey[al]{verse}{mode}{fullvoc, voc, novoc,
147 trans}{\def\al@mode{#1}}
148 \presetkeys[al]{verse}{metre=LR{\vskip -\baselineskip}, utf=false,
149 delim=false}{}

```

Then follows the environment itself:

```

150 \NewDocumentEnvironment{arabverse}{0{}}%
151 {\par\centering\noindent\bgroup\setkeys[al]{verse}{metre}{#1}%
152 \ifx\al@mode\al@mode@trans%
153 \ifal@verse@utf\setRL\else\setLR\fi%
154 \else\setRL\fi}%
155 {\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 *ʿağuz*. 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.

```

156 \NewDocumentCommand{\arb@utf}{m}{%
157 \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
158 \def\al@hemistich@delim{*}
159 \NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
160 \def\al@verse@stroke{\leavevmode\xleaders\hbox{\arb{--}}\hfill\kern0pt}
161 \NewDocumentCommand{\bayt}{m o m}{%
162 \ifdefined\savenotes\savenotes\else\fi%
163 \edef\al@tatweel{--}%
164 \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#1}}%

```

```

165 \IfNoValueTF{#2}{%
166   \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
167   \else%
168   \hspace{\al@gutter@width}%
169   \fi
170 }{%
171   \edef\@tempa{#2}%
172   \ifx\@tempa\al@tatweel%
173   \ifx\al@mode\al@mode@trans%
174   \hspace{\al@gutter@width}%
175   \else%
176   \makebox[\al@gutter@width][s]{\al@verse@stroke}%
177   \fi%
178   \else%
179   \ifx\al@mode\al@mode@trans%
180   \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#2}}%
181   \else%
182   \makebox[\al@gutter@width][s]{\arb@utf{#2}}%
183   \fi\fi}%
184 \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
185 \ifdefined\spewnotes\spewnotes\else\fi%
186 }

```

`\abjad` `\abjad{⟨number⟩}` expresses its argument in Arabic letters in accordance with the *abjad* 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.

```

187 \AtBeginDocument{%
188   \ifdefined\abjad%
189   \RenewDocumentCommand{\abjad}{m}%
190   {\luairect{tex.sprint(abjadify(#1))}}%
191   \else%
192   \NewDocumentCommand{\abjad}{m}%
193   {\luairect{tex.sprint(abjadify(#1))}}
194 \fi}

```

`\arbnul` The `\arbnul` command does nothing by itself. It is processed only if it is found in Arabic context so as to put back on contextual analysis in case it has been broken by other commands.

```

195 \NewDocumentCommand{\arbnul}{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`.

```

196 \NewDocumentCommand{\abraces}{+m}{%
197   \luairect{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.

```

198 \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.

```

199 \DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pdir
200   TLT\LR{\footnote{#1}}\egroup}
201 \DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pdir
202   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.

```

203 \NewDocumentCommand{\FixArbFtnmk}{}{%
204   \@ifpackageloaded{scrextend}%
205   {\AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%
206   {\RequirePackage{scrextend}
207    \AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%

```

That is it. Say goodbye before leaving.

Patches

```

208 \NeedsTeXFormat{LaTeX2e}
209 \ProvidesPackage{arabluatex-patch}%
210 [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:

```

211 \def\list#1#2{%
212   \ifnum \@listdepth >5\relax
213     \@toodeep
214   \else
215     \global\advance\@listdepth\@ne
216   \fi
217   \rightmargin\z@
218   \listparindent\z@
219   \itemindent\z@
220   \csname @list\romannumeral\the\@listdepth\endcsname
221   \def\@itemlabel{#1}%
222   \let\makelabel\@mklab
223   \nmblistfalse
224   #2\relax
225   \@trivlist
226   \parskip\parsep
227   \parindent\listparindent

```

