

# arabluatex

ArabT<sub>E</sub>X for LuaL<sup>A</sup>T<sub>E</sub>X

Robert Alessi

<mailto:alessi@robertalessi.net>

2023/05/21 – v1.21-dev

## Contents

License and disclaimer	2	4.10.1 Underlining words or numbers	25
<b>1 Introduction</b>	<b>3</b>	<b>5 Arabic poetry</b>	<b>25</b>
1.1 arabluatex is for LuaL <sup>A</sup> T <sub>E</sub> X	4	Scaling and distortion of characters	27
<b>2 The basics of arabluatex</b>	<b>4</b>	Footnotes	27
2.1 Activating arabluatex	4	Critical Notes	28
Font setup	4	Line numbering	28
2.2 Options	5	5.1 Example	28
2.2.1 Classic contrasted with modern typesetting of Arabic	5	<b>6 Special applications</b>	<b>30</b>
2.3 Typing Arabic	7	Linguistics	30
2.3.1 Local options	8	Brackets	30
<b>3 Standard ArabT<sub>E</sub>X input</b>	<b>8</b>	Additional Arabic marks	30
3.1 Consonants	8	The ‘Zero width joiner’ character (U+200D)	31
3.2 Additional characters	9	6.1 The Qur’ān	32
3.3 Vowels	10	Caveat	33
3.3.1 Long vowels	10	<b>7 Color</b>	<b>33</b>
3.3.2 Short vowels	11	7.1 Tricks of the trade	34
<b>4 arabluatex in action</b>	<b>11</b>	Diacritics	34
4.1 The vowels and diphthongs	11	<i>tanwīn</i>	35
Short vowels	11	<i>waslah</i> and <i>maddah</i>	36
<i>‘alif maqṣūrah</i>	12	<i>šaddah</i>	36
<i>‘alif otiosum</i>	12	The definite article and the euphonic <i>tašdīd</i>	36
<i>‘alif maḥdūfah</i> and defective <i>ū, ī</i>	12	<i>hamzah</i>	37
Silent <i>و, ي</i>	12	<b>8 Transliteration</b>	<b>37</b>
<i>‘Amr<sup>um</sup></i> , and the silent <i>ي</i>	12	Convention	37
<i>tanwīn</i>	13	Style	37
4.2 Other orthographic signs	13	Font	37
<i>tā’ marbū‘ah</i>	13	Proper names	38
<i>hamzah</i>	13	Hyphenation	38
<i>maddah</i>	15	‘Long’ proper names	38
<i>šaddah</i>	15	Proper names outside Arabic environments	38
The definite article and the <i>‘alif<sup>u</sup> ‘l-waṣl<sup>i</sup></i>	16	8.1 Additional note on <i>dmg</i> convention	39
4.3 Special orthographies	18	<i>‘irāb</i> boundaries	40
The name of God	18	Discarding the <i>‘irāb</i>	40
The conjunctive <i>الَّذِي</i>	18	Uncertain short vowels	41
4.4 Quoting	19	8.2 Examples	41
novoc	19	<b>9 Buckwalter input scheme</b>	<b>41</b>
voc	20	‘base’, ‘xml’ and ‘safe’ schemes	42
fullvoc	20	Transliteration	43
4.4.1 Quoting the <i>hamzah</i>	20	<b>10 Unicode Arabic input</b>	<b>43</b>
The ‘pipe’ character ( )	21	<b>11 L<sup>A</sup>T<sub>E</sub>X Commands in Arabic environments</b>	<b>44</b>
4.6 Putting back on broken contextual analysis rules	21	General principle	44
4.7 Stretching characters: the <i>taṭwīl</i>	23	11.1 New commands	46
4.8 Digits	23	11.2 Environments	47
4.8.1 Numerical figures	23	11.2.1 Lists	47
4.8.2 The <i>abjad</i>	23	رموز الكأب	48
4.9 Additional characters	24	Caveat	49
4.10 Arabic emphasis	24		

11.3	csquotes . . . . .	50	12.3	Further processing of Unicode converted files . . . . .	53
11.4	Two-argument special commands . . . . .	50			
	textcolor 50 reledmac 51				
11.5	quran . . . . .	51	13	Future work	53
12	Exporting Unicode Arabic to an external file	51	14	Implementation	54
12.1	Commands and environments . . . . .	52	15	References	68
	export global option 52 Exporting running paragraphs 52 Appending words or commands to the external file only 52 Exporting Arabic poetry 53		16	Change History	69
12.2	Nested Arabic environments . . . . .	53	17	Index	70

## List of Tables

1	Standard ArabTeX (consonants) . . . . .	8	5	“Quoted” <i>hamzah</i> . . . . .	21
2	Standard ArabTeX (additional characters) . . . . .	10	6	Additional Arabic codings . . . . .	24
3	Standard ArabTeX (long vowels) . . . . .	10	7	Additional Arabic marks . . . . .	30
4	Standard ArabTeX (short vowels) . . . . .	11	8	ArabTeX diacritics for <code>\arbcolor</code> . . . . .	34
			9	Buckwalter input scheme . . . . .	42

### Abstract

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

## License and Disclaimer

`OpenBSD` `arabluatex` is licensed under the terms of the so-called `OpenBSD` license, as it is modelled after the `ISC` copyright, which is functionally equivalent to a two-term `BSD` copyright with language removed that is made unnecessary by the Berne convention.<sup>1</sup>

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

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

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

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

<sup>1</sup>More information about the `OpenBSD` policy to which `arabluatex` adheres: <https://www.openbsd.org/policy.html>.

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

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

This release of arabluatex consists of the following source files:

- arabluatex.ins
- arabluatex.dtx
- arabluatex.lua
- arabluatex\_voc.lua
- arabluatex\_fullvoc.lua
- arabluatex\_novoc.lua
- arabluatex\_trans.lua
- arabluatex.el

## 1 Introduction

In comparison to Prof. Lagally’s outstanding Arab $\TeX$ ,<sup>2</sup> ArabLua $\TeX$  is at present nothing more than a modest piece of software. Hopefully—if I may say so—it will eventually provide all of its valuable qualities to the Lua $\LaTeX$  users.

arab $\TeX$  dates back to 1992. As far as I know, it was then the first and only way to typeset Arabic texts with  $\TeX$  and  $\LaTeX$ . To achieve that, arab $\TeX$  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>3</sup> Even if the same can be said about Unicode, Arab $\TeX$  ASCII input notation still surpasses Unicode input, in my opinion, when it comes to typesetting complex documents, such as scientific documents or critical editions where footnotes and other kind of annotations can be particularly abundant. It must also be said that most text editors have trouble in displaying Arabic script connected with preceding or following  $\LaTeX$  commands: it often happens that commands seem misplaced, not to mention punctuation marks, or opening or closing braces, brackets or parentheses that are unexpectedly displayed in the wrong direction. Of course, some text editors provide ways to get around such difficulties by inserting invisible Unicode characters, such as LEFT-TO-RIGHT or RIGHT-TO-LEFT MARKS (U+200E, U+200F), RTL/LTR “embed” characters (U+202B, U+202A) and RLO/LRO “bidi-override” characters (U+202E, U+202D).<sup>4</sup> Nonetheless, it remains that inserting all the time these invisible characters in complex documents rapidly becomes confusing and cumbersome.

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

---

<sup>2</sup>See <http://ctan.org/pkg/arabtex>

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

<sup>4</sup>Gáspár Sinai’s Yudit probably has the best Unicode support. See <http://www.yudit.org>.

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, ArabT<sub>E</sub>X input can be full vocalized Arabic (*scriptio plena*), vocalized Arabic or non-vocalized Arabic (*scriptio defectiva*); it further can be transliterated into whichever romanization standard the user may choose.

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

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

## 1.1 `arabluatex` is for LuaL<sup>A</sup>T<sub>E</sub>X

It goes without saying that `arabluatex` requires LuaL<sup>A</sup>T<sub>E</sub>X. T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X have `arabtex`, and X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X has `arabxetex`. Both of them are much more advanced than `arabluatex`, as they can process a number of different languages,<sup>5</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<sub>E</sub>X input on the fly into Unicode Arabic script, whereas the latter passes ArabT<sub>E</sub>X input on to a set of Lua functions. At first, L<sup>A</sup>T<sub>E</sub>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<sub>E</sub>X the converted strings—which may in turn contain some other ArabT<sub>E</sub>X input yet to be processed—for further processing.

## 2 The basics of `arabluatex`

### 2.1 Activating `arabluatex`

`arabluatex` is loaded the usual way:

```
\usepackage{arabluatex}
```

The only requirement of `arabluatex` is LuaL<sup>A</sup>T<sub>E</sub>X; it will complain if the document is compiled with another engine. That aside, `arabluatex` does not load packages such as `polyglossia`. Although it can work with `polyglossia`, it does not require it.

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

---

<sup>5</sup>To date, both packages support Arabic, Maghribi, Urdu, Pashto, Sindhi, Kashmiri, Uighuric and Old Malay; in addition to these, `arabtex` also has a Hebrew mode, including Judeo-Arabic and Yiddish.

```
\newfontfamily\arabicfont{<fontname>}[Script=Arabic]
```

where *<fontname>* is the standard name of the Arabic font to be used.

By default, if no Arabic font is selected, `arabluatex` will issue a warning message and attempt to load the Amiri font<sup>6</sup> like so:—

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

REM. *a* By default Amiri places the *kasrah* in combination with the *tašdīd* below the consonant, like so: َ.

That is correct, as at least in the oldest manuscripts َ may stand for َ as well as َ. See Wright.<sup>7</sup>

The placement of the *kasrah* above the consonant may be obtained by selecting the `ss05` feature of the Amiri font, like so:—<sup>8</sup>

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

Other Arabic fonts may behave differently.

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

## 2.2 Options

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

`voc`

Default

In this mode, which is the one selected by default, every short vowel written generates its corresponding diacritical mark: *ḍammah* (َ), *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 (أَ).

`fullvoc`

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

`novoc`

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

`trans`

This mode transliterates the ArabTeX input into one of the accepted standards. At present, three standards are supported (see below [sect. 8 on page 37](#) for more details):

**dmg/dmg+** *Deutsche Morgenländische Gesellschaft* `dmg` is selected by default;

**loc** *Library of Congress*;

**arabica** *Arabica*.

More standards will be included in future releases of `arabluatex`.

`export`      `export=true|false`

Default: false

*New feature v.1.13*

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

<sup>6</sup>Khaled Hosny, *Amiri* (Dec. 13, 2017), <http://www.amirifont.org/>.

<sup>7</sup>W. LL.D Wright, *A Grammar of the Arabic Language*, rev. W. Robertson Smith and M. J. de Goeje, with a foreword by Pierre Cachia, 2 vols. (3rd edn., Beirut: Librairie du Liban, 1896), i. 14 C–D.

<sup>8</sup>See the documentation of `amiri`, *ibid.*, 6.

### 2.2.1 Classic contrasted with modern typesetting of Arabic

*New feature v.1.2* By default, arabxetex 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 سَمَاءٌ *samā<sup>un</sup>*, jA'a جَاءَ *jā'a*, yataSA'alUna يَتَسَاءَلُونَ *yatasā'alūna*<sup>9</sup>  
(see on page 15 for further details).

- The euphonic *tašdīd* is generated (see on page 15).
- In `fullvoc` mode, the *sukūn* is expressed.
- In such words as *شَيْئًا* and *ظَمًا* and the like, the *hamzah* alone is not written over the letter *yā* with no diacritical points below as in *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`.

`\SetArbEasy` Such refinements as ‘classic’ *maddah* may be discarded by the `\SetArbEasy` command, *New feature v1.4.4* either globally in the preamble or locally at any point of the document. The difference

`\SetArbEasy*` between `\SetArbEasy` and its ‘starred’ version `\SetArbEasy*` is that the former keeps the *sukūn* that is generated by the `fullvoc` mode, while the latter further takes it away.

`\SetArbDflt` Default ‘classic’ rules may be set back at any point of the document with the `\SetArbDflt` *New feature v1.6* command. Assimilation rules laid on (b) on page 15 may also be applied by the ‘starred’

`\SetArbDflt*` version of this command `\SetArbDflt*` either in the preamble or at any point of the document.<sup>10</sup> Examples follow:—

- (a) `\SetArbDflt`:
- i. voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
  - ii. fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
  - iii. trans *wa-māta 'stisqā<sup>an</sup> qabla 'an yutimma kitāba-hu fī nuǧūm<sup>i</sup> 's-samā<sup>i</sup>*
- (b) `\SetArbDflt*`:
- i. voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
  - ii. fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
  - iii. trans *wa-māta 'stisqā<sup>an</sup> qabla 'ay yutimma kitāba-hu fī nuǧūm<sup>i</sup> 's-samā<sup>i</sup>*
- (c) `\SetArbEasy`:
- i. voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
  - ii. fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
  - iii. trans *wa-māta 'stisqā<sup>an</sup> qabla 'an yutimma kitāba-hu fī nuǧūm<sup>i</sup> 's-samā<sup>i</sup>*
- (d) `\SetArbEasy*`:
- i. voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ

<sup>9</sup>Note that in old mss. such forms as *سَمَاءٌ*, *جَاءَ* are also found; see W. LL.D Wright, *A Grammar of the Arabic Language*, rev. W. Robertson Smith and M. J. de Goeje, with a foreword by Pierre Cachia, 2 vols. (3rd edn., Beirut: Librairie du Liban, 1896), i. 24 D.

<sup>10</sup>For an example, see [sect. 5.1 on page 28](#).

- ii. fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتَمَّ كِتَابُهُ فِي نُجُومِ السَّمَاءِ
- iii. trans *wa-māta 'stisqā<sup>an</sup> qabla 'an yutimma kitāba-hu fī nuǧūm<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:<sup>a</sup>— Arabic, like Hebrew and Syriac, is written and read from right to left. The letters of the alphabet (حُرُوفُ الْمَجَامِي، حُرُوفُ التَّهْجِي، حُرُوفُ الْمَجَامِي، or حُرُوفُ الْمُجَمِّم) are twenty-eight in number and are all consonants, though three of them are also used as vowels (see § 3).

<sup>a</sup>Wright, see n. 7, i. 1 A.

The following example comes from Wright:<sup>11</sup>—

```

1 \begin{enumerate}[label=\Roman*., start=16]
2 \item \arb{faw`ilu}*.
3 \begin{enumerate}[label=\arabic*.]
4 \item \arb{fA`alUN}; as \arb{_hAtamuN} \emph{a
5 signet-ring}, ...
6 \end{enumerate}
7 \end{enumerate}
```

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

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

```

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:
```

<sup>11</sup>Wright, see n. 7, i. 213 C.

```

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 جُحَا يَطْلُبُ مِنْهُ حِمَارَهُ لِيُرْكَبَهُ فِي سَفَرَةٍ قَصِيرَةٍ فَقَالَ لَهُ: "سَوْفَ أُعِيدُهُ إِلَيْكَ فِي الْمَسَاءِ وَأَدْفَعُ لَكَ أُجْرَةً." فَقَالَ جُحَا: "أَنَا أَسَفٌ جِدًّا أَنِّي لَا أَسْتَطِيعُ أَنْ أُحَقِّقَ لَكَ رَغْبَتَكَ فَالْحِمَارُ لَيْسَ هُنَا الْيَوْمَ." وَقَبْلَ أَنْ يُتِمَّ جُحَا كَلَامَهُ بَدَأَ الْحِمَارُ يَنْهَقُ فِي إِصْطِلِهِ. فَقَالَ لَهُ صَدِيقُهُ: "إِنِّي أَسْمَعُ حِمَارَكَ يَا جُحَا يَنْهَقُ." فَقَالَ لَهُ جُحَا: "غَرِيبٌ أَمْرُكَ يَا صَدِيقِي أَتَصَدِّقُ الْحِمَارَ وَتُكَذِّبُنِي؟"

### 2.3.1 Local options

As seen above in [sect. 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]{\Arabic text};
novoc    - \arb[novoc]{\Arabic text};
trans    - \arb[trans]{\Arabic text}.

```

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

## 3 Standard ArabTeX input

### 3.1 Consonants

[Table 1](#) gives the ArabTeX equivalents for all of the Arabic consonants.

Letter	Transliteration <sup>12</sup>			ArabTeX notation
	dmg+	loc	arabica	
أ ؤ	'u, 'a, 'i	'u, 'a, 'i	'u, 'a, 'i	'u or 'a or 'i
ب	b	b	b	b
ت	t	t	t	t
ث	<u>t</u>	th	<u>t</u>	_t

Table 1: Standard ArabTeX (consonants)

<sup>12</sup>See below [sect. 8 on page 37](#).

<sup>13</sup>See below, *Rem. a*. For 'alif as a consonant, see Wright, *A Grammar of the Arabic Language*, i. 16 D. The hamzah itself is encoded <'> and may be followed by either <u, a> or <i>. See below [sect. 4.2 on page 13](#).



Letter	Transliteration		ArabT <sub>E</sub> X notation	
	dmg+	loc	arabica	
ج	<i>ǧ</i>	<i>j</i>	<i>ǧ</i>	$\hat{g}$ or j
ح	<i>ḥ</i>	<i>ḥ</i>	<i>ḥ</i>	.h
خ	<i>ḫ</i>	<i>kh</i>	<i>ḫ</i>	_h or x
د	<i>d</i>	<i>d</i>	<i>d</i>	d
ذ	<i><u>d</u></i>	<i>dh</i>	<i><u>d</u></i>	_d
ر	<i>r</i>	<i>r</i>	<i>r</i>	r
ز	<i>z</i>	<i>z</i>	<i>z</i>	z
س	<i>s</i>	<i>s</i>	<i>s</i>	s
ش	<i>š</i>	<i>sh</i>	<i>š</i>	$\hat{s}$
ص	<i>ṣ</i>	<i>ṣ</i>	<i>ṣ</i>	.s
ض	<i><u>ḍ</u></i>	<i><u>ḍ</u></i>	<i><u>ḍ</u></i>	.d
ط	<i><u>ṭ</u></i>	<i><u>ṭ</u></i>	<i><u>ṭ</u></i>	.t
ظ	<i><u>ẓ</u></i>	<i><u>ẓ</u></i>	<i><u>ẓ</u></i>	.z
ع	<i>ġ</i>	<i>gh</i>	<i>ġ</i>	.g
ف	<i>f</i>	<i>f</i>	<i>f</i>	f
ق	<i>q</i>	<i>q</i>	<i>q</i>	q
ك	<i>k</i>	<i>k</i>	<i>k</i>	k
ل	<i>l</i>	<i>l</i>	<i>l</i>	l
م	<i>m</i>	<i>m</i>	<i>m</i>	m
ن	<i>n</i>	<i>n</i>	<i>n</i>	n
ه	<i>h</i>	<i>ḥ</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>14</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 ʿl-waṣli* is expressed only by the vowel that accompanies the omitted *hamzah*: ⟨*u, a, i*⟩ as in *wa-inhazama* وَأَنْهَزَمَ *wa-ʿnhazama*. For more details on the definite article and the *ʿalifu ʿl-waṣli* see [sect. 4.2 on page 16](#).

That said, *l* 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* وَإِنْ *wa-.n* (where the dot symbolizes the absence of vowel), *wa-.Aan* وَإِنْ *wa-an*, *wa-.Ain* وَإِنْ *wa-in*.

REM. b The letter *ي* with two points below, أَلْيَاءُ الْمُنْتَهَى مِنْ تَحْتِهَا, may also be written without diacritical points as *ى*. When it is used as a consonant, it is encoded *aY*, where *a* recalls the *fathah* placed above the preceding letter in vocalized Arabic, like so: *qaY'uN* قَائِي *qayʿun*, *ˆsaY'uN* سَائِي *ṣayʿun*, *ˆsaY'aN* سَائِي *ṣayʿan*.

The same result may be achieved by encoding this letter as *.y*, like so: *qa.y'uN* قَائِي *qayʿun*, *ˆsa.y'uN* سَائِي *ṣayʿun*, *ˆsa.y'aN* سَائِي *ṣayʿan*.

### 3.2 Additional characters

*New feature v1.8.5* [Table 2 on the next page](#) gives the ArabT<sub>E</sub>X equivalents for some additional Persian characters.

<sup>14</sup>For the letter *ى* with no diacritical points below, see [REM. b](#) below.

Letter	Transliteration <sup>15</sup>			ArabT <sub>E</sub> X notation
	dmg+	loc	arabica <sup>16</sup>	
پ	<i>p</i>	<i>p</i>	<i>p</i>	P
چ	<i>č</i>	<i>ch</i>	<i>č</i>	$\hat{c}$
ژ	<i>ž</i>	<i>zh</i>	<i>ž</i>	$\hat{z}$
ف <sup>17</sup>	<i>v</i>	<i>v</i>	<i>v</i>	v
گ	<i>g</i>	<i>g</i>	<i>g</i>	g
ن <sup>18</sup>	<i>ñ</i>	<i>ñ</i>	<i>ñ</i>	$\hat{n}$

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

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

### 3.3 Vowels

#### 3.3.1 Long vowels

Table 3 gives the ArabT<sub>E</sub>X equivalents for the Arabic long vowels.

Letter	Transliteration <sup>19</sup>			ArabT <sub>E</sub> X notation
	dmg+	loc	arabica	
ا	<i>ā</i>	<i>ā</i>	<i>ā</i>	A
و	<i>ū</i>	<i>ū</i>	<i>ū</i>	U
ي	<i>ī</i>	<i>ī</i>	<i>ī</i>	I <sup>20</sup>
ي <sup>21</sup>	<i>ā</i>	<i>á</i>	<i>ā</i>	_A or Y
اَ	<i>ā</i>	<i>ā</i>	<i>ā</i>	_a
وَ	<i>ū</i>	<i>ū</i>	<i>ū</i>	_u
يَ	<i>ī</i>	<i>ī</i>	<i>ī</i>	_i

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

REM. *a* The long vowels *ā*, *ū*, *ī*, otherwise called *ḥurūf<sup>u</sup> ʾl-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. قَالٌ *qāla*, قِيلٌ *qīla*, يَقُولٌ *yaqūlu*.

<sup>15</sup>See below [sect. 8 on page 37](#).

<sup>16</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>17</sup>This character is not found in Carl Brockelmann et al., “Die Transliteration der arabischen Schrift in ihrer Anwendung auf die Hauptliteratursprachen der islamischen Welt”, in *Denkschrift dem 19. internationalen Orientalistenkongreß in Rom vorgelegt von der Transkriptionkommission der Deutschen Morgenländischen Gesellschaft*, in collab. with Ph. S. van Ronkel and Otto Spies (Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockhaus, 1935), [http://www.naher-osten.uni-muenchen.de/studium\\_lehr\\_e/werkzeugkasten/dmgtransliteration.pdf](http://www.naher-osten.uni-muenchen.de/studium_lehr_e/werkzeugkasten/dmgtransliteration.pdf), 2. It is taken from the *Information and Documentation - Romanization of the Arabic Alphabet for Arabic, Ottoman-Turkish, Persian, Kurdish, Urdu and Pushto* (July 2011), <http://www.din.de> standard.

<sup>18</sup>See [17](#).

<sup>19</sup>See below [sect. 8 on page 37](#).

<sup>20</sup>For the letter ي with no diacritical points, see *Rem. c.* below.

<sup>21</sup>= *al-ʾalif<sup>u</sup> ʾl-maqṣūrat<sup>u</sup>*.

- REM. *b* Defective writings, such as  $\text{ا}$ , *al-ʿalif*<sup>u</sup> *ʿl-mahdūfat*<sup>u</sup>, or defective writings of  $\bar{u}$  and  $\bar{i}$  are encoded `_a_u` and `_i` respectively, e.g. `_d_alika`  $\text{ذَكَ$ , `al-mal_a'ikaT-u`  $\text{لِـرَا.هْمَانـو}$   $\text{الْمَلِكِ الرَّحْمَنِ}$ , `.hu_dayfaT-u bn-u`  $\text{بْنِ الْيَمَانِ}$   $\text{حَدِيقَةُ بِنِ الْيَمَانِ}$  for *Hudayfat*<sup>u</sup> *bn*<sup>u</sup> *ʿl-Yamānī*, etc.
- REM. *c* The letter  $\text{ي}$  with two points below,  $\text{أَلْيَاءُ الْمُنَاةُ مِنْ تَحْتِهَا}$ , may also be written without diacritical points as  $\text{ي}$ . 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`  $\text{لِي}$  *lī*, `yam^siY`  $\text{يَمِشِي}$  *yamšī*.

### 3.3.2 Short vowels

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

Letter	Transliteration <sup>22</sup>			ArabTeX notation
	dmg+	loc	arabica	
$\text{ـ}$				
$\text{ـ}$	<i>a</i>	<i>a</i>	<i>a</i>	<code>a</code>
$\text{ـ}$	<i>u</i>	<i>u</i>	<i>u</i>	<code>u</code>
$\text{ـ}$	<i>i</i>	<i>i</i>	<i>i</i>	<code>i</code>
$\text{ـ}$	<i>an</i>	<i>an</i>	<i>an</i>	<code>aN</code>
$\text{ـ}$	<i>un</i>	<i>un</i>	<i>un</i>	<code>uN</code>
$\text{ـ}$	<i>in</i>	<i>in</i>	<i>in</i>	<code>iN</code>

Table 4: Standard ArabTeX (short vowels)

Whether Arabic texts be vocalized or not is essentially a matter of personal choice. So one may use `voc` mode and decide not to write vowels except at some particular places for disambiguation purposes, or use `novoc` mode, not write vowels—as `novoc` normally does not show them—except, again, where disambiguation is needed.<sup>23</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 [sect. 4.4 on page 19](#));
- when `arabluatex` needs to perform a contextual analysis to determine the carrier of the *hamzah*;
- in the various transliteration modes, as vowels are always expressed in romanized Arabic.

## 4 arabluatex in action

### 4.1 The vowels and diphthongs

**Short vowels** As said above, they are written  $\langle a, u, i \rangle$ :

<sup>22</sup>See below [sect. 8 on page 37](#).

<sup>23</sup>See below [sect. 4.4 on page 19](#).

\_halaqa (or xalaqa) خَلَقَ *ḥalaqa*, ~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 حُجَّةٌ *ḥuǧḡat<sup>un</sup>*.

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

qAla قَالَ *qāla*, bI`a بِيَعُ *bī'a*, .tUruN طُورٌ *tūr<sup>un</sup>*, .tInuN طِينٌ *tīn<sup>un</sup>*, murU'aTuN مَرُوءَةٌ *murū'at<sup>un</sup>*.

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

al-fat\_A الْفَاتَى *al-fatā*, al-maqh\_A الْمَقْهَى *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 *yaǧzū*, ramaW رَمَوْا *ramaw*, banaWA بَنَوْا *banaw*.

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

al-l\_ah-u اللهُ *al-lāh<sup>u</sup>*, 'il\_ahuN إِلَهٌ *'ilāh<sup>un</sup>*.  
 al-ra.hm\_an-u الرَّحْمَنُ *ar-raḥmān<sup>u</sup>*, l\_akin لَكِن *lākin*, h\_ahunA هَهُنَا *hāhunā*,  
 .hunayn-u bn-u 'is.h\_aq-a إِسْحَقُ بْنُ إِسْحَاقَ *Hunayn<sup>u</sup> bn<sup>u</sup> 'Ishāq<sup>a</sup>*, rabb\_i رَبِّ *rabbī*, al-`A.s\_i الْعَاصِ *al-Āṣī*.

**Silent و/ي** Some words ending with ةَ are usually written حَوةَ or حوة instead of ةَ: see Wright.<sup>24</sup> *arabluatex* preserves that particular writing; the same applies to words ending in يةَ for ةَ. Long vowels ⟨U, I⟩ shall receive no *sukūn* after a 'alif maḥdūfah and are discarded in *trans* mode:

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

And so also: al-rib\_aIT-u الرِّبِيَّةُ *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.<sup>25</sup> In no way this affects the sound of the *tanwīn*, so it has to be discarded in *trans* mode:

<sup>24</sup>Wright, see n. 7, i. 12 A.

<sup>25</sup>*Ibid.*, i. 12 C.

`amruNU عمرو *'amr<sup>un</sup>*, `amraNU عمروا *'amr<sup>an</sup>*, `amriNU عمرو *'amr<sup>in</sup>*.

When the *tanwīn* falls away (Wright, *A Grammar of the Arabic Language*, i. 249 B): `amr-uU bn-u mu.hammadiN عمرو بن محمد *'Amr<sup>u</sup> bn<sup>u</sup> Muḥammad<sup>in</sup>*, mu.hammad-u bn-u `amr-iU bn-i\_hAlidiN خالد بن عمرو *Muḥammad<sup>u</sup> bn<sup>u</sup> 'Amr<sup>i</sup> bn<sup>i</sup> Ḥālid<sup>in</sup>*.

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

**tanwīn** The marks of doubled short vowels, ُ, ِ, ٍ, are written ⟨uN, aN, iN⟩ respectively. arabuatex deals with special cases, such as ِ taking an ِ after all consonants except ة, and *tanwīn* preceding ى as in هدى, which is written ⟨aN\_A⟩ or ⟨aN<sup>Y</sup>⟩:

mAluN مأل *māl<sup>un</sup>*, bAbaN بابأ *bāb<sup>an</sup>*, madInaTan مَدِينَة *madīnat<sup>an</sup>*, bintiN بنت *bint<sup>in</sup>* maqhaN\_A مَقَهَى *maqḥa<sup>n</sup>*, fataNY فَتَى *fata<sup>n</sup>*.

arabuatex is aware of special orthographies: َsay'uN شَيْءٌ *šay<sup>un</sup>*, َsay'aN شَيْئاً *šay<sup>an</sup>*, َsay'iN شَيْءٍ *šay<sup>in</sup>*.

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

al-qA.dI الْقَاضِي *al-qāḍī*, qA.diyaN قَاضِيًا *qāḍiy<sup>an</sup>*, qA.diNI قَاضٍ *qāḍi<sup>n</sup>*.

## 4.2 Other orthographic signs

**tā' marbūṭah** It is written ⟨T⟩:

madInaTuN مَدِينَةٌ *madīnat<sup>un</sup>*, madInaTan مَدِينَةً *madīnat<sup>an</sup>*, madInaTiN مَدِينَةٍ *madīnat<sup>in</sup>*.

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

**Initial hamzah:** 'asaduN أَسَدٌ *'asad<sup>un</sup>*, 'u\_htuN أُخْتُ *'uht<sup>un</sup>*, 'iqlIduN إِقْلِيدٌ *'iqlīd<sup>un</sup>*, 'anna أَنْ *'anna*, 'inna إِنْ *'inna*.

*hamzah* followed by the long vowel و is encoded '\_U: '\_U1\_A أُوْلَى *'ulā*, '\_U1U أُوْلُ *'ulū*, '\_U1A'ika أَوْلَاكَ *'ulā'ika*.

*hamzah* followed by the long vowel ي is encoded '\_I: '\_ImAnuN إِيْمَانٌ *'imān<sup>un</sup>*<sup>26</sup>.

<sup>26</sup>For another way of encoding the initial *hamzah* followed by a long vowel, see the *taḥfīf<sup>u</sup> 'l-hamzat<sup>i</sup>* on the next page.

**Middle hamzah:** xA.ti'-Ina خَاطِئِينَ *ḥāṭi'ina*, ru'UsuN رُءُوسٌ *ru'ūs<sup>un</sup>*, xa.tI'aTuN خَاطِئَةٌ *ḥaṭi'at<sup>un</sup>*, su'ila سُئِلَ *su'ila*, 'as'ilaTuN أَسْئَلَةُ *as'ilat<sup>un</sup>*, mas'alaTuN مَسْئَلَةٌ *mas'alat<sup>un</sup>*, 'as'alu أُسْئِلَ *as'alu*, yatasA'alUna يَتَسَاءَلُونَ *yatasā'alūna*, murU'aTuN مُرُوءَةٌ *murū'at<sup>un</sup>*, taw'amUN تَوَّامٌ *taw'am<sup>un</sup>*, ta'xIruN تَأْخِيرٌ *ta'hīr<sup>un</sup>*, ta'axxara تَأَخَّرَ *ta'ahhara*, ji'tu-ka جِئْتُكَ *ḡi'tu-ka*, qA'iluN قَائِلٌ *qā'il<sup>un</sup>*, .hIna'i\_diN هِنَا *ḥina'id<sup>in</sup>* حِينَدٌ *ḥay'aTuN* هَيْئَةٌ *hay'at<sup>un</sup>*, hay'AtuN هَيَاتٌ *hay'āt<sup>un</sup>*.  
 From Wright:<sup>27</sup>— All consonants, whatsoever, not even *ʿalif ḥemzatum* excepted, admit of being doubled and take *tašdīd*. 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'āḡ<sup>un</sup>*.

**Final hamzah:** xa.ta'uN خَطَأٌ *ḥaṭa'<sup>un</sup>*, xa.ta'aN خَطَأٌ *ḥaṭa'<sup>an</sup>*, xa.ta'iN خَطِئًا *ḥaṭa'<sup>in</sup>*, 'aqra'u أَقْرَأُ *'aqra'u*, taqra'Ina تَقْرَأِينَ *taqra'ina*, taqra'Una تَقْرَأُونَ *taqra'ūna*, yaqra'na يَقْرَأَنَّ *yaqra'na*, yaxba'Ani يَخْبَأَنَّ *yahbā'āni*, xaba'A خَبَأَ *ḥaba'ā*, xubi'a خُبِيَ *ḥubi'a*, xubi'UA خُبِيَوا *ḥubi'ū*, jA'a جَاءَ *ḡā'a*, ridA'uN رَدَأٌ *ridā'<sup>un</sup>*, ridA'aN رَدَأٌ *ridā'<sup>an</sup>*, jI'a جِيَ *ḡi'a*, radI'iN رَدِيَ *radī'<sup>in</sup>*, sU'uN سُوءٌ *sū'<sup>un</sup>*, .daw'uN ذُوءٌ *daw'<sup>un</sup>*, qay'iN قِيءٌ *qay'<sup>in</sup>*, ^sifA'I شِفَائِي *šifā'ī*, man^sa'I مَنْسَأِي *manša'ī*, nisA'uN نِسَاءٌ *nisā'<sup>un</sup>*, nisA'u-hu نِسَاءُهُ *nisā'u-hu*, nisA'i-hi نِسَائِهِ *nisā'ī-hi*, nisA'I نِسَائِي *nisā'ī*.

^say'uN سَيٌّ *šay'<sup>un</sup>*, ^say'aN سَيًّا *šay'<sup>an</sup>*, ^say'iN سَيِيٌّ *šay'<sup>in</sup>*, al-^say'-u السَّيِّءُ *aš-šay'<sup>u</sup>*, 'a^syA'-u أَشْيَاءُ *ašyā'<sup>u</sup>*, 'a^syA'-a أَشْيَاءُ *ašyā'<sup>a</sup>*, .zim'aN ظِمًّا *ḡim'<sup>an</sup>*, radI'aN رَدِيًّا *radī'<sup>an</sup>*.

**taḥfīf<sup>u</sup> ʿl-hamzat<sup>i</sup>:** if the *hamzah* has *ḡazmah* and is preceded by *ʿalif hamzatum*, it must be changed into the letter of prolongation that is homogeneous with the preceding vowel; hence: 'a'mana آمَنَ *āmana*, 'u'minu أَوْمِنُ *ūminu*, 'i'mAnu إِيْمَانٌ *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 *taḥfīf<sup>u</sup> ʿl-hamzat<sup>i</sup>:* i'sir اِسِرْ *īsir*, i'\_dan اِذْنْ *īdan*, u'mul اُوْمَلْ *ūmul*. arabluatex also provides ways of encoding those words when the initial *ʿalif* comes into *waṣl*, so as to make the *ʿalif waṣl* fall away when preceded by و or ف: wa-'sir وَاِسِرْ *wa-'sir*, fa-'\_dan فَادَنْ *fa-'dan*, fa-'ti فَاتِ *fa-'ti*, wa-'tamirUA وَاَتْمِرُوا *wa-'tamirū*; or be retained outside the imperative, as in fa-i'tazarat فَاتْتَزَرْتُ *fa-'tazarat*, ba`da i'tilAfīn بَعْدَ اِتِّلَافٍ *ba`da 'tilāf<sup>in</sup>*.

<sup>27</sup>Wright, see n. 7, i. 14 B.

The strange spelling of *mi'at<sup>un</sup>*: mi'aTuN مائة *mi'at<sup>un</sup>*,  
 mi'atAni مائتان *mi'atāni*, mi'atayni مائتين *mi'atayni*, mi'Una مئون *mi'ūna*,  
 mi'AtuN مئآت *mi'ātun*, mi'aN\_A مأي *mi'a<sup>n</sup>*. Of course, the ‘pipe’ character  
 can be used to prevent this rule from being applied (see [sect. 4.5 on page 21](#)):  
 mi'a|TuN مئة *mi'at<sup>un</sup>*.

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

Hence one should keep to this distinction and encode ‘a'kulu أَكُلُ *‘akulu* and ‘AkilUN أَكِلُ *‘akil<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 *‘akilūna*, 'a'mannA آمَنَّا *‘āmannā*, al-  
 qur'An-u الْقُرْآنُ *al-qur‘ān<sup>u</sup>*.

jA'a جَاءَ *ğā‘a*, yatasA'alUna يَتَسَاءَلُونَ *yatasā‘alūna*, ridA'uN رِدَاءٌ *ridā‘un*,  
 xaba'A خَبَأَ *ħaba‘ā*, yaxba'Ani يَخْبَأُنِي *yahba‘āni*.

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

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

‘allaqa عَلَّقَ *‘allaqa*, mAdduN مَادَّ *mādd<sup>un</sup>*, ‘ammara أَمَّرَ *ammara*, murruN مُرَّرَ *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 ([ibid.](#), i. 15 B–16 C):—

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

Unlike arabtex and arabxetex, arabluatex *never* requires the solar letter to be written twice, as it automatically generates the euphonic *tašdīd* above the letter that carries it, whether the article be written in the assimilated form or not, e.g. al-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-raħmān<sup>u</sup>*, al-.zulm-u  
 الظُّلْمُ *až-žulm<sup>u</sup>*, al-lu.gaT-u اللُّغَةُ *al-luğat<sup>u</sup>*.

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

Note the absence of *sukūn* above the passed over ن in the following examples, each of which is accompanied by a consistent transliteration: min\_rabbi-hi مِنْ رَبِّهِ *mir rabbi-hi*, min layliN مِنْ لَيْلٍ *mīl 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 [sect. 2.2.1 on page 5](#) for further details. Other kinds of assimilations, including the various cases of *'idjām*, 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. لَبِثْتُ for لَبِثْتُ *labittu*, will be discarded here, as it is largely condemned by the grammarians (see Wright, *A Grammar of the Arabic Language*, i. 16 B–C).

**The definite article and the 'alif<sup>u</sup> 'l-waṣl<sup>i</sup>** At the beginning of a sentence, ا is never written, as اَلْحَمْدُ لِلَّهِ; 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 لِلَّهِ *al-ḥamd<sup>u</sup> li-l-lāh<sup>i</sup>*.

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

- (a) Definite article at the beginning of a sentence is encoded `al-`, or `a<solar letter>-` if one wishes to mark the assimilation—which is in no way required, as *arabluatex* will detect all cases of assimilation.
- (b) Definite article inside sentences is encoded `'l-` or `'<solar letter>-`.
- (c) In all remaining cases of elision, the 'alif<sup>u</sup> 'l-waṣl<sup>i</sup> is expressed by the vowel that accompanies the omitted *hamzah*:  $\langle u, a, i \rangle$ .

**Article:** bAb-u 'l-madrasaT-i بَابُ الْمَدْرَسَةِ *bāb<sup>u</sup> 'l-madrasat<sup>i</sup>*, al-maqAlaT-u 'l-'\_U1\_A الْمَقَالَةُ الْأُولَى *al-maqālat<sup>u</sup> 'l-'ūlā*, al-lu.gaT-u 'l-'arabiyyaT-u فِي صِنَاعَةِ اللُّغَةِ الْعَرَبِيَّةِ *al-luḡat<sup>u</sup> 'l-'arabiyyat<sup>u</sup>*, fI .sinA`aT-i 'l-.tibb-i فِي صِنَاعَةِ الطِّبِّ إِلَى الْأَنْتِقَاصِ *ilā 'l-intiqād<sup>i</sup>*, fI 'l-ibtidA'-i فِي الْإِبْتِدَاءِ *fi 'l-ibtidā'<sup>i</sup>*, 'abU 'l-wazIr-i أَبُو الْوَزِيرِ *abu 'l-wazīr<sup>i</sup>*, fa-lammA ra'aW 'l-najm-a فَلَمَّا رَأَوْا النَّجْمَ *fa-lammā ra'awu 'n-naǧm<sup>a</sup>*.

**Particles:**—

- (a) *li-*: 'alif<sup>u</sup> 'l-waṣl<sup>i</sup> is omitted in the article اَل when it is preceded by the preposition لِ: li-l-rajul-i لِلرَّجُلِ *li-r-raǧul<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 لِللَّيْلِ *li-l-laylat<sup>i</sup>*.



- (b) *la-*: the same applies to the affirmative particle لَ: la-l-.haqq-u لَقِيَ *la-l-haqq<sup>u</sup>*.
- (c) With the other particles, *'alif<sup>u</sup> 'l-waṣl<sup>i</sup>* is expressed: fI 'l-madInaT-i فِي الْمَدِينَةِ *fi '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-ru`b<sup>i</sup>*.

**Perfect active, imperative, nomen actionis:** qAla isma قَالَ أَسْمَعُ *qāla 'sma<sup>ʿ</sup>*, qAla uqtul قَالَ أَقْتُلُ *qāla 'qtul*, huwa inhazama هُوَ أَنْهَزَمَ *huwa 'nhazama*, wa-ustu`mila وَأَسْتَعْمِلُ *wa-'stu`mila*, qadi in.sarafa قَدْ أَنْصَرَفَ *qadi 'nṣarafa*, al-iqtidar-u الْأَقْتِدَارُ *al-iqtidār<sup>u</sup>*, 'il\_A 'l-intiqA.d-i إِلَى الْأَنْتِقَاصِ *ila 'l-intiqād<sup>i</sup>*, law istaqbala لَوْ اسْتَقْبَلَ *lawi 'staqbala*.

**Other cases:** 'awi ismu-hu أَوْ اسْمُهُ *'awi 'smu-hu*, zayduN ibn-u `amriNU زَيْدُ عَمْرٍو *Zayd<sup>uni</sup> 'bn<sup>u</sup> 'Amr<sup>in</sup>*,<sup>28</sup> `umar-u ibn-u 'l-\_ha.t.tAb-i عُمَرُ بْنُ أَحْطَابِ *'Umar<sup>u</sup> 'bn<sup>u</sup> 'l-Haṭṭāb<sup>i</sup>*,<sup>29</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 يُؤْمِنُ اللَّهُ *la-'ymun<sup>u</sup> 'l-lāh<sup>i</sup>*.

**'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, *A Grammar of the Arabic Language*, 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-ilāl<sup>i</sup>*, maqh\_A 'l-'amIr-i مَقْبَى الْأَمِيرِ *maqha '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 (*ibid.*, 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 *fi 'aynaya 'l-malik<sup>i</sup>*, ix`say 'l-qawm-a إِخْشَيْ الْقَوْمِ *iḥṣayi 'l-qawm<sup>a</sup>*, mu.s.tafaw 'l-l\_ah-i مُصْطَفَوُ اللَّهِ *muṣṭafawu 'l-lāh<sup>i</sup>*.

ramaW 'l-.hijAraT-a رَمَوْا الْحِجَارَةَ *ramawu 'l-ḥiġārat<sup>a</sup>*, fa-lamma ra'aW 'l-najm-a فَلَمَّا رَأَوْا النَّجْمَ *fa-lammā ra'awu 'n-naġm<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 (*ibid.*, i. 22 A–C), it is encoded explicitly, like so:—

<sup>28</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>29</sup> “Umar is the son of al-Haṭṭāb” (see n. 28).

'antumum 'l-kA\_dib-Una أَنْتُمُ الْكَادِبُونَ *antumum 'l-kādib<sup>āna</sup>*, ra'aytumum 'l-rajul-a رَأَيْتُمُ الرَّجُلَ *ra'aytumum 'r-rağul<sup>a</sup>*, mani 'l-ka\_d\_dAb-u مِنَ الْكُذَّابِ *mani 'l-kad\_dAb-u* *kaddāb<sup>u</sup>*, qatalati 'l-rUm-u قَتَلَتِ الرُّومَ *qatalati 'r-Rūm<sup>u</sup>*.

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

mu.hammaduN 'l-nabI مُحَمَّدٌ النَّبِيُّ *Muhammad<sup>uni</sup> 'n-nabī*, salAmuN ud\_hulUA سَلَامٌ أَدْخُلُوا *salām<sup>umu</sup> 'dhulū*, qa.sIdata-hu fI qatl-i \uc{'a}bI \uc{m}uslimiN قَاتِلِ ابْنِ مُسَلِمٍ الَّذِي يَقُولُ فِيهَا *qaṣīdata-hu fī qatlī 'Abī Muslim<sup>ini</sup> 'latī yaqūlu fī-hā*.

### 4.3 Special orthographies

**The name of God** The name of God, اللهُ, is compounded of the article اَلْ, and اِلَٰهُ (noted اِلهٌ with the defective *ʿalif*) so that it becomes اِلَٰهٌ; then the *hamzah* is suppressed, its vowel being transferred to the ل before it, so that there remains اللهُ (I refer to Edward William Lane, *An Arabic-English lexicon*, 8 vols. [London – Edinburgh: Williams and Norgate, 1863–1893] [henceforth Lane, *Lexicon*], I. 83 col. 1). Finally, the first ل is made quiescent and incorporated into the other, hence the *tašdīd* above it. As *arabluatex* never requires a solar letter to be written twice (see above, on page 15), the name of God is therefore encoded *al-l\_ah-u* or *'l-l\_ah-u*:—

al-l\_ah-u اللهُ *al-lāh<sup>u</sup>*, yA|<sup>30</sup> al-l\_ah-u يَا اللهُ *yā al-lāh<sup>u</sup>*, 'a-fa|<sup>31</sup>-al-l\_ah-i لا-تأ-غ-الاننا أَفَإِنَّ اللَّهَ لَتَنَعَلَنَّ *'a-fa-al-lāh<sup>i</sup> la-tağ'alanna*, bi-'l-l\_ah-i بِاللَّهِ *bi-'l-lāh<sup>i</sup>*, wa-'l-l\_ah-i وَاللَّهِ *wa-'l-lāh<sup>i</sup>*, bi-sm-i 'l-l\_ah-i بِسْمِ اللَّهِ *bi-sm<sup>i</sup> 'l-lāh<sup>i</sup>*, al-.hamd-u li-l-l\_ah-i الْحَمْدُ لِلَّهِ *al-ḥamd<sup>u</sup> li-l-lāh<sup>i</sup>*, li-l-l\_ah-i 'l-qA'il-u لِيَلْقَاكَ اللهُ *li-l-lāh<sup>i</sup> 'l-qā'il<sup>u</sup>*.

**The conjunctive الَّذِي** Although it is compounded of the article اَلْ, the demonstrative letter ل and the demonstrative pronoun ذَا, both masculine and feminine forms that are written defectively are encoded *alla\_dI* and *allatI* respectively. Forms starting with the connective *ʿalif* are encoded *'lla\_dI* and *'llatI*:—

'a\_hAfu mina 'l-malik-i 'lla\_dI يا.زليمة 'l-nAs-a أَخَافُ مِنَ الْمَلِكِ الَّذِي *'ahāfu mina 'l-malik<sup>i</sup> 'lladī yağlimu 'n-nās<sup>a</sup>*, `udtu 'l-~say\_h-a 'lla\_dI huwa marI.duN هُوَ مَرِيضٌ عُدْتُ الشَّيْخَ الَّذِي *'udtu 'š-šayh<sup>a</sup> 'lladī huwa marīd<sup>un</sup>*, mA 'anA bi-'lla\_dI qA'iluN لا-كأ-سأ-ان مَا أَنَا بِالَّذِي قَاتِلٌ لَكَ شَيْئًا *mā 'anā bi-'lladī qā'il<sup>un</sup> la-ka ṣay'an*.

<sup>30</sup>Note the “pipe” character ‘|’ here after *yA* and below after *fa* before footnote mark *31*: it is needed by the *dmg* transliteration mode as in this mode any vowel at the commencement of a word preceded by a word that ends with a vowel, either short or long, is absorbed by this vowel viz. *ʿala 'l-tariq<sup>i</sup>*. See [sect. 4.5 on page 21](#) on the “pipe” and [sect. 8 on page 37](#) on *dmg* mode.

<sup>31</sup>See [30](#).

'ari-nA 'lla\_dayni 'a.dallA-nA mina 'l-jinn-i wa-'l-'ins-i أَرَنَا الَّذِينَ  
 'ari-na 'lladayni 'adallā-nā mina 'l-ǧinn<sup>i</sup> wa-'l-'ins<sup>i</sup>.  
 أضلانا من الجن والإنس

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

fa-'innA na\_dkuru 'l-.sawt-ayni 'l-la\_dayni rawaynA-humA `an ja.h.zaT-  
 a فَإِنَّا نَذْكُرُ الصَّوْتَيْنِ اللَّذَيْنِ رَوَيْنَاهُمَا عَنْ بَحْظَةَ  
 fa-'innā naḍkuru 's-ṣawt<sup>ayni</sup> 'l-ladayni  
 rawaynā-humā `an Ġahẓat<sup>a</sup>.

And also: al-la\_dAni اللذَانِ al-ladāni, al-la\_dayni اللذَيْنِ al-ladayni, al-  
 latAni اللتَانِ al-latāni, al-latayni اللتَيْنِ al-latayni, al-lAtI اللَّاتِي al-lātī, al-  
 lA' |Ati<sup>32</sup> اللَّاتِ al-lā'āti, al-lA'I اللَّاتِي al-lā'ī, and so forth.

#### 4.4 Quoting

It is here referred to “quoting” after the arabtex package.<sup>33</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: علم علم الهيئة; is it علم, he was taught the science of astronomy, or علم, he taught the science of astronomy? In order to disambiguate this clause, it may be sensible to put a dammah above the first ع: علم علم الهيئة, which is achieved by “quoting” the vowel u, like so: `u"ullima, or, with no other vowel than the required u: `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:—

`u"ullima `ilm-a 'l-hay'aT-i علم 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 ندرکنا  
 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:—

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

<sup>33</sup>See Lagally, ArabTeX, 22

qAla<sub>q</sub>isma`" قال اسمعُ *qāla 'sma'*, jA'at" hinduN هِنْدُ جَاءَتْ *ġā'at Hind<sup>un</sup>*,  
 ^sabIhuN bi-man q"u.ti`at" qadamA-hu قَدَمَاهُ بِن قُطِعَتْ *šabīh<sup>un</sup> bi-*  
*man qūṭi'at qadamā-hu.*

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

wa-"ust"u`mila وَأَسْتَعْمَلُ *wa-'stu'mila*, huwa "inhazama هُوَ أَنهَزَمَ *huwa*  
 'nhazama, al-"intiqA.d-u الأَنْتِقَاضُ *al-intiqād<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<sub>q</sub>"A'iluN قَالَ قَاتِلُ *qāla qā'il<sup>un</sup>*, ibn-u<sub>q</sub>'abI<sub>q</sub>'u.saybi`aT-"a إِبْنُ  
 أُبَيِّ أَصْبَيْعَةَ *Ibn<sup>u</sup> 'Abī 'Uṣaybi'at<sup>a</sup>.*

- (b) *tanwīn*:—

madInaT"aN مَدِينَةٌ *madīnat<sup>an</sup>*, bAb"aN بَابًا *bāb<sup>an</sup>*, hud"aN\_A هُدًى *huda<sup>n</sup>*,  
 ^say'"iN شَيْءٌ *šay<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<sup>i</sup>:—

fI<sub>q</sub>"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 قَالَ أَهْرَبُ فَلَنْ تُقْتَلَ *qāla 'hrub fa-lan*  
*tuqtala.*

- (b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala قَالَ أَقْتُلْ فَلَنْ تُقْتَلَ *qāla 'qtul fa-lan tuqtala*,  
 mA jA'at" mini imra'aTiN مِنْ امْرَأَةٍ *mā ġā'at mini 'mra'at<sup>in</sup>*, kam"  
 qad" ma.dat" min" laylaTiN لَيْلَةٍ *kam qad maḍat min*  
*laylat<sup>in</sup>.*

**fullvoc** In this mode, “quoting” can be used to take away any short vowel (or *tanwīn*, as seen above) or any *sukūn*:—

al-jamr-u 'l-.sayfiyy-u 'lla\_dI kAna bi-q"rAn"|nUn-a الْجَمْرُ الصَّيْفِيُّ الَّذِي  
 كَانَ يَقْرَأُونَ *al-ġamr<sup>u</sup> 'ṣ-ṣayfiyy<sup>u</sup> 'ladī kāna bi-Qrānnūn<sup>a</sup>.*

#### 4.4.1 Quoting the *hamzah*

As said above in [sect. 4.2 on page 13](#), the *hamzah* is always written ⟨ ' ⟩, its carrier being determined by contextual analysis. “Quoting” that straight single quote character like so: ⟨ " ' ⟩ allows to determine the carrier of the *hamzah* freely, without any consideration for the context. [Table 5 on the next page](#) gives the equivalents for all the possible carriers the *hamzah* may take.

Letter	Transliteration <sup>34</sup>			ArabTeX notation
	dmg+	loc	arabica	
ء	ʾ	ʾ	ʾ	''
آ	ā	ā	ā	A''
أ	ʾ	ʾ	ʾ	a''
أ	ʾ	ʾ	ʾ	u''
ؤ	ʾ	ʾ	ʾ	w''
إ	ʾ	ʾ	ʾ	i''
ي	ʾ	ʾ	ʾ	y''

Table 5: “Quoted” *hamzah*

As one can see from [table 5](#), the carrier of the *hamzah* is inferred from the letter that precedes the straight double quote (<">). Of course, any “quoted” *hamzah* may take a short vowel, which is to be written *after* the ArabTeX equivalent for the *hamzah* itself, namely (<'>). For example, ؤ is encoded <w''a>, while إ is encoded <w''i>. In the latter example, the second straight double quote encodes the *sukūn* in voc mode in accordance with the rule laid above on pages [19–20](#).

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

#### 4.5 The ‘pipe’ character (|)

In the terminology of ArabTeX, the “pipe” character ‘|’ is referred to as the “invisible consonant”. Hence, as already seen above in [sect. 4.4.1 on the preceding page](#), its usage to encode the *hamzah* alone, with no carrier: |'' ء.

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

bi-qrAn|nUn-a بقرآنونَ *bi-Qrānnūn*<sup>a</sup>, “in Crannon” (Thessaly, Greece).<sup>35</sup>

As one can see, the “pipe” character between the two <n> prevents the necessary *tašdīd* rule ( [on page 15](#)) from being applied.

#### 4.6 Putting back on broken contextual analysis rules

*New feature v1.7* In complex documents such as critical editions where footnotes and other kind of annotations can be particularly abundant, the contextual analysis rules that are described above may be broken by 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 which interferes with
4     the contextual analysis.} 'l-na`gma...
```

<sup>34</sup>See below [sect. 8 on page 37](#).

<sup>35</sup>See more context on the previous page.

5 `\end{arab}`

This is wrong:

فَلَمَّا رَأَوْا<sup>a</sup> النَّجْمَ...

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

According to the rule stated on page 17, 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.

`\arbnnull` The `\arbnnull` 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, `\arbnnull` may be found just after or before Arabic words.

In any case, *no space must be left* after or before the Arabic word that `\arbnnull` 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\arbnnull{'l-na^gma}\LRfootnote{A footnote
3     which interferes with the contextual analysis.}
4   'l-na^gma...
5
6   qAla\LRfootnote{A footnote which interferes with the
7     contextual analysis.} \arbnnull{qAla}uhrub fa-lan tuqtala.
8
9   \uc{z}ayduN\arbnnull{ibnu}\LRfootnote{A footnote which
10     interferes with the contextual analysis.}
11   \arbnnull{zayduN}ibn-u \uc{'a}mriNU.\LRfootnote{See
12     \vref{fn:zayd-is-son}.}
13 \end{arab}
14
15 \begin{arab}[trans]
16   \uc{z}ayduN\arbnnull{ibnu}\LRfootnote{A footnote which
17     interferes with the contextual analysis.}
18   \arbnnull{zayduN}ibn-u \uc{'a}mriNU.\LRfootnote{See
19     \vref{fn:zayd-is-son}.}
20 \end{arab}
```

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

*Zayd<sup>u</sup>nie 'bn<sup>u</sup> 'Amr<sup>i</sup>n.f*

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

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

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

<sup>d</sup>See 28 on page 17.

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

<sup>f</sup>See 28 on page 17.

## 4.7 Stretching characters: the *taṭwīl*

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

.hunayn-u bn-u 'is.h--\_aq-a حُنَيْنُ بْنُ 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

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

```

1 \usepackage{arabluatex}
2 \SetArbNumbers{Arabic}
3 % use '+anum' for Arabic numbers or '-anum' for Indian numbers:
4 \newfontfamily\arabicfont{Amiri}[Script=Arabic, RawFeature={+anum}]

```



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

### 4.8.2 The *abjad*

The numbers may also be expressed with letters from right to left arranged in accordance with the order of the Hebrew and Aramaic alphabets (see Wright, *A Grammar of the Arabic Language*, i. 28 B–C). The *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 مَهْ كِتَابُهُ 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 *abjad* numbers in Roman numerals if it finds the `\abjad` command in any of the transliteration modes.

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

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

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

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

`\abjad` may also be used to convert values of counters into *ʿabjad* numbers, like so:—

```

1 The \arb[trans]{ʿabjad} number for the current page (\thepage) is
2 \abjad{\thepage}.

```

The *ʿabjad* number for the current page (23) is ٢٣.

This technique can be used to produce abjad-numbered lists as will be demonstrated on page 48.

## 4.9 Additional characters

In the manuscripts, the unpointed letters, *al-ḥurūf al-muḥmalat*<sup>36</sup>, are sometimes further distinguished from the pointed by various contrivances, as explained in Wright.<sup>36</sup> 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>37</sup>			ArabT <sub>E</sub> X notation
	dmg+	loc	arabica	
ب	<i>ḃ</i>	<i>b</i>	<i>b</i>	.b
د	<i>ḋ</i>	<i>d</i>	<i>d</i>	~d
ف	<i>ḟ</i>	<i>f</i>	<i>f</i>	.f
ق	<i>q̇</i>	<i>q</i>	<i>q</i>	.q
ك	<i>k̇</i>	<i>k</i>	<i>k</i>	.k
ن	<i>ṅ</i>	<i>n</i>	<i>n</i>	.n
ه	<i>ah</i>	<i>ah</i>	<i>a</i>	H <sup>38</sup>
ي	<i>ẏ</i>	<i>y</i>	<i>y</i>	.y <sup>39</sup>
﴿	(	(	(	((
﴾	)	)	)	))

Table 6: Additional Arabic codings

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

## 4.10 Arabic emphasis

As already seen in [sect. 4.8.2 on the preceding page](#), the *ʿabjad* numbers are distinguished from the surrounding words by a stroke placed over them. This technique is used to distinguish further words that are proper names or book titles.

<sup>36</sup>Wright, see n. 7, i. 4 B–C.

<sup>37</sup>See below [sect. 8 on page 37](#).

<sup>38</sup>This letter can be used to encode the *tāʾ marbūṭah* devoid of diacritical points as it is found in some manuscripts, with the same *tanwīn* and the same short vowels as the standard *tāʾ marbūṭah* with two points above, e. g.

al-madInaHa, madInaHaN, مَدِينَة، المَدِينَة.

<sup>39</sup>See above [REM. b on page 9](#).



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

`\abjad{45}: kitAbu-hu \aemph{fI 'l-`AdAt-i}` مة: كِتَابُهُ فِي الْعَادَاتِ 45: *kitābu-hu fi 'l-`Adāt*'.

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

*New feature v1.9.2*

`\aemph*`

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

#### 4.10.1 Underlining words or numbers

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

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

*New feature v1.19*

(b) `\aoline*`, which is the same as `\aoline`, but better suited for *ʿabjad* numbers.<sup>40</sup>

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

## 5 Arabic poetry

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

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

```
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 [sect. 2.2 on page 5](#).

`width` `width=⟨length⟩`

Default: `0.3\linewidth`

The default width of each hemistich that the verse consists of. It may be expressed in any accepted unit of measurement, such as `4cm` or `2in`. However, one must keep in mind that the total length of the two hemistichs added to the one of the gutter that separates them must not exceed the length of the base line, unless one wishes to have the hemistichs distributed on subsequent lines.

`gutter` `gutter=⟨width⟩`

Default: `0.15 x (hemistich width)`

The gutter consists of the blank space that is between the two hemistichs. By default, it is commensurate with the width of the hemistich, but it may be expressed in any accepted unit of measurement as well.

`metre` `metre=⟨name⟩`

Default: `none`

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

`delim` `delim=true|false`

Default: `false`

<sup>40</sup>See the example provided above [sect. 4.8.2 on page 23](#).

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

`utf utf=true|false` Default: false

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

`color color=(color name)` Default: not set

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

`export export=true|false` Default: false

*New feature v.1.13* This named argument does not need a value as it defaults to `true` if it is used. If `export` is set as a global option as well (see above on page 5), all the lines will be converted to Unicode and exported to the external selected file. See below [sect. 12 on page 51](#) for more details.

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

`\bayt{<šadr>}[<tadwīr>] {<ağuz>}\\`

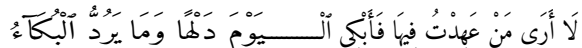
That two subsequent hemistichs should be connected with one another is technically named *tadwīr*. Should that happen, either the *šadr* or the *ağuz* or both of them, may be connected to one another by letters that are naturally bound to the following or the preceding ones over the *tadwīr*. The optional 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 [sect. 4.7 on page 23](#)). Of course, the ending word of the *šadr* 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{[A 'ar_A man `ahidtu fI-hA fa-'abkI 'l---][--]{---yawma
3 dalhaN wa-mA yaruddu 'l-bukA'u}\\
4 \end{arabverse}

```



As one can see, *triple hyphens* have been used. In the *šadr*, the first hyphen triggers the rules that are related to the definite article and the ‘*alif*’ *l-wasl*,<sup>42</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:—

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

<sup>42</sup>See [sect. 4.2 on page 16](#).

```

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 `arā man `ahidtu fī-hā fa-'abki 'l-    -yawma dalh<sup>an</sup> wa-mā yaruddu 'l-bukā'u

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

```

غَيْرِ أَنِّي قَدْ أَسْتَعِينُ عَلَى الْهَمِّ    إِذَا خَفَّ بِالثَّوْبِ النَّجَاءُ  
بِزَفُوفٍ كَانَهَا هَقْلَةً أُمَّمُ    رِثَالٍ دَوِيَّةٍ سَقْفَاءُ

In the first line, the word  $\text{أَلْهَمِّ}$  should be split into  $\text{أَلْهَمِّ}$  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  $\text{أُمَّمُ}$  should be split into  $\text{أُمَّمُ}$ , so that the only way to avoid splitting it into two parts is to write it all inside the *tadwīr*. In that case, as the word is to be placed in the middle, it has been surrounded by spaces.

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

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

Default: true

*New feature v1.20* `\StretchBayt` takes one optional argument, either `true` or `false` and can be used to remove the stretching form lines of Arabic poetry. As a side effect, there will be more space between words, but this can be compensated by inserting double hyphens between letters (on this technique, see [sect. 4.7 on page 23](#)). Should it be desired to extend further the strokes, four hyphens may be inserted (----), viz. a multiple of two. `\StretchBayt` may be used at any point of the document, even between two subsequent lines of poetry. Note that `\StretchBayt[false]` may require to carefully adjust the width of the hemistichs to avoid overlapping.

**Footnotes** Footnotes are not set by default inside the `\bayt` command, but there are two easy ways to have them printed.

If they are little in number, each footnote may be split into pairs of `\footnotemark{}` (please mind the braces or “declare” `footnotemark` using `\MkArbBreak` to take it out of the Arabic environment<sup>43</sup>) 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>44</sup>

`\bayt+` **Critical Notes** If the `ekdosis` package be loaded,<sup>45</sup> the `\bayt` command also accepts a + *New feature v1.21* optional argument that can be used to let critical notes be inserted in lines of poetry. Details on how to use this command are provided in the relevant section of the documentation of the `ekdosis` package.<sup>46</sup>

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

## 5.1 Example

Here follow the first lines of Imru’u ‘l-Qaysi’s *Mu‘allaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the *‘idjām* that is fit to this declamatory poetry:—<sup>47</sup>

```

1 \begin{arab}[fullvoc]
2   qAla imru'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-siq.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
18    lA tahluk 'asaN_A wa-ta^gammali}\\
19    \bayt{wa-'inna ^sifA'I `abraTuN muharAqaTuN}{fa-hal `inda
20    rasmiN dArisiN min mu`awwali}\\
21   \end{linenumbers*}
22 \end{arabverse}

```

`\StretchBayt[true]` (Default):—

<sup>43</sup>See [sect. 11.1 on page 46](#).

<sup>44</sup>The footnote package can also be used for the same effect. However, it must be loaded *after* `arabluatex`.

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

<sup>46</sup>*Ibid.*, see “Arabic Poetry”.

<sup>47</sup>Please note that for the time being only the assimilation rules that are laid on (b) on page 15 are applied. See [sect. 2.2.1 on page 5](#) for more information. None of the editions of the *Mu‘allaqāt* that I know of feature the *‘idjām* in the Arabic text, although it is often strongly marked in declamation.

قَالَ أَمْرُ الْقَيْسِ فِي مُعَلَّتِهِ:

1 قَفَا نَبِكَ مِنْ ذِكْرِي حَيْبٍ وَمَنْزِلٍ  
2 فَتَوَضَّحَ فَاَلْمُقْرَأَةَ لَمْ يَعْفُ رَسْمَهَا  
3 تَرَى بَعَرَ الْأَرَامِ فِي عَرَصَاتِهَا  
4 كَأَنِّي غَدَاةَ الْبَيْنِ يَوْمَ تَحْمَلُوا  
5 وَوَقُوفًا بِهَا صَحِيٍّ عَلَيَّ مَطِيئِهِمْ  
6 وَإِنَّ شِفَائِي عِبْرَةٌ مَهْرَاقَةٌ  
بِسْفَطِ الْوَلَى بَيْنَ الدَّخُولِ حَوْمَلِ  
لَمَّا نَسَجَتْهَا مِنْ جَنُوبٍ وَشَمَلِ  
وَقِيَعَانَهَا كَأَنَّهُ حَبُّ فُلْفُلِ  
لَدَى سَمَرَاتِ الْحَيِّ نَاقِفُ حَنْظَلِ  
يَقُولُونَ لَا تَهْلِكُ أَسَى وَتَجَمَلِ  
فَهَلْ عِنْدَ رَسْمِ دَارِسٍ مِّنْ مُّعْوَلِ

(الضَّرْبُ الثَّانِي مِنَ الْعُرُوضِ الْأُولَى مِنَ الطَّوِيلِ)

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

1 qifā nabki min dīkrā ḥabīb<sup>iw</sup> wa-manzilī  
2 fa-Tūḍiḥa fa-'l-Miqrāti lam ya'fu rasmu-hā  
3 tarā ba'ara 'l-'ar'āmi fī 'araṣāti-hā  
4 ka-'annī ḡadāta 'l-bayni yawma taḥammalū  
5 wuqūf<sup>am</sup> bi-hā ṣaḥbī 'alayya maṭiyya-hum  
6 wa-'inna šifā'ī 'abrat<sup>um</sup> muḥarāqat<sup>un</sup>  
bi-siḡṭi 'l-liwā bayna 'd-Daḥūli fa-Hawmali  
limā nasaḡat-hā min ḡanūb<sup>iw</sup> wa-ṣam'ali  
wa-ḡi'āni-hā ka-'anna-hu ḥabbu fulfuli  
ladā samurāti 'l-ḥayyi nāḡifu ḥanzali  
yaḡulūna lā tahlīk 'aṣa<sup>w</sup> wa-taḡammali  
fa-hal 'inda rasm<sup>in</sup> dāris<sup>im</sup> mim mu'awwali  
(aḡ-ḡarbu 'l-'tāni mina 'l-'arūḡi 'l-'ulā mina 'l-'tawīli)

\StretchBayt [false]:—

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

قَالَ أَمْرُ الْقَيْسِ فِي مُعَلَّتِهِ:

1 قَفَا نَبِكَ مِنْ ذِكْرِي حَيْبٍ وَمَنْزِلٍ  
2 فَتَوَضَّحَ فَاَلْمُقْرَأَةَ لَمْ يَعْفُ رَسْمَهَا  
3 تَرَى بَعَرَ الْأَرَامِ فِي عَرَصَاتِهَا  
4 كَأَنِّي غَدَاةَ الْبَيْنِ يَوْمَ تَحْمَلُوا  
5 وَوَقُوفًا بِهَا صَحِيٍّ عَلَيَّ مَطِيئِهِمْ  
6 وَإِنَّ شِفَائِي عِبْرَةٌ مَهْرَاقَةٌ  
بِسْفَطِ الْوَلَى بَيْنَ الدَّخُولِ حَوْمَلِ  
لَمَّا نَسَجَتْهَا مِنْ جَنُوبٍ وَشَمَلِ  
وَقِيَعَانَهَا كَأَنَّهُ حَبُّ فُلْفُلِ  
لَدَى سَمَرَاتِ الْحَيِّ نَاقِفُ حَنْظَلِ  
يَقُولُونَ لَا تَهْلِكُ أَسَى وَتَجَمَلِ  
فَهَلْ عِنْدَ رَسْمِ دَارِسٍ مِّنْ مُّعْوَلِ

(الضَّرْبُ الثَّانِي مِنَ الْعُرُوضِ الْأُولَى مِنَ الطَّوِيلِ)

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

1 qifā nabki min dīkrā ḥabīb<sup>iw</sup> wa-manzilī  
2 fa-Tūḍiḥa fa-'l-Miqrāti lam ya'fu rasmu-hā  
3 tarā ba'ara 'l-'ar'āmi fī 'araṣāti-hā  
4 ka-'annī ḡadāta 'l-bayni yawma taḥammalū  
5 wuqūf<sup>am</sup> bi-hā ṣaḥbī 'alayya maṭiyya-hum  
6 wa-'inna šifā'ī 'abrat<sup>um</sup> muḥarāqat<sup>un</sup>  
bi-siḡṭi 'l-liwā bayna 'd-Daḥūli fa-Hawmali  
limā nasaḡat-hā min ḡanūb<sup>iw</sup> wa-ṣam'ali  
wa-ḡi'āni-hā ka-'anna-hu ḥabbu fulfuli  
ladā samurāti 'l-ḥayyi nāḡifu ḥanzali  
yaḡulūna lā tahlīk 'aṣa<sup>w</sup> wa-taḡammali  
fa-hal 'inda rasm<sup>in</sup> dāris<sup>im</sup> mim mu'awwali  
(aḡ-ḡarbu 'l-'tāni mina 'l-'arūḡi 'l-'ulā mina 'l-'tawīli)

## 6 Special applications

**Linguistics** The same horizontal stroke as the *taṭwīl* (see [sect. 4.7 on page 23](#)) 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" ..

*New feature v1.4.3* **Brackets** The various bracket symbols are useful in technical documents such as critical editions for indicating that some words or some letters must be added or removed. `arabluatex` will automatically fit those symbols to the direction of the text. For the time being, the following symbols are supported:

- parentheses: `()`
- square brackets: `[]`
- angle brackets: `<>`
- braces: `{}`

`\abraces` Parentheses, square and angle brackets may be input directly at the keyboard; however, words or letters that are to be read between braces must be passed as arguments to the `\abraces` command:—

```

1 \begin{arab}
2   \abraces{wa-qAla} 'inna 'abI kAna mina 'l-muqAtilaTi
3   wa-kAna--<--> 'ummI min `u.zamA'i buyUti 'l-zamAzimaTi.
4 \end{arab}

```

{وَقَالَ} إِنَّ أَبِي كَانَ مِنَ الْمُقَاتِلَةِ وَكَانَتْ < أَبِي مِنْ عُظْمَاءِ بِيُوتِ الزَّمَانِمَةِ.

**Additional Arabic marks** In addition to common letters, many symbols and ligatures are encoded in Arabic Unicode standard, such as honorifics consisting of complex ligatures, and annotation signs used in the *Qurʿān* or in classical poetry.

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

Codepoint	Shorthand	Glyph	Transliteration
FDFD	bismillah	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ	bi-'smi 'Llāhi 'r-rahmāni 'r-rahīmi
FDF5	salam	صَلِّع	ṣalla 'Llāhu 'alay-hi wa-sallama
FDF4	slm	صَلِّع	ṣalla 'Llāhu 'alay-hi wa-sallama
FDFB	jalla	جَلَّال	ġalla ġalāla-hu

Table 7: Additional Arabic marks

*New feature v1.13* The mark to be inserted is determined by contextual analysis, or by an optional argument, either `r1` to have the Arabic glyph printed, or `l1` to print the transliterated equivalent.

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

```
\newarbookmark{<shorthand>}{<RTL codepoint>}{<LTR rendition>}
```

As regards the right-to-left codepoint, it may be either typed in Unicode or selected as Unicode codepoint. To that end, the L<sup>A</sup>T<sub>E</sub>X command `\symbol{"XYZT}` or its plain T<sub>E</sub>X variant `\char"XYZT\relax` may be used, where XYZT are uppercase hex digits (0 to 9 or A to F).

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

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

It must be noted that `\newarbookmark` expects ArabT<sub>E</sub>X input scheme inside `\arb[trans]{}` to the exclusion of buckwalter input scheme.

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

```
1 \SetArbDflt*
2 \newarbookmark{sly}{\arbcolor[red]{~~~~06d6}}{}
3 \newarbookmark{jim}{\arbcolor[red]{~~~~06da}}{}
4 \begin{arab}
5   sUraTu 'l-nisA'i, 19:
6 \end{arab}
7 \begin{center}
8   \begin{arab}
9     \arbmark{bismillah}
10    \end{arab}
11  \end{center}
12 \begin{arab}[fullvoc]
13   y_a'ayyuhA 'lla_dIna 'a'manUA lA ya.hillu la-kum 'an tari_tUA
14   'l-nisA'a karhaN\arbmark{sly} wa-lA ta`.dulU-hunna li-ta_dhabUA
15   bi-ba`.di mA 'a'taytumU-hunna 'illa 'an ya'tIna bi-fA.hi^saTiN
16   mubayyinaTiN\arbmark{jim} wa-`A^sirU-hunna
17   bi-'l-ma`rUfi\arbmark{jim} fa-'in karihtumU-hunna fa-`as_A_a
18   'an takrahUA ^say'aN wa-ya^g`ala 'l-l_ahu fI-hi _hayraN
19   ka_tIran ((19))
20 \end{arab}
```

سُورَةُ النَّسَاءِ، ١٩:

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا يَحِلُّ لَكُمْ أَنْ تَرِثُوا النِّسَاءَ كَرِهًا وَلَا تَعْضُلُوهُنَّ لِتَذَهَبُوا بِبَعْضِ مَا آتَيْتُمُوهُنَّ إِلَّا أَنْ يَأْتِيَنَّ بِفَاحِشَةٍ مُبِينَةٍ وَعَاشِرُوهُنَّ بِالْمَعْرُوفِ فَإِنْ كَرِهْتُمُوهُنَّ فَمَسِيءٌ أَنْ تَكْرَهُوا شَيْئًا وَيَجْعَلَ اللَّهُ فِيهِ خَيْرًا كَثِيرًا ﴿١٩﴾

New feature v1.18 The ‘Zero width joiner’ character (U+200D) The ‘Zero width joiner’ character (U+200D)

belongs to the ‘General Punctuation’ block (range 2000–206F) of the Unicode standard. It is a non-printing character which, when it is placed between two characters that would for some reason not be connected, causes them to be printed in their connected forms.

It is encoded & in ArabTeX scheme.

In elegantly printed books where many of the letters are interwoven with one another so as to form ligatures, it may be convenient to bring the letters into line in some instances. In the following example, the ‘zero width joiner’ is used to prevent two adjacent letters, viz. س and ح, from standing one above the other in the name of ‘Ishāq (إِسْحَاقُ):<sup>48</sup>—

```


1 \begin{arab}[fullvoc]
2 huwa 'abU zaydiN .hunaynu bnu 'is&\underline{&.h_a}qa
3 'l-`a\underline{bA}diyyu bi-fat.hi 'l-`ayni wa-ta_hfIfi 'l-bA'i.
4
5 huwa 'abU zaydiN .hunaynu bnu 'is&\highlight{&.h_a}qa
6 'l-`a\highlight{bA}diyyu bi-fat.hi 'l-`ayni wa-ta_hfIfi 'l-bA'i.
7 \end{arab}


```

هو أبو زيد حنين بن إسحاق العبادي بفتح العين وتخفيف الباء.  
هو أبو زيد حنين بن إسحاق العبادي بفتح العين وتخفيف الباء.

## 6.1 The Qur’ān

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

*New feature v1.15* For the time being, `\ayah{<3-digit number>}` is provided so as to typeset the number of the *āyah* that it is referred to inside the dedicated mark—Unicode U+06DD: —in Arabic script or inside parentheses in romanized Arabic:—

`\ayah{123}`  (123).

An example follows:—

```

1 \SetArbDflt*
2 \newarbookmark{alifsp}{\arabic{0627}}{\arb[trans]{'alif} }
3 \newarbookmark{lamspace}{\arabic{0644}\arabic{0653}}{\arb[trans]{lAm} }
4 \newarbookmark{mim}{\arabic{0645}\arabic{0653}}{\arb[trans]{mIm}}
5 \begin{arab}[fullvoc]
6 min ((sUraTi \uc{'l-b}aqaraTi)):
7 \end{arab}
8 \begin{arab}[fullvoc]
9 \arbmark{alifsp}\arbmark{lamspace}\arbmark{mim}~\ayah{1}
10 _d_alika 'l-kit_abu lA rayba fI-hi hudaN_A
11 li-l-muttaqIna~\ayah{2} 'lla_dIna yu'minUna bi-'l-.gaybi
12 wa-yuqImUna 'l-.sal_aUTa wa-mimma razaqn_a-hum
13 yunfiqUna~\ayah{3}
14 \end{arab}

```

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



مِنْ ﴿سُورَةِ الْبَقَرَةِ﴾: ﴿الَّذِينَ يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَاةَ وَمِمَّا رَزَقْنَاهُمْ يُنْفِقُونَ﴾

*min (sūratī 'l-Baqarati):*  
*'alif lām mīm (1) dālika 'l-kitābu lā rayba fī-hi huḍā' li-l-muttaqīna (2) 'lādīna*  
*yu'minūna bi-'l-ġaybi wa-yuqīmūna 's-ṣalāta wa-mimmā razaqnā-hum yun-*  
*fīqūna (3)*

**Caveat** For some reason, most of the Arabic fonts do not show the number properly: some are only able to display at most two digits, while others display the digits outside the ‘end of *āyah*’ sign, let alone those that print the digits stacked. To the knowledge of the writer, this should be reported to the developers of those fonts.

## 7 Color

*New feature v1.12* arabuatex is able to render in color either words, parts of words or diacritics. As the techniques implemented in this section may lead to some complexity, the reader should first become well acquainted with the following points:<sup>49</sup>—

- The “pipe” character (|, [sect. 4.5 on page 21](#));
- ‘Quoting’ technique ([sect. 4.4 on page 19](#)), and more specifically ‘quoting the *hamzah*’ (on page 20);
- Putting back on broken contextual analysis rules ([sect. 4.6 on page 21](#));
- Arabic marks ([sect. 6 on page 30](#)).

`\arbcolor` `\arbcolor` takes the text to be colored into `<color>` as an argument:—

`\arbcolor[<color>]{<Arabic text>}`

```

1 \begin{arab}
2   \arbcolor[red]{al-bAbu 'l-hAmisu} fI .tabaqAti 'l-'a.tibbA'i
3   'lla_dIna kAnUA mun_du zamAni \uc{^gAlInUsa} wa-qarIbaN
4   min-hu. \arbcolor[red]{\uc{^gAlInUsu}}: wa-l-na.da` 'awwalaN
5   kalAmaN kulliyyaN fI 'a_hbAri \uc{^gAlInUsa} wa-mA kAna
6   `alay-hi...
7 \end{arab}
8 \begin{arab}[trans]
9   \arbcolor[red]{al-bAbu 'l-hAmisu} fI .tabaqAti 'l-'a.tibbA'i
10  'lla_dIna kAnUA mun_du zamAni \uc{^gAlInUsa} wa-qarIbaN
11  min-hu. \arbcolor[red]{\uc{^gAlInUsu}}: wa-l-na.da` 'awwalaN
12  kalAmaN kulliyyaN fI 'a_hbAri \uc{^gAlInUsa} wa-mA kAna
13  `alay-hi...
14 \end{arab}

```

الْبَابُ الْخَامِسُ فِي طَبَقَاتِ الْأَطِبَاءِ الَّذِينَ كَانُوا مِنْدُ زَمَانِ جَالِينُوسَ وَقَرِيْبًا مِنْهُ. جَالِينُوسُ: وَلَنْضَعُ أَوَّلًا كَلَامًا  
 كَلِمًا فِي أَخْبَارِ جَالِينُوسَ وَمَا كَانَ عَلَيْهِ...

<sup>49</sup>Regarding the colors themselves and the way new colors can be defined in addition to those that are already available, please refer to the `xcolor` package.

*al-bābu 'l-hāmisu fī tabaqāti 'l-ʾaṭibbāʾi 'lādīna kānū mundu zamāni Ğālīnūsa wa-qarīb<sup>an</sup> min-hu. Ğālīnūsu: wa-l-nadaʾ ʾawwal<sup>an</sup> kalām<sup>an</sup> kulliy<sup>an</sup> fī ʾaḥbāri Ğālīnūsa wa-mā kāna ʾalay-hi...*

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

**‘voc’ mode:**

```
i^stara\arbcOLOR[brown]{y}tu-hu bi- tama\arbcOLOR[red]{niN}
'a`\arbcOLOR[blue]{^ga}ba-ka أَعْجَبَكَ بِتَمَنٍّ إِشْتَرَيْتَهُ ištaraytu-hu bi-tamanin
ʾaġaba-ka.
```

**‘fullvoc’ mode:**

```
i^stara\arbcOLOR[brown]{y}tu-hu bi- tama\arbcOLOR[red]{niN}
'a`\arbcOLOR[blue]{^ga}ba-ka أَعْجَبَكَ بِتَمَنٍّ إِشْتَرَيْتَهُ ištaraytu-hu bi-tamanin
ʾaġaba-ka.
```

## 7.1 Tricks of the trade

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

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

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

Diacritic	Transliteration <sup>50</sup>			ArabTeX notation
	dmg+	loc	arabica	
˘				
-	<i>a</i>	<i>a</i>	<i>a</i>	.a
˙				
-	<i>u</i>	<i>u</i>	<i>u</i>	.u
˘				
-	<i>i</i>	<i>i</i>	<i>i</i>	.i
◌◌◌				
◌◌◌				o

Table 8: ArabTeX diacritics for `\arbcOLOR`

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

**‘voc’ mode:**

```
i^staraytu-hu bi- taman\arbcOLOR[red]{iN} 'a`^g\arbcOLOR[red]{.a}ba-
ka أَعْجَبَكَ بِتَمَنٍّ إِشْتَرَيْتَهُ ištaraytu-hu bi-tamanin ʾaġaba-ka.
```

<sup>50</sup>See below [sect. 8 on page 37](#).

i^staraytu-hu bi-\_tama\arbcOLOR[red]{n}iN 'a`\arbcOLOR[red]{^g}.aba-ka  
 ka أَعْجَبَكَ بِمَنْ اشْتَرَيْتَهُ *ištaraytu-hu bi-ṭaman<sup>in</sup> 'a'ġaba-ka.*

i^staraytu-hu bi-\_tama\arbcOLOR[red]{n}\arbcOLOR[blue]{iN}  
 'a`\arbcOLOR[red]{^g}\arbcOLOR[blue]{.a}ba-ka أَعْجَبَكَ بِمَنْ اشْتَرَيْتَهُ  
*ištaraytu-hu bi-ṭaman<sup>in</sup> 'a'ġaba-ka.*

**‘fullvoc’ mode:**

i^staray"\arbcOLOR[red]{o}tu-hu bi-\_taman"\arbcOLOR[red]{iN}  
 'a`^g"\arbcOLOR[red]{.a}ba-ka أَعْجَبَكَ بِمَنْ اشْتَرَيْتَهُ *ištaraytu-hu bi-ṭaman<sup>in</sup>*  
*'a'ġaba-ka.*

i^stara\arbcOLOR[red]{y}otu-hu bi-\_tama\arbcOLOR[red]{n"}iN  
 'a`\arbcOLOR[red]{^g"}.aba-ka أَعْجَبَكَ بِمَنْ اشْتَرَيْتَهُ *ištaraytu-hu bi-ṭaman<sup>in</sup>*  
*'a'ġaba-ka.*

i^stara\arbcOLOR[red]{y"}\arbcOLOR[blue]{o}tu-hu  
 bi-\_tama\arbcOLOR[red]{n"}\arbcOLOR[blue]{iN} 'a`\arbcOLOR[red]  
 {^g"}\arbcOLOR[blue]{.a}ba-ka أَعْجَبَكَ بِمَنْ اشْتَرَيْتَهُ *ištaraytu-hu bi-ṭaman<sup>in</sup>*  
*'a'ġaba-ka.*

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

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

mu`allim\arbcOLOR[red]{\arbnUL{m}aN} مُعَلِّمًا *mu'allim<sup>an</sup>,*  
 istisqA'\arbcOLOR[red]{\arbnUL{A'}aN} اِسْتِسْقَاءٌ *istisqā'<sup>an</sup>,*  
 ^say'\arbcOLOR[red]{\arbnUL{ay'}aN} شَيْئًا *šay'<sup>an</sup>,*  
 ^gAmi`aT|\arbcOLOR[red]{\arbnUL{T}aN} جَامِعَةٌ *ġāmi'at<sup>an</sup>.*

REM. Note that in the last example (*ġāmi'at<sup>an</sup>*), the ‘pipe’ character has been inserted before \arbcOLOR. Otherwise, the dmG mode of the transliteration rules would have interpreted the *tā' marbūṭah* as *final* (e.g. *h* instead of the expected *t*).<sup>51</sup>

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

```

1 \newarbookmark{Y}{\color{red}0649}{}
2 \arb{hud\arbcOLOR[red]{a\arbnUL{A}}\arbookmark{Y}}
3 \arb{trans}{hud\arbcOLOR[red]{a\arbnUL{A}}\arbookmark{Y}}

```

---

هُدًى *hudā<sup>n</sup>*

<sup>51</sup>See also on page 40 “Discarding the *i'rāb*” for more information.

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

```
wa-\arbcolor[red]{\arbnul{wa}i}stisqA'uN وَأَسْتَسْقَاءُ wa-'stisqāum52.
fI "al".i-\arbcolor[red]{\arbnul{'l-}i}btidA'i
فِي الْأَبْتِدَاءِ fi 'li-'btidā'i.

\arbcolor[red]{'a'\arbnul{k}}kulu أَكُلُ 'ākulu,
\arbcolor[red]{'A\arbnul{k}}kiluN أَكِلُ 'ākilum.
```

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

```
1 \newarbookmark{alifmaddahred}{~~~~0627\arbcolor[red]{~~~~0653}}%
2 {\arb[trans]{\arbcolor[red]{'a'\arbnul{k}}}}
3 \arb{\arbookmark{alifmaddahred}kulu}
4 \arb[trans]{\arbookmark{alifmaddahred}kulu}.
```

أَكُلُ 'ākulu.

REM. In the preceding example, any consonant could have been passed as argument to the `\arbnul` command.

*šaddah* In the following example, it is assumed that the *šaddah* above the letter ل in المعْلُونِ <sup>المعلون</sup>, *al-mu'allimūna*, is to be rendered in red. Thus the Arabic mark must generate the *šaddah* alone—of which the Unicode codepoint is 0651—in Arabic script and the letter 'l' in transliteration:—

```
1 \newarbookmark{lamshaddah}{~~~~0651}{l}
2 \arb[fullvoc]{al-mu'al"\arbcolor[red]{\arbookmark{lamshaddah}}.imUna}
3 \arb[trans]{al-mu'al"\arbcolor[red]{\arbookmark{lamshaddah}}.imUna}.
```

المعلونِ al-mu'allimūna.

**The definite article and the euphonic *tašdīd*** The intricate business of rendering in color the initial *alif al-waṣl* of the definite article followed by a solar consonant must be unraveled.

From the examples provided above, in fIل'1-nAsi فِي النَّاسِ <sup>فِي النَّاسِ</sup> *fi 'n-nāsi*, the initial *alif<sup>w</sup>* *l-waṣl<sup>i</sup>* can be rendered in red like so: `\arbcolor[red]{\arbnul{al-}a}` أَ. Then, the following two letters, namely l-n, must print the string *lām + nūn + šaddah* in Arabic, and exactly *n-n* in transliteration. Thus an Arabic mark is needed:—

```
1 \newarbookmark{lnn}{~~~~0644~~~~0646~~~~0651}{n-n}
2 \arb[fullvoc]{fI\arbnul{al-}
3 \arbcolor[red]{\arbnul{al-}a}\arbookmark{lnn}Asi}
4 \arb[trans]{fI\arbnul{al-}
5 \arbcolor[red]{\arbnul{al-}a}\arbookmark{lnn}Asi}.
```

فِي النَّاسِ <sup>فِي النَّاسِ</sup> *fi 'n-nāsi*.

<sup>52</sup>To the knowledge of the writer, the *waṣlah* alone is not part of the Arabic Unicode block.

*hamzah* The ‘quoting’ technique provides an easy way to determine the carrier of the *hamzah*, as shown in [table 5 on page 21](#):—

```
yatasA\arbnul{ 'a}\arbcolor[red]{|''}.aUna يَتَسَاءُلُونَ yatasā'alūna,
^say\arbcolor[red]{|''}\arbnul{' }aN شَيْئًا šayam, ^say\arbcolor[red]
{|''}iN شَيْءٍ šayin, \arbcolor[red]{a''}.as\arbcolor[red]{y''}.ilaTuN
أَسْئَلَةٌ as'ilatun.
```

## 8 Transliteration

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

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

**dmg** *Deutsche Morgenländische Gesellschaft*, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.<sup>53</sup> **dmg** transliteration convention is selected by default;

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

**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>54</sup>

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

More standards will be included in future releases of *arabluatex*.

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

```
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, e.g. `\upshape`, `\itshape`, `\slshape`, and so forth.

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

<sup>53</sup>See Brockelmann et al., “Die Transliteration der arabischen Schrift”.

<sup>54</sup>See <http://www.loc.gov/catdir/cpso/roman.html> for the source document concerning Arabic language.

<sup>55</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).

```

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

```

`\uc` **Proper names** Proper names or book titles that must have their first letters uppercased may be passed as arguments to the `\uc{<word>}` command. `\uc` is a clever command, for it will give the definite article *al-* in lower case in all positions. Moreover, if the 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 إِسْحَاقَ Hunaynu bnu 'Ishāqa,
\uc{`u_tm_an-u} عُمَنْ Utmānu, .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
لِلَّهِ عَبْدُ اللَّهِ عَوْفُ بْنُ عَدِّ بْنِ خَلْدٍ سَعْدُ بْنُ عَدِّ بْنِ زَيْدِ بْنِ خَلْدٍ daraba Zaydu bnu Hālidin Sa`da bna 'Awfi bni
'Abdi 'Llāhi.

```

However, `\uc` must be used cautiously in some very particular cases, for the closing brace of its argument may prevent a rule from being applied. To take an example, as seen above on page 18, the transliteration of مُحَمَّدٌ 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 زَكْرِيَاءَ الرَّازِيّ Abū Bakrin Muḥammadu
bnu Zakariyyāa 'r-Rāziyyu.

```

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

```

\arb[trans]{\uc{'abU zaydiN .hunaynu bnu 'is.h_aqa 'l-`ibAdiyuu}}
'Abū Zaydin Hunaynu bnu 'Ishāqa 'l-'Tbādiyyu.

```

*New feature v1.10* `\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>56</sup>—

```

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

```

<sup>56</sup>Just as `\uc`, `\prname` is also able to process proper names consisting of several subsequent words.

```

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

```

From Wright:<sup>a</sup> — If the name following <sup>أ</sup>بن be that of the mother or the grandfather, the <sup>أ</sup> is retained; as عيسى ابن مريم, “Jesus the son of Mary”; عمار ابن منصور, “Ammār the (grand)son of Manṣūr”.

<sup>a</sup>Wright, see n. 7, i. 23 C.

The following example shows how `\prname` can be used in conjunction with the `nameauth` package to have Arabic proper names printed first in full then in partial forms:<sup>57</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 ‘Iṣḥāq al-‘Ibādī, ‘Abū Bakr Muḥammad ibn Zakariyyā’ ar-Rāzī. Then as Ḥunayn, ar-Rāzī.

`\prname*` REM. `arabluatex` also provides `\prname*` which only renders in upright roman style already transliterated proper names without applying any further processing. It is mostly used internally and applied to proper names exported in Unicode to an external selected file.<sup>58</sup>

## 8.1 Additional note on `dmg` convention

*New feature v1.3* According to Brockelmann et al.,<sup>59</sup> Arabic *ʿrāb* may be rendered into `dmg` 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 case one wishes to apply rule (a). The default rule (b) can be set back with `\ArbUpDflt` at any point of the document.

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

<sup>57</sup>See the documentation of `nameauth` for more details: <https://ctan.org/pkg/nameauth>

<sup>58</sup>See below [sect. 12 on page 51](#) for more details.

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

```

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

```

Arabic dmg transliteration for رَأَيْتُ جَامِعًا مَهْدَمَةً مَثَدْتَهُ: *ra'aytu ḡāmi`<sub>an</sub> 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.

**īrāb 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`allimūN مَعْلِمٌ *mu'allim<sup>un</sup>*, kA'inuN كَاتِبٌ *kā'in<sup>un</sup>*, kA'inAtuN كَاتِبَاتٌ *kā'in<sup>ātun</sup>*, \uc{-amraNU} عَمْرًا *Amr<sup>an</sup>*, fataN\_A فَتًى *fata<sup>n</sup>*, qA.diNI قَاضٍ *qāḍi<sup>n</sup>*.

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

REM. *a* As the *tanwīn* is passed over in pronunciation when it is followed by the letters ر, ل, م, و, ي (see (b) on page 15), it may be desirable to further distinguish it by putting it above the line, but not to do the same for *ḡayr munṣarif* terminations. This can be achieved by simply omitting the hyphen before any *ḡayr munṣarif* termination:—

kAna .ḡaniyyaN l\_akinna-hu labisa ḡubbaTaN mumazzaqaN 'aydu-hA كَانَ غَنِيًّا لَكِنَّهُ لَيْسَ جَبَّةً *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 مَنَّ النَّاسُ كَافَّةً مِنْ مَنْعِ النَّاسِ كَافَّةً *mana`a 'n-nās<sup>a</sup> kāffat<sup>an</sup> min muḡāḡabati-hi 'aḡad<sup>un</sup> bi-sayyidi-nā*.

**Discarding the īrāb** As said above ((c) on the previous page), the *īrāb* may be discarded in some cases, as in transliterated proper names or book titles. `arabluatex` is able to render words ending with *tā' marbūḡah* 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 *ḡikmat Allāh*, fi.d.daT al-darAhim fīḡḡat ad-darāhim.

REM. It may so happen, as in the absence of the article before the annexed word, that `arabluatex` be unable to determine which of the above two cases the word ending with *tā' marbūḡah* falls into. The 'pipe' character (see sect. 4.5 on page 21) may be appended to that word to indicate that what follows is in the construct state: \uc{r}isAlaT fI tartIb qira'aT| kutub \uc{ḡ}AlInUs *Risālah fī tartīb qirā'at kutub ḡālīnūs*.



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

## 8.2 Examples

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

**‘dmg+’ standard:** ‘atā ṣadīq<sup>un</sup> ilā Ğuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safrat<sup>in</sup> qaṣīrat<sup>in</sup> fa-qāla la-hu: “sawfa u’idu-hu ilay-ka fī l-masā’i wa-’adfa’u la-ka uḡrat<sup>an</sup>.” fa-qāla Ğuḥā: “anā āsif<sup>un</sup> ḡidd<sup>an</sup> annī lā ’astatī’u an uḥaqqiqa la-ka raḡbata-ka fa-l-ḥimār<sup>u</sup> laysa huna l-yawm<sup>a</sup>.” wa-qabla an yutimma Ğuḥā 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 Ğuḥā yanhaqu.” fa-qāla la-hu Ğuḥā: “ḡarīb<sup>un</sup> amru-ka yā ṣadīqī a-tuṣaddiqu l-ḥimār<sup>a</sup> wa-tukaddiba-nī?”

**‘loc’ standard:** atā ṣadīqun ilā Juḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safrat<sup>in</sup> qaṣīrat<sup>in</sup> fa-qāla la-hu: “sawfa u’idu-hu ilay-ka fī al-masā’i wa-’adfa’u la-ka ujratan.” fa-qāla Juḥā: “anā āsifun ḡiddan annī lā astatī’u an uḥaqqiqa la-ka raḡbata-ka fa-al-ḥimāru laysa hunā al-yawma.” wa-qabla an yutimma Juḥā kalāma-hu bada’a al-ḥimāru yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: “innī asma’u ḥimāra-ka yā Juḥā yanhaqu.” fa-qāla la-hu Juḥā: “ḡharībun amru-ka yā ṣadīqī a-tuṣaddiqu al-ḥimāra wa-tukaddhiba-nī?”

**‘arabica’ standard:** atā ṣadīqun ilā Ğuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safrat<sup>in</sup> qaṣīrat<sup>in</sup> fa-qāla la-hu: “sawfa u’idu-hu ilay-ka fī l-masā’i wa-’adfa’u la-ka uḡratan.” fa-qāla Ğuḥā: “anā āsifun ḡiddan annī lā astatī’u an uḥaqqiqa la-ka raḡbata-ka fa-l-ḥimāru laysa hunā l-yawma.” wa-qabla an yutimma Ğuḥā kalāma-hu bada’a l-ḥimāru yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: “innī asma’u ḥimāra-ka yā Ğuḥā yanhaqu.” fa-qāla la-hu Ğuḥā: “ḡarībun amru-ka yā ṣadīqī a-tuṣaddiqu l-ḥimāra wa-tukaddiba-nī?”

## 9 Buckwalter input scheme

*New feature v1.4* Even though arabluatex is primarily designed to process the ArabT<sub>E</sub>X notation, it can also process the Buckwalter input scheme to a large extent.<sup>60</sup> The Buckwalter scheme is actually processed in two steps, as it is first converted into ArabT<sub>E</sub>X. Then, once this is accomplished, the ArabT<sub>E</sub>X scheme is processed through the above described functions. In this way, the Buckwalter input scheme can make the most of the arabluatex special features that are presented in [sect. 2.2 on page 5](#).

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

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

<sup>60</sup>See <http://www.qamus.org/transliteration.htm>

‘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.<sup>61,62</sup> However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode  $\{$  and  $\}$ , must be replaced with square brackets viz. [ and ] respectively.

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

Table 9 gives the Buckwalter equivalents that are currently used by arablutext. The additional characters that are defined in table 6 on page 24 are also available.

Letter	Transliteration <sup>63</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>ḍ</i>	<i>dh</i>	<i>ḍ</i>	*	V
ر	<i>r</i>	<i>r</i>	<i>r</i>	r	r
ز	<i>z</i>	<i>z</i>	<i>z</i>	z	z
س	<i>s</i>	<i>s</i>	<i>s</i>	s	s
ش	<i>š</i>	<i>sh</i>	<i>š</i>	\$	c
ص	<i>ṣ</i>	<i>ṣ</i>	<i>ṣ</i>	S	S
ط	<i>ṭ</i>	<i>ṭ</i>	<i>ṭ</i>	D	D
ظ	<i>ẓ</i>	<i>ẓ</i>	<i>ẓ</i>	T	T
ع	<i>ʿ</i>	<i>ʿ</i>	<i>ʿ</i>	Z	Z
غ	<i>ǧ</i>	<i>gh</i>	<i>ǧ</i>	E	E
ف	<i>f</i>	<i>f</i>	<i>f</i>	g	g
ق	<i>q</i>	<i>q</i>	<i>q</i>	f	f
ك	<i>k</i>	<i>k</i>	<i>k</i>	q	q
ل	<i>l</i>	<i>l</i>	<i>l</i>	k	k
م	<i>m</i>	<i>m</i>	<i>m</i>	l	l
ن	<i>n</i>	<i>n</i>	<i>n</i>	m	m
ه	<i>h</i>	<i>h</i>	<i>h</i>	n	n
و	<i>w</i>	<i>w</i>	<i>w</i>	h	h
ي	<i>y</i>	<i>y</i>	<i>y</i>	w	w
ى	<i>ā</i>	<i>á</i>	<i>ā</i>	y	y
ة	<i>ah</i>	<i>ah</i>	<i>a</i>	Y	Y
ء	<i>ʾ</i>	<i>ʾ</i>	<i>ʾ</i>	p	p
آ	<i>ʾā</i>	<i>ʾā</i>	<i>ʾā</i>	ʾ	C
أ	<i>ʾ</i>	<i>ʾ</i>	<i>ʾ</i>		M
ؤ	<i>ʾ</i>	<i>ʾ</i>	<i>ʾ</i>	>	O
	<i>ʾ</i>	<i>ʾ</i>	<i>ʾ</i>	&	W

Table 9: Buckwalter input scheme

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

<sup>62</sup>I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in arablutext.

<sup>63</sup>See sect. 8 on page 37.

Letter	Transliteration			Buckwalter notation	
	dmg+	loc	arabica	base/xml	safe
ا	’	’	’	<	I
آ	’	’	’	]	Q
ـ	—	—	—	~	~
أ	’	’	—	[	L
ا	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>
و	<i>u</i>	<i>u</i>	<i>u</i>	<i>u</i>	<i>u</i>
ي	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>
ان	<i>an</i>	<i>an</i>	<i>an</i>	F	F
ان	<i>un</i>	<i>un</i>	<i>un</i>	N	N
ان	<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
- (taṭwīl)	—	—	—	-	-

Table 9: Buckwalter input scheme

**Transliteration** The Buckwalter notation can also be transliterated into any accepted romanization standard of Arabic. See above [sect. 8 on page 37](#) for more information. However, it should be pointed out again that only accurate coding produces accurate transliteration. It is therefore at the very least highly advisable to use the hyphen for tying the definite article and the inseparable particles (viz. prepositions, adverbs and conjunctions) to words, like so:—

Al-EaAlamu الْعَالَمُ *al-‘ālam<sup>u</sup>*, Al-camsu الشَّمْسُ *aš-šams<sup>u</sup>*, bi-SinaAEapi Al-T-ib~i, بِصِنَاعَةِ الطِّبِّ *bi-ṣinā‘at<sup>i</sup> ṭ-ṭibb<sup>i</sup>*.

wa-Al-l~ehi وَاللَّهِ *wa-‘l-lāh<sup>i</sup>*, Al-Hamdu li-l~ehi الْحَمْدُ لِلَّهِ *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> ṭ-mamdūdat<sup>i</sup>*’ and the ‘*alif<sup>u</sup> ṭ-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_A1-LntiqAADI` gives أَلَى الْأَيْتِقَاضِ as expected, but only `<iLY_A1-intiqADI` can be transliterated as *‘ilā ṭ-intiqāḍi* with the correct vowel ⟨i⟩ in place of the ‘*alif<sup>u</sup> ṭ-waṣl<sup>i</sup>*’.

## 10 Unicode Arabic input

*New feature v1.5* As said above in [sect. 9 on page 41](#) about the Buckwalter input scheme, even though `arabluatex` is primarily designed to process the `ArabTeX` notation, it also accepts Unicode Arabic input. It should be noted that `arabluatex` does in no way interfere with Unicode Arabic input: none of the `voc`, `fullvoc`, `novoc` or `trans` options will have any effect on plain Unicode Arabic for the time being.

- That said, there are two ways of inserting Unicode Arabic:
- `\txarb` (a) The `\txarb{<Unicode Arabic>}` command for inserting Unicode Arabic text in paragraphs;
- `txarab (env.)` (b) The `txarab` environment for inserting running paragraphs of Arabic text, like so:—

```

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

```

## 11 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 any single-argument command with up to *two optional arguments*—that is: `\command[<opt1>][<opt2>]{<arg>}`—such as `\emph{<text>}`, `\textbf{<text>}` and the like, is assumed to have Arabic text in its mandatory argument:—

`\abjad{45} kitAbu-hu \emph{fI 'l-\uc{`AdAt-i}}` مة 45 كَابُهُ فِي الْعَادَاتِ  
*kitābu-hu fi 'l-Ādāt*.<sup>64</sup>

`\arb{\abjad{45}\rlframebox[1in][s]{kitAbu-hu_fI_'l-`AdAti}}`

مة<sup>65</sup> كَابُهُ فِي الْعَادَاتِ

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 أَيَّ كَانَ مِنَ الْمُقَاتِلَةِ<sup>أ</sup>، وَكَانَتْ أُمِّي مِنْ عُظَمَاءِ بِيوتِ الزَّمَازِمَةِ<sup>ب</sup>.

<sup>أ</sup>الْمُقَاتِلَةُ: الْمُقَاتِلِينَ.  
<sup>ب</sup>الزَّمَازِمَةُ: طَائِفَةٌ مِنَ الْفُرْسِ.

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

- `\LR{<arg>}` is designed to typeset its argument from left to right. It may be used in an Arabic environment, either `\arb{<Arabic text>}` or `\begin{arab} <Arabic text> \end{arab}`, for short insertions of left-to-right text, or to insert any L<sup>A</sup>T<sub>E</sub>X 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.

<sup>64</sup>This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.

<sup>65</sup>`\rlframebox` has been adapted from `\framebox` for insertions of right-to-left text.

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

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

```

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

```

زيد<sup>أ</sup> ابن عمرو

<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 L<sup>A</sup>T<sub>E</sub>X 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 `scrxtext` package.<sup>66</sup>

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

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

```

بِرْزَوِيهِ لِبُرْجُمِهْرَ بْنِ الْبُخْتِكَانَ  
قَالَ بِرْزَوِيهِ بْنُ أَزْهَرَ، رَأْسُ أَطْبَاءِ فَارِسٍ...

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

## 11.1 New commands

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

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

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

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

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

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

```
1 \MkArbBreak{fvred}
2 \newcommand{\fvred}[1]{\arbcOLOR[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 اتَّالَقَ ذُو الْقَرْنَيْنِ إِلَى أُمَّةٍ أُخْرَى فِي ﴿مَطْلَعِ الشَّمْسِ﴾ وَلَا بِنَاءَ لَهُمْ يَوْمَئِذٍ مِنَ الشَّمْسِ.

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

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

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

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

## 11.2 Environments

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

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

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

```
1 \setquotestyle{arabic}
2 \begin{arab}[fullvoc]
3   kAna \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_akiNa-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 bi-'l-'imsAki 'l-`sadIdi. fa-qAla: \enquote{wa-kayfa
9     ra'ayta yA \uc{'abA} \uc{'imrAna} tilka 'l-dajAjaTa?} qAla:
10  \enquote{kAnat `ajabaN mina 'l-`ajabi!} fa-yaqUlu:
11  \begin{quoting}[begintext=\textquotedblright,
12    endtext=\textquotedblleft]
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 bi-'ayyi `say'iN kunnA nusamminu-hA? wa-fI 'ayyi
16    makAniN kunnA na`lifu-hA?
17  \end{quoting}
18  fa-lA yazAlu fI h_a_dA wa-'l-'A_haru ya.d.haku .da.hkaN
19  na`rifu-hu na.hnu wa-lA ya`rifu-hu \uc{'abU} \uc{'l-hu_dayli}.
20 \end{arab}
```

كَانَ أَبُو الْهَدَيْلِ أَهْدَى إِلَى مُوسَى دَجَاجَةً. وَكَانَتْ دَجَاجَتُهُ الَّتِي أَهْدَاهَا دُونَ مَا كَانَ يَتَّخَذُ لِمُوسَى. وَلَكِنَّهُ بَكَرَمِهِ  
وَبِحُسْنِ خَلْقِهِ أَظْهَرَ التَّعَجُّبَ مِنْ سَمِّيَّهَا وَطَيْبِ لِحْمِهَا. وَكَانَ <أَبُو الْهَدَيْلِ> يَعْرِفُ بِالْإِمْسَاكِ الشَّدِيدِ. فَقَالَ: "وَكَيْفَ  
رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ الدَّجَاجَةَ؟" قَالَ: "كَانَتْ عَجَبًا مِنَ الْعَجَبِ!" فَيَقُولُ:

" وَتَدْرِي مَا جِنْسُهَا؟ وَتَدْرِي مَا سَمِّيُّهَا؟ فَإِنَّ الدَّجَاجَةَ إِذَا تَطَيَّبَ بِالْجِنْسِ وَالسِّنِّ. وَتَدْرِي بِأَيِّ شَيْءٍ كُنَّا نَسْمِيْنَهَا؟ وَفِي أَيِّ مَكَانٍ كُنَّا  
نَعْلِفُهَا؟"

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

### 11.2.1 Lists

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

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

```

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

```

رموز الكتاب  
 ب مخطوط المكتبة الأهلية بباريس ٢٨٦٠ عربي.  
 س مخطوط المكتبة الأهلية بباريس ٢٨٥٩ عربي.  
 م مخطوط مجلس شوراى مى 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}

```

القانونُ في الطبِّ  
 - الفنُّ الأوَّلُ في حدِّ الطبِّ وموضوعاته من الأمور الطبيعيَّةِ وشتمِلُ على سِتَّةِ تعاليمٍ  
 - التَّعليمُ الأوَّلُ [وهو فصلان]  
 - الفصلُ الأوَّلُ

As a third example, abjad-numbered lists can be typeset in conjunction with the `enumitem` package,<sup>67</sup> like so:—

```

1 % preamble:---
2 \usepackage{enumitem}
3 \newlist{enumabjad}{enumerate}{10}
4 \setlist[enumabjad]{nosep, label={\abjad{\arabic*}}}
5 \usepackage{multicol}

```

<sup>67</sup>See the documentation of `enumitem` for more details: <https://ctan.org/pkg/enumitem>



```

1 From \textcite[i. 29 B--C]{Wright}:--- The derived forms of the
2 trilateral verb are usually reckoned fifteen in number, but the
3 learner may pass over the last four, because (with the exception
4 of the twelfth) they are of very rare occurrence.
5 \RLmulticolcolumns
6 \begin{multicols}{3}
7   \begin{arab}[fullvoc]
8     \begin{enumabjad}
9       \item fa`ala
10      \item fa``ala
11      \item fA`ala
12      \item `af`ala
13      \item tafa``ala
14      \item tafA`ala
15      \item infa`ala
16      \item ifta`ala
17      \item if`alla
18      \item istaf`ala
19      \item if`Alla
20      \item if`aw`ala
21      \item if`awwala
22      \item if`anlala
23      \item if`anl_A
24     \end{enumabjad}
25   \end{arab}
26 \end{multicols}

```

From Wright:<sup>a</sup> — The derived forms of the trilateral verb are usually reckoned fifteen in number, but the learner may pass over the last four, because (with the exception of the twelfth) they are of very rare occurrence.

يَا إِفْعَالٌ يَبْ إِفْعُوْعَلٌ يَجْ إِفْعُوْعَلٌ يَدْ إِفْعِنَلَلٌ يَهْ إِفْعِنَلِي	وَ تَفَاعَلٌ زَ إِفْعَلٌ حَ إِفْعَلٌ طَ إِفْعَلٌ يَ إِسْتَفْعَلٌ	أَ فَعَلٌ بَ فَعَلٌ جَ فَاعَلٌ دَ أَفْعَلٌ هَ تَفَعَّلٌ
--	--	---

<sup>a</sup>Wright, see n. 7, i. 29 B–C.

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

```

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

```

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

### 11.3 csquotes

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

```
1 \usepackage{csquotes}
2 \DeclareQuoteStyle{arabic}
3 {\textquotedblright}{\textquotedblleft}
4 {\textquoteright}{\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 جَحَا: "غَرِيبٌ أَمْرُكَ يَا صَدِيقِي أَتَصَدَّقُ الْحِمَارَ 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.

### 11.4 Two-argument special commands

`textcolor` The two-argument command `\textcolor{<color>}{<Arabic text>}` is supported inside `\begin{arab} ... \end{arab}`. One simple example follows:<sup>68</sup>—

```
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 عَبْدِ الرَّحِيمِ بْنِ عَلِيٍّ هُوَ شَيْخُنَا الإِمَامُ الصَّدْرُ الكَبِيرُ العَالِمُ الفَاضِلُ مَهْدَبُ الدِّينِ أَبُو مُحَمَّدٍ عَبْدِ الرَّحِيمِ بْنِ عَلِيٍّ بْنِ حَامِدٍ وَيَعْرَفُ بِالدَّخْوَرِ.

<sup>68</sup>arabluatex provides its own `\varbcolor` command which is able to render syllables or diacritics in colors. See [sect. 7 on page 33](#).

*Muhaddabu 'd-Dīni 'Abdu 'r-Raḥīmi bnu 'Aliyy<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}`.<sup>69</sup> As an example, one may get `arabluatex` and `reledmac` to work together like so:—

```

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

```

## 11.5 quran

`arabluatex` is compatible with the `quran` package so that both can be used in conjunction with one another for typesetting the *Qur'ān*. As `quran` draws the text of the *Qur'ān* from a Unicode encoded database, its commands have to be passed as arguments to the `\txarb` command for short insertions in left-to-right paragraphs, or inserted inside the `txarab` environment for typesetting running paragraphs of *Qur'ānic* text (see above [sect. 10 on page 43](#) for more details). Please note that `arabluatex` takes care of formatting the Arabic: therefore, it is recommended to load the `quran` package with the `nopar` option, after `arabluatex` itself has been loaded, like so:—

```

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﴾

## 12 Exporting Unicode Arabic to an external file

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

<sup>69</sup>`\pstart` and `\pend` are also supported inside the `arab` environment.

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

## 12.1 Commands and environments

`export` **export global option** First, `arabluatex` must be loaded with the `export` global option enabled,<sup>70</sup> like so:—

```

1 % preamble
2 \usepackage[export]{arabluatex}
3 % or:
4 \usepackage[export=true]{arabluatex}

```

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

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

`arabexport (env.)` **Exporting running paragraphs** Then, the `arabexport` environment is provided to actually exporting running paragraphs with or without Arabic environments to the external selected file, like so:—

```

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

```

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

- (a) Unicode Arabic text, either in Arabic script or in transliteration, is inserted as argument of `\txarb`<sup>71</sup> or `\txtrans`<sup>72</sup> accordingly.
- `\arbparidir` (b) Additionally, Arabic paragraphs may receive `\arbparidir`, which `arabluatex` uses to determine the direction of Arabic paragraphs to be set by default, or either `\setRL` or `\setLR` depending on what may have been set locally.<sup>73</sup>
- `\prname*` (c) Proper names are inserted as arguments of `\prname*`.<sup>74</sup>

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

<sup>70</sup>See above on page 5 for more information.

<sup>71</sup>See above [sect. 10 on page 43](#).

<sup>72</sup>`\txtrans` is used internally by several Lua functions to format transliterated Arabic. Therefore, it is not documented.

<sup>73</sup>See above on page 45.

<sup>74</sup>See above on page 39.

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

Please note that inside `arabverse` environments `\bayt` is replaced with `\bayt*`.<sup>75</sup>

## 12.2 Nested Arabic environments

The exporting mechanism described above converts only the outermost level of nested Arabic environments. This may be sufficient in some cases, but if nested Arabic environments be found in the original `.tex` source file, then the Unicode converted file must be opened and compiled in turn, and so on until the innermost Arabic environment be converted and exported. In such cases, `arabluatex` issues a warning, so that authors do not have to check the entire file that just has been exported:—

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

Where  $\langle jobname \rangle$  is the name of the original `.tex` source file, and  $\langle suffix \rangle$  the suffix appended to the file that is to be opened and compiled again.

## 12.3 Further processing of Unicode converted files

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

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

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

```
1 % preamble:
2 \usepackage{arabluatex} % note that 'export' has been removed
3 \renewcommand{\txarb}[1]{#1}
4 \renewcommand{\txtrans}[1]{\emph{#1}}
5 \renewcommand{\arbup}[1]{\textsuperscript{#1}}
6 % now that \prname{} has been replaced with \prname*{} it should
7 % be safe to say:
8 \renewcommand{\prname}[2]{#2}
9 % &c
```

## 13 Future work

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

- (a) Short-term:

---

<sup>75</sup>See above 41 on page 26 for more information.

<sup>76</sup>See <http://pandoc.org/>

- i. TEI xml support: arabluatex will interoperate with TEI xml through new global and local options that will output Arabic in a TEI xml compliant file in addition to the usual PDF output: see on page 4.
- (b) Medium-term:
  - i. More languages: the list of supported languages will eventually be the same as arabtex: see 5 on page 4.
  - ii. Formulate propositions for extending the ArabTeX notation and the transliteration tables. Include them in arabluatex. See sect. 4.9 on page 24.

## 14 Implementation

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

```
1 \RequirePackage{iftex}
```

arabluatex requires LuaL<sup>A</sup>T<sub>E</sub>X of course. Issue a warning if the document is processed with another engine.

```
2 \RequireLuaTeX
```

Declare the global options, and define them:

```
3 \RequirePackage{xkeyval}
4 \DeclareOptionX{voc}{\def\al@mode{voc}}
5 \DeclareOptionX{fullvoc}{\def\al@mode{fullvoc}}
6 \DeclareOptionX{novoc}{\def\al@mode{novoc}}
7 \DeclareOptionX{trans}{\def\al@mode{trans}}
8 \define@boolkey{arabluatex.sty}[@pkg@]{export}[true]{%
9   \if@pkg@export%
10  \AtBeginDocument{\luadirect{arabluatex.openstream()}}%
11    \MkArbBreak{@al@ob,@al@cb,@al@cb@sp}}
12  \AtEndDocument{\luadirect{arabluatex.closestream()}}
13  \else\fi}
14 \ExecuteOptionsX{voc}
15 \ProcessOptionsX\relax
16 \def\al@mode@voc{voc}
17 \def\al@mode@fullvoc{fullvoc}
18 \def\al@mode@novoc{novoc}
19 \def\al@mode@trans{trans}
```

Packages that are required by arabluatex:

```
20 \RequirePackage{xcolor}
21 \RequirePackage{luacolor}
22 \RequirePackage{etoolbox}
23 \RequirePackage{arabluatex-patch}
24 \RequirePackage{fontspec}
25 \RequirePackage{luacode}
26 \RequirePackage{xparse}
27 \RequirePackage{adjustbox}
28 \RequirePackage{xstring}
29 \RequirePackage{lualua}
```

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

```
30 \providebool{al@rlmode}
```

Here begins the real work: load arabluatex.lua:

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

Font setup. If no Arabic font is selected, issue a warning message and attempt to load the Amiri font which is included in  $\TeX$ live:

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

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

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

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

```
39 \AtBeginDocument{\def\setLR{\boolfalse{al@rlmode}\pardir TLT%
40   \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:

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

`\SetArbNumbers`

```
44 \NewDocumentCommand{\SetArbNumbers}{m}{%
45   \luadirect{arabluatex.setnums(\luastringN{#1})}%
46 }
```

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

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

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

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

```
50 \NewDocumentCommand{\MkArbBreak}{s m}{%
51   \IfBooleanTF{#1}
52   {\luadirect{arabluatex.mkarbbreak(\luastringN{#2}, "out")}}
53   {\luadirect{arabluatex.mkarbbreak(\luastringN{#2}, "dflt")}}
54 }
```

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

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

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

```

55 \newunderlinetype\@aoverLine{\leaders\vrule height 3ex depth -2.9ex}
56 \def\@oline{\ifstar\@oline\@@oline}
57 \def\@oline#1{\ensuremath{\overline{\mbox{#1}}}}
58 \def\@@oline#1{\@aoverLine#1}
59 \newunderlinetype\@aunderLine{\leaders\vrule height -.65ex depth .75ex}
60 \def\@uline#1{\@aunderLine#1}
61 \AtBeginDocument{\ifdef\@emph}%
62   {\RenewDocumentCommand\@emph}{s m}{%
63     \IfBooleanTF{#1}{%
64       \luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
65         "over"))}}
66     {\luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
67       "dflt"))}}}
68   {\NewDocumentCommand\@emph}{s m}{%
69     \IfBooleanTF{#1}{%
70       \luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
71         "over"))}}
72     {\luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
73       "dflt"))}}}

```

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

```

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

```

`\SetInputScheme` arabluatex is designed for processing Arab<sub>TEX</sub> 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’.

```

76 \def\al@input@scheme{arabtex}
77 \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`, `fullvoc` or `trans`. Such refinements can be discarded with `\SetArbEasy`, either globally in the preamble or at any point of the document. Note that `\SetArbEasy` keeps the *sukūn* that is generated, while the starred version `\SetArbEasy*` takes it away. Default complex rules can be set back at any point of the document with `\SetArbDflt`.

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

```

78 \def\al@arb@rules{dflt}
79 \NewDocumentCommand\SetArbEasy{s}{%
80   \IfBooleanTF{#1}
81   {\def\al@arb@rules{easynosukun}}
82   {\def\al@arb@rules{easy}}}
83 \NewDocumentCommand\SetArbDflt{s}{%
84   \IfBooleanTF{#1}
85   {\def\al@arb@rules{idgham}}
86   {\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.

```

87 \def\al@trans@font{\rmfamily}%
88 \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:

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

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

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

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

```
92 \def\al@trans@convention{dmg}
93 \NewDocumentCommand{\SetTranslitConvention}{m}{%
94   \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. `\ArbUpDflt` The default rule can be set back with `\ArbUpDflt` at any point of the document. Finally `\SetArbUp` can be used to customize the way *tanwīn* is displayed: this command takes the formatting directives as argument, like so: `\SetArbUp{<code>}`.

```
95 \NewDocumentCommand{\al@arbup@dflt}{m}{\textsuperscript{#1}}%
96 \NewDocumentCommand{\al@arbup}{m}{\al@arbup@dflt{#1}}
97 \NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
98 \NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt}
99 \NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\al@arbup}{m}{##1}}
100 \NewDocumentCommand{\SetArbUp}{m}{%
101   \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.

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

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

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

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

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

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

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

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

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

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

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

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

```
117 \NewDocumentEnvironment{txarabtr}{}{%
118   \par%
119   \pardir TLT\textdir TLT%
120   \altrfont}{\par}
```

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

```
121 \NewDocumentCommand{\arb}{0{\al@mode} +m}{%
122   {\edef\@tempa{#1}%
123     \ifx\@tempa\al@mode@voc%
124       \ifvmode\leavevmode\fi%
125       \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
126       \luadirect{tex.sprint(arabluatex.processvoc(\luastringN{#2},
127         \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
128     }else%
129     \ifx\@tempa\al@mode@fullvoc%
130       \ifvmode\leavevmode\fi%
131       \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
132       \luadirect{tex.sprint(arabluatex.processfullvoc(\luastringN{#2},
133         \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
134     }else%
135     \ifx\@tempa\al@mode@novoc%
136       \ifvmode\leavevmode\fi%
137       \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
138       \luadirect{tex.sprint(arabluatex.processnovoc(\luastringN{#2},
139         \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
140     }else%
141     \ifx\@tempa\al@mode@trans%
142       \bgroup\textdir TLT%
143       \luadirect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
144         \luastringO{\al@trans@convention},
145         \luastringO{\al@arb@rules},
146         \luastringO{\al@input@scheme}))}\egroup%
147     }else%
```

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

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

```
149 \NewDocumentCommand{\arbmark}{0{} m}{%
150   \bgroup%
151   \SetInputScheme{arabtex}%
152   \luadirect{tex.sprint(arabluatex.processarbmarks(\luastringN{#2},
153     \luastringN{#1}))}%
154   \egroup}
```

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

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

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

```
158 \NewDocumentEnvironment{arab}{!0{\al@mode} +b}{%
159   {\par\edef\@tempa{#1}%
160     \ifx\@tempa\al@mode@voc%
161       \booltrue{al@rlmode}%
162       \bgroup\pdir TRT\textdir TRT\arabicfont%
163       \luadirect{tex.sprint(arabluatex.processvoc(\luastringN{#2},
164         \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
165       \else%
166         \ifx\@tempa\al@mode@fullvoc%
167           \booltrue{al@rlmode}%
168           \bgroup\pdir TRT\textdir TRT\arabicfont%
169           \luadirect{tex.sprint(arabluatex.processfullvoc(\luastringN{#2},
170             \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
171           \else%
172             \ifx\@tempa\al@mode@novoc%
173               \booltrue{al@rlmode}%
174               \bgroup\pdir TRT\textdir TRT\arabicfont%
175               \luadirect{tex.sprint(arabluatex.processnovoc(\luastringN{#2},
176                 \luastringO{\al@arb@rules}, \luastringO{\al@input@scheme}))}\egroup%
177               \else%
178                 \ifx\@tempa\al@mode@trans%
179                   \bgroup\pdir TLT\textdir TLT%
180                   \luadirect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
181                     \luastringO{\al@trans@convention},
182                     \luastringO{\al@arb@rules},
183                     \luastringO{\al@input@scheme}))}\egroup%
184                   \else \fi\fi\fi\fi}{\par}
```

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

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

```

187 \newlength{\al@verse@twidth}
188 \setlength{\al@bayt@width}{.3\textwidth}
189 \setlength{\al@gutter@width}{.15\al@bayt@width}
190 \define@key[al]{verse}{width}{\setlength{\al@bayt@width}{#1}}
191 \define@key[al]{verse}{gutter}{\setlength{\al@gutter@width}{#1}}
192 \define@key[al]{verse}{metre}{\def\al@verse@metre@value{\arb{#1}}}
193 \define@key[al]{verse}{color}[]{\color{#1}}
194 \define@boolkey[al]{verse}{utf}[true]{}
195 \define@boolkey[al]{verse}{delim}[true]{}
196 \define@boolkey[al]{verse}{export}[true]{}
197 \define@choicekey[al]{verse}{mode}{fullvoc, voc, novoc,
198   trans}{\def\al@mode{#1}}
199 \presetkeys[al]{verse}{utf=false, delim=false}{}

```

Then follows the environment itself:

```

200 \NewDocumentEnvironment{arabverse}{!0{}}%
201 {\bgroup\setkeys[al]{verse}[width, gutter, color, utf, delim,
202   metre]{#1}%
203   \if@pkg@export
204     \ifal@verse@export
205       \ArbOutFile{\begin{arabverse}}%
206       % \ifx\al@mode\al@mode@trans%
207       % \luadirect{arabluatex.tooutfile(\luastringN{[#1]})}%
208       % \else%
209       \IfSubStr[1]{#1}{utf}%
210       {\luadirect{arabluatex.tooutfile(\luastringN{[#1]})}}%
211       {\luadirect{arabluatex.tooutfile(\luastringN{[#1, utf]})}}%
212       % \fi
213     \else
214     \fi
215   \else
216   \fi
217   \egroup
218   \centering\noindent\bgroup\setkeys[al]{verse}[metre]{#1}%
219   % \ifx\al@mode\al@mode@trans%
220   % \ifal@verse@utf\setRL\else\setLR\fi%
221   % \else\setRL\fi%
222   \ifal@verse@utf
223     \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi
224   \else
225     \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi
226   \fi
227   \addtolength{\al@verse@twidth}{2\al@bayt@width}%
228   \addtolength{\al@verse@twidth}{\al@gutter@width}%
229   \arabv@export[#1]
230 }%
231 {\endarabv@export
232   \setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
233   export]{#1}%
234   \ifdefined\al@verse@metre@value\hfill\al@verse@metre@value\fi
235 \egroup
236 \bgroup\setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
237   metre]{#1}%
238 \if@pkg@export\ifal@verse@export
239   \ArbOutFile{\end{arabverse}}%
240 \else\fi\else\fi\egroup}

```

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

`\StretchBayt` `\StretchBayt[⟨true/false⟩]` Allows to remove stretching and undesirable warping effect from Arabic lines of poetry. This command accepts one fixed optional argument, either `true` or `false`, and may be used either in the preamble or at any point of the document. By default, it is set to `true`.

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

```

241 \newif\ifal@warp@bayt
242 \al@warp@bayttrue
243 \NewDocumentCommand{\StretchBayt}{0{true}}{
244   \edef\oarg@true{true}
245   \edef\oarg@false{false}
246   \edef\@tempa{#1}
247   \ifx\@tempa\oarg@true\al@warp@bayttrue
248   \else
249   \ifx\@tempa\oarg@false\al@warp@baytfalse
250   \else
251   \PackageError{arabluatex}{\string\StretchBayt\space must be
252     either 'true' or 'false'}}{
253   \fi
254   \fi
255 }
256 \NewDocumentCommand{\arb@utf}{m}{%
257   \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
258 \def\al@hemistich@delim{*}
259 \NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
260 \def\al@verse@stroke{\leavevmode\xleaders\hbox{\arb{--}}\hfill\kernOpt}
261 \providebool{ekd@state}
262 \NewDocumentCommand{\bayt}{t+ s m o m}{%
263   \IfBooleanTF{#1}{%
264     \ifekd@state
265     \leavevmode
266     \stepcounter{ekd@lab}%
267     \zlabel{ekd:\theekd@lab}%
268     \luadirect{ekdosis.storeabspg(
269       \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}}{%
270     \add@apparatus
271     \fi
272   }{\relax}%
273   \IfBooleanTF{#2}{\relax}{\relax}%
274   \ifdefined\savenotes\savenotes\else\fi%
275   \edef\al@tatweel{--}%
276   \ifal@warp@bayt%
277     \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
278   \else%
279     \makebox[\al@bayt@width] [s]{\arb@utf{#3}}%
280   \fi%
281   \IfNoValueTF{#4}{%

```

```

282 \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
283 \else%
284 \hspace{\al@gutter@width}%
285 \fi
286 }{%
287 \edef\@tempa{#4}%
288 \ifx\@tempa\al@tatweel%
289 \ifx\al@mode\al@mode@trans%
290 \hspace{\al@gutter@width}%
291 \else%
292 \makebox[\al@gutter@width][s]{\al@verse@stroke}%
293 \fi%
294 \else%
295 \ifx\al@mode\al@mode@trans%
296 \ifal@warp@bayt%
297 \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#4}}%
298 \else%
299 \makebox[\al@gutter@width][s]{\arb@utf{#4}}%
300 \fi%
301 \else%
302 \makebox[\al@gutter@width][s]{\arb@utf{#4}}%
303 \fi\fi}%
304 \ifal@warp@bayt%
305 \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#5}}%
306 \else%
307 \makebox[\al@bayt@width][s]{\arb@utf{#5}}%
308 \fi%
309 \ifdefined\spewnotes\spewnotes\else\fi%
310 }

```

`\arind` `\arind{<root>}` is a command specialized in the construction of indexes. As a mandatory argument, it takes the Arabic root under which a given word is to be indexed. Additionally, it may receive three optional ‘named’ arguments: `index`, `root` and `form`.

```

311 \NewDocumentCommand{\SetDefaultIndex}{m}{
312 \edef\@tempa{#1}
313 \ifx\@tempa\empty
314 \def\al@default@index{\jobname}
315 \else
316 \def\al@default@index{#1}
317 \fi
318 }

319 \def\al@index@mode{\al@mode}
320 \NewDocumentCommand{\SetIndexMode}{m}{
321 \def\al@index@mode{#1}
322 }

323 \define@cmdkeys[al]{index}[alind@]{index,root,form,pipe}
324 \NewDocumentCommand{\arind}{o m}{%
325 \IfNoValueTF{#1}{%
326 \ifdefined\al@default@index%
327 \csname index\endcsname[\al@default@index]{#2}%
328 \else%
329 \csname index\endcsname{#2}%
330 \fi%
331 }{%
332 \bgroup

```

```

333 \setkeys[al]{index}{#1}%
334 \def\al@one{%
335   \ifdefined\alind@root
336     \ifnum\alind@root < 10%
337       !\LR{0\alind@root}%
338     \else
339       !\LR{\alind@root}%
340     \fi
341   \else
342     !\LR{01}%
343   \fi}%
344 \def\al@two{%
345   \ifdefined\alind@form @\arb[\al@index@mode]{\alind@form}\else\fi}%
346 \ifdefined\alind@index%
347   \ifdefined\alind@pipe
348     \csname index\endcsname[\alind@index]{#2\al@one\al@two|\alind@pipe}%
349   \else
350     \csname index\endcsname[\alind@index]{#2\al@one\al@two}%
351   \fi
352 \else%
353   \ifdefined\al@default@index%
354     \ifdefined\alind@pipe
355       \csname index\endcsname[\al@default@index]{%
356         #2\al@one\al@two|\alind@pipe}%
357     \else
358       \csname index\endcsname[\al@default@index]{%
359         #2\al@one\al@two}%
360     \fi
361   \else%
362     \ifdefined\alind@pipe
363       \csname index\endcsname{#2\al@one\al@two|\alind@pipe}%
364     \else
365       \csname index\endcsname{#2\al@one\al@two}%
366     \fi
367   \fi%
368 \fi%
369 \egroup}}

```

`\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.

```

370 \AtBeginDocument{%
371   \ifdefined\abjad%
372     \RenewDocumentCommand{\abjad}{m}%
373     {\ifbool{al@rlmode}%
374       {\aoline*{%
375         \luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}}
376     {\luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}}
377 \else%
378   \NewDocumentCommand{\abjad}{m}%
379   {\ifbool{al@rlmode}%
380     {\aoline*{%
381       \luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}}
382   {\luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}}
383 \fi}

```



`\ayah` `\ayah{<number>}` prints up to 3-digit numbers inside ‘end of Ayah’ sign (U+06DD) or inside parentheses depending on the mode which is selected.

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

`\arbnnull` The `\arbnnull` 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.

```
386 \NewDocumentCommand{\arbnnull}{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`.

```
387 \NewDocumentCommand{\abraces}{+m}{%
388   \luadirect{tex.sprint(arabluatex.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.

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

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

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

```
393 \DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pdir
394   TLT\textdir TLT\footnote{#1}\egroup}
395 \DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pdir
396   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.

```
397 \NewDocumentCommand{\FixArbFtnmk}{}{%
398   \@ifpackageloaded{scrextend}%
399   {\AtBeginDocument{%
400     \deffootnote{2em}{1.6em}{\LR{thefootnotemark}.\enskip}}}%
401   {\RequirePackage{scrextend}
402     \AtBeginDocument{%
403     \deffootnote{2em}{1.6em}{\LR{thefootnotemark}.\enskip}}}}
```

### Exporting Unicode Arabic to external file

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

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

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



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

```

406 \NewDocumentCommand{\ArbOutFile}{s O{no} +m}{%
407   \if@pkg@export%
408   \IfBooleanTF{#1}{%
409     #3\luadirect{arabluatex.tooutfile(\luastringN{#3}, "#2")}}{%
410     \luadirect{arabluatex.tooutfile(\luastringN{#3}, "#2")}}%
411   \else\IfBooleanTF{#1}{#3}\fi}

```

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

```

412 \NewDocumentEnvironment{arabexport}{+b}{%
413   \if@pkg@export%
414   \par
415   #1
416   \luadirect{arabluatex.doexport("yes")}
417   \luadirect{tex.sprint(arabluatex.arbtoutf(\luastringN{#1}))}
418   \luadirect{arabluatex.doexport("no")}
419   \else\par#1\fi
420 }{\par}

```

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

```

421 \NewDocumentEnvironment{arab@v@export}{0{} +b}{%
422   \setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
423   metre]{#1}
424   \if@pkg@export
425     \ifal@verse@export
426       \ifekd@state
427         \begin{ekdverse}[width=\al@verse@twidth]
428           #2
429         \end{ekdverse}
430       \else
431         #2
432       \fi
433       \luadirect{arabluatex.doexport("arabverse")}%
434       \luadirect{tex.sprint(arabluatex.arbtoutf(\luastringN{#2}))}%
435       \luadirect{arabluatex.doexport("no")}%
436     \else
437       \ifekd@state
438         \begin{ekdverse}[width=\al@verse@twidth]
439           #2
440         \end{ekdverse}
441       \else
442         #2
443       \fi
444     \fi
445   \else
446     \ifekd@state
447       \begin{ekdverse}[width=\al@verse@twidth]
448         #2
449       \end{ekdverse}
450     \else

```

```

451     #2
452     \fi
453 \fi
454 }{}

```

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

```

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

```

## Errors and Warnings

```

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

```

That is it. Say goodbye before leaving.

## Patches

```

466 \NeedsTeXFormat{LaTeX2e}
467 \ProvidesPackage{arabluatex-patch}%
468 [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:

```

469 \def\list#1#2{%
470   \ifnum \@listdepth >5\relax
471     \@toodeep
472   \else
473     \global\advance\@listdepth\@ne
474   \fi
475   \rightmargin\z@
476   \listparindent\z@
477   \itemindent\z@
478   \csname @list\romannumeral\the\@listdepth\endcsname
479   \def\@itemlabel{#1}%
480   \let\makelabel\@mklab
481   \@nmbrolistfalse
482   #2\relax
483   \@trivlist
484   \parskip\parsep
485   \parindent\listparindent
486   \advance\linewidth -\rightmargin
487   \advance\linewidth -\leftmargin

```

patch begins:

```

488 \ifbool{al@rlmode}{\advance\@totalleftmargin \rightmargin}%
489 {\advance\@totalleftmargin \leftmargin}
patch ends.
490 \parshape \@ne \@totalleftmargin \linewidth
491 \ignorespaces}
492 \def\@item[#1]{%
493 \if@noperitem
494 \@donoperitem
495 \else
496 \if@inlabel
497 \indent \par
498 \fi
499 \ifhmode
500 \unskip\unskip \par
501 \fi
502 \if@newlist
503 \if@nobreak
504 \@nbitem
505 \else
506 \addpenalty\@beginparpenalty
507 \addvspace\@topsep
508 \addvspace{-\parskip}%
509 \fi
510 \else
511 \addpenalty\@itempenalty
512 \addvspace\itemsep
513 \fi
514 \global\@inlabeltrue
515 \fi
516 \everypar{%
517 \@minipagefalse
518 \global\@newlistfalse
519 \if@inlabel
520 \global\@inlabelfalse
521 {\setbox\z@\lastbox
522 \ifvoid\z@
523 \kern-\itemindent
524 \fi}%
525 \box\@labels
526 \penalty\z@
527 \fi
528 \if@nobreak
529 \@nobreakfalse
530 \clubpenalty \@M
531 \else
532 \clubpenalty \@clubpenalty
533 \everypar{}%
534 \fi}%
535 \if@noitemarg
536 \@noitemargfalse
537 \if@nbrlist
538 \refstepcounter\@listctr
539 \fi
540 \fi
patch begins:

```

```

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

```

patch ends.

```

545 \unhbox\@labels
546 \hskip \itemindent
547 \hskip -\labelwidth
548 \hskip -\labelsep
549 \ifdim \wd\@tempboxa >\labelwidth
550 \box\@tempboxa
551 \else
552 \hbox to\labelwidth {\unhbox\@tempboxa}%
553 \fi
554 \hskip \labelsep}%
555 \ignorespaces}

```

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

```

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

```

## 15 References

- Alessi, Robert, *The Ekdosis package: Typesetting TEI-xml compliant Critical Editions* (version 1.4) (Nov. 21, 2021), <http://www.ekdosis.org>.
- Brockelmann, Carl, et al., “Die Transliteration der arabischen Schrift in ihrer Anwendung auf die Hauptliteratursprachen der islamischen Welt”, in *Denkschrift dem 19. internationalen Orientalistenkongreß in Rom vorgelegt von der Transkriptionkommission der Deutschen Morgenländischen Gesellschaft*, in collab. with Ph. S. van Ronkel and Otto Spies (Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockhaus, 1935), [http://www.naher-osten.uni-muenchen.de/studium\\_lehre/we rkzeugkasten/dmgtransliteration.pdf](http://www.naher-osten.uni-muenchen.de/studium_lehre/we rkzeugkasten/dmgtransliteration.pdf).
- Habash, Nizar Y., *Introduction to Arabic Natural Language Processing* (Synthesis Lectures on Human Language Technologies, 10; Toronto: Morgan & Claypool Publishers, 2010).
- Hosny, Khaled, *Amiri* (Dec. 13, 2017), <http://www.amirifont.org/>.
- Information and Documentation - Romanization of the Arabic Alphabet for Arabic, Ottoman-Turkish, Persian, Kurdish, Urdu and Pushto* (July 2011), <http://www.din.de>.
- Krüger, Marcel, *The Lua-ul package: Underlining for LuaLaTeX* (version 0.0.1) (Mar. 12, 2020), <http://www.ctan.org/pkg/lua-ul>.
- Lagally, Klaus, *ArabTeX: Typesetting Arabic and Hebrew* (version 4.00) [User Manual Version 4.00] (Nov. 3, 2004), <http://mirrors.ctan.org/language/arabic/arabtex/doc/html/arabtex.htm>.
- Lane, Edward William, *An Arabic-English lexicon*, 8 vols. (London – Edinburgh: Williams and Norgate, 1863–1893).
- Wright, W. LL.D, *A Grammar of the Arabic Language*, rev. W. Robertson Smith and M. J. de Goeje, with a foreword by Pierre Cachia, 2 vols. (3rd edn., Beirut: Librairie du Liban, 1896).

## 16 Change History

v1.0.	General: Initial release . . . . .	1	\SetHemistichDelim: New \SetHemistichDelim command for changing the default delimiter between hemistichs . . . . .	61
v1.0.1.	General: Minor update of the documentation . . . . .	1		
v1.1.	\abjad: New and more flexible \abjad command. . . . .	63	v1.7. \arbnnull: New \arbnnull command for putting back on any contextual analysis rule broken by other commands. . . . .	64
v1.2.	\SetArbEasy: New \SetArbEasy/\SetArbDflt for 'modern' or 'classic' Arabic styles. . .	56	v1.8. General: arabica transliteration standard is now supported . . . . .	37
v1.3.	\arbut: <i>ʾrāb</i> is now written as superscript text in dmg mode by default. . . . .	57	v1.8.5. General: Six additional Persian characters are now available . . . . .	9
v1.4.	\SetInputScheme: \SetInputScheme can be used to process other input schemes such as 'Buckwalter' . . . .	56	v1.9. \MkArbBreak: New \MkArbBreak command for inserting user-defined macros in Arabic environments . . .	55
	\SetTranslitFont: For selecting a specific font for transliterated texts	56	v1.9.2. \aemph*: Starred version which always puts the stroke over its argument . .	55
v1.4.3.	\abraces: New \abraces command which expresses its argument between braces. . . . .	64	v1.10. General: \uc supersedes \cap . . . . .	38
v1.4.4.	\SetArbEasy*: this starred version discards the <i>sukūn</i> in addition to what is already discarded by \SetArbEasy. . . . .	56	\prname: New command for typesetting Arabic proper names in transliteration . . . . .	57
v1.5.	General: Compatibility with the quran package . . . . .	51	v1.11. \arbookmark: New command for inserting additional marks in Arabic environments . . . . .	59
	Environments may be nested inside the arab environment . . . . .	47	\newarbookmark: Allows defining additional sets of Arabic marks . . . . .	59
	txarab: New txarab environment for typesetting running paragraphs in Unicode Arabic . . . . .	58	v1.12. General: \abjad can now process L <sup>A</sup> T <sub>E</sub> X counters . . . . .	24
v1.6.	arabverse: New environment arabverse for typesetting Arabic poetry . . . . .	59	\arbcOLOR: Standard color command for Arabic environments . . . . .	56
	\bayt: New macro \bayt for typesetting each verse inside the arabverse environment . . . . .	61	\MkArbBreak*: 'starred' version which closes Arabic environments before processing declared commands. . . .	55
	\SetArbDflt*: This starred version applies the assimilation rules in addition to what \SetArbDflt already does. . . . .	56	v1.13. arabexport: Processes and print its argument in the current file and exports it in full Unicode in the external selected .tex file. . . . .	65
			arabverse: New options color and export to arabverse environment. . .	59
			\arbookmark: New optional argument: either rl or lr . . . . .	59
			\ArbOutFile: Silently exports its argument in the selected external file.	64

<code>\arbpardir</code> : Sets the direction of Arabic paragraphs once they are converted to Unicode. . . . .	66	v1.19.	<code>\aemph</code> : Now uses lua-ul . . . . .	55
<code>\prname*</code> : Renders proper names already converted to Unicode in upright roman style . . . . .	57	<code>\auline</code> : Non context-sensitive command to underline Arabic words is provided . . . . .	55	
<code>\SetArbOutSuffix</code> : Sets a suffix to be appended to the filename of the external Unicode file. . . . .	64	v1.20.	<code>\StretchBayt</code> : Optionally removes stretching from lines of poetry . . .	61
v1.15.		v1.21.	<code>\altrfont</code> : new command <code>\altrfont</code> to store font and shape information (trans mode) . . . . .	57
<code>\ayah</code> : Prints End of Ayah sign . . . . .	63	<code>\SetArbNumbers</code> : selects Indian or Arabic numbers . . . . .	55	
v1.16.				
<code>\aemph</code> : Now uses ulem . . . . .	55			
v1.18.				
<code>\arind</code> : New command <code>\arind</code> for building indexes . . . . .	62			

## 17 Index

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

Symbols		
<code>\@@aoline</code> . . . . .	56, 58	<code>\@tempboxa</code> . . . . .
<code>\@M</code> . . . . .	530	541, 542, 549, 550, 552
<code>\@aoline</code> . . . . .	56, 57	<code>\@toodeep</code> . . . . .
<code>\@aoverLine</code> . . . . .	55, 58	471
<code>\@aunderLine</code> . . . . .	59, 60	<code>\@topsep</code> . . . . .
<code>\@beginparpenalty</code> . . . . .	506	507
<code>\@clubpenalty</code> . . . . .	532	<code>\@totalleftmargin</code> . . . . .
<code>\@donoparitem</code> . . . . .	494	488, 489, 490
<code>\@ifpackageloaded</code> . . . . .	398	<code>\@trivlist</code> . . . . .
<code>\@ifstar</code> . . . . .	56	483
<code>\@inlabelfalse</code> . . . . .	520	<code>\_</code> . . . . .
<code>\@inlabeltrue</code> . . . . .	514	34
<code>\@item</code> . . . . .	492	
<code>\@itemlabel</code> . . . . .	479	
<code>\@itempenalty</code> . . . . .	511	
<code>\@labels</code> <i>525, 543, 544, 545</i>		
<code>\@listctr</code> . . . . .	538	
<code>\@listdepth</code> <i>470, 473, 478</i>		
<code>\@minipagefalse</code> . . . . .	517	
<code>\@mklab</code> . . . . .	480	
<code>\@nbitem</code> . . . . .	504	
<code>\@ne</code> . . . . .	473, 490	
<code>\@newlistfalse</code> . . . . .	518	
<code>\@nmbriestfalse</code> . . . . .	481	
<code>\@nbreakfalse</code> . . . . .	529	
<code>\@noitemargfalse</code> . . . . .	536	
<code>\@tempa</code> <i>122, 123, 129, 135,</i>		
<i>141, 159, 160, 166,</i>		
<i>172, 178, 246, 247,</i>		
<i>249, 287, 288, 312, 313</i>		
		<b>A</b>
		<code>\abjad</code> . . . . .
		<i>23, 370</i>
		<code>\abraces</code> . . . . .
		<i>30, 387</i>
		<code>\add@apparatus</code> . . . . .
		<i>270</i>
		<code>\addpenalty</code> . . . . .
		<i>506, 511</i>
		<code>\addtolength</code> . . . . .
		<i>227, 228</i>
		<code>\addvspace</code> . . . . .
		<i>507, 508, 512</i>
		<code>\adjustbox</code> . . . . .
		<i>277, 297, 305</i>
		<code>\advance</code> . . . . .
		<i>473,</i>
		<i>486, 487, 488, 489</i>
		<code>\aemph</code> . . . . .
		<i>25, 55</i>
		<code>\aemph*</code> . . . . .
		<i>25, 55</i>
		<code>\al@arb@rules</code> . . . . .
		. . . . .
		<i>78, 81, 82, 85,</i>
		<i>86, 127, 133, 139,</i>
		<i>145, 164, 170, 176, 182</i>
		<code>\al@arbup</code> <i>96, 97, 98, 99, 101</i>
		<code>\al@arbup@dfilt</code> . <i>95, 96, 98</i>
		<code>\al@bayt@width</code> . . . . .
		<i>185, 188, 189, 190,</i>
		<i>227, 277, 279, 305, 307</i>
		<code>\al@default@index</code> . . . . .
		<i>314, 316,</i>
		<i>326, 327, 353, 355, 358</i>
		<code>\al@error</code> . . . . .
		<i>458, 459</i>
		<code>\al@gutter@width</code> . . . . .
		<i>186, 189,</i>
		<i>191, 228, 282, 284,</i>
		<i>290, 292, 297, 299, 302</i>
		<code>\al@hemistich@delim</code> . . . . .
		<i>258, 259, 282</i>
		<code>\al@index@mode</code> <i>319, 321, 345</i>
		<code>\al@input@scheme</code> . . . . .
		<i>76,</i>
		<i>77, 127, 133, 139,</i>
		<i>146, 164, 170, 176, 183</i>
		<code>\al@mode</code> . . . . .
		<i>4, 5, 6, 7, 121, 158,</i>
		<i>198, 206, 219, 223,</i>
		<i>225, 289, 295, 319, 456</i>
		<code>\al@mode@fullvoc</code> <i>17, 129, 166</i>
		<code>\al@mode@novoc</code> <i>18, 135, 172</i>
		<code>\al@mode@trans</code> . . . . .
		<i>19,</i>
		<i>141, 178, 206, 219,</i>
		<i>223, 225, 289, 295, 456</i>
		<code>\al@mode@voc</code> . . . . .
		<i>16, 123, 160</i>
		<code>\al@one</code> . . . . .
		<i>334, 348, 350,</i>
		<i>356, 359, 363, 365</i>
		<code>\al@tatweel</code> . . . . .
		<i>275, 288</i>
		<code>\al@trans@convention</code> . . . . .
		<i>92, 94, 144, 181</i>
		<code>\al@trans@font</code> . . . . .
		<i>87, 88, 91</i>
		<code>\al@trans@style</code> <i>89, 90, 91</i>

`\al@two` . 344, 348, 350, 356, 359, 363, 365  
`\al@verse@metre@value`  
    ..... 192, 234  
`\al@verse@stroke` 260, 292  
`\al@verse@width` . 187,  
    227, 228, 427, 438, 447  
`\al@warning` .... 457, 463  
`\al@warp@baytfalse` .. 249  
`\al@warp@bayttrue` 242, 247  
`\al@wrong@mark` ..... 463  
`\al@wrong@nesting` ... 459  
Alessi, Robert ..... 28  
`\alind@form` ..... 345  
`\alind@index` 346, 348, 350  
`\alind@pipe` ..... 347,  
    348, 354, 356, 362, 363  
`\alind@root` .....  
    .. 335, 336, 337, 339  
`\altrfont` ..... 91, 112, 120  
*Amiri* ..... 5  
amiri (package) ..... 5  
`\aoline` .. 25, 55, 374, 380  
`\aoline*` ..... 25, 55  
arab (env.) ..... 7, 158  
`\arab@v@export` ..... 229  
`\arab@v@export (env.)` .. 421  
*ArabTeX* ..... 3, 19  
arabexport (env.) .. 52, 412  
*Arabic-English Lexicon*,  
    *An* ..... 18  
`\arabicfont` ..... 32,  
    34, 36, 110, 116, 125,  
    131, 137, 162, 168, 174  
arabluatex (package) 2–5,  
    7, 8, 10–19, 23–26,  
    28, 30, 32–34,  
    37–46, 49–57, 64–66  
arabtex (package) .....  
    ..... 3, 4, 15, 19, 54  
arabulatex (package) .... 16  
arabverse (env.) .. 25, 185  
arabxetex (package) ... 4, 15  
`\arb` ..... 7, 107,  
    121, 192, 257, 260, 345  
`\arb@utf` . 256, 277, 279,  
    297, 299, 302, 305, 307  
`\arbcolor` ..... 33, 74  
`\arbmark` .... 30, 149, 464  
`\arbnull` ..... 22, 386  
`\ArbOutFile` .....  
    ... 52, 205, 239, 406  
`\ArbOutFile*` ..... 52, 406  
`\arbpardir` ..... 52, 455  
`\arbut` ..... 39, 95  
`\ArbUpDflt` ..... 39, 95  
`\arind` ..... 311  
`\AtEndDocument` ..... 12  
`\aoline` ..... 25, 55  
`\ayah` ..... 32, 384  

### B

babel (package) ..... 49  
babel-french (package) .. 49  
`\bayt` ..... 26, 241  
`\bayt+` ..... 28  
`\begin` .. 205, 427, 438, 447  
bidi (package) ..... 68  
`\boolfalse` ..... 39  
`\booltrue` . 37, 115, 125,  
    131, 137, 161, 167, 173  
`\box` ..... 525, 550  
Brockelmann, Carl 10, 37, 39  

### C

`\centering` ..... 218  
`\clubpenalty` .... 530, 532  
`\color` ..... 193  
color (option) ..... 26  
`\color@endgroup` ..... 557  
`\color@setgroup` ..... 557  
csquotes (package) .... 50  

### D

`\DeclareOptionX` . 4, 5, 6, 7  
`\deffootnote` .... 400, 403  
`\define@boolkey` .....  
    ..... 8, 194, 195, 196  
`\define@choicekey` ... 197  
`\define@cmdkeys` ..... 323  
`\define@key` 190, 191, 192, 193  
`\delim (option)` ..... 25  

### E

`\empty` ..... 313  
`\end` ... 239, 429, 440, 449  
`\endarab@v@export` ... 231  
`\ensuremath` ..... 57  
enumitem (package) .. 47–49  
environments:  
    arab ..... 7, 158  
    arab@v@export .... 421  
    arabexport .... 52, 412  
    arabverse .... 25, 185  
    txarab ..... 44, 113  
    txarabtr ..... 117  
`\everypar` ..... 516, 533  
`\ExecuteOptionsX` ..... 14  
export (option) ... 5, 26, 52  

### F

`\FixArbFtnmk` ..... 45, 397  
fontspec (package) . 4, 37, 56  
`\footnote` .. 394, 396, 460  
footnote (package) ..... 28  
footnotehyper (package) .. 28  
fullvoc (option) ..... 5, 8  

### G

`\global` ..... 473,  
    514, 518, 520, 543, 544  
*Grammar of the Arabic*  
    *Language, A* 5–8,  
    12–17, 23, 24, 39, 49  
gutter (option) ..... 25  

### H

Habash, Nizar Y. .... 42  
`\hbox` 260, 543, 544, 552, 556  
`\Height` ..... 277, 297, 305  
`\hfill` ..... 234, 260  
Hosny, Khaled ..... 5  
`\hskip` . 546, 547, 548, 554  
`\hspace` ..... 284, 290  

### I

`\if@inlabel` .... 496, 519  
`\if@newlist` ..... 502  
`\if@nmbrrlist` ..... 537  
`\if@nobreak` .... 503, 528  
`\if@noitemarg` ..... 535  
`\if@noparitem` ..... 493  
`\if@pkg@export` .... 9,  
    203, 238, 407, 413, 424  
`\ifal@verse@delim` ... 282  
`\ifal@verse@export` ..  
    ..... 204, 238, 425  
`\ifal@verse@utf` .....  
    ..... 220, 222, 257  
`\ifal@warp@bayt` .....  
    .. 241, 276, 296, 304  
`\ifbool` .....  
    373, 379, 488, 541, 543  
`\IfBooleanTF` .....  
    51, 63, 69, 80, 84,  
    107, 263, 273, 408, 411  
`\ifdim` ..... 549  
`\ifekd@state` .....  
    .. 264, 426, 437, 446  
`\ifhmode` ..... 499  
`\IfNoValueTF` .....  
    ... 75, 281, 325, 390  
`\ifnum` ..... 336, 470  
`\IfSubStr` ..... 209



`\ifvmode` . 109, 124, 130, 136  
`\ifvoid` . . . . . 522  
`\ignorespaces` . . . 491, 555  
`\indent` . . . . . 497  
*Information and Documentation - Romanization of the Arabic Alphabet for Arabic, Ottoman-Turkish, Persian, Kurdish, Urdu and Pushto* . 10  
*Introduction to Arabic Natural Language Processing* . . . . . 42  
`\itemindent` 477, 523, 546  
`\itemsep` . . . . . 512  
`\itshape` . . . . . 89  
**J**  
`\jobname` . . . . . 314  
**K**  
`\kern` . . . . . 260, 523  
KOMA-script (package) . . 45  
Krüger, Marcel . . . . . 32  
**L**  
`\labelsep` . . . . . 548, 554  
`\labelwidth` 547, 549, 552  
Lagally, Klaus . . . . . 3, 19  
Lane, Edward William . . . 18  
`\lastbox` . . . . . 521  
`\leaders` . . . . . 55, 59  
`\leavevmode` . . . . . 109, 124, 130, 136, 260, 265  
`\leftmargin` . . . . . 487, 489  
lineno (package) . . . . . 28  
`\linewidth` . 486, 487, 490  
`\list` . . . . . 469  
`\listparindent` . . 476, 485  
`\long` . . . . . 556  
`\LR` . . . . . 41, 44, 337, 339, 342, 400, 403, 462  
`\LRfootnote` . . . . . 45, 393  
`\LRmarginpar` . . . . . 45, 389  
lua-ul (package) . . . . . 32, 55  
`\luadirect` . . . . . 10, 12, 31, 45, 52, 53, 64, 66, 70, 72, 103, 126, 132, 138, 143, 152, 153, 156, 157, 163, 169, 175, 180, 207, 210, 211, 268, 375, 376, 381, 382, 385, 388, 405, 409, 410, 416, 417, 418, 433, 434, 435  
luaotfload (package) . . . . . 37  
`\luastring` . . . . . 269, 375, 376, 381, 382  
`\luastringN` . . . . . 45, 52, 53, 64, 66, 70, 72, 103, 126, 132, 138, 143, 152, 153, 156, 157, 163, 169, 175, 180, 207, 210, 211, 385, 388, 405, 409, 410, 417, 434  
`\luastring0` 127, 133, 139, 144, 145, 146, 164, 170, 176, 181, 182, 183  
**M**  
`\makebox` . . . . . 279, 282, 292, 299, 302, 307  
`\makelabel` . . 480, 541, 542  
`\marginpar` . . . . . 391, 392  
`\mbox` . . . . . 57  
metre (option) . . . . . 25  
`\MkArbBreak` . . . . . 11, 46, 50  
`\MkArbBreak*` . . . . . 46, 50  
mode (option) . . . . . 25  
**N**  
nameauth (package) . . . . . 39  
`\newarbookmark` . . . . . 34, 155  
`\NewDocumentEnvironment` . . . . . 113, 117, 158, 200, 412, 421  
`\newfontfamily` . . . . . 36  
`\newlength` . . 185, 186, 187  
`\newunderlinetype` . 55, 59  
`\NoArbUp` . . . . . 39, 95  
`\noindent` . . . . . 218  
novoc (option) . . . . . 5, 8  
**O**  
`\oarg@false` . . . . . 245, 249  
`\oarg@true` . . . . . 244, 247  
options:  
color . . . . . 26  
delim . . . . . 25  
export . . . . . 5, 26, 52  
fullvoc . . . . . 5, 8  
gutter . . . . . 25  
metre . . . . . 25  
mode . . . . . 25  
novoc . . . . . 5, 8  
trans . . . . . 5, 8  
utf . . . . . 26  
voc . . . . . 5, 8  
width . . . . . 25  
`\overline` . . . . . 57  
**P**  
`\PackageInfo` . . . . . 33  
`\par` . . . . . 114, 116, 118, 120, 159, 184, 414, 419, 420, 497, 500  
paralist (package) . . . 47, 49  
`\pardir` . . . . . 37, 39, 116, 119, 162, 168, 174, 179, 393, 395  
`\parindent` . . . . . 485  
`\parsep` . . . . . 484  
`\parshape` . . . . . 490  
`\parskip` . . . . . 484, 508  
`\penalty` . . . . . 526  
polyglossia (package) . . 4, 63  
`\presetkeys` . . . . . 199  
`\prname` . . . . . 38, 105  
`\prname*` . . . . . 39, 52, 105  
`\ProcessOptionsX` . . . . . 15  
`\providebool` . . . . . 30, 261  
`\ProvidesPackage` . . . . 467  
**Q**  
quoting (package) . . . . . 47  
quran (package) . . . . . 51  
**R**  
`\refstepcounter` . . . . . 538  
reledmac (package) . . . . . 51  
`\RequireLuaTeX` . . . . . 2  
`\RequirePackage` . . . . . 1, 3, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 401  
`\reset@font` . . . . . 42, 43  
`\rightmargin` 475, 486, 488  
`\RL` . . . . . 44, 47, 461, 462  
`\RLfootnote` . . . . . 45, 393  
`\rmfamily` . . . . . 48, 49, 87  
`\romannumeral` . . . . . 478  
**S**  
`\savenotes` . . . . . 274  
`\sbox` . . . . . 542  
scrextend (package) . . 45, 64  
`\SetArbDflt` . . . . . 6, 78  
`\SetArbDflt*` . . . . . 6, 78  
`\SetArbEasy` . . . . . 6, 78  
`\SetArbEasy*` . . . . . 6, 78  
`\SetArbNumbers` . . . . . 23, 44  
`\SetArbOutSuffix` . 52, 404  
`\SetArbUp` . . . . . 39, 95  
`\setbox` . 521, 543, 544, 556



<code>\SetDefaultIndex</code> . . . . .	311		
<code>\SetHemistichDelim</code> ..			
. . . . .	26, <u>241</u> , 259		
<code>\SetIndexMode</code> . . . . .	320		
<code>\SetInputScheme</code> <i>41</i> , <u>76</u> , 151			
<code>\setkeys</code> . . . . .	201,		
218, 232, 236, 333, 422			
<code>\setlength</code> 188, 189, 190, 191			
<code>\setLR</code> . . . . .	<u>39</u> ,		
45, 220, 223, 225, 456			
<code>\setRL</code> . . . . .	<u>37</u> , 45,		
220, 221, 223, 225, 456			
<code>\SetTranslitConvention</code>			
. . . . .	37, <u>92</u>		
<code>\SetTranslitFont</code> ..	<u>37</u> , <u>87</u>		
<code>\SetTranslitStyle</code> . . .			
. . . . .	37, <u>89</u> , 106		
<code>\space</code> . . . . .	251, 460		
<code>\spewnotes</code> . . . . .	309		
<code>\sRLbox</code> . . . . .	541, 556		
<code>\stepcounter</code> . . . . .	266		
<code>\StretchBayt</code> . . . . .			
. . . . .	27, <u>241</u> , 243, 251		
		<b>T</b>	
		<code>\textcolor</code> . . . . .	75
		<code>\textdir</code> 38, 40, 42, 43,	
		48, 49, 110, 112, 116,	
		119, 125, 131, 137,	
		142, 162, 168, 174,	
		179, 391, 392, 394, 396	
		<code>\textsuperscript</code> . . . . .	95
		<code>\textwidth</code> . . . . .	188
		<code>\the</code> . . . . .	478
		<i>The Ekdotis package</i> . . . . .	28
		<i>The Lua-ul package</i> . . . . .	32
		<code>\theekd@lab</code> . . . . .	267, 269
		<code>\thefootnotemark</code> 400, 403	
		<code>trans</code> (option) . . . . .	5, 8
		<i>Transliteration der arabis-</i>	
		<i>chen Schrift, Die</i>	
		. . . . .	10, 37, 39
		<code>txarab</code> (env.) . . . . .	44, <u>113</u>
		<code>txarab</code> (environment) . . . . .	51
		<code>txarabtr</code> (env.) . . . . .	<u>117</u>
		<code>\txarb</code> . . . . .	44, <u>108</u> , 257
		<code>\txtrans</code> . . . . .	107, <u>108</u>
			<b>U</b>
		<code>\Uc</code> . . . . .	<u>104</u>
		<code>\uc</code> . . . . .	38, <u>102</u> , 104, <u>107</u>
		<code>\unhbox</code> . . . . .	545, 552
		<code>\unskip</code> . . . . .	500
		<code>utf</code> (option) . . . . .	26
			<b>V</b>
		<code>voc</code> (option) . . . . .	5, 8
		<code>\vrule</code> . . . . .	55, 59
			<b>W</b>
		<code>\wd</code> . . . . .	549
		<code>width</code> (option) . . . . .	25
		Wright, W. LL.D . . . . .	5–8,
		12–17, 23, 24, 39, 49	
			<b>X</b>
		<code>xcolor</code> (package) . . . . .	33
		<code>\xleaders</code> . . . . .	260
			<b>Z</b>
		<code>\z@</code> . . . . .	475,
		476, 477, 521, 522, 526	
		<code>\zlabel</code> . . . . .	267
		<code>\zref@extract</code> . . . . .	269