

# The arabluatex package v1.10.3 – 2018/02/12

Robert Alessi  
[alessi@robertalessi.net](mailto:alessi@robertalessi.net)\*  
CNRS UMR 8167 Paris (France)

## Contents

<b>License and disclaimer</b>	<b>2</b>	4.8.2 The <i>abjad</i> . . . . .	25
<b>1 Introduction</b>	<b>3</b>	4.9 Additional characters . . . . .	26
1.1 arabluatex is for Lua <sup>®</sup> TeX . . . . .	4	4.10 Arabic emphasis . . . . .	26
<b>2 The basics of arabluatex</b>	<b>4</b>	<b>5 Arabic poetry</b>	<b>27</b>
2.1 Activating arabluatex . . . . .	4	5.1 Example . . . . .	30
2.2 Options . . . . .	5	<b>6 Special applications</b>	<b>31</b>
2.2.1 Classic contrasted with modern typesetting of Arabic	6	<b>7 Transliteration</b>	<b>32</b>
2.3 Typing Arabic . . . . .	7	7.1 Additional note on <b>dmg</b> convention .	34
2.3.1 Local options . . . . .	8	7.2 Examples . . . . .	36
<b>3 Standard ArabTeX input</b>	<b>9</b>	<b>8 Buckwalter input scheme</b>	<b>36</b>
3.1 Consonants . . . . .	9	<b>9 Unicode Arabic input</b>	<b>39</b>
3.2 Additional characters . . . . .	10	<b>10 L<sup>A</sup>T<sub>E</sub>X Commands in Arabic environments</b>	<b>39</b>
3.3 Vowels . . . . .	11	10.1 New commands . . . . .	41
3.3.1 Long vowels . . . . .	11	10.2 Environments . . . . .	42
3.3.2 Short vowels . . . . .	11	10.2.1 Lists . . . . .	43
<b>4 arabluatex in action</b>	<b>12</b>	10.3 csquotes . . . . .	45
4.1 The vowels and diphthongs . . . . .	12	10.4 Two-argument commands . . . . .	45
4.2 Other orthographic signs . . . . .	14	10.5 quran . . . . .	46
4.3 Special orthographies . . . . .	20	<b>11 Future work</b>	<b>47</b>
4.4 Quoting . . . . .	21	<b>12 Implementation</b>	<b>47</b>
4.4.1 Quoting the <i>hamzah</i> . . . . .	22	<b>References</b>	<b>57</b>
4.5 The “pipe” character ( ) . . . . .	23	<b>Change History</b>	<b>57</b>
4.6 Putting back on broken contextual analysis rules . . . . .	23	<b>Index</b>	<b>58</b>
4.7 Stretching characters: the <i>taṭwīl</i> .	25		
4.8 Digits . . . . .	25		
4.8.1 Numerical figures . . . . .	25		

---

\*Institutional email: [robert.alessi@cnrs.fr](mailto:robert.alessi@cnrs.fr)

## List of Tables

1	ArabTeX consonants . . . . .	9	5	“Quoted” <i>hamzah</i> . . . . .	23
2	ArabTeX additional characters . .	10	6	Additional Arabic codings . . . . .	26
3	ArabTeX long vowels . . . . .	11	7	Buckwalter scheme . . . . .	37
4	ArabTeX short vowels . . . . .	11			

### Abstract

This package provides for Lua<sup>A</sup>TeX 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<sup>A</sup>TeX.

Copyright © 2016–2018 Robert Alessi

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

- email: <[alessi@robertalessi.net](mailto: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 © 2016–2018 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,<sup>1</sup> 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<sup>A</sup>TeX 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<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X. 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”.<sup>2</sup> 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<sup>A</sup>T<sub>E</sub>X 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).<sup>3</sup> 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<sub>E</sub>X “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

<sup>1</sup> See <http://ctan.org/pkg/arabtex>

<sup>2</sup> Lagally (2004, p. 2).

<sup>3</sup> 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<sup>L</sup>A<sup>T</sup>E<sup>X</sup>

It goes without saying that `arabluatex` requires Lua<sup>L</sup>A<sup>T</sup>E<sup>X</sup>. T<sup>E</sup>X and L<sup>A</sup>T<sup>E</sup>X have `arabxetex`, and X<sub>Y</sub>L<sup>A</sup>T<sup>E</sup>X has `arabxetex`. Both of them are much more advanced than `arabluatex`, as they can process a number of different languages,<sup>4</sup> 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<sup>E</sup>X input on the fly into Unicode Arabic script, whereas the latter passes ArabT<sup>E</sup>X input on to a set of Lua functions. At first, L<sup>A</sup>T<sup>E</sup>X 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<sup>E</sup>X the converted strings—which may in turn contain some other ArabT<sup>E</sup>X 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<sup>L</sup>A<sup>T</sup>E<sup>X</sup>; 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.

---

<sup>4</sup>To 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 font<sup>5</sup> like so:—

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

REM. By default Amiri places the *kasrah* in combination with the *tašdīd* below the consonant, like so: `كـ`. That is correct, as at least in the oldest manuscripts `كـ` may stand for `كـ` as well as `كـ`. 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:—<sup>6</sup>

```
\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 21).

`trans`

This mode transliterates the ArabT<sub>E</sub>X input into one of the accepted standards. At present, three standards are supported (see below section 7 on page 32 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`.

<sup>5</sup>Hosny (2015).

<sup>6</sup>See the documentation of `amiri`, Hosny (2015, p. 5).

New feature  
v.1.2

## 2.2.1 Classic contrasted with modern typesetting of Arabic

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 (ل), then a mere *hamzah* is written on the line, and a *maddah* is placed over the ‘*alif*’, like so:—

samA'uN سَمَاءُ *samā<sup>un</sup>*, jA'a جَاءَ *ǧā'a*, yatasA'alUna يَتَسَاءَلُونَ *yatasā'alūna*<sup>7</sup>  
(see on page 16 for further details).

- The euphonic *tašdīd* is generated (see on page 16).
- In *fullvoc* mode, the *sukūn* is expressed.
- In such words as ظِمًا, شَيْئًا and the like, the *hamzah* alone is not written over the letter *yā* with no diacritical points below as in ظِمًا, شَيْئًا, but over a horizontal stroke placed in the continuation of the preceding letter.

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

New feature  
v1.4.4

`\SetArbEasy`  
`\SetArbEasy*`  
`\SetArbDflt`

Such refinements as ‘classic’ *maddah* may be discarded by the `\SetArbEasy` command, either globally in the preamble or locally at any point of the document. The difference between `\SetArbEasy` and its ‘starred’ version `\SetArbEasy*` is that the former keeps the *sukūn* that is generated by the *fullvoc* mode, while the latter further takes it away. Default ‘classic’ rules may be set back at any point of the document with the `\SetArbDflt` command. Assimilation rules laid on item b on page 17 may also be applied by the ‘starred’ version of this command `\SetArbDflt*` either in the preamble or at any point of the document.<sup>8</sup> Examples follow:—

(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<sup>an</sup> qabla 'an yutimma kitāba-hu fī nuǧūm<sup>i</sup> 's-samā<sup>i</sup>*

(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<sup>an</sup> qabla 'ay yutimma kitāba-hu fī nuǧūm<sup>i</sup> 's-samā<sup>i</sup>*

(c) `\SetArbEasy`:

- voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ

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

<sup>8</sup>For an example, see section 5.1 on page 30.

- iii. trans *wa-māta 'stisqā<sup>'an</sup> qabla 'an yutimma kitāba-hu fī nuḡūm<sup>i</sup> 's-samā<sup>i</sup>*
- (d) \SetArbEasy\*:
  - i. voc وَمَاتَ اسْتَسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
  - ii. fullvoc وَمَاتَ اسْتَسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
  - iii. trans *wa-māta 'stisqā<sup>'an</sup> qabla 'an yutimma kitāba-hu fī nuḡūm<sup>i</sup> 's-samā<sup>i</sup>*

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

## 2.3 Typing Arabic

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

```

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 الْحِجَاءِ, حُرُوفُ الْهَجَاءِ, 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 جِحَا يَطْلُبُ مِنْهُ حِمَارَهُ لِيَرْكَبَهُ فِي سَفَرَةٍ قَصِيرَةٍ فَقَالَ لَهُ: "سَوْفَ أُعِيدُهُ إِلَيْكَ فِي الْمَسَاءِ  
وَأَدْفَعُ لَكَ أُجْرَةً." فَقَالَ جِحَا: "أَنَا آسَفٌ جِدًّا أَنِّي لَا أَسْتَطِيعُ أَنْ أُحَقِّقَ لَكَ رَغْبَتَكَ فَالْحِمَارُ لَيْسَ هُنَا  
الْيَوْمَ." وَقَبْلَ أَنْ يَتِمَّ جِحَا كَلَامَهُ بَدَأَ الْحِمَارُ يَنْهَقُ فِي إِصْطِلِهِ. فَقَالَ لَهُ صَدِيقُهُ: "إِنِّي أَسْمَعُ حِمَارَكَ يَا جِحَا  
يَنْهَقُ." فَقَالَ لَهُ جِحَا: "غَرِيبٌ أَمْرُكَ 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 argument, like so:—

voc — `\arb[voc]{Arabic text}`;



`fullvoc`    – `\arb[fullvoc]{\langle Arabic text \rangle};`  
`novoc`     – `\arb[novoc]{\langle Arabic text \rangle};`  
`trans`     – `\arb[trans]{\langle Arabic text \rangle}.`

The same optional arguments may be passed to the environment `arab`: one may have `\begin{arab}[\langle mode \rangle] \dots \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 <sup>9</sup>			ArabTeX notation
	dmg	loc	arabica	
أ <sup>10</sup>	'u 'a 'i	'u, 'a, 'i	'u, 'a, 'i	'u or 'a or 'i
ب	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>
ت	<i>t</i>	<i>t</i>	<i>t</i>	<i>t</i>
ث	<i>ṭ</i>	<i>th</i>	<i>ṭ</i>	<i>_t</i>
ج	<i>ǧ</i>	<i>j</i>	<i>ǧ</i>	<i>^g</i> or <i>j</i>
ح	<i>ḥ</i>	<i>ḥ</i>	<i>ḥ</i>	<i>.h</i>
خ	<i>ḫ</i>	<i>kh</i>	<i>ḫ</i>	<i>_h</i> or <i>x</i>
د	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>
ذ	<i>ḍ</i>	<i>dh</i>	<i>ḍ</i>	<i>_d</i>
ر	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>
ز	<i>z</i>	<i>z</i>	<i>z</i>	<i>z</i>
س	<i>s</i>	<i>s</i>	<i>s</i>	<i>s</i>
ش	<i>š</i>	<i>sh</i>	<i>š</i>	<i>^s</i>
ص	<i>ṣ</i>	<i>ṣ</i>	<i>ṣ</i>	<i>.s</i>
ض	<i>ḍ</i>	<i>ḍ</i>	<i>ḍ</i>	<i>.d</i>
ط	<i>ṭ</i>	<i>ṭ</i>	<i>ṭ</i>	<i>.t</i>
ظ	<i>ẓ</i>	<i>ẓ</i>	<i>ẓ</i>	<i>.z</i>
ع	<i>ʿ</i>	<i>ʿ</i>	<i>ʿ</i>	<i>`</i>
غ	<i>ǧ</i>	<i>gh</i>	<i>ǧ</i>	<i>.g</i>
ف	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>
ق	<i>q</i>	<i>q</i>	<i>q</i>	<i>q</i>
ك	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>
ل	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>
م	<i>m</i>	<i>m</i>	<i>m</i>	<i>m</i>
ن	<i>n</i>	<i>n</i>	<i>n</i>	<i>n</i>

<sup>9</sup>See below section 7 on page 32.

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

Letter	Transliteration			ArabT <sub>E</sub> X notation
	dmg	loc	arabica	
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 <sup>11</sup>
ة	<i>ah</i>	<i>ah</i>	<i>a</i>	T

Table 1: Standard ArabT<sub>E</sub>X (consonants)

REM. *a*. Please note that in all cases of elision, the *ʿalifu ʾl-waṣli* is expressed only by the vowel that accompanies the omitted *hamzah*: ⟨*u, a, i*⟩ as in *wa-inhazama* <sup>وَأَنهَزَمَ</sup> *wa-ʾnhazama*. For more details on the definite article and the *ʿalifu ʾl-waṣli* see section 4.2 on page 17.

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 *ʿalif* 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, أَلِفَاءُ 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: qaYʾuN قَيَّ qayʾun, ^saYʾuN شَيَّ sayʾun, ^saYʾaN شَيَّان sayʾan.

The same result may be achieved by encoding this letter as .y, like so: qa.yʾuN قَيَّ qayʾun, ^sa.yʾuN شَيَّ sayʾun, ^sa.yʾaN شَيَّان sayʾan.

### 3.2 Additional characters

Table 2 gives the ArabT<sub>E</sub>X equivalents for some additional Persian characters.