```

228 \advance\linewidth -\rightmargin
229 \advance\linewidth -\leftmargin
patch begins:
230 \ifbool{al@rlmode}{\advance\@totalleftmargin \rightmargin}%
231 {\advance\@totalleftmargin \leftmargin}
patch ends.
232 \parshape \@ne \@totalleftmargin \linewidth
233 \ignorespaces}
234 \def\@item[#1]{%
235 \if@noperitem
236 \@donoperitem
237 \else
238 \if@inlabel
239 \indent \par
240 \fi
241 \ifhmode
242 \unskip\unskip \par
243 \fi
244 \if@newlist
245 \if@nobreak
246 \@nbitem
247 \else
248 \addpenalty\@beginparpenalty
249 \addvspace\@topsep
250 \addvspace{-\parskip}%
251 \fi
252 \else
253 \addpenalty\@itempenalty
254 \addvspace\itemsep
255 \fi
256 \global\@inlabeltrue
257 \fi
258 \everypar{%
259 \@minipagefalse
260 \global\@newlistfalse
261 \if@inlabel
262 \global\@inlabelfalse
263 {\setbox\z@\lastbox
264 \ifvoid\z@
265 \kern-\itemindent
266 \fi}%
267 \box\@labels
268 \penalty\z@
269 \fi
270 \if@nobreak
271 \@nobreakfalse
272 \clubpenalty \@M
273 \else
274 \clubpenalty \@clubpenalty

```

```

275     \everypar{%
276     \fi}%
277 \if@noitemarg
278     \@noitemargfalse
279     \if@nmbrrlist
280         \refstepcounter\@listctr
281     \fi
282 \fi

patch begins:
283 \ifbool{al@rlmode}{\sRLbox\@tempboxa{\makelabel{#1}}}{%
284 \sbox\@tempboxa{\makelabel{#1}}}%
285 \ifbool{al@rlmode}{\global\setbox\@labels\hbox dir TRT}{%
286 {\global\setbox\@labels\hbox}{%

patch ends.
287     \unhbox\@labels
288     \hskip \itemindent
289     \hskip -\labelwidth
290     \hskip -\labelsep
291     \ifdim \wd\@tempboxa >\labelwidth
292         \box\@tempboxa
293     \else
294         \hbox to\labelwidth {\unhbox\@tempboxa}%
295     \fi
296     \hskip \labelsep}%
297 \ignorespaces}

This is adapted from Vafa Khalighi's bidi package. Thanks to him.
298 \long\def\sRLbox#1#2{\setbox#1\hbox dir TRT{%
299     \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 Transkriptionskommission der Deutschen Morgenländischen Gesellschaft*. In collab. with Ph. S. van Ronkel and Otto Spies. Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockhaus. 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/>.
- DIN 31 635 (2011). *Information and Documentation - Romanization of the Arabic Alphabet for Arabic, Ottoman-Turkish, Persian, Kurdish, Urdu and Pushto*. URL: <http://www.din.de>.

- 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

v1.0	General: Initial release	1	v1.5	General: Compatibility with the quran package	44
v1.0.1	General: Minor update of the documentation	1		Environments may be nested inside the <code>arab</code> environment . .	40
v1.1	<code>\abjad</code> : New and more flexible <code>\abjad</code> command.	51		<code>txarab</code> : New <code>txarab</code> environment for typesetting running paragraphs in unicode Arabic .	48
v1.2	<code>\SetArbEasy</code> : New <code>\SetArbEasy/\SetArbDflt</code> for ‘modern’ or ‘classic’ Arabic styles.	47	v1.6	<code>arabverse</code> : New environment <code>arabverse</code> for typesetting Arabic poetry	50
v1.3	<code>\arbup</code> : <i>ʾrāb</i> is now written as superscript text in <code>dmg</code> mode by default.	48		<code>\bayt</code> : New macro <code>\bayt</code> for typesetting each verse inside the <code>arabverse</code> environment	50
v1.4	<code>\SetInputScheme</code> : <code>\SetInputScheme</code> may be used to process other input schemes such as ‘Buckwalter’	47		<code>\SetArbDflt*</code> : This starred version applies the assimilation rules in addition to what <code>\SetArbDflt</code> already does.	47
	<code>\SetTranslitFont</code> : For selecting a specific font for transliterated texts	47		<code>\SetHemistichDelim</code> : New <code>\SetHemistichDelim</code> command for changing the default delimiter between hemistichs .	50
v1.4.3	<code>\abraces</code> : New <code>\abraces</code> command which expresses its argument between braces.	51	v1.7	<code>\arbnul1</code> : New <code>\arbnul1</code> command for putting back on any contextual analysis rule broken by other commands.	51
v1.4.4	<code>\SetArbEasy*</code> : this starred version discards the <i>sukūn</i> in addition to what is already discarded by <code>\SetArbEasy</code>	47	v1.8	General: <code>arabica</code> transliteration standard is now supported . . .	31
			v1.8.5	General: Six additional Persian characters are now available . .	10

