

arabluatex

ArabTeX for LuaL^AT_EX

Robert Alessi

<mailto:alessi@robertalessi.net>

2026/02/04 – v1.21-dev

Contents

License and disclaimer	2	5.8.2	The <i>abjad</i>	23
1 Introduction	3	5.9	Additional characters	24
1.1 arabluatex is for LuaL ^A T _E X	4	5.10	Arabic emphasis	25
2 The basics of arabluatex	4	5.10.1	Underlining words or numbers	25
2.1 Activating arabluatex	4	6 Arabic poetry	25	
Font setup	5	Scaling and distortion of characters	28	
2.2 Options	5	Footnotes	28	
2.2.1 Classic contrasted with modern typesetting of Arabic	6	Critical Notes	28	
2.3 Typing Arabic	7	Line numbering	28	
2.3.1 Local options	8	6.1 Example	29	
3 Standard ArabTeX input	8	7 Special applications	30	
3.1 Consonants	8	Linguistics	30	
3.2 Additional characters	10	Brackets	30	
3.3 Vowels	10	Additional Arabic marks	31	
3.3.1 Long vowels	10	The ‘Zero width joiner’ character (U+200D)	32	
3.3.2 Short vowels	11	7.1 The Qur’ān	33	
4 Variants	12	Caveat	33	
Maghribi	12	8 Color	33	
5 arabluatex in action	12	8.1 Tricks of the trade	34	
5.1 The vowels and diphthongs	12	Diacritics	34	
Short vowels	12	<i>tanwīn</i>	36	
Long vowels	12	<i>waṣṣah</i>	36	
‘ <i>alif maqṣūrah</i> ’	12	and <i>maddah</i>	36	
‘ <i>alif otiosum</i> ’	12	<i>šaddah</i>	36	
‘ <i>alif maḥdūfah</i> ’ and defective <i>ū</i> , <i>ī</i>	12	The definite article and the euphonic <i>tašdīd</i>	37	
Silent <i>و</i> / <i>ي</i>	13	<i>hamzah</i>	37	
‘ <i>Amr^{un}</i> ’, and the silent <i>و</i>	13	9 Transliteration	37	
<i>tanwīn</i>	13	Convention	38	
5.2 Other orthographic signs	13	Style	38	
<i>tā’ marbūṭah</i>	13	Font	38	
<i>hamzah</i>	14	Proper names	38	
<i>maddah</i>	15	Hyphenation	39	
<i>šaddah</i>	15	‘Long’ proper names	39	
The definite article and the ‘ <i>alif^u</i> ’ <i>l-waṣṣlⁱ</i>	16	Proper names outside Arabic environments	39	
5.3 Special orthographies	18	9.1 Additional note on <i>dmg</i> convention	40	
The name of God	18	<i>īrāb</i> boundaries	40	
The conjunctive <i>الَّذِي</i>	19	Discarding the <i>īrāb</i>	41	
5.4 Quoting	19	Uncertain short vowels	41	
<i>novoc</i>	19	9.2 Examples	41	
<i>voc</i>	20	10 Buckwalter input scheme	42	
<i>fullvoc</i>	21	‘base’, ‘xml’ and ‘safe’ schemes	42	
5.4.1 Quoting the <i>hamzah</i>	21	Transliteration	43	
5.5 The ‘pipe’ character ()	21	11 Unicode Arabic input	44	
5.6 Putting back on broken contextual analysis rules	22			
5.7 Stretching characters: the <i>taṭwīl</i>	23			
5.8 Digits	23			
5.8.1 Numerical figures	23			

12	LaTeX Commands in Arabic environments	44			
	General principle	44			
12.1	New commands	46			
12.2	Environments	47			
12.2.1	Lists	48			
	رموز الـكـاـب 48 Caveat	50			
12.3	csquotes	50			
12.4	Two-argument special commands	51			
	textcolor 51 reledmac 51				
12.5	quran	51			
13	Exporting Unicode Arabic to an external file	52			
13.1	Commands and environments	52			
	export global option 52 Exporting				
	running paragraphs 52 Appending				
	words or commands to the exter-				
	nal file only 53 Exporting Arabic				
	poetry 53				
13.2	Nested Arabic environments	53			
13.3	Further processing of Unicode con-				
	verted files	53			
14	Future work	54			
15	Implementation	54			
16	References	69			
17	Change History	70			
18	Index	71			