Letter	Transliteration <sup>12</sup>			ArabT <sub>E</sub> X notation
	dmg	loc	arabica <sup>13</sup>	
پ	<i>p</i>	<i>p</i>	<i>p</i>	p
چ	<i>č</i>	<i>ch</i>	<i>č</i>	^c
ژ	<i>ž</i>	<i>zh</i>	<i>ž</i>	^z
ف <sup>14</sup>	<i>v</i>	<i>v</i>	<i>v</i>	v
گ	<i>g</i>	<i>g</i>	<i>g</i>	g
ن <sup>15</sup>	<i>ñ</i>	<i>ñ</i>	<i>ñ</i>	^n

Table 2: Standard ArabT<sub>E</sub>X (additional characters)

<sup>11</sup>For the letter ى with no diacritical points below, see Rem. *b*. below.

<sup>12</sup>See below section 7 on page 32.

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

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

<sup>15</sup>See footnote 14.

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 ArabT<sub>E</sub>X equivalents for the Arabic long vowels.

Letter	Transliteration <sup>16</sup>			ArabT <sub>E</sub> X notation
	dmg	loc	arabica	
ا	$\bar{a}$	$\bar{a}$	$\bar{a}$	A
و	$\bar{u}$	$\bar{u}$	$\bar{u}$	U
ي	$\bar{i}$	$\bar{i}$	$\bar{i}$	I <sup>17</sup>
ى <sup>18</sup>	$\bar{a}$	$\bar{a}$	$\bar{a}$	_A or Y
أ	$\bar{a}$	$\bar{a}$	$\bar{a}$	_a
و	$\bar{u}$	$\bar{u}$	$\bar{u}$	_u
ي	$\bar{i}$	$\bar{i}$	$\bar{i}$	_i

Table 3: Standard ArabT<sub>E</sub>X (long vowels)

REM. a. The long vowels  $\bar{a}$ ,  $\bar{u}$ ,  $\bar{i}$ , otherwise called *hurūf<sup>u</sup> 'l-madd<sup>i</sup>*, 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*, قَالَ يَقُولُ *yaqūlu*.

REM. b. Defective writings, such as ا, *al-'alif<sup>u</sup> 'l-mahḍūfat<sup>u</sup>*, or defective writings of  $\bar{u}$  and  $\bar{i}$  are encoded \_a\_u and \_i respectively, e.g. \_d\_alika ذَلِكَ, al-mal\_a'ikaT-u 'l-ra.hm\_an-u الْمَلِكَةُ الرَّحْمَنُ, .hu\_dayfaT-u bn-u 'l-yamAn\_i الْحَيَاءُ الْمُنَاةُ مِنْ نَحْبِهَا for *Huḍayfat<sup>u</sup> bn<sup>u</sup> 'l-Yamānī*, etc.

REM. c. The letter ي with two points below, أَلْيَاءُ الْمُنَاةُ مِنْ نَحْبِهَا, may also be written without diacritical points as ى. When it is used as a long vowel, it is encoded iY, where i recalls the *kasrah* placed below the preceding letter in vocalized Arabic, like so: liY لِي *lī*, yam<sup>^</sup>siY يَمْسِي *yamṣī*.

#### 3.3.2 Short vowels

Table 4 gives the ArabT<sub>E</sub>X equivalents for the Arabic short vowels.

Letter	Transliteration <sup>19</sup>			ArabT <sub>E</sub> X 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

<sup>16</sup>See below section 7 on page 32.

<sup>17</sup>For the letter ي with no diacritical points, see *Rem. c.* below.

<sup>18</sup>= *al-'alif<sup>u</sup> 'l-maqṣūrat<sup>u</sup>*.

<sup>19</sup>See below section 7 on page 32.

Letter	Transliteration			ArabT <sub>E</sub> X notation
	dmg	loc	arabica	
ـ	<i>i</i>	<i>i</i>	<i>i</i>	i
ا	<i>an</i>	<i>an</i>	<i>an</i>	aN
u	<i>un</i>	<i>un</i>	<i>un</i>	uN
ي	<i>in</i>	<i>in</i>	<i>in</i>	iN

Table 4: Standard ArabT<sub>E</sub>X (short vowels)

Whether Arabic texts be vocalized or not is essentially a matter of personal choice. So one may use `voc` mode and decide not to write vowels except at some particular places for disambiguation purposes, or use `novoc` mode, not write vowels—as `novoc` normally does not show them—except, again, where disambiguation is needed.<sup>20</sup>

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 21);
- 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 شَمْسُ *šams<sup>uN</sup>*, karImuN كَرِيمٌ *Karīm<sup>uN</sup>*.  
bi-hi بِهِ *bi-hi*, 'aqi.tuN أَقِيْتُ *'aqiṭ<sup>uN</sup>*.  
la-hu لَهُ *la-hu*, .hujjaTuN هُجَّةٌ *huğğat<sup>uN</sup>*.

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

<sup>20</sup>See below section 4.4 on page 21.

qAla قَالَ *qāla*, bI`a بَعَّ *bī'a*, .tUruN طُور *ṭūr<sup>un</sup>*, .tInuN طِين *ṭīn<sup>un</sup>*,  
murU'aTuN مُرُوَّةٌ *murū'at<sup>un</sup>*.

**'alif maqṣūrah** It is written ⟨\_A⟩ or ⟨Y⟩:

al-fat\_A أَلْفَى *al-fatā*, al-maqh\_A أَلْفَى *al-maqhā*, 'il\_A إِلَى *ilā*.

**'alif otiosum** Said *'alif<sup>u</sup> l-wiqāyat<sup>i</sup>*, “the guarding *'alif*”, after و at the end of a word, both when preceded by *ḍammah* and by *fathah* is written ⟨UA⟩ or ⟨aW, aWA⟩:

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

**'alif maḥḍūfah and defective ū, ī** They are written ⟨\_a, \_i \_u⟩:

al-l\_ah-u أَلَهُ *al-lāh<sup>u</sup>*, 'il\_ahUN إِلَهُ *ilāh<sup>un</sup>*.  
al-ra.hm\_an-u أَلْرَحْمَنُ *ar-raḥmān<sup>u</sup>*, l\_akin لَكِنْ *lākin*, h\_ahunA هَاهُنَا *hāhunā*,  
.hunayn-u bn-u 'is.h\_aq-a حَنِينُ بْنُ إِسْحَقَ *Hunayn<sup>u</sup> bn<sup>u</sup> 'Ishāq<sup>a</sup>*, rabb\_i رَبِّ *rabbī*, al-`A.s\_i أَلْعَاصِ *al-Āṣī*.

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

.hay\_aUTuN حَيَوَةٌ *hayāt<sup>un</sup>*, .sal\_aUTuN صَلَوَةٌ *ṣalāt<sup>un</sup>*, mi~sk\_aUTuN مَشْكُورَةٌ *miš-kāt<sup>un</sup>*, tawr\_aITuN تَوْرِيَةٌ *tawrāt<sup>un</sup>*.

And so also: al-rib\_aIT-u أَلْرِبِيَّةُ *ar-ribāt<sup>u</sup>*.

**'Amr<sup>un</sup>, and the silent و** To that name a silent و is added to distinguish it from *'Umar<sup>u</sup>*: see Wright (1896, i. 12 C). In no way this affects the sound of the *tanwīn*, so it has to be discarded in **trans** mode:

`amruNU عَمَرُوا *amr<sup>un</sup>*, `amraNU عَمَرُوا *amr<sup>an</sup>*, `amriNU عَمَرُوا *amr<sup>in</sup>*.

When the *tanwīn* falls away (Wright 1896, i. 249 B): `amr-uU bn-u mu.hammadIN عَمَرُوا بْنُ مُحَمَّدٍ *Amr<sup>u</sup> bn<sup>u</sup> Muḥammad<sup>in</sup>*, mu.hammad-u bn-u `amr-iU bn-i\_hAlidiN عَمَرُوا بْنُ خَالِدٍ *Muḥammad<sup>u</sup> bn<sup>u</sup> Amr<sup>i</sup> bn<sup>i</sup> Hālid<sup>in</sup>*.

And so also: al-rib\_aUA أَلْرِبَا *ar-ribā*, ribaNU رِبَا *rib<sup>an</sup>*.



yatasā'alūna, murU'aTuN مُرُوءٌ, taw'amuN تَوَّعٌ taw'am<sup>un</sup>, murū'at<sup>un</sup>, ta'xIruN تَأْخِيرٌ ta'hīr<sup>un</sup>, ta'axxara تَأَخَّرَ ta'ahhara, ji'tu-ka جِئْتُكَ ġi'tu-ka, qA'iluN قَائِلٌ qā'il<sup>un</sup>, .hIna'i\_diN hīna'id<sup>in</sup> حِينَدِ, hay'aTuN هَيْتَ hay-'at<sup>un</sup>, hay'AtuN هَيْآتَ hay'āt<sup>un</sup>.

From Wright (1896, i. 14 B):— All consonants, whatsoever, not even 'alifhəmzatum excepted, admit of being doubled and take *tašdid*. Hence we speak and write ra'AsuN رَأْسٌ ra'ās<sup>un</sup>, sa'AluN سَأَلٌ sa'al<sup>un</sup>, na'AjuN نَأَجٌ na'āġ<sup>un</sup>.

**Final hamzah:** xa.ta'uN خَطَأٌ haṭa'<sup>un</sup>, xa.ta'aN خَطَأٌ haṭa'<sup>an</sup>, xa.ta'iN خَطِئٌ haṭa'<sup>in</sup>, 'aqra'u أَقْرَأُ aqra'u, taqra'Ina تَقْرَأِينَ taqra'ina, taqra'Una تَقْرَأُونَ taqra'una, yaqra'na يَقْرَأْنَ yaqra'na, yaxba'Ani يَخْبَأَنِي yaḥba'āni, xaba'A خَبَأَ haḥba'ā, xubi'a خُبِئَ hubi'a, xubi'UA خُبِئُوا hubi'ū, jA'a جَاءَ ġā'a, ridA'uN رَدَأَ ridā'<sup>un</sup>, ridA'aN رَدَأَ ridā'<sup>an</sup>, jI'a جِئَ ġī'a, radI'iN رَدِئَ radī'<sup>in</sup>, sU'uN سُوءٌ sū'<sup>un</sup>, .daw'uN دَوَّءٌ daw'<sup>un</sup>, qay'iN قَيَّ قay'<sup>in</sup>, ~sifA'I شَفَاتِي šifā'ī, man~sa'I مَنْشَى manša'ī, nisA'uN نِسَاءٌ ni-sā'<sup>un</sup>, nisA'u-hu نِسَاؤُهُ nisā'u-hu, nisA'i-hi نِسَائِهِ nisā'ī-hi, nisA'I نِسَائِي nisā'ī.

~say'uN سَيَّ سāy'<sup>un</sup>, ~say'aN سَيَّ سāy'<sup>an</sup>, ~say'iN سَيَّ سāy'<sup>in</sup>, al-~say'-u أَلْسَيَّ aš-sāy'<sup>u</sup>, 'a~syA'-u أَشْيَاءُ ašyā'<sup>u</sup>, 'a~syA'-a أَشْيَاءُ ašyā'<sup>a</sup>, .zim'aN ظَمَأَ ḡim'<sup>an</sup>, radI'aN رَدِئَ radī'<sup>an</sup>.

**taḥfif<sup>u</sup> 'l-hamzat<sup>i</sup>:** if the *hamzah* has *ġazmah* 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 أَمِنَ āmana, 'u'minu أَوْمِنُ ūminu, 'i'mAnuN إِيمَانٌ īmān<sup>un</sup>. For other possible ways of encoding such sequences, see on the preceding page (*hamzah* followed by و and ي) and the *maddah* on the next page.

Imperatives of verbs that have the *hamzah* as the first radical are other cases of *taḥfif<sup>u</sup> 'l-hamzat<sup>i</sup>:* i'sir إِيسِرْ īsir, i'\_dan إِذَنْ īdan, u'mul أَوْمِلْ ūmul. arabluatex also provides ways of encoding those words when the initial 'alif comes into *waṣl*, so as to make the 'alif *waṣl* fall away when preceded by وَ or فَ: wa-'sir وَأَسِرْ wa-'sir, fa-'\_dan فَادَنْ fa-'dan,

fa-'ti فَاْتِ fa-'ti, wa-'tamirUA وَأَتَمِّرُوا wa-'tamirū; or be retained outside the imperative, as in fa-i-'tazarat فَاتَّزَرْتُ fa-'tazarat, ba`da i'tilAfīn بَعْدَ اِتِّلَافٍ ba`da 'tilāf<sup>in</sup>.

**The strange spelling of mi'at<sup>un</sup>:** mi'aTuN مَائَةٌ mi'at<sup>un</sup>, mi'atAni مَائَتَانِ mi'atāni, mi'atayni مَائَتَيْنِ mi'atayni, mi'Una مِئُونٌ mi'ūna, mi'AtuN مِئَاتٌ mi'ātun, mi'aN\_A مَائِي mi'a<sup>n</sup>. Of course, the 'pipe' character may be used to prevent this rule from being applied (see section 4.5 on page 23): mi'a|TuN مِئَةٌ mi'at<sup>un</sup>.

**maddah** At the beginning of a syllabe, 'alif with *hamzah* and *fathah* (أ) followed by 'alifu 'l-maddi ('alif of prolongation) or 'alif with *hamzah* and *ğazmah* (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<sup>un</sup> respectively.

arabluatex otherwise determines *al-'alif<sup>u</sup> 'l-mamdūdat<sup>u</sup>* by context analysis.

'is'AduN إِسَادٌ 'is'ād<sup>un</sup>, 'AkilUna أَكِلُونُ ākilūna, 'a'manna أَمَنَّا āmannā, al-qur'An-u الْقُرْآنُ al-qur'ān<sup>u</sup>.  
jA'a جَاءَ ġā'a, yatasA'alUna يَتَسَاءَلُونَ yatasā'alūna, ridA'uN رِدَاءٌ ridā<sup>un</sup>, xaba'A حَبَا ħaba'a, yaxba'Ani يَخْبَانُ yaḥba'āni.

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

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

`allaqa عَلَّقَ `allaqa, mAdduN مَادَّ mād<sup>dun</sup>, 'ammara أَمَّرَ ammara, murruN مَرَّرَ murr<sup>un</sup>.

**The euphonic tašdīd** always follows a vowelless consonant which is passed over in pronunciation and assimilated to a following consonant. It may be found (Wright 1896, i. 15 B–16 C):—

- (a) With the *solar* letters ت, ث, د, ذ, ر, ز, س, ش, ص, ض, ط, ظ, ل, ن, after the article اَلْ:—



Unlike `arabtex` and `arabxetex`, `arabluatex` *never requires the solar letter to be written twice*, as it automatically generates the euphonic *tašdīd* above the letter that carries it, whether the article be written in the assimilated form or not, e.g. `al-ˆsams-u` الشَّمْسُ *aš-šams<sup>u</sup>*, or `aˆs-ˆsams-u` الشَّمْسُ *aš-šams<sup>u</sup>*.

`al-tamr-u` التَّمْرُ *at-tamr<sup>u</sup>*, `al-ra.hm_an-u` الرَّحْمَنُ *ar-rahmān<sup>u</sup>*, `al-zulm-u` الظُّلْمُ *aẓ-ẓulm<sup>u</sup>*, `al-lu.gaT-u` اللُّغَةُ *al-luġat<sup>u</sup>*.

- (b) With the letters ر, ل, م, 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 رَبِّي *mir rabbi-hi*, `min layliN` مِنْ لَيْلٍ *mil layl<sup>in</sup>*, `'an yaqtula` أَنْ يَقْتُلَ *'ay yaqtula*.

With *tanwīn*: `kitAbuN mubInuN` كِتَابٌ مُبِينٌ *kitāb<sup>um</sup> mubīn<sup>un</sup>*.

REM. This particular feature must be put into operation by the `\SetArbDflt*` command explicitly. See above section 2.2.1 on page 6 for further details. Other kinds of assimilations, including the various cases of *ʾidġām*, will be included in `arabluatex` gradually.

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

**The definite article and the 'alif<sup>u</sup> 'l-waṣl<sup>i</sup>** At the beginning of a sentence, ٱ is never written, as اَلْحَمْدُ لِلّٰهِ; instead, to indicate that the 'alif is a connective 'alif ('alif<sup>u</sup> 'l-waṣl<sup>i</sup>), the *hamzah* is omitted and only its accompanying vowel is expressed:

`al-.hamd-u li-l-l_ah-i` اَلْحَمْدُ لِلّٰهِ *al-ḥamd<sup>u</sup> li-l-lāh<sup>i</sup>*.

As said above on section 2.2, `fullvoc` is the mode in which `arabluatex` expresses the *sukūn* and the *waṣlah*. `arabluatex` will take care of doing that automatically provided that the vowel which is to be absorbed by the final vowel of the preceding word be properly encoded, like so:—

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

`al-`, or `a<solar letter>-`

if one wishes to mark the assimilation—which is in no way required, as `arabluatex` will detect all cases of assimilation.

- (b) Definite article inside sentences is encoded

`'l-` or `'<solar letter>-`.

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

**Article:** bAb-u 'l-madrasaT-i الْمَدْرَسَةُ بَابُ *bāb*<sup>u</sup> 'l-madrasat<sup>i</sup>, al-maqA laT-u 'l-'\_Ul\_A الْمَقَالَةُ الْأُولَى *al-maqālat*<sup>u</sup> 'l-ʿulā, al-lu.gaT-u 'l-`ara biyyaT-u اللُّغَةُ الْعَرَبِيَّةُ *al-luġat*<sup>u</sup> 'l-ʿarabiyyat<sup>u</sup>, fI .sinA`aT-i 'l-.tibb-i فِي صِنَاعَةِ الطِّبِّ *fī ṣināʿat*<sup>i</sup> 't-ṭibb<sup>i</sup>, 'il\_A 'l-intiqA.d-i إِلَى الْإِنْتِقَاضِ *ʿilā* 'l-intiqāḍ<sup>i</sup>, fI 'l-ibtidA'-i فِي الْإِبْتِدَاءِ *fī* 'l-ibtidā'<sup>i</sup>, 'abU 'l-wazIr-i أَبُو الْوَزِيرِ *ʿabu* 'l-wazīr<sup>i</sup>, fa-lammA ra'aW 'l-najm-a فَلَمَّا رَأَوْا النَّجْمَ *fa-lammā* raʿawu 'n-naġm<sup>a</sup>.

**Particles:—**

- (a) *li-*: *ʿalif*<sup>u</sup> *ʿl-waṣl*<sup>i</sup> is omitted in the article أَلْ when it is preceded by the preposition لِ: li-l-rajul-i لِلرَّجُلِ *li-r-raġul*<sup>i</sup>.  
If the first letter of the noun be ل, then the ل of the article also falls away, but arabluatex is aware of that: li-l-laylaT-i لِلَّيْلَةِ *li-l-laylat*<sup>i</sup>.
- (b) *la-*: the same applies to the affirmative particle لَ: la-l-.haqq-u لِلْحَقِّ *la-l-ḥaqq*<sup>u</sup>.
- (c) With the other particles, *ʿalif*<sup>u</sup> *ʿl-waṣl*<sup>i</sup> is expressed: fI 'l-madInaT-i فِي الْمَدِينَةِ *fī* 'l-madīnat<sup>i</sup>, wa-'l-rajul-u وَالرَّجُلُ *wa-*'r-raġul<sup>u</sup>, bi-'l-qalam-i بِالْقَلَمِ *bi-*'l-qalam<sup>i</sup>, bi-'l-ru`b-i بِالرُّعْبِ *bi-*'r-ruḅ<sup>i</sup>.

**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 وَأَسْتُعْمِلُ *wa-*'stu'mila, qad-i in.sarafa قَدْ أَنْصَرَفَ *qadi* 'nṣarafa, al-iqtidAr-u الْأَقْتِدَارُ *al-iqtidār*<sup>u</sup>, 'il\_A 'l-inti qA.d-i إِلَى الْإِنْتِقَاضِ *ʿilā* 'l-intiqāḍ<sup>i</sup>, law istaqbala لَوْ اسْتَقْبَلَ *lawi* 'staqbala.

**Other cases:** 'awi ismu-hu أَوْ اسْمُهُ *ʿawi* 'smu-hu, zayduN ibn-u `amriNU زَيْدُ بْنُ عَمْرٍو *Zayd*<sup>uni</sup> 'bn<sup>u</sup> 'Amr<sup>in</sup>,<sup>22</sup> `umar-u ibn-u 'l-\_ha.t.tAb-إِمْرُؤُ بْنُ أَحْطَابٍ *ʿUmar*<sup>u</sup> 'bn<sup>u</sup> 'l-Ḥaṭṭāb<sup>i</sup>,<sup>23</sup> imru'-u 'l-qays-i إِمْرُؤُ الْقَيْسِ *Imru*<sup>u</sup> 'l-Qays<sup>i</sup>, la-aymun-u 'l-l\_ah-i لَا يَمُنُ اللَّهُ *la-*'ymun<sup>u</sup> 'l-lāh<sup>i</sup>.

<sup>22</sup> “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 زَيْدُ بْنُ عَمْرٍو, “Zayd, son of ‘Amr”.

<sup>23</sup> “Umar is the son of al-Ḥaṭṭāb” (see footnote 22).

**'alif<sup>u</sup> 'l-waṣl<sup>i</sup> 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 arablumatex takes it into account in some transliteration standards:—

fI 'l-nAs-i فِي النَّاسِ *fi 'n-nās<sup>i</sup>*, 'abU 'l-wazIr-i أَبُو الْوَزِيرِ *'abu 'l-wazīr<sup>i</sup>*,  
fI 'l-ibtidA'-i فِي الْإِبْتِدَاءِ *fi 'l-ibtidā'<sup>i</sup>*, \_dU 'l-i`lAl-i ذُو الْأَعْلَالِ *du 'l-i`lāl<sup>i</sup>*,  
maqH\_A 'l-'amIr-i مَقْبَحِي الْأَمِيرِ *maqḥa 'l-'amīr<sup>i</sup>*.

**'alif<sup>u</sup> 'l-waṣl<sup>i</sup> 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*. arablumatex detects the cases in which this rule applies:—

fI `aynay 'l-malik-i فِي عَيْنِي الْمَلِكِ *fī 'aynayi 'l-malik<sup>i</sup>*, ix^say 'l-qawm-a اخْشَى الْقَوْمَ *iḥšay<sup>i</sup> 'l-qawm<sup>a</sup>*, mu.s.tafaw 'l-l\_ah-i مُصْطَفَوُ اللَّهِ *muṣṭa-fawu 'l-lāh<sup>i</sup>*.  
ramaW 'l-.hiJAraT-a رَمَوْا الْحِجَارَةَ *ramawu 'l-ḥiğārat<sup>a</sup>*, fa-lamma ra'aW 'l-naJm-a فَلَمَّا رَأَوْا النَّجْمَ *fa-lammā ra'awu 'n-nağm<sup>a</sup>*.

**'alif<sup>u</sup> 'l-waṣl<sup>i</sup> 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 أَنْتُمْ الْكَاذِبُونَ *'antumU 'l-kādib<sup>ūna</sup>*, ra'aytumU 'l-raJul-a رَأَيْتُمُ الرَّجُلَ *ra'aytumU 'r-rağul<sup>a</sup>*, mani 'l-ka\_d\_dAb-u مَنِ الْكَادِبِ *mani 'l-kaddāb<sup>u</sup>*, qatalati 'l-rUm-u قَتَلَتِ الرُّومُ *qatalati 'r-Rūm<sup>u</sup>*.

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. arablumatex takes care of that automatically:—

mu.hammaduN 'l-nabI مُحَمَّدُ النَّبِيِّ *Muḥammad<sup>uni</sup> 'n-nabī*, salAmuN ud\_hulUA سَلَامٌ أَدْخُلُوا *salām<sup>umu</sup> 'dhulū*, qa.sIdata-hu fI qatl-i قَصِيدَتُهُ فِي قَتْلِ أَبِي مُسْلِمٍ الَّتِي يَقُولُ فِيهَا *qaṣīdata-hu fī qatl<sup>i</sup>*  
'llatI yaqUlu fI-hA *'llatī yaqūlu fī-hā*.



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 اللَّائِي *al-*  
*lāti*, al-lA' |Ati<sup>26</sup> اللَّاءَاتِ *al-lā'āti*, al-lA' I اللَّائِي *al-lā'i*, and so forth.

#### 4.4 Quoting

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

`ullima `ilm-a 'l-hay'aT-i علم الهيئة 'ullima 'ilm<sup>a</sup> 'l-hay'at<sup>i</sup>,  
 ya.gz"UA يغزوا *yaǧzū*.

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

wa-'innA sawfa tudriku-nA 'l-manAyA muqadd"araT"aN وإنا سوف  
 تدرِكنا المنايا مقدرةً, *wa-'innā sawfa tudriku-na 'l-manāyā muqaddarat<sup>an</sup>*.

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

qAla isma`" قال اسمعُ qāla 'sma', jA'at" hinduN جاءتْ هندُ ḡā'at  
 Hind<sup>un</sup>, ṣabIhuN bi-man q"u.ti`at" qadamA-hu شبيه بن قُطعتْ qadamā-hu.  
 ṣabīh<sup>un</sup> bi-man quṭi'at qadamā-hu.

<sup>26</sup>Note here the “pipe” character '|’: as already stated on page 16, the sequence 'A usually encodes *‘alif* with *hamzah* followed by *‘alif* of prolongation, which is represented in writing *‘alif* with *maddah*: ٲ. The “pipe” character prevents this rule from being applied. See section 4.5 on page 23.

<sup>27</sup>See Lagally (2004, p. 22)

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

wa-"ust"u`mila وَأَسْتَعْمَلُ *wa-'stu'mila*, huwa "inhazama هو أَنَهَزَمَ *huwa 'nhazama*, al-"intiqA.d-u الْإِنْتِقَاضُ *al-intiqāḍ<sup>u</sup>*.

**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 قَالِ قَائِلٌ *qāla qā'il<sup>un</sup>*, ibn-u 'abI 'u.saybi`aT-"a  
ابْنُ أَبِي أُصَيْبَةَ *Ibn<sup>u</sup> 'Abī 'Uṣaybi'at<sup>a</sup>*.

- (b) *tanwīn*:—

madInaT"aN مَدِينَةٌ *madīnat<sup>an</sup>*, bAb"aN أَبَا *bāb<sup>an</sup>*, hud"aN\_A هُدًى *hudā<sup>n</sup>*,  
سَيِّئٌ *say<sup>'in</sup>*.

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<sup>u</sup> 'l-waṣl'*:—

fI "istiq.sA'-iN فِي اسْتِقْصَاءٍ *fi 'stiqṣā'<sup>in</sup>*, wa-"istiq.sA'-uN وَأَسْتِقْصَاءٌ  
wa-'stiqṣā'<sup>un</sup>, qAla "uhrub fa-lan tuqtala قَالَهُ هَرُبَ فَلَنْ تُقْتَلَ  
'hrub fa-lan tuqtala.

- (b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala قَالَهُ أَقْتُلْ فَلَنْ تُقْتَلَ *qāla 'qtul fa-lan tuq-*  
*tala*, mA jA'at" mini imra'aTiN مَا جَاءَتْ مِنْ امْرَأَةٍ *mā ḡā'at mini*  
'mra'at<sup>in</sup>, kam" qad" ma.dat" min" laylaTiN كَمْ قَدْ مَضَتْ مِنْ لَيْلَةٍ *kam*  
*qad maḍat min laylat<sup>in</sup>*.

**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 الْجَمْرُ الصَّيْفِيُّ  
الَّذِي كَانَ يَقْرَأُونَ *al-ḡamr<sup>u</sup> 'ṣ-ṣayfiyy<sup>u</sup> 'llaḍī kāna bi-Qrānnūn<sup>a</sup>*.

#### 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 <sup>28</sup>			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 21–22.

’a`dA’ukum أَدَاؤُكُمْ ’a`dā’ukum, ’a`dA|”’ukum أَدَاءُكُمْ ’a`dā’ukum, ’a`dA’ikum أَدَائِكُمْ ’a`dā’ikum, ’a`dA|”’ikum أَدَاءُكُمْ ’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 on the previous page, its usage to encode the *hamzah* alone, with no carrier: |”’ ء.

Aside from that usage, the “pipe” character is used to prevent almost any of the contextual analysis rules that are described above from being applied. Two examples have already been given to demonstrate how that particular mechanism works in footnote 24 on page 20 and in footnote 26 on page 21. One more example follows:—

bi-qraN|nUn-a بِقْرَانُونٌ *bi-Qrānnūn*<sup>a</sup>, “in Crannon” (Thessaly, Greece).<sup>29</sup>

As one can see, the “pipe” character between the two (n) prevents the necessary *tašdīd* rule (section 4.2) 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

<sup>28</sup>See below section 7 on page 32.

<sup>29</sup>See more context on the preceding page.

annotations can be particularly abundant, the contextual analysis rules that are described above may be broken by L<sup>A</sup>T<sub>E</sub>X 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<sup>a</sup> النَّجْمَ...

---

<sup>a</sup>A footnote that interferes with the contextual analysis.

According to the rule stated on page 19, the diphthong in *ra'aw* must be resolved into two simple vowels before the *'alif<sup>u</sup> l-wasl<sup>i</sup>*, 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     \uc{z}aydu\arbnul{ibnu}\LRfootnote{A footnote that interferes
9       with the contextual analysis.} \arbnul{zayduN}ibn-u
10    \uc{'a}mriNU.\LRfootnote{See \vref{fn:zayd-is-son}.}
11  \end{arab}
12  \begin{arab}[trans]
13    \uc{z}aydu\arbnul{ibnu}\LRfootnote{A footnote that interferes
14      with the contextual analysis.} \arbnul{zayduN}ibn-u
15    \uc{'a}mriNU.\LRfootnote{See \vref{fn:zayd-is-son}.}
16  \end{arab}

```



فَلَمَّا رَأَوْا<sup>a</sup> النَّجْمَ...  
 قَالَ<sup>b</sup> أَهْرَبْ فَلَنْ تَقْتُلَ.  
 زَيْدٌ<sup>c</sup> ابْنُ عَمْرٍو<sup>d</sup>.

*Zayd<sup>unie</sup> 'bn<sup>u</sup> Amr<sup>in</sup>!*<sup>f</sup>

<sup>a</sup>A footnote that interferes with the contextual analysis.

<sup>b</sup>A footnote that interferes with the contextual analysis.

<sup>c</sup>A footnote that interferes with the contextual analysis.

<sup>d</sup>See footnote 22 on page 18.

<sup>e</sup>A footnote that interferes with the contextual analysis.

<sup>f</sup>See 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<sup>u</sup> bn<sup>u</sup> 'Ishāq<sup>a</sup>*

## 4.8 Digits

### 4.8.1 Numerical figures

The *Indian numbers*, *ar-raqam<sup>u</sup> 'l-hindiyy<sup>u</sup>*, 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:—

\abjad{45} kitAbu-hu fI 'l-`AdAt-i مَكَّابُهُ فِي الْعَادَاتِ 45 *kitābu-hu fi*  
*'l-ādāt<sup>i</sup>.*

REM. *a.* As can be seen in the above given example, `arabluatex` expresses the *ʿabǧad* 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:—

The `\arb[trans]{ʿabjad}` number for 1874 is `\abjad{1874}` The *ʿabǧad* number for 1874 is غصعد.

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

## 4.9 Additional characters

In the manuscripts, the unpointed letters, *al-ḥurūfu ʿl-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 <sup>30</sup>			ArabTeX 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
ⵍ	(	(	(	((
ⵍ	)	)	)	))

Table 6: Additional Arabic codings

`ʿafAman.tUs Gal.(M)` .fmn.n.ts (sic) Gal.(E1), أفامنطوس Gal.(M) منطس (sic) Gal.(E1), *ʿafāmanṭūs Gal.(M)* *fmnṇṭs* (sic) Gal.(E1).

## 4.10 Arabic emphasis

As already seen in section 4.8.2 on the preceding page, the *ʿabǧad* 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

<sup>30</sup>See below section 7 on page 32.

emphasize words, like so:—

`\abjad{45}: kitAbu-hu \aemph{fI 'l-`AdAt-i}` 45: مة: كُتَابُهُ فِي الْعَادَاتِ  
*kitābu-hu fi 'l-`Ādāt<sup>i</sup>.*

New feature  
v1.9.2

`\aemph*`

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

REM. *b.* `\aemph*` is also provided should one wish to always have the horizontal stroke printed over the emphasized words, like so: `\abjad{45}: kitAbu-hu \aemph*{fI 'l-`AdAt-i}`  
45: مة: كُتَابُهُ فِي الْعَادَاتِ  
*kitābu-hu fi 'l-`Ādāt<sup>i</sup>.*

## 5 Arabic poetry

New feature  
v1.6

`arabverse`

`arabluatex` provides a special environment for typesetting Arabic poetry. Every line in this environment must end with `\\`.

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 39 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 25). 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{[LA 'ar_A man `ahidtu fI-hA fa-'abkI 'l---][--]{---yawma
3 dalhaN wa-mA yaruddu 'l-bukA'u}\\
4 \end{arabverse}
```

لَا أَرَى مَنْ عَهِدْتُ 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*<sup>31</sup> *ʿl-waṣl*<sup>31</sup> 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{[LA 'ar_A man `ahidtu fI-hA fa-'abkI 'l---][--]{---yawma
3 dalhaN wa-mA yaruddu 'l-bukA'u}\\
4 \end{arabverse}
```

*lā* ʿarā man ʿahidtu fī-hā fa-ʿabki ʿl- -yawma dālḥ<sup>an</sup> wa-mā yaruddu ʿl-bukāʿu

<sup>31</sup>See section 4.2 on page 17

- (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-_tawiyyi '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 أَنِّي قَدْ اسْتَعِينُ عَلَى الْهَمِّ إِذَا خَفَّ بِالْثَوِيِّ النَّجَاءُ  
 بِزَفْوْفٍ كَانَهَا هِقْلَةً أُمُّ رِثَالٍ دَوِيَّةً سَقَفَاءُ

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 *‘ağuz*. 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.

**Sealing 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.<sup>32</sup>

<sup>32</sup>The `footnote` package may also be used for the same effect. However, it must be loaded *after* `arabluatex`.

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

## 5.1 Example

Here follow the first lines of Imru'u 'l-Qaysi's *Mu'allaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the *ʿidgām* that is fit to this declamatory poetry:—<sup>33</sup>

```

1 \begin{arab}[fullvoc]
2   qAla \uc{i}mru'u 'l-\uc{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 \uc{'l-d}a_hUli fa-\uc{.h}awmali}\\
11     \bayt{fa-\uc{t}U.di.ha fa-'l-\uc{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 'asaN_A wa-ta^gammali}\\
19     \bayt{wa-'inna ^sifa'I `abraTuN muharAqaTuN}{fa-hal `inda
20      rasmiN
21     dAsiriN min mu`awwali}\\
22   \end{linenumbers*}
23 \end{arabverse}

```

قَالَ أَمْرُؤُ الْقَيْسِ فِي مَعْلَقَتِهِ:

1	بَسَقَطَ اللَّوَى بَيْنَ الدَّخُولِ وَالْخَوَالِ	قَفَا نَبِكُ مِنْ ذِكْرِي حَبِيبٍ وَمَنْزِلِ
2	لَمَّا نَسَجْتَهَا مِنْ جَنُوبٍ وَشَمَالِ	فَتَوَضَّحَ فَأَلْمَقَرَّةَ لَمْ يَعْفُ رَسْمَهَا
3	وَقِيْعَانَهَا كَأَنَّهُ حَبٌّ فَلَفْلُ	تَرَى بَعْرَ الْأَرَامِ فِي عَرَصَاتِهَا
4	لَدَى سُرَرَاتِ الْحَيِّ نَاقِفٌ حَنْظَلِ	كَأَنِّي غَدَاةَ الْبَيْنِ يَوْمَ تَحْمَلُوا
5	يُقُولُونَ لَا تَهْلِكُ أَسَى وَيَتَجَمَّلِ	وُقُوفًا بِهَا صَحْبِي عَلَيَّ مَطْمَعِهِمْ

<sup>33</sup>Please note that for the time being only the assimilation rules that are laid on item **b** on page 17 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 *ʿidgām* 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 مِّنَ الْعُرُوضِ الْأُولَى مِنَ الطَّوِيلِ)  
qāla Imru'u 'l-Qaysi fī mu'allaqati-hi:

1      gifā nabki min dīkrā ḥabīb<sup>iw</sup> wa-manzili      bi-saqti 'l-liwā bayna 'd-Daḥūli fa-Ḥawmali  
2      fa-Tūḍiḥa fa-'l-Migrāti lam ya'fu rasmu-hā      limā nasaḡat-hā min ḡanūb<sup>iw</sup> wa-šam'ali  
3      tarā ba'ara 'l-'ar'āmi fī 'araṣāti-hā      wa-qī'āni-hā ka-'anna-hu ḥabbu fulfuli  
4      ka-'annī ḡadāta 'l-bayni yawma taḥammalū      ladā samurāti 'l-ḥayyi nāḡifu ḥanzali  
5      wuḡūf<sup>un</sup> bi-hā ṣaḥbī 'alayya maṭiyya-hum      yaḡūlūna lā tahlik 'aṣa<sup>w</sup> wa-taḡammali  
6      wa-'inna šifā'i 'abrat<sup>um</sup> muḥarāḡat<sup>un</sup>      fa-hal 'inda rasm<sup>in</sup> dāsir<sup>im</sup> mim mu'awwali  
(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 25) 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  
 $\overset{\text{u}}{\text{u}}$   $\overset{\text{a}}{\text{a}}$   $\overset{\text{i}}{\text{i}}$ , 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. *arabuatex* 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:  $\{ \}$

$\backslash$ 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  $\backslash$ 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.<sup>34</sup> **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;<sup>35</sup>

**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.<sup>36</sup>

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{<font selection command>}` allows any specific font to be selected for rendering transliterated text with the font-selecting commands of the `fontspec` or `luaotfload` package. Of course, this font must have been defined properly. To take one example, here is how the *Gentium Plus* font may be used for rendering transliterated text:—

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

<sup>34</sup>See Brockelmann et al. (1935).

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

<sup>36</sup>See [http://www.brill.nl/files/brill.nl/specific/authors\\_instructions/ARAB.pdf](http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf).



**\uc Proper names** Proper names or book titles that must have their first letters uppercased may be passed as arguments to the `\uc{⟨word⟩}` command. `\uc` is a clever command, for it will give the definite article *al-* in lower case in all positions. Moreover, if the initial letter, apart from the article, cannot be uppercased, viz. ‘ or ‘, the letter next to it will be uppercased:—

`\uc{.hunayn-u} bn-u \uc{'is.h_aq-a}` حُنَيْنُ بْنُ إِسْحَقَ *Hunayn<sup>u</sup> bn<sup>u</sup>*  
`'Ishāqa, \uc{`u_tm_an-u}` عُثْمَانُ *Utmān<sup>u</sup>*, .daraba `\uc{zayd-u} bn-u`  
`\uc{`h_alidiN} \uc{sa`d-a} bn-a \uc{`awf-i} bn-i \uc{`abd-i}`  
`\uc{'l-l_ah-i}` ضَرَبَ زَيْدُ بْنُ خَلْدٍ سَعْدَ بْنَ عَوْفٍ بْنَ عَبْدِ اللَّهِ *daraba Zayd<sup>u</sup> bn<sup>u</sup>*  
*Hālid<sup>in</sup> Sa`d<sup>a</sup> bn<sup>a</sup> Awf<sup>i</sup> bn<sup>i</sup> Abd<sup>i</sup> 'Llāh<sup>i</sup>.*

However, `\uc` must be used cautiously in some very particular cases, for the closing brace of its argument may prevent a rule from being applied. To take an example, as seen above on page 19, the transliteration of مُحَمَّدٌ النَّبِيُّ must be *Muḥammad<sup>uni</sup> 'n-nabī*, as nouns having the *tanwīn* take a *kasrah* in pronunciation before *'alifu 'l-waṣli*. In that case, encoding مُحَمَّدٌ like so: `\uc{mu.hammaduN}` is wrong, because the closing brace would prevent `arabluatex` from detecting the sequence `⟨-uN⟩` immediately followed by `⟨'l-⟩`. Fortunately, this can be circumvented in a straightforward way by inserting only part of the noun in the argument of `\uc` viz. up to the first letter that is to be uppercased, like so: `\uc{m}u.hammaduN`.

**Hyphenation** In case transliterated Arabic words break the T<sub>E</sub>X hyphenation algorithm, one may use the `\-` command to insert discretionary hyphens. This command will be discarded in all of the Arabic modes of `arabluatex`, but will be processed by any of the transliteration modes:—

`\uc{'abU} \uc{bakriN} \uc{mu\-.ham\-.madu} bnu \uc{za\-.ka \-`  
`riy\-.yA'a} \uc{'l-rAziyyu}` أَبُو بَكْرٍ مُحَمَّدُ بْنُ زَكْرِيَّا الرَّازِيُّ *'Abū Bakr<sup>in</sup> Mu-*  
*hammad<sup>u</sup> bn<sup>u</sup> Zakariyyā<sup>a</sup> 'r-Rāziyyu.*

**‘Long’ proper names** `\uc` is also able to process proper names consisting of several subsequent words:—

`\arb[trans]{\uc{'abU zaydiN .hunaynu bnu 'is.h_aqa 'l-`ibAdiyyu}}`  
*'Abū Zayd<sup>in</sup> Hunaynu bnu 'Ishāqa 'l-'Ibādiyyu.*

**\prname Proper names outside Arabic environments** Transliterated proper names inserted in paragraphs of English text should be printed in the same typeface as the surrounding text. `\prname{⟨Arabic proper name⟩}` is provided to that effect:<sup>37</sup>—

<sup>37</sup>Just as `\uc`, `\prname` is also able to process proper names consisting of several subsequent words.

```

1 From \textcite[i. 23 C]{Wright}:--- If the name following
2 \arb[fullvoc]{ibnuN} be that of the mother or the grandfather, the
3 \arb[fullvoc]{a} is retained; as \arb[fullvoc]{Is_A ibnu
   maryama},
4 \enquote{Jesus the son of Mary}; \arb[fullvoc]{ammAru ibnu
5   man.sUriN}, \enquote{\prname{ammAr} the (grand)son of
6   \prname{man.sUr}}.

```

From Wright (1896, i. 23 C):— If the name following <sup>ابن</sup> be that of the mother or the grandfather, the <sup>ا</sup> is retained; as عيسى ابن مريم, “Jesus the son of Mary”; عمار ابن منصور, “Ammār the (grand)son of Maṣṣūr”.

The following example shows how `\prname` can be used in conjunction with the `nameauth` package to have Arabic proper names printed first in full then in partial forms:<sup>38</sup>—

```

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

```

On first occurrence, proper names are printed as 'Abū Zayd Ḥunayn ibn 'Ishāq al-'Ibādī, 'Abū Bakr Muḥammad ibn Zakariyyā' ar-Rāzī. Then as Ḥunayn, ar-Rāzī.

## 7.1 Additional note on `dmḡ` convention

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

- (a) In full: *Amrun*;
- (b) As superscript text: *Amr<sup>un</sup>*;
- (c) Discarded: *Amr*.

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

`\NoArbUp` `\NoArbUp` may be used either in the preamble or at any point of the document in

`\ArbUpDflt`

<sup>38</sup>See the documentation of `nameauth` for more details: <https://ctan.org/pkg/nameauth>

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 ḡāmi`aN
3 muhaddamaTaN mi`_danatu-hu}: \arb[trans]{ra'aytu
4 ḡāmi`aN muhaddamaTaN mi`_danatu-hu.}

```

Arabic dmg transliteration for رَأَيْتُ جَامِعًا مَهْدَمَةً مِثْلَهُ: *ra'aytu ḡāmi`an muhaddamat<sub>an</sub> 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<sup>un</sup>* or *ḡayr<sup>u</sup> munṣarif<sup>in</sup>*. The former is automatically parsed by `arabluatex`, whereas the latter has to be delimited with an hyphen, like so:—

**munṣarif:** *mu`allim<sup>uN</sup> مُعَلِّمٌ mu`allim<sup>un</sup>, ka`inuN كَائِنٌ kā'in<sup>un</sup>, ka`inAtuN كَائِنَاتٌ kā'in<sup>ātun</sup>, \uc{`amraNU} عَمَرُوا `Amr<sup>an</sup>, fataN\_A فَتَى fata<sup>n</sup>, qA.diNI قَاضٍ qāḍi<sup>n</sup>.*

**ḡayr munṣarif:** *al-mu`allim-u الْمُعَلِّمُ al-mu`allim<sup>u</sup>, kitAb-Ani كِتَابَانِ kitāb<sup>āni</sup>, ra`sa'-Ani رَاسَانِ raśā`āni, sAriq-Una سَارِقُونَ sāriq<sup>āna</sup>, qA.d-Una قَاضُونَ qāḍ<sup>āna</sup>, al-.zulm-Atu الظُّلْمَاتُ aḡ-ḡulm<sup>ātu</sup>.*

REM. *a*. As the *tanwīn* is passed over in pronunciation when it is followed by the letters ر, ل, و, ي (see item [b](#) on page [17](#)), it may be desirable to further distinguish it by putting it above the line, but not to do the same for *ḡayr munṣarif* terminations. This can be achieved by simply omitting the hyphen before any *ḡayr munṣarif* termination:—

*kAna .ganiyyaN l\_akinna-hu labisa ḡubbaTaN mumazzaqaN 'aydu-hA كَانَ غَنِيًّا لَكِنَّهُ لَيْسَ جَبَّةً مَزَقًا kāna ḡaniyy<sup>an</sup> lākinna-hu labisa ḡubbat<sup>an</sup> mumazzaq<sup>an</sup> '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 مَنَّعَ النَّاسَ كُلَّهُ مِّنَ mana`a 'n-nās<sup>a</sup> kāffat<sup>an</sup> min muḡaḡabati-hi `aḡad<sup>un</sup> bi-sayyidi-nā.*

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

- (a) Nouns followed by an adjective in apposition: `madInaT kabIraT madīnah kabīrah`, `al-madInaT al-kabIraT al-madīnah al-kabīrah`.
- (b) Nouns followed by another noun in the genitive (construct state): `.hikmaT al-l_ah hikmat Allāh`, `fi.d.daT al-darAhim fidḍat ad-darāhim`.

REM. It may so happen, as in the absence of the article before the annexed word, that `arablu-atex` be unable to determine which of the above two cases the word ending with *tā' marbūṭah* falls into. The ‘pipe’ character (see section 4.5 on page 23) may be appended to that word to indicate that what follows is in the construct state: `\uc{r}isAlaT fI tartIb qirA'aT kutub \uc{g}AlInUs Risālah fī tartīb qirā'at kutub Ġālīnūs`.

**Uncertain short vowels** In some printed books, it may happen that more than one short vowel be placed on a consonant in cases where the vocalization is uncertain or ambiguous, like so: *فَعِلَ*. In transliteration, the uncertain vowels go between slashes and are separated by commas: `fa`uaila فَعِلَ fa`/u,a,i/la`.

## 7.2 Examples

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

**‘dmg’ standard:** `atā ṣadīqun ilā Ġuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratin qaṣīratin fa-qāla la-hu:` “*sawfa uḥdu-hu ilay-ka fī l-masā’i wa-adfa’u la-ka uḡrat<sup>an</sup>.*” `fa-qāla Ġuḥā:` “*anā āsīf<sup>un</sup> ḡidd<sup>an</sup> annī lā astatī’u an uḥaqqīqa la-ka raḡbata-ka fa-l-ḥimār<sup>u</sup> laysa huna l-yawm<sup>a</sup>.*” `wa-qāla la-hu` *yutimma Ġuḥā kalāma-hu bada’a l-ḥimār<sup>u</sup> yanhaqu fī iṣṭabli-hi.* `fa-qāla la-hu` *ṣadīqu-hu:* “*innī asma’u ḥimāra-ka yā Ġuḥā yanhaqu.*” `fa-qāla la-hu Ġuḥā:` “*ḡarīb<sup>un</sup> amru-ka yā ṣadīqī a-tuṣaddiqu l-ḥimār<sup>a</sup> 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 uḥdu-hu ilay-ka fī al-masā’i wa-adfa’u la-ka uḡratan.*” `fa-qāla Juḥā:` “*anā āsīfun ḡiddan annī lā astatī’u an uḥaqqīqa la-ka raḡbata-ka fa-al-ḥimāru laysa hunā al-yawma.*” `wa-qāla 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:` “*ḡharībun amru-ka yā ṣadīqī a-tuṣaddiqu al-ḥimāra wa-tukaddhība-nī?*”

**‘arabica’ standard:** `atā ṣadīqun ilā Ġuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratin qaṣīratin fa-qāla la-hu:` “*sawfa uḥdu-hu ilay-ka fī l-masā’i wa-adfa’u la-ka uḡratan.*” `fa-qāla Ġuḥā:` “*anā āsīfun ḡiddan annī lā astatī’u an uḥaqqīqa la-ka raḡbata-ka fa-l-ḥimāru laysa hunā l-yawma.*” `wa-qāla 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:` “*ḡarībun 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

\SetInputScheme

can also process the Buckwalter input scheme to a large extent.<sup>39</sup> 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 arablutext 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{arablutext}
2 \SetInputScheme{buckwalter}
```

**‘base’, ‘xml’ and ‘safe’ schemes** arablutext can use any of the so-called Buckwalter ‘base’, ‘xml’ or ‘safe’ schemes as they are described in Habash (2010, pp. 25–26).<sup>40</sup> However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode ا and ؤ, 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 arablutext. The additional characters that are defined in table 6 on page 26 are also available.

Letter	Transliteration <sup>41</sup>			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>ḥ</i>	<i>ḥ</i>	<i>ḥ</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>ḏ</i>	<i>dh</i>	<i>ḏ</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

<sup>39</sup>See <http://www.qamus.org/transliteration.htm>

<sup>40</sup>I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in arablutext.

<sup>41</sup>See section 7 on page 32.

Letter	Transliteration			Buckwalter notation	
	dmg	loc	arabica	base/xml	safe
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>ġ</i>	<i>gh</i>	<i>ġ</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
ء	’	’	’	’	C
آ	<i>’ā</i>	<i>’ā</i>	<i>’ā</i>		M
أ	’	’	’	>	O
ؤ	’	’	’	&	W
إ	’	’	’	<	I
ئ	’	’	’	]	Q
ـ	—	—	—	~	~
ـ	’	’	—	[	L
ا	<i>a</i>	<i>a</i>	<i>a</i>	a	a
و	<i>u</i>	<i>u</i>	<i>u</i>	u	u
ي	<i>i</i>	<i>i</i>	<i>i</i>	i	i
ان	<i>an</i>	<i>an</i>	<i>an</i>	F	F
ون	<i>un</i>	<i>un</i>	<i>un</i>	N	N
ين	<i>in</i>	<i>in</i>	<i>in</i>	K	K
و	—	—	—	o	o
ا	<i>ā</i>	<i>ā</i>	<i>ā</i>	`	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 32 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<sup>u</sup>*, Al-camsu الشَّمْسُ *aš-šams<sup>u</sup>*, bi-SinaAEapi  
 Al-T~ib~i, بِصَنَاعَةِ الطِّبِّ *bi-ṣinā‘at<sup>i</sup> ‘l-ṭibb<sup>i</sup>*.  
 wa-Al-l~ehi وَاللَّهِ *wa-‘l-lāh<sup>i</sup>*, Al-Hamdu li-l~ehi لِلَّهِ الْحَمْدُ *al-ḥamd<sup>u</sup>  
 li-llāh<sup>i</sup>*.

Similarly, it is not advisable to use | and [ (‘base’ and ‘xml’ schemes) or M and L (‘safe’ scheme) to encode the ‘*alif<sup>u</sup>*’ *‘l-mamdūdāt<sup>i</sup>* and the ‘*alif<sup>u</sup>*’ *‘l-waṣl<sup>i</sup>* 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-LntiqāADi` gives اِلَى الْاِنْتِقَاضِ as expected, but only `<ilY Al-intiqADi` can be transliterated as *‘ilā ‘l-intiqāḍi* with the correct vowel ⟨*i*⟩ in place of the ‘*alif<sup>u</sup>*’ *‘l-waṣl<sup>i</sup>*.

## 9 Unicode Arabic input

New feature  
v1.5

As said above in section 8 on page 36 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<sup>A</sup>T<sub>E</sub>X Commands in Arabic environments

**General principle** L<sup>A</sup>T<sub>E</sub>X 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-\uc{`AdAt-i}} مَهْ كِتَابُهُ فِي الْعَادَاتِ 45  
*kitābu-hu fi l-‘Ādāt*.<sup>42</sup>

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.}.
8 \end{arab}

```

إِنَّ أَيْ كَانَ مِنَ الْمُقَاتِلَةِ<sup>a</sup>، وَكَانَتْ أُمِّي مِنْ عُظَمَاءِ بُيُوتِ الزَّمَازِمَةِ<sup>b</sup>.

<sup>a</sup>المُقَاتِلَةُ: الْمُقَاتِلِينَ.  
<sup>b</sup>الزَّمَازِمَةُ: 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]{\uc{z}ayd-uN\LRfootnote{%
2 \enquote{\arb[trans]{\uc{z}ayd} is the son of
3 \arb[trans]{\uc{`a}mr}}: the second
4 noun is not in apposition to the first, but forms
5 part of the predicate\ldots} "ibn-u \uc{`a}mr-iNU}

```

<sup>42</sup>This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.



زَيْدٌ<sup>a</sup> ابْنُ حَمْرٍو

<sup>a</sup> “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.<sup>43</sup>

`\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 39 applies.

`\setRL` `\setLR` and `\setLR` may be used to change the direction of paragraphs, either  
`\setLR` 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}
```

بَرْزَوِيهِ لِبُزْرَجْمَهَرِ بْنِ الْبُخْتِكَانِ  
 قَالَ بَرْزَوِيهِ بْنُ أَزْهَرَ، رَأْسُ أَطِبَّاءِ فَارِسَ...

## 10.1 New commands

In some particular cases, it may be useful to define new commands to be inserted in Arabic environments. From the general principle laid on page 39, it follows that any command that is found inside an Arabic environment is assumed to have Arabic

<sup>43</sup>See <http://ctan.org/pkg/koma-script>; read the documentation of KOMA-script for details about the `\deffootnotemark` and `\deffootnote` commands.

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, ...}
```

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:—<sup>44</sup>

```
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 ﴿١﴾ وَلَا بِنَاءَ لَهُمْ يَوْمَئِذٍ مِنَ الشَّمْسِ.

## 10.2 Environments

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]
```

<sup>44</sup>See below section 10.4 on page 45 for more information on two-argument commands.

```

3 kAna \uc{'abU} \uc{'l-hu_dayli} 'ahd_A 'il_A \uc{muwaysiN}
4 dajAjaTaN. wa-kAnat dajAjatu-hu 'llatI 'ahdA-hA dUna mA kAna
5 yuttaxa_du li-\uc{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 <\uc{'abU} \uc{'l-hu_dayli}>
8 yu`rafu
9 bi-'l-'imsAki 'l-`sadIdi. fa-qAla: \enquote{wa-kayfa ra'ayta yA
10 \uc{'abA} \uc{'imrAna} tilka 'l-dajAjaTa?} qAla:
11 \enquote{kAnat `ajabaN mina 'l-`ajabi!} fa-yaqUlu:
12 \begin{quoting}[beginntext=», endntext=«]
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.
15 wa-tadrI
16 bi-'ayyi `say'iN kunna nusamminu-hA? wa-fI 'ayyi makAniN
17 kunna
18 na`lifu-hA?
19 \end{quoting}
20 fa-lA yazAlu fI h_a_dA wa-'l-'A_haru ya.d.haku .da.hkaN
21 na`rifu-hu
22 na.hnu wa-lA ya`rifu-hu \uc{'abU} \uc{'l-hu_dayli}.
23 \end{arab}

```

كَانَ أَبُو الْهَدَيْلِ أَهْدَى إِلَى مُوسَى دَجَاجَةً. وَكَانَتْ دَجَاجَتُهُ الَّتِي أَهْدَاهَا دُونَ مَا كَانَ يَتَّخِذُ لِمُوسَى. وَلَكِنَّهُ بِكَرَمِهِ وَبِحُسْنِ خُلُقِهِ أَظْهَرَ التَّعَجُّبَ مِنْ سَمَنِهَا وَطِيبِ حَمِّهَا. وَكَانَ <أَبُو الْهَدَيْلِ> يَعْرِفُ بِالْأَمْسَاكِ الشَّدِيدِ. فَقَالَ: "وَكَيْفَ رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ الدَّجَاجَةَ؟" قَالَ: "كَانَتْ عَجْبًا مِنَ الْعَجَبِ!" فَيَقُولُ:

« وَتَدْرِي مَا جَنَسُهَا؟ وَتَدْرِي مَا سَنِهَا؟ فَإِنَّ الدَّجَاجَةَ إِذَا تَطِيبُ بِالْجَنَسِ وَالسِّنِّ. وَتَدْرِي بِأَيِّ شَيْءٍ كُنَّا نُسَمِّيْنَهَا؟ وَفِي أَيِّ مَكَانٍ كُنَّا نَعْلِفُهَا؟ »

فَلَا يَزَالُ فِي هَذَا وَالْآخِرِ يَضْحَكُ ضَحْكًا نَعْرِفُهُ نَحْنُ وَلَا يَعْرِفُهُ أَبُو الْهَدَيْلِ.

### 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-\uc{bArIs} 2860
5 `arabiyuN.
6 \item[s] max.tU.tu 'l-maktabaTi 'l-'ahliyyaTi bi-\uc{bArIs} 2859
7 `arabiyuN.
8 \item[m] max.tU.tu majlisi \arb[novoc]{^sUrAY maly} .tahrAna
9 521.
9 \end{description}
10 \end{arab}

```

رموز الكتاب

ب مخطوط المكتبة الأهلية بباريس ٢٨٦٠ عربيّ.

س مخطوط المكتبة الأهلية بباريس ٢٨٥٩ عربيّ.

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 وَيَشْتَمِلُ عَلَى سِتَّةِ تَعَالِيمٍ

- التَّعْلِيمُ الْأَوَّلُ [وَهُوَ فَصْلَانِ]

- الْفَصْلُ الْأَوَّلُ

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

```
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: “غَرِيبٌ أَمْرُكَ يَا صَدِيقِي أَتَصَدِّقُ الْجَمَّارَ 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}{\uc{m}uha_d_dabu \uc{'l-d}Ini \uc{'a}bdu
3     \uc{'l-r}a.hImi bnu \uc{'a}liyyiN} huwa ^say_hu-nA 'l-'imAmu
```

```

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

```

مَهْدَبُ الدِّينِ عَبْدُ الرَّحِيمِ بْنِ عَلِيٍّ هُوَ شَيْخُنَا الْإِمَامُ الصَّدْرُ الْكَبِيرُ الْعَالِمُ الْفَاضِلُ مَهْدَبُ الدِّينِ أَبُو مُحَمَّدٍ عَبْدُ  
الرَّحِيمِ بْنِ عَلِيٍّ بْنِ حَامِدٍ وَيَعْرِفُ بِالْدَّخْوَرِ.

*Muhaddabu 'd-Dīni 'Abdu 'r-Raḥīmi bnu 'Aliyyi<sup>in</sup> huwa šayḥu-na 'l-  
imāmu 'ṣ-ṣadru 'l-kabīru 'l-ālimu 'l-fādilu Muhaddabu 'd-Dīni 'Abū  
Muḥammad<sup>in</sup> 'Abdu 'r-Raḥīmi bnu 'Aliyyi bni Ḥāmid<sup>in</sup> 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 arablutax 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

arablutax 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 39 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 *sūrat 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 26.

## 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 [2018/02/12 v1.10.3 An ArabTeX-like interface for LuaLaTeX]
4 \RequirePackage{ifluatex}
```

arabluatex requires Lua<sup>A</sup>T<sub>E</sub>X 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 \RequirePackage{xkeyval}
13 \DeclareOptionX{voc}{\def\al@mode{voc}}
14 \DeclareOptionX{fullvoc}{\def\al@mode{fullvoc}}
15 \DeclareOptionX{novoc}{\def\al@mode{novoc}}
16 \DeclareOptionX{trans}{\def\al@mode{trans}}
17 \ExecuteOptionsX{voc}
18 \ProcessOptionsX\relax
19 \def\al@mode@voc{voc}
20 \def\al@mode@fullvoc{fullvoc}
21 \def\al@mode@novoc{novoc}
22 \def\al@mode@trans{trans}

```

Packages that are required by arabluatex:

```

23 \RequirePackage{etoolbox}
24 \RequirePackage{arabluatex-patch}
25 \RequirePackage{fontspec}
26 \RequirePackage{amsmath}
27 \RequirePackage{luacode}
28 \RequirePackage{xparse}
29 \RequirePackage{environ}
30 \RequirePackage{adjustbox}

```

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 T<sub>E</sub>Xlive:

```

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})}}

```

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

`\aemph*` The ‘starred’ version of this command alway puts the stroke over its argument.

```

48 \AtBeginDocument{\ifdef{\aemph}%
49   {\RenewDocumentCommand{\aemph}{s m}{%
50     \IfBooleanTF{#1}{%
51       $\overline{\text{#2}}$}
52       {\luadirect{tex.sprint(aemph(\luastringN{#2}))}}}
53   {\NewDocumentCommand{\aemph}{s m}{%
54     \IfBooleanTF{#1}{%
55       $\overline{\text{#2}}$}
56       {\luadirect{tex.sprint(aemph(\luastringN{#2}))}}}}}

```

`\SetInputScheme` `arabluatex` is designed for processing Arab<sub>T</sub><sub>E</sub>X input notation. `\SetInputScheme` may be used in the preamble or at any point of the document should the user wish to use a different notation such as the ‘Buckwalter scheme’.

```

57 \def\al@input@scheme{arabt看}
58 \NewDocumentCommand{\SetInputScheme}{m}{\def\al@input@scheme{#1}}

```

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

`\SetArbDflt*` As of v1.6, arabuatex does not apply any more the assimilation rules that are laid on item [b](#) on page [17](#); a new starred version `\SetArbDflt*` is now available to the user should he wish to apply them.

```
59 \def\al@arb@rules{dflt}
60 \NewDocumentCommand{\SetArbEasy}{s}{%
61   \IfBooleanTF{#1}
62   {\def\al@arb@rules{easynosukun}}
63   {\def\al@arb@rules{easy}}}
64 \NewDocumentCommand{\SetArbDflt}{s}{%
65   \IfBooleanTF{#1}
66   {\def\al@arb@rules{idgham}}
67   {\def\al@arb@rules{dflt}}}
```

`\SetTranslitFont` By default, the font that is used for transliterated text is the main font of the document. Any other font may also be selected with the font-selecting commands of the `fontspec` package.

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

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

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

`\SetTranslitConvention` `\SetTranslitConvention{<convention>}` may be used to change the transliteration convention, which is `dmg` by default:

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

`\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 `dmg` mode. `\NoArbUp` may be used either in the preamble or at any point of the document in case one wishes to have the *tanwīn* on the line. The default rule can be set back with `\ArbUpDflt` at any point of the document. Finally `\SetArbUp` may be used to customize the way *tanwīn* is displayed: this command takes the formatting directives as argument, like so: `\SetArbUp{<code>}`.

```
74 \NewDocumentCommand{\al@arbup@dflt}{m}{\textsuperscript{\thinspace#1}}%
75 \NewDocumentCommand{\al@arbup}{m}{\al@arbup@dflt{#1}}
76 \NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
77 \NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt}
78 \NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\al@arbup}{m}{##1}}
79 \NewDocumentCommand{\SetArbUp}{m}{\RenewDocumentCommand{\al@arbup}{m}{#1}}
```

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

```
80 \NewDocumentCommand{\uc}{m}%
81   {\luairect{tex.sprint(uc(\luastringN{#1}))}}
```

`\Uc` `\uc` 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. `\Uc` does the same as `\uc` except that *it is never discarded*. For that reason, `\Uc` *should never be used outside the `trans` mode*. `arabluatex` uses `\Uc` internally so as to prevent `\uc` from being discarded in case words that are to be transliterated are inserted into Arabic commands or environments where transliteration is not required. Therefore, it is not documented.

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

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

```
83 \NewDocumentCommand{\prname}{m}{%
84   \bgroup\SetTranslitStyle{\relax}\arb[trans]{\uc{#1}}\egroup}
```

`\txarb` `\txarb` sets the direction to right-to-left and selects the Arabic font. It is used internally by several Lua functions, but available to the user should he wish to insert utf8 Arabic text in his document.

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

```
85 \DeclareDocumentCommand{\txarb}{+m}{\bgroup\textdir
86   TRT\arabicfont#1\egroup}
87 \DeclareDocumentCommand{\txtrans}{+m}{\bgroup\textdir
88   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.

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

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

```
93 \DeclareDocumentCommand{\arb}{0{\al@mode} +m}%
94 {\edef\@tempa{#1}%
95   \ifx\@tempa\al@mode@voc%
96     \bgroup\textdir TRT\arabicfont%
97     \luadirect{tex.sprint(processvoc(\luastringN{#2},
98       \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
99   \else%
100    \ifx\@tempa\al@mode@fullvoc%
101      \bgroup\textdir TRT\arabicfont%
102      \luadirect{tex.sprint(processfullvoc(\luastringN{#2},
103        \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
104    \else%
105    \ifx\@tempa\al@mode@novoc%
```

```

106 \bgroup\textdir TRT\arabicfont%
107 \luadirect{tex.sprint(processnovoc(\luastringN{#2},
108   \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
109 \else%
110 \ifx\@tempa\al@mode@trans%
111 \bgroup\textdir TLT\al@trans@style%
112 \luadirect{tex.sprint(processstrans(\luastringN{#2},
113   \luastring0{\al@trans@convention},
114   \luastring0{\al@arb@rules},
115   \luastring0{\al@input@scheme}}))\egroup%
116 \else%
117 \fi\fi\fi\fi}

```

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

```

118 \NewEnviron{arab}[1][\al@mode]%
119 {\par\edef\@tempa{#1}%
120   \ifx\@tempa\al@mode@voc%
121     \booltrue{al@rlmode}%
122     \bgroup\pardir TRT\textdir TRT\arabicfont%
123     \luadirect{tex.sprint(processvoc(\luastring0{\BODY},
124       \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
125     \else%
126       \ifx\@tempa\al@mode@fullvoc%
127         \booltrue{al@rlmode}%
128         \bgroup\pardir TRT\textdir TRT\arabicfont%
129         \luadirect{tex.sprint(processfullvoc(\luastring0{\BODY},
130           \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
131         \else%
132           \ifx\@tempa\al@mode@novoc%
133             \booltrue{al@rlmode}%
134             \bgroup\pardir TRT\textdir TRT\arabicfont%
135             \luadirect{tex.sprint(processnovoc(\luastring0{\BODY},
136               \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
137             \else%
138               \ifx\@tempa\al@mode@trans%
139                 \bgroup\pardir TLT\textdir TLT\al@trans@style%
140                 \luadirect{tex.sprint(processstrans(\luastring0{\BODY},
141                   \luastring0{\al@trans@convention},
142                   \luastring0{\al@arb@rules},
143                   \luastring0{\al@input@scheme}}))\egroup%
144                 \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:

```

145 \newlength{\al@bayt@width}
146 \newlength{\al@gutter@width}
147 \setlength{\al@bayt@width}{.3\textwidth}

```

```

148 \setlength{\al@gutter@width}{.15\al@bayt@width}
149 \define@key[al]{verse}{width}{\setlength{\al@bayt@width}{#1}}
150 \define@key[al]{verse}{gutter}{\setlength{\al@gutter@width}{#1}}
151 \define@key[al]{verse}{metre}{\arb{#1}}
152 \define@boolkey[al]{verse}{utf}[true]{}
153 \define@boolkey[al]{verse}{delim}[true]{}
154 \define@choicekey[al]{verse}{mode}{fullvoc, voc, novoc,
155   trans}{\def\al@mode{#1}}
156 \presetkeys[al]{verse}{metre={}, utf=false,
157   delim=false}{}

```

Then follows the environment itself:

```

158 \NewDocumentEnvironment{arabverse}{0{}}%
159 {\par\centering\noindent\bgroup\setkeys[al]{verse}[metre]{#1}%
160   \ifx\al@mode\al@mode@trans%
161     \ifal@verse@utf\setRL\else\setLR\fi%
162     \else\setRL\fi}%
163 {\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.

```

164 \NewDocumentCommand{\arb@utf}{m}{%
165   \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
166 \def\al@hemistich@delim{*}
167 \NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
168 \def\al@verse@stroke{\leavevmode\xleaders\hbox{\arb{--}}\hfill\kern0pt}
169 \NewDocumentCommand{\bayt}{m o m}{%
170   \ifdefined\savenotes\savenotes\else\fi%
171   \edef\al@tatweel{--}%
172   \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#1}}%
173   \IfNoValueTF{#2}{%
174     \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
175     \else%
176       \hspace{\al@gutter@width}%
177       \fi
178   }{%
179     \edef\@tempa{#2}%
180     \ifx\@tempa\al@tatweel%
181       \ifx\al@mode\al@mode@trans%
182         \hspace{\al@gutter@width}%
183       \else%
184         \makebox[\al@gutter@width][s]{\al@verse@stroke}%
185       \fi%

```

```

186 \else%
187 \ifx\al@mode\al@mode@trans%
188 \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#2}}%
189 \else%
190 \makebox[\al@gutter@width][s]{\arb@utf{#2}}%
191 \fi\fi}%
192 \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
193 \ifdefined\spewnotes\spewnotes\else\fi%
194 }

\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.
195 \AtBeginDocument{%
196 \ifdefined\abjad%
197 \RenewDocumentCommand{\abjad}{m}%
198 {\luadirect{tex.sprint(abjadify(#1))}}%
199 \else%
200 \NewDocumentCommand{\abjad}{m}%
201 {\luadirect{tex.sprint(abjadify(#1))}}
202 \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.
203 \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.
204 \NewDocumentCommand{\abraces}{+m}{%
205 \luadirect{tex.sprint(abraces(\luastringN{#1}))}}

\LRmarginpar \LRmarginpar is supposed to be inserted in an Arabic environment. It typesets his
argument in a marginal note from left to right.
206 \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.
207 \DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pdir
208 TLT\textdir TLT\footnote{#1}\egroup}
209 \DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pdir
210 TRT\textdir TRT\footnote{#1}\egroup}

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

```

```

211 \NewDocumentCommand{\FixArbFtnmk}{-}{%
212   \@ifpackageloaded{scrextend}%
213   {\AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%
214   {\RequirePackage{scrextend}
215    \AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}}

```

That is it. Say goodbye before leaving.

## Patches

```

216 \NeedsTeXFormat{LaTeX2e}
217 \ProvidesPackage{arabluatex-patch}%
218 [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:

```

219 \def\list#1#2{%
220   \ifnum \@listdepth >5\relax
221     \@toodeep
222   \else
223     \global\advance\@listdepth\@ne
224   \fi
225   \rightmargin\z@
226   \listparindent\z@
227   \itemindent\z@
228   \csname @list\romannumeral\the\@listdepth\endcsname
229   \def\@itemlabel{#1}%
230   \let\makelabel\@mklab
231   \@nmbrrlistfalse
232   #2\relax
233   \@trivlist
234   \parskip\parsep
235   \parindent\listparindent
236   \advance\linewidth -\rightmargin
237   \advance\linewidth -\leftmargin

```

patch begins:

```

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

```

patch ends.

```

240   \parshape \@ne \@totalleftmargin \linewidth
241   \ignorespaces}
242 \def\@item[#1]{%
243   \if@nopritem
244     \@donopritem
245   \else

```

```

246 \if@inlabel
247 \indent \par
248 \fi
249 \ifhmode
250 \unskip\unskip \par
251 \fi
252 \if@newlist
253 \if@nobreak
254 \@nbitem
255 \else
256 \addpenalty\@beginparpenalty
257 \addvspace\@topsep
258 \addvspace{-\parskip}%
259 \fi
260 \else
261 \addpenalty\@itempenalty
262 \addvspace\itemsep
263 \fi
264 \global\@inlabeltrue
265 \fi
266 \everypar{%
267 \@minipagefalse
268 \global\@newlistfalse
269 \if@inlabel
270 \global\@inlabelfalse
271 {\setbox\z@\lastbox
272 \ifvoid\z@
273 \kern-\itemindent
274 \fi}%
275 \box\@labels
276 \penalty\z@
277 \fi
278 \if@nobreak
279 \@nobreakfalse
280 \clubpenalty \@M
281 \else
282 \clubpenalty \@clubpenalty
283 \everypar{}%
284 \fi}%
285 \if@noitemarg
286 \@noitemargfalse
287 \if@nmbrlist
288 \refstepcounter\@listctr
289 \fi
290 \fi
patch begins:
291 \ifbool{al@rlmode}{\sRLbox\@tempboxa{\makelabel{#1}}}{%
292 \sbox\@tempboxa{\makelabel{#1}}}%
293 \ifbool{al@rlmode}{\global\setbox\@labels\hbox dir TRT}%

```



```

294 {\global\setbox\@labels\hbox}{%
patch ends.
295 \unhbox\@labels
296 \hskip \itemindent
297 \hskip -\labelwidth
298 \hskip -\labelsep
299 \ifdim \wd\@tempboxa >\labelwidth
300 \box\@tempboxa
301 \else
302 \hbox to\labelwidth {\unhbox\@tempboxa}%
303 \fi
304 \hskip \labelsep}%
305 \ignorespaces}

```

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

```

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

```

## References

- Brockelmann, Carl et al. (1935). “Die Transliteration der arabischen Schrift in ihrer Anwendung auf die Hauptliteratursprachen der islamischen Welt”. In: *Denkschrift dem 19. internationalen Orientalistenkongreß in Rom vorgelegt von der Transkriptionkommission der Deutschen Morgenländischen Gesellschaft*. In collab. with Ph. S. van Ronkel and Otto Spies. Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockhaus. URL: [http://www.naher-osten.uni-muenchen.de/studium\\_lehre/werkzeugkasten/dmgtransliteration.pdf](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 (Sept. 20, 2015). *Amiri*. URL: <http://www.amirifont.org/>.
- DIN 31 635 (July 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 (Nov. 3, 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	v1.6.	<b>arabverse</b> : New environment <b>arabverse</b> for typesetting Arabic poetry . . . . .	52
v1.0.1.	General: Minor update of the documentation . . . . .	1	<b>\bayt</b> : New macro <b>\bayt</b> for typesetting each verse inside the <b>arabverse</b> environment . . . . .	53	
v1.1.	<b>\abjad</b> : New and more flexible <b>\abjad</b> command. . . . .	54	<b>\SetArbDflt*</b> : This starred version applies the assimilation rules in addition to what <b>\SetArbDflt</b> already does. . . . .	50	
v1.2.	<b>\SetArbEasy</b> : New <b>\SetArbEasy</b> / <b>\SetArbDflt</b> for ‘modern’ or ‘classic’ Arabic styles. . . . .	49	<b>\SetHemistichDelim</b> : New <b>\SetHemistichDelim</b> command for changing the default delimiter between hemistichs . . .	53	
v1.3.	<b>\arabup</b> : <i>ʾrāb</i> is now written as superscript text in <b>dmg</b> mode by default. . . . .	50	v1.7.	<b>\arbnull</b> : New <b>\arbnull</b> command for putting back on any contextual analysis rule broken by other commands. . . . .	54
v1.4.	<b>\SetInputScheme</b> : <b>\SetInputScheme</b> may be used to process other input schemes such as ‘Buckwalter’ . . . . .	49	v1.8.	General: <b>arabica</b> transliteration standard is now supported . . .	32
	<b>\SetTranslitFont</b> : For selecting a specific font for transliterated texts . . . . .	50	v1.8.5.	General: Six additional Persian characters are now available . .	10
v1.4.3.	<b>\abraces</b> : New <b>\abraces</b> command which expresses its argument between braces. . . . .	54	v1.9.	<b>\MkArbBreak</b> : New <b>\MkArbBreak</b> command for inserting user-defined macros in Arabic environments . . . . .	49
v1.4.4.	<b>\SetArbEasy*</b> : this starred version discards the <i>sukūn</i> in addition to what is already discarded by <b>\SetArbEasy</b> . . . . .	49	v1.9.2.	<b>\aemph*</b> : Starred version which always puts the stroke over its argument . . . . .	49
v1.5.	General: Compatibility with the <b>quran</b> package . . . . .	46	v1.10.	General: <b>\uc</b> supersedes <b>\cap</b> . . . .	33
	Environments may be nested inside the <b>arab</b> environment . .	42	<b>\prname</b> : New command for typesetting Arabic proper names in transliteration . . . . .	51	
	<b>txarab</b> : New <b>txarab</b> environment for typesetting running paragraphs in unicode Arabic . .	51			

# 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		
\@M . . . . .	280	\adjustbox 172, 188, 192
\@beginparpenalty .	256	\advance . . . . . 223, 236, 237, 238, 239
\@clubpenalty . . . . .	282	\aemph . . . . . 26, <u>48</u>
\@donoparitem . . . . .	244	\aemph* . . . . . 27, <u>48</u>
\@ifpackageloaded .	212	\al@arb@rules 59, 62, 63, 66, 67, 98, 103, 108, 114, 124, 130, 136, 142
\@inlabelfalse . . . . .	270	\al@arbup . . . . . . 75, 76, 77, 78, 79
\@inlabeltrue . . . . .	264	\al@arbup@dflt . . . . . . . . . . 74, 75, 77
\@item . . . . .	242	\al@bayt@width . . . . . . . . . . 145, 147, 148, 149, 172, 192
\@itemlabel . . . . .	229	\al@gutter@width . . . . . . . . . . 146, 148, 150, 174, 176, 182, 184, 188, 190
\@itempenalty . . . . .	261	\al@hemistich@delim . . . . . . . . . . 166, 167, 174
\@labels . . . . .	275, 293, 294, 295	\al@input@scheme . . . . . . . . . . 57, 58, 98, 103, 108, 115, 124, 130, 136, 143
\@listctr . . . . .	288	\al@mode . . . . . 13, 14, 15, 16, 93, 118, 155, 160, 181, 187
\@listdepth 220, 223, 228		\al@mode@fullvoc . . . . . . . . . . 20, 100, 126
\@minipagefalse . . . . .	267	\al@mode@novoc . . . . . . . . . . 21, 105, 132
\@mklab . . . . .	230	\al@mode@trans . . . . . . . . . . 22, 110, 138, 160, 181, 187
\@nbitem . . . . .	254	\al@mode@voc 19, 95, 120
\@ne . . . . .	223, 240	\al@tatweel . . . . . 171, 180
\@newlistfalse . . . . .	268	\al@trans@convention . . . . . . . . . . 72, 73, 113, 141
\@nmbrrlistfalse . . . . .	231	\al@trans@font . . . . . . . . . . 68, 69, 88
\@nbreakfalse . . . . .	279	\al@trans@style . . . . . . . . . . 70, 71, 111, 139
\@noitemargfalse . . . . .	286	\al@verse@stroke . . . . . . . . . . 168, 184
\@tempa . . . . .	94, 95, 100, 105, 110, 119, 120, 126, 132, 138, 179, 180	<i>Amiri</i> . . . . . 5
\@tempboxa . . . . .	291, 292, 299, 300, 302	amiri (package) . . . . . 5
\@toodeep . . . . .	221	arab (environment) 8, <u>118</u>
\@topsep . . . . .	257	<i>ArabTeX</i> . . . . . 3, 21
\@totalleftmargin . . . . .	238, 239, 240	<i>Arabic-English Lexicon</i> , <i>An</i> . . . . . 20
\@trivlist . . . . .	233	\arabicfont . . . . . . . . . . 34, 35, 37, 86, 92, 96, 101, 106, 122, 128, 134
\_ . . . . .	35	arabluatex (package) . . . . . 2, 4–8, 11–21, 26, 27, 29, 31– 37, 39–42, 45–51
A		arabtex (package) . . . . . . . . . . 3, 4, 17, 21, 47
\abjad . . . . .	25, <u>195</u>	arabulatex (package) . . . . . 17
\abraces . . . . .	31, <u>204</u>	arabverse (environ- ment) . . . . . 27, <u>145</u>
\addpenalty . . . . .	256, 261	arabxetex (package) . . . . . 4, 17
\addvspace 257, 258, 262		\arb . . . . . 7, 84, <u>93</u> , 151, 165, 168
		\arb@utf . . . . . 164, 172, 188, 190, 192
		\arbnul . . . . . 24, <u>203</u>
		\arbup . . . . . 34, <u>74</u>
		\ArbUpDflt . . . . . 34, <u>74</u>
		B
		babel (package) . . . . . 45
		babel-french (package) . . . . . 45
		\bayt . . . . . 28, <u>164</u>
		bidi (package) . . . . . 57
		\BODY 123, 129, 135, 140
		\boolfalse . . . . . 39
		\booltrue . . . . . 38, 91, 121, 127, 133
		\box . . . . . 275, 300

Brockelmann, Carl ...  
       ..... 10, 32, 34

**C**

\centering ..... 159  
\clubpenalty .. 280, 282  
\color@endgroup ... 307  
\color@setgroup ... 307  
csquotes (package) ... 45

**D**

\DeclareOptionX ...  
       ..... 13, 14, 15, 16  
\deffootnote .. 213, 215  
\define@boolkey 152, 153  
\define@choicekey . 154  
\define@key 149, 150, 151  
delim (option) ..... 27

**E**

environments:  
   arab ..... 8, 118  
   arabverse .. 27, 145  
   txarab ..... 39, 89  
\everypar .... 266, 283  
\ExecuteOptionsX ... 17

**F**

\FixArbFtnmk ... 41, 211  
fontspec (package) ...  
       ..... 5, 32, 50  
\footnote .... 208, 210  
footnote (package) .... 29  
footnotehyper (package) 29  
fullvoc (option) .... 5, 9

**G**

\global ... 223, 264,  
       268, 270, 293, 294  
*Grammar of the Arabic*  
*Language, A* ..  
       5–7, 9, 13, 15–  
       17, 19, 25, 26, 34  
gutter (option) ..... 27

**H**

Habash, Nizar Y. .... 37  
\hbox ..... 168,  
       293, 294, 302, 306  
\Height .. 172, 188, 192

\hfill ..... 163, 168  
Hosny, Khaled ..... 5  
\hskip 296, 297, 298, 304  
\hspace ..... 176, 182

**I**

\if@inlabel .. 246, 269  
\if@newlist ..... 252  
\if@nmbrrlist ..... 287  
\if@nobreak .. 253, 278  
\if@noitemarg ..... 285  
\if@noparitem ..... 243  
\ifal@verse@delim . 174  
\ifal@verse@utf 161, 165  
\ifbool .. 238, 291, 293  
\IfBooleanTF .....  
       ..... 50, 54, 61, 65  
\ifdim ..... 299  
\ifhmode ..... 249  
\ifluatex ..... 5  
\IfNoValueTF ..... 173  
\ifnum ..... 220  
\ifvoid ..... 272  
\ignorespaces . 241, 305  
\indent ..... 247

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

*Introduction to Arabic  
Natural Lan-  
guage Process-  
ing* ..... 37

\itemindent 227, 273, 296  
\itemsep ..... 262  
\itshape ..... 70

**K**

\kern ..... 168, 273  
KOMAScript (package) 41

**L**

\labelsep .... 298, 304  
\labelwidth 297, 299, 302

Lagally, Klaus ..... 3, 21  
Lane, Edward William . 20  
\lastbox ..... 271  
\leavevmode ..... 168  
\leftmargin .. 237, 239  
lineno (package) ..... 30  
\linewidth 236, 237, 240  
\list ..... 219  
\listparindent 226, 235  
\long ..... 306  
\LR ... 40, 40, 213, 215  
\LRfootnote ... 40, 207  
\LRmarginpar ... 41, 206  
luabidi (package) ... 4, 48  
\luadirect 32, 33, 47,  
       52, 56, 81, 97,  
       102, 107, 112,  
       123, 129, 135,  
       140, 198, 201, 205  
luaotfload (package) ... 32  
\luastringN .... 47,  
       52, 56, 81, 97,  
       102, 107, 112, 205  
\luastringO .....  
       . 98, 103, 108,  
       113, 114, 115,  
       123, 124, 129,  
       130, 135, 136,  
       140, 141, 142, 143

**M**

\makebox . 174, 184, 190  
\makelabel 230, 291, 292  
\marginpar ..... 206  
metre (option) ..... 27  
\MkArbBreak .... 42, 46  
mode (option) ..... 27

**N**

nameauth (package) ... 34  
\newbool ..... 31  
\NewDocumentEnvironment  
       ..... 89, 158  
\NewEnviron ..... 118  
\newfontfamily ..... 37  
\newlength ... 145, 146  
\NoArbUp ..... 34, 74  
\noindent ..... 159  
novoc (option) ..... 5, 9

<b>O</b>		<code>\rightmargin</code> . . . . .	<code>\textsuperscript</code> . . . 74
options:		. . . . . 225, 236, 238	<code>\textwidth</code> . . . . . 147
<code>delim</code> . . . . . 27	<code>\RL</code> . . . . . 40, 43	<code>\the</code> . . . . . 228	<code>\thefootnotemark</code> . . . . . 213, 215
<code>fullvoc</code> . . . . . 5, 9	<code>\RLfootnote</code> . . . 40, 207	<code>\thinspace</code> . . . . . 74	<code>trans</code> (option) . . . . . 5, 9
<code>gutter</code> . . . . . 27	<code>\rmfamily</code> . . . . .	<i>Transliteration der arabischen Schrift, Die</i> . . . 10, 32, 34	<code>txarab</code> (environment) . . . . . 39, 89
<code>metre</code> . . . . . 27	. 41, 42, 44, 45, 68		<code>txarab</code> (environment) . . 46
<code>mode</code> . . . . . 27	<code>\romannumeral</code> . . . . 228		<code>\txarb</code> . . . . . 39, 85, 165
<code>novoc</code> . . . . . 5, 9	<b>S</b>		<code>\txtrans</code> . . . . . 85
<code>trans</code> . . . . . 5, 9	<code>\savenotes</code> . . . . . 170	<b>U</b>	
<code>utf</code> . . . . . 27	<code>\sbox</code> . . . . . 292	<code>\Uc</code> . . . . . 82	
<code>voc</code> . . . . . 5, 8	<code>scrextend</code> (package) 41, 54	<code>\uc</code> . . . . . 33, 80, 82, 84	
<code>width</code> . . . . . 27	<code>\SetArbDflt</code> . . . . . 6, 59	<code>\unhbox</code> . . . . . 295, 302	
<code>\overline</code> . . . . . 51, 55	<code>\SetArbDflt*</code> . . . . . 6, 59	<code>\unskip</code> . . . . . 250	
<b>P</b>		<code>utf</code> (option) . . . . . 27	
<code>\par</code> . . . . . 90, 92, 119, 144, 159, 247, 250	<code>\SetArbEasy</code> . . . . . 6, 59	<b>V</b>	
<code>paralist</code> (package) . 43, 45	<code>\SetArbEasy*</code> . . . . . 6, 59	<code>voc</code> (option) . . . . . 5, 8	
<code>\pardir</code> . . . . . 38, 39, 92, 122, 128, 134, 139, 207, 209	<code>\SetArbUp</code> . . . . . 35, 74	<b>W</b>	
<code>\parindent</code> . . . . . 235	<code>\setbox</code> . . . . . 271, 293, 294, 306	<code>\wd</code> . . . . . 299	
<code>\parsep</code> . . . . . 234	<code>\SetHemistichDelim</code> . . . . . 27, 164, 167	<code>width</code> (option) . . . . . 27	
<code>\parshape</code> . . . . . 240	<code>\SetInputScheme</code> . 37, 57	<code>Wright, W. LL.D</code> . . . . . 5–7, 9, 13, 15–17, 19, 25, 26, 34	
<code>\parskip</code> . . . . . 234, 258	<code>\setkeys</code> . . . . . 159, 163		
<code>\penalty</code> . . . . . 276	<code>\setlength</code> . . . . . 147, 148, 149, 150	<b>X</b>	
<code>polyglossia</code> (package) . . . . . 4, 48, 54	<code>\setLR</code> . . . . . 39, 41, 161	<code>\xleaders</code> . . . . . 168	
<code>\presetkeys</code> . . . . . 156	<code>\setRL</code> . 38, 41, 161, 162	<b>Z</b>	
<code>\prname</code> . . . . . 33, 83	<code>\SetTranslitConvention</code> . . . . . 32, 72	<code>\z@</code> . . . . . 225, 226, 227, 271, 272, 276	
<code>\ProcessOptionsX</code> . . . 18	<code>\SetTranslitFont</code> 32, 68		
<code>\ProvidesPackage</code> 2, 217	<code>\SetTranslitStyle</code> . . . . . 32, 70, 84		
<b>Q</b>			
<code>quoting</code> (package) . . . . 42	<code>\spewnotes</code> . . . . . 193		
<code>quran</code> (package) . . 46, 47	<code>\sRLbox</code> . . . . . 291, 306		
<b>R</b>		<b>T</b>	
<code>\refstepcounter</code> . . . 288	<code>\text</code> . . . . . 51, 55		
<code>reledmac</code> (package) . . . 46	<code>\textdir</code> . . . . . 38, 39, 41, 42, 44, 45, 85, 87, 92, 96, 101, 106, 111, 122, 128, 134, 139, 206, 208, 210		
<code>\RequirePackage</code> . . . . . 4, 12, 23, 24, 25, 26, 27, 28, 29, 30, 214			