v1.9

`\MkArbBreak`: New `\MkArbBreak`
command for inserting
user-defined macros in Arabic
environments 46

Index

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

Symbols	
<code>\@M</code>	272
<code>\@beginparpenalty</code> .	248
<code>\@clubpenalty</code>	274
<code>\@donoparitem</code>	236
<code>\@ifpackageloaded</code> .	204
<code>\@inlabelfalse</code>	262
<code>\@inlabeltrue</code>	256
<code>\@item</code>	234
<code>\@itemlabel</code>	221
<code>\@itempenalty</code>	253
<code>\@labels</code>	267, 285, 286, 287
<code>\@listctr</code>	280
<code>\@listdepth</code> 212, 215, 220	
<code>\@minipagefalse</code> ...	259
<code>\@mklab</code>	222
<code>\@nbitem</code>	246
<code>\@ne</code>	215, 232
<code>\@newlistfalse</code>	260
<code>\@nmbrlistfalse</code> ...	223
<code>\@nobreakfalse</code>	271
<code>\@noitemargfalse</code> ..	278
<code>\@tempa</code>	86, 87, 92, 97, 102, 111, 112, 118, 124, 130, 171, 172
<code>\@tempboxa</code>	283, 284, 291, 292, 294
<code>\@toodeep</code>	213
<code>\@topsep</code>	249
<code>\@totalleftmargin</code> .	230, 231, 232
<code>\@trivlist</code>	225
<code>_</code>	35
A	
<code>\abjad</code>	24, 187
<code>\abraces</code>	30, 196
<code>\addpenalty</code> ..	248, 253
<code>\addvspace</code> 249, 250, 254	
<code>\adjustbox</code> 164, 180, 184	
<code>\advance</code>	215, 228, 229, 230, 231
<code>\aemph</code>	26, 48
<code>\al@arb@rules</code> 53, 56, 57, 60, 61, 90, 95, 100, 106, 116, 122, 128, 134	
<code>\al@arbup</code>	69, 70, 71, 72, 73
<code>\al@arbup@dflt</code>	68, 69, 71
<code>\al@bayt@width</code>	137, 139, 140, 141, 164, 184
<code>\al@gutter@width</code> ..	138, 140, 142, 166, 168, 174, 176, 180, 182
<code>\al@hemistich@delim</code>	158, 159, 166
<code>\al@input@scheme</code> ..	51, 52, 90, 95, 100, 107, 116, 122, 128, 135
<code>\al@mode</code>	12, 13, 14, 15, 85, 110, 147, 152, 173, 179
<code>\al@mode@fullvoc</code> ..	19, 92, 118
<code>\al@mode@novoc</code>	20, 97, 124
<code>\al@mode@trans</code>	21, 102, 130, 152, 173, 179
<code>\al@mode@voc</code> 18, 87, 112	
<code>\al@tatweel</code> ..	163, 172
<code>\al@trans@convention</code>	66, 67, 105, 133
<code>\al@trans@font</code>	62, 63, 80
<code>\al@trans@style</code> ...	64, 65, 103, 131
<code>\al@verse@stroke</code> ..	160, 176
<i>Amiri</i>	5
<i>amiri</i> (package)	5
<i>arab</i> (environment) 7, 110	
<i>ArabTeX</i>	3, 20
<i>Arabic-English Lexicon</i> , <i>An</i>	19
<code>\arabicfont</code>	34, 35, 37, 78, 84, 88, 93, 98, 114, 120, 126
<i>arabluatex</i> (package) .	2, 4– 8, 10–13, 15– 20, 25, 26, 28, 30–39, 42, 44–48

arabtex (package) ...
... 3, 4, 16, 20, 45
arabulatex (package) .. 17
arabverse (environ-
ment) ... 26, 137
arabxetex (package) . 4, 16
\arb 7, 85, 143, 157, 160
\arb@utf 156,
164, 180, 182, 184
\arbnul 23, 195
\arbus 32, 68
\ArbUpDflt 32, 68

B
babel (package) 42
babel-french (package) . 42
\baselineskip 148
\bayt 27, 156
bidi (package) 54
\BODY 115, 121, 127, 132
\boolfalse 39
\booltrue 38,
83, 113, 119, 125
\box 267, 292
Brockelmann, Carl ...
..... 10, 31, 32

C
\Cap 76
\cap 32, 74, 76
\centering 151
\clubpenalty .. 272, 274
\color@endgroup ... 299
\color@setgroup ... 299
csquotes (package) ... 42

D
\DeclareOption
.... 12, 13, 14, 15
\deffootnote .. 205, 207
\define@boolkey 144, 145
\define@choicekey . 146
\define@key 141, 142, 143
delim (option) 26

E
environments:
arab 7, 110
arabverse .. 26, 137
txarab 37, 81

\everypar 258, 275
\ExecuteOptions 16

F
\FixArbFtnmk ... 38, 203
fontspec (package) ...
..... 5, 31, 47
footnote (package) 28
\footnote 200, 202
footnotehyper (package) 28
fullvoc (option) 5, 8

G
\global ... 215, 256,
260, 262, 285, 286
Grammar of the Arabic
Language, A ..
..... 5–7, 9,
13–16, 18, 24, 25
gutter (option) 26

H
Habash, Nizar Y. 35
\hbox 160,
285, 286, 294, 298
\Height .. 164, 180, 184
\hfill 155, 160
Hosny, Khaled 5
\hskip 288, 289, 290, 296
\hspace 168, 174

I
\if@inlabel .. 238, 261
\if@newlist 244
\if@nmbrrlist 279
\if@nobreak .. 245, 270
\if@noitemarg 277
\if@noparitem 235
\ifal@verse@delim . 166
\ifal@verse@utf 153, 157
\ifbool .. 230, 283, 285
\IfBooleanTF 55, 59
\ifdim 291
\ifhmode 241
\ifluatex 5
\IfNoValueTF 165
\ifnum 212
\ifvoid 264
\ignorespaces . 233, 297
\indent 239

*Information and Docu-
mentation - Ro-
manization of
the Arabic Al-
phabet for Ara-
bic, Ottoman-
Turkish, Per-
sian, Kur-
dish, Urdu and
Pushko* 10

*Introduction to Arabic
Natural Lan-
guage Process-
ing* 35

\itemindent 219, 265, 288
\itemsep 254
\itshape 64

K
\kern 160, 265
KOMA-script (package) 39

L
\labelsep 290, 296
\labelwidth 289, 291, 294
Lagally, Klaus 3, 20
Lane, Edward William . 19
\lastbox 263
\leavevmode 160
\leftmargin .. 229, 231
lineno (package) 29
\linewidth 228, 229, 232
\list 211
\listparindent 218, 227
\long 298
\LR 38, 40, 148,
200, 202, 205, 207
\LRfootnote ... 38, 199
\LRmarginpar ... 39, 198
luabidi (package) ... 4, 46
\luadirect
32, 33, 47, 75,
89, 94, 99, 104,
115, 121, 127,
132, 190, 193, 197
luaotfload (package) ... 31
\luastringN . 47, 75,
89, 94, 99, 104, 197

<code>\luastring0</code>	<code>\parskip</code> 226, 250	<code>\SetTranslitStyle</code> 31, 64
.. 90, 95, 100,	<code>\penalty</code> 268	<code>\spewnotes</code> 185
105, 106, 107,	<code>polyglossia</code> (package) .	<code>\sRLbox</code> 283, 298
115, 116, 121, 4, 46, 51	
122, 127, 128,	<code>\presetkeys</code> 148	T
132, 133, 134, 135	<code>\ProcessOptions</code> 17	<code>\text</code> 49, 50
	<code>\ProvidesPackage</code> 2, 209	<code>\textdir</code> . 38, 39, 41,
M		42, 44, 45, 77,
<code>\makebox</code> . 166, 176, 182	Q	79, 84, 88, 93,
<code>\makelabel</code> 222, 283, 284	<code>quoting</code> (package) 40	98, 103, 114,
<code>\marginpar</code> 198	<code>quran</code> (package) 44	120, 126, 131, 198
<code>metre</code> (option) 26		<code>\textsuperscript</code> ... 68
<code>\MkArbBreak</code> 39, 46	R	<code>\textwidth</code> 139
<code>mode</code> (option) 26	<code>\refstepcounter</code> ... 280	<code>\the</code> 220
	<code>reledmac</code> (package) ... 44	<code>\thefootnotemark</code> ..
N	<code>\RequirePackage</code> 205, 207
<code>\newbool</code> 31 4, 22,	<code>\thinspace</code> 68
<code>\NewDocumentEnvironment</code>	23, 24, 25, 26,	<code>trans</code> (option) 5, 8
..... 81, 150	27, 28, 29, 30, 206	<i>Transliteration der ara-</i>
<code>\NewEnviron</code> 110	<code>\rightmargin</code>	<i>bischen Schrift,</i>
<code>\newfontfamily</code> 37 217, 228, 230	<i>Die</i> ... 10, 31, 32
<code>\newlength</code> ... 137, 138	<code>\RL</code> 38, 43	<code>txarab</code> (environment) .. 44
<code>\NoArbUp</code> 32, 68	<code>\RLfootnote</code> ... 38, 199	<code>txarab</code> (environment)
<code>\noindent</code> 151	<code>\rmfamily</code> 37, 81
<code>novoc</code> (option) 5, 8	. 41, 42, 44, 45, 62	<code>\txarb</code> 37, 77, 157
	<code>\romannumeral</code> 220	<code>\txtrans</code> 77
O		
options:	S	U
<code>delim</code> 26	<code>\savenotes</code> 162	<code>\unhbox</code> 287, 294
<code>fullvoc</code> 5, 8	<code>\sbox</code> 284	<code>\unskip</code> 242
<code>gutter</code> 26	<code>scrextend</code> (package) 39, 52	<code>utf</code> (option) 26
<code>metre</code> 26	<code>\SetArbDflt</code> 6, 53	
<code>mode</code> 26	<code>\SetArbDflt*</code> 6, 53	V
<code>novoc</code> 5, 8	<code>\SetArbEasy</code> 6, 53	<code>voc</code> (option) 5, 8
<code>trans</code> 5, 8	<code>\SetArbEasy*</code> 6, 53	<code>\vskip</code> 148
<code>utf</code> 26	<code>\SetArbUp</code> 33, 68	
<code>voc</code> 5, 8	<code>\setbox</code>	W
<code>width</code> 26	263, 285, 286, 298	<code>\wd</code> 291
<code>\overline</code> 49, 50	<code>\SetHemistichDelim</code>	<code>width</code> (option) 26
 26, 156, 159	<code>Wright, W. LL.D</code> ...
P	<code>\SetInputScheme</code> . 34, 51 5–7, 9,
<code>\par</code> 82, 84, 111,	<code>\setkeys</code> 151, 155	13–16, 18, 24, 25
136, 151, 239, 242	<code>\setlength</code>	
<code>paralist</code> (package) . 41, 42	139, 140, 141, 142	X
<code>\pardir</code> 38, 39,	<code>\setLR</code> 39, 39, 153	<code>\xleaders</code> 160
84, 114, 120,	<code>\setRL</code> . 38, 39, 153, 154	
126, 131, 199, 201	<code>\SetTranslitConvention</code>	Z
<code>\parindent</code> 227 31, 66	<code>\z@</code> 217, 218,
<code>\parsep</code> 226	<code>\SetTranslitFont</code> 31, 62	219, 263, 264, 268
<code>\parshape</code> 232		