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	31
4	Standard ArabTeX (short vowels)	11	8	ArabTeX diacritics for \arbcOLOR	35
			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.¹

arabluatex -- ArabTeX for LuaLaTeX

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

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above

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

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

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

- email: <mailto:alessi@roberalessi.net>
- website: <https://sr.ht/~ralessi/arabluatex/>
- development: <http://git.robertalessi.net/arabluatex>
- comments, feature requests, bug reports: <https://todo.sr.ht/~ralessi/arabluatex>

This release of arabluatex consists of the following source files:

- arabluatex.ins
- arabluatex.dtx
- arabluatex.lua
- arabluatex_voc.lua
- arabluatex_fullvoc.lua
- arabluatex_novoc.lua
- arabluatex_trans.lua
- arabluatex.el

1 Introduction

In comparison to Prof. Lagally’s outstanding ArabTeX,² ArabLuaTeX is at present nothing more than a modest piece of software. Hopefully—if I may say so—it will eventually provide all of its valuable qualities to the LuaL^ATeX users.

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

²See <http://ctan.org/pkg/arabtex>

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

characters (U+202B, U+202A) and RLO/LRO “bidi-override” characters (U+202E, U+202D).⁴ Nonetheless, it remains that inserting all the time these invisible characters in complex documents rapidly becomes confusing and cumbersome.

The great advantage of ArabTeX notation is that it is immune from all these difficulties, let alone its being clear and straightforward. One also must remember that computers are designed to process code. ArabTeX notation is a way of encoding Arabic language, just as TeX “mathematics mode” is a way of processing code to display mathematics. As such, not only does it allow greater control over typographical features, but it also can be processed in several different ways: so without going into details, depending on one’s wishes, ArabTeX input can be full vocalized Arabic (*scriptio plena*), vocalized Arabic or non-vocalized Arabic (*scriptio defectiva*); it further can be transliterated into whichever romanization standard the user may choose.

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

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

1.1 arabluatex is for LuaL^ATeX

It goes without saying that `arabluatex` requires LuaL^ATeX. TeX and L^ATeX have `arabtex`, and X_YL^ATeX has `arabxetex`. Both of them are much more advanced than `arabluatex`, as they can process a number of different languages,⁵ whereas `arabluatex` can process only Arabic for the time being. More languages will be included in future releases of `arabluatex`.

In comparison to `arabxetex`, `arabluatex` works in a very different way. The former relies on the **TECkit** engine which converts ArabTeX input on the fly into Unicode Arabic script, whereas the latter passes ArabTeX input on to a set of Lua functions. At first, L^ATeX 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 TeX the converted strings—which may in turn contain some other ArabTeX input yet to be processed—for further processing.

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

⁵To 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.

2 The basics of arabluatex

2.1 Activating arabluatex

arabluatex is loaded the usual way:

```
\usepackage{arabluatex}
```

The only requirement of arabluatex is Lua¹TeX; 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:

```
\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 font⁶ 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.⁷

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

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

Other Arabic fonts may behave differently.

REM. b \newfontfamily can be used to have either Indian 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: *dammah* (َ), *fatḥah* (َ) and *kasrah* (ِ). If a vowel is followed by N, viz. <uN, aN, iN>, then the corresponding *tanwīn* (ُ, ِ, ِ, ِ or ِ) is generated. Finally, <u, a, i> at the commencement of a word indicate a “connective *ʿalif*” (*ʿalifu ʿl-waṣli*), but voc mode does not show the *waṣlah* above the *ʿalif*; instead, the accompanying vowel may be expressed at the beginning of a sentence (أَ 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. 5.4 on page 19](#)).

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

⁷W. 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.

⁸See the documentation of amiri, *ibid.*, 6.

trans

This mode transliterates the ArabTeX input into one of the accepted standards. At present, three standards are supported (see below [sect. 9 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. 13 on page 52](#) for more information.

2.2.1 Classic contrasted with modern typesetting of Arabic

New feature v.1.2

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

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

samA'uN سَمَاءُ^{un}, jA'a جَاءَ^a, yatasA'alUna يَتَسَاءَلُونَ⁹
(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 ظَمِنًا, شَيْئًا and the like, the *hamzah* alone is not written over the letter *yā*’ with no diacritical points below as in ظَمِنًا, 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, either globally in the preamble or locally at any point of the document. The difference

New feature v1.4.4

\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` command. Assimilation rules laid on (b) on page 16 may also be applied by the ‘starred’

New feature v1.6

\SetArbDflt*

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

(a) `\SetArbDflt`:

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

⁹Note that in old mss. such forms as سَمَاءُ, 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.

¹⁰For an example, see [sect. 6.1 on page 29](#).

- (b) `\SetArbDflt*`:
- i. voc وَمَاتَ اسْتِسْقَاءٌ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
 - ii. fullvoc وَمَاتَ اسْتِسْقَاءٌ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
 - iii. trans *wa-māta 'stisqā^{an} qabla 'ay yutimma kitāba-hu fī nuḡūmⁱ 's-samāⁱ*
- (c) `\SetArbEasy`:
- i. voc وَمَاتَ اسْتِسْقَاءٌ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
 - ii. fullvoc وَمَاتَ اسْتِسْقَاءٌ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
 - iii. trans *wa-māta 'stisqā^{an} qabla 'an yutimma kitāba-hu fī nuḡūmⁱ 's-samāⁱ*
- (d) `\SetArbEasy*`:
- i. voc وَمَاتَ اسْتِسْقَاءٌ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
 - ii. fullvoc وَمَاتَ اسْتِسْقَاءٌ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
 - iii. trans *wa-māta 'stisqā^{an} qabla 'an yutimma kitāba-hu fī nuḡūmⁱ 's-samāⁱ*

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:^a— Arabic, like Hebrew and Syriac, is written and read from right to left. The letters of the alphabet (حُرُوفُ الْمَجَائِدِ, حُرُوفُ التَّهْجِي, حُرُوفُ الْمَجَائِدِ, or حُرُوفُ الْمُعْجَمِ) are twenty-eight in number and are all consonants, though three of them are also used as vowels (see § 3).

^aWright, see n. 7, i. 1 A.

The following example comes from Wright:¹¹—

```

1 \begin{enumerate}[label=\Roman*, start=16]
2 \item \arb{fawA`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}
```

¹¹Wright, see n. 7, i. 213 C.

XVI. *فَوَاعِلْ.

1. فَاعِلْ; 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:
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 جَحَا يَطْلُبُ مِنْهُ حِمَارَهُ لِيَرْكَبَهُ فِي سَفَرَةٍ قَصِيرَةٍ فَقَالَ لَهُ: "سَوْفَ أُعِيدُهُ إِلَيْكَ فِي الْمَسَاءِ وَأَدْفَعُ لَكَ أُجْرَةً." فَقَالَ جَحَا: "أَنَا آسَفٌ جِدًّا أَنِّي لَا أَسْتَطِيعُ أَنْ أُحَقِّقَ لَكَ رَغْبَتَكَ فَالْحِمَارُ لَيْسَ هُنَا الْيَوْمَ." وَقَبْلَ أَنْ يَتِمَّ جَحَا كَلَامَهُ بَدَأَ الْحِمَارُ يَنْهَقُ فِي إِصْطِلِهِ. فَقَالَ لَهُ صَدِيقُهُ: "إِنِّي أَسْمَعُ حِمَارَكَ يَا جَحَا يَنْهَقُ." فَقَالَ لَهُ جَحَا: "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 ¹²		ArabTeX notation	
	dmg+	loc	arabica	
أ ¹³	'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
ج	ǧ	j	ǧ	ˆg or j
ح	ḥ	ḥ	ḥ	.h
خ	ḫ	kh	ḫ	_h or x
د	d	d	d	d
ذ	<u>d</u>	dh	<u>d</u>	_d
ر	r	r	r	r
ز	z	z	z	z
س	s	s	s	s
ش	š	sh	š	ˆs
ص	ṣ	ṣ	ṣ	.s
ض	<u>ḍ</u>	<u>ḍ</u>	<u>ḍ</u>	.d
ط	<u>t</u>	<u>t</u>	<u>t</u>	.t
ظ	ẓ	ẓ	ẓ	.z
ع	ʿ	ʿ	ʿ	`
غ	ǧ	gh	ǧ	.g
ف	f	f	f	f
ق	q	q	q	q
ك	k	k	k	k
ل	l	l	l	l
م	m	m	m	m
ن	n	n	n	n
ه	h	h	h	h
و	w	w	w	w
ي	y	y	y	y ¹⁴
ة	ah	ah	a	T

Table 1: Standard ArabTeX (consonants)

REM. *a* Please note that in all cases of elision, the *ʿalifu ʿl-waṣli* is expressed only by the vowel that accompanies the omitted *hamzah*: *⟨u, a, i⟩* as in *wa-inhazama* ^{واِنْهَزَمَ} *wa-ʿnhazama*. For more details on the definite article and the *ʿalifu ʿl-waṣli* see [sect. 5.2 on page 16](#).

¹²See below [sect. 9 on page 37](#).

¹³See below, *Rem. a*. For *ʿalif* 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. 5.2 on page 14](#).

¹⁴For the letter *ي* with no diacritical points below, see *REM. b* below.

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* وإن *wa-.n* (where the dot symbolizes the absence of vowel), *wa-.Aan* وإن *wa-an*, *wa-.Ain* وإن *wa-in*.

REM. *b* The letter ي with two points below, الياء المثناة من تحتها, 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* قاي'ن *qay'un*, *saY'uN* ساي'ن *say'un*, *saY'aN* ساي'ان *say'an*.

The same result may be achieved by encoding this letter as *.y*, like so: *qa.y'uN* قاي'ن *qay'un*, *sa.y'uN* ساي'ن *say'un*, *sa.y'aN* ساي'ان *say'an*.

3.2 Additional characters

New feature v1.8.5 Table 2 gives the ArabTeX equivalents for some additional Persian characters.

Letter	Transliteration ¹⁵			ArabTeX notation
	dmg+	loc	arabica ¹⁶	
پ	<i>p</i>	<i>p</i>	<i>p</i>	<i>P</i>
چ	<i>č</i>	<i>ch</i>	<i>č</i>	<i>^c</i>
ژ	<i>ž</i>	<i>zh</i>	<i>ž</i>	<i>^z</i>
ف ¹⁷	<i>v</i>	<i>v</i>	<i>v</i>	<i>v</i>
گ	<i>g</i>	<i>g</i>	<i>g</i>	<i>g</i>
ن ¹⁸	<i>ñ</i>	<i>ñ</i>	<i>ñ</i>	<i>^n</i>

Table 2: Standard ArabTeX (additional characters)

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

3.3 Vowels

3.3.1 Long vowels

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

Letter	Transliteration ¹⁹			ArabTeX notation
	dmg+	loc	arabica	
ا	<i>ā</i>	<i>ā</i>	<i>ā</i>	<i>A</i>

Table 3: Standard ArabTeX (long vowels)

¹⁵See below [sect. 9 on page 37](#).

¹⁶The 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.

¹⁷This 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 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, 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.

¹⁸See [17](#).

¹⁹See below [sect. 9 on page 37](#).

Letter	Transliteration			ArabTeX notation
	dmg+	loc	arabica	
و	$\bar{u}$	$\bar{u}$	$\bar{u}$	U
ي	$\bar{i}$	$\bar{i}$	$\bar{i}$	I ²⁰
ى ²¹	$\bar{a}$	$\bar{a}$	$\bar{a}$	_A or Y
أ	$\bar{a}$	$\bar{a}$	$\bar{a}$	_a
أ	$\bar{u}$	$\bar{u}$	$\bar{u}$	_u
أ	$\bar{i}$	$\bar{i}$	$\bar{i}$	_i

Table 3: Standard ArabTeX (long vowels)

- REM. *a* The long vowels $\bar{a}$, $\bar{u}$, $\bar{i}$, otherwise called *hurūf^u ‘l-maddⁱ*, the letters of prolongation, involve the placing of the short vowels *a*, *u*, *i* before the letters و, ي respectively. *arabluatex* does that automatically in case any from *voc*, *fullvoc* or *trans* modes is selected e.g. قَالَ *qāla*, قِيلَ *qīla*, يَقُولُ *yaqūlu*.
- REM. *b* Defective writings, such as ا, *al-‘alif^u ‘l-maḥḍūfat^u*, or defective writings of $\bar{u}$ and $\bar{i}$ are encoded _a_u and _i respectively, e.g. _d_alika ذَلِكَ, al-mal_a'ikaT-u 'l-ra.hm_an-u الْمَلِكُ الرَّحْمَنُ, .hu_dayfaT-u bn-u 'l-yamAn_i حَذَفَةُ بْنُ الْيَمَانِ for *Hudayfat^u bn^u ‘l-Yamānī*, etc.
- REM. *c* The letter ي with two points below, أَلْيَاءُ الْمُتَأَنِّ مِنْ تَحْتِهَا, may also be written without diacritical points as ى. When it is used as a long vowel, it is encoded iY, where i recalls the *kasrah* placed below the preceding letter in vocalized Arabic, like so: liY لِي *lī*, yam^siY يَمْنَى *yamšī*.

3.3.2 Short vowels

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

Letter	Transliteration ²²			ArabTeX notation
	dmg+	loc	arabica	
ا	<i>a</i>	<i>a</i>	<i>a</i>	a
و	<i>u</i>	<i>u</i>	<i>u</i>	u
ي	<i>i</i>	<i>i</i>	<i>i</i>	i
ان	<i>an</i>	<i>an</i>	<i>an</i>	aN
un	<i>un</i>	<i>un</i>	<i>un</i>	uN
in	<i>in</i>	<i>in</i>	<i>in</i>	iN

Table 4: Standard ArabTeX (short vowels)

Whether Arabic texts be vocalized or not is essentially a matter of personal choice. So one may use *voc* mode and decide not to write vowels except at some particular places for disambiguation purposes, or use *novoc* mode, not write vowels—as *novoc* normally does not show them—except, again, where disambiguation is needed.²³

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

²¹= *al-‘alif^u ‘l-maḥḍūrat^u*.

²²See below [sect. 9 on page 37](#).

²³See below [sect. 5.4 on page 19](#).

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. 5.4 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 Variants

While primarily intended for writing documents in classical and modern Arabic, `arabluatex` will gradually provide support for other variants and languages written in Arabic script, as described n. [5 on page 4](#).

`\SetArab` The commands described in this section must be given either in the preamble or at any point of the document, provided they are inserted outside of Arabic commands and environments. They operate on all subsequent of these. To return to classical Arabic, `\SetArab` shall be used.

`\SetMaghribi` **Maghribi** `\SetMaghribi` activates a different writing convention. *fāʾ* is written with one dot below the letter, *qāf* with one dot above and *vāʾ* with three dots below, like so: فَاءٌ, قَافٌ, وَافٌ.

5 arabluatex in action

5.1 The vowels and diphthongs

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

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

bi-hi بِهٍ *bi-hi*, 'aqi.tuN أَقِطٌ *'aqiṭ^{un}*.

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

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

qAla قَالَ *qāla*, bI`a بَعِ *bī`a*, .tUruN طُورٌ *ṭūr^{un}*, .tInuN طِينٌ *ṭīn^{un}*, murU'aTuN مَرُوءَةٌ *murū'at^{un}*.

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

al-fat_A أَلْفَتَى *al-fatā*, al-maqh_A أَلْمَقْهَى *al-maqhā*, 'il_A إِلَى *ilā*.

'alif otiosum Said 'alif^u 'l-wiqāyatⁱ, “the guarding 'alif”, after و at the end of a word, both when preceded by *dammah* and by *fathah* is written ⟨U⟩ or ⟨aW, aWA⟩:

na.sarUA نَصَرُوا *naṣarū*, katabUA كَتَبُوا *katabū*, ya.gzUA يَغْزُوا *yağzū*, 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^u*, 'il_ahUN إِلَهٍ *'ilāh^{un}*.
 al-ra.hm_an-u الرَّحْمَنُ *ar-rahmān^u*, l_akin لَكِنْ *lākin*, h_ahunA هَهُنَا *hāhunā*,
 .hunayn-u bn-u 'is.h_aq-a إِسْحَقُ بْنُ إِسْحَاقَ *Hunayn^u bn^u 'Ishāq^a*, rabb_i رَبِّ *rabbī*, al-`A.s_i الْعَاصِ *al-Āṣī*.

Silent ي/و Some words ending with لَآ are usually written حَوَة or نَوَة instead of لَآ: see Wright.²⁴ arablutext preserves that particular writing; the same applies to words ending in يَآ for لَآ. Long vowels ⟨U, I⟩ shall receive no *sukūn* after a 'alif maḥdūfah and are discarded in *trans* mode:

.hay_aUTuN حَيَوة *hayāt^{un}*, .sal_aUTuN صَلَوة *ṣalāt^{un}*, mi^sk_aUTuN مَشْكُوة *miš-kāt^{un}*, tawr_aITuN تَوْرَية *tawrāt^{un}*.

And so also: al-rib_aIT-u الرِّبِية *ar-ribāt^u*.

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

`amruNU عَمَرُوا *'amr<sup>un</sup>*, `amraNU عَمَرُوا *'amr^{an}*, `amriNU عَمَرُوا *'amrⁱⁿ*.

When the *tanwīn* falls away (Wright, *A Grammar of the Arabic Language*, i. 249 B): `amr-uU bn-u mu.ḥammadiN عَمَرُوا بْنُ مُحَمَّدٍ *'Amr<sup>u</sup> bn<sup>u</sup> Muḥammad<sup>in</sup>*, mu.ḥammad-u bn-u `amr-iU bn-i_hAlidiN مُحَمَّدُ بْنُ عَمْرٍو بْنِ خَالِدٍ *Muḥammad^u bn^u 'Amrⁱ bnⁱ Ḥālidiⁱⁿ*.

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

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

mAluN مَالٌ *māl^{un}*, bAbaN أَبَا *bāb^{an}*, madInaTaN مَدِينَةٌ *madīnat^{an}*, bintiN بِنْتٍ *bintⁱⁿ* maqhaN_A مَقْهَى *maqḥaⁿ*, fataNY فَتًى *fataⁿ*.

arablutext is aware of special orthographies: ^say'uN شَيْءٌ *šay^{un}*, ^say'aN شَيْئًا *šay^{an}*, ^say'iN شَيْءٍ *šayⁱⁿ*.

²⁴Wright, see n. 7, i. 12 A.

²⁵Ibid., i. 12 C.

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

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

5.2 Other orthographic signs

tā' marbūṭah It is written ⟨T⟩:

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

hamzah It is written ⟨'⟩, its carrier being determined by contextual analysis. In case one wishes to bypass this mechanism, he can use the “quoting” feature that is described below in [sect. 5.4 on page 19](#).

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

hamzah followed by the long vowel و is encoded '_U: '_U1_A أُولى *'ulā*, '_U1U أولُو *'ulū*, '_U1A' ika أُولَئِكَ *'ulā'ika*.

hamzah followed by the long vowel ي is encoded '_I: '_ImAnuN إِيْمَانٌ *'īmān^{un}*²⁶.

Middle hamzah: xA.ti'-Ina خَاطِئِينَ *ḫāṭi'ina*, ru'UsuN رُءُوسٌ *ru'ūs^{un}*, xA.tI'aTuN خَاطِئَةً *ḫaṭi'at^{un}*, su'ila سُئِلَ *su'ila*, 'as'ilaTuN أَسْئَلَةُ *'as'ilat^{un}*, mas'alaTuN مَسْأَلَةٌ *mas'alat^{un}*, 'as'alu أَسْأَلَ *'as'alu*, yatasA'alUna يَتَسَاءَلُونَ *yatasā'alūna*, murU'aTuN مُرُوءَةٌ *murū'at^{un}*, taw'amUN تَوَّعَمَ *taw'am^{un}*, ta'xIruN تَأَخَّرَ *ta'ḫīr^{un}*, ta'axxara تَأَخَّرَ *ta'ahḫara*, ji'tu-ka جِئْتُكَ *gi'tu-ka*, qA'iluN قَائِلٌ *qā'il^{un}*, .hIna'i_diN هِنَا'يدٍ *hinā'idⁱⁿ*, hay'aTuN هَيَّأَ *hay'at^{un}*, hay'AtuN هَيَّاتَ *hay'āt^{un}*.

From Wright:²⁷— All consonants, whatsoever, not even *'alif hēmzatum* excepted, admit of being doubled and take *tašdīd*. Hence we speak and write ra' 'AsuN رَأْسٌ *ra'ās^{un}*, sa' 'AluN سَأَلَ *sa'al^{un}*, na' 'AjuN نَأَى *na'ağ^{un}*.

Final hamzah: xA.ta'uN خَطَأٌ *ḫaṭa^{un}*, xA.ta'aN خَطَأٌ *ḫaṭa^{an}*, xA.ta'iN خَطِئًا *ḫaṭa'in*, 'aqra'u أَقْرَأُ *'aqra'u*, taqra'Ina تَقْرَأِينَ *taqra'ina*, taqra'Una تَقْرَؤُونَ *taqra'una*, yaqra'na يَقْرَأْنَ *yaqra'na*, yaxba'Ani يَخْبَأَنَّ *yahba'āni*, xaba'A خَبَأَ *ḫaba'ā*, xubi'a خُبِيَ *ḫubi'a*, xubi'UA خُبِيَوا *ḫubi'ū*, jA'a جَاءَ *ğā'a*, ridA'uN رَدَأَ *ridā^{un}*, ridA'aN رَدَأَ *ridā'an*, jI'a جِيَ *ğī'a*, radI'iN رَدِيَ *radī'in*, sU'uN سُوءٌ *sū'ū*.

²⁶For another way of encoding the initial *hamzah* followed by a long vowel, see the *taḥfīf^u 'l-hamzatⁱ* on this page.

²⁷Wright, see n. 7, i. 14 B.

sū^{un}, *daw'uN* دَاوْ *daw^{un}*, *qay'iN* قَيَّ *qayⁱⁿ*, *˘sifA'I* شِفَايَ *šifāⁱ*, *man˘sa'I* مَنَسَايَ *manšāⁱ*, *nisaA'uN* نِسَاءُ *nisā^{un}*, *nisa'u-hu* نِسَاؤُهُ *nisā'u-hu*, *nisa'i-hi* نِسَائِهِ *nisā'i-hi*, *nisa'I* نِسَائِي *nisāⁱ*.

˘say'uN شَيَّ *šay^{un}*, *˘say'aN* شَيْئًا *šay^{an}*, *˘say'iN* شَيْءٍ *šayⁱⁿ*, *al-˘say'-u* أَلشَّيْءُ *aš-šay^u*, *'a˘syA'-u* أَشْيَاءُ *ašyā^u*, *'a˘syA'-a* أَشْيَاءَ *ašyā^a*, *.zim'aN* ظِمًّا *ẓim^{an}*, *radI'aN* رَدِيًّا *radī^{an}*.

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

Imperatives of verbs that have the *hamzah* as the first radical are other cases of *tahfif^u 'l-hamzatⁱ*: i'sir اِسِرْ *isir*, i'dan اِذَنْ *īdan*, u'mul اُوْمَلْ *ūmul*. arabluatex also provides ways of encoding those words when the initial 'alif comes into *waṣl*, so as to make the 'alif *waṣl* fall away when preceded by و or ف: wa-'sir اَسِرْ *wa-'sir*, fa-'dan فَاذَنْ *fa-'dan*, fa-'ti فَاتِ *fa-'ti*, wa-'tamirUA اَتَمِّرُوا *wa-'tamirū*; or be retained outside the imperative, as in fa-i'tazarat فَاتَتَزَرَّتْ *fa-'tazarat*, ba'da i'tilAfIn بَعْدَ اِتِّخْلَافٍ *ba'da 'tilāfⁱⁿ*.

The strange spelling of *mi'at^{un}*: *mi'aTuN* مَائَةٌ *mi'at^{un}*, *mi'atAni* مَائَتَانِ *mi'atāni*, *mi'atayni* مَائَتَيْنِ *mi'atayni*, *mi'Una* مِئُونُ *mi'ūna*, *mi'AtuN* مِئَاتُ *mi'ātun*, *mi'aN_A* مَائِي *mi'āⁿ*. Of course, the 'pipe' character can be used to prevent this rule from being applied (see [sect. 5.5 on page 21](#)): *mi'a|TuN* مِئَةٌ *mi'at^{un}*.

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

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

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

'is'AduN إِسَادُ *is'ād^{un}*, 'AkilUna أَكِلُونُ *ākilūna*, 'a'mannA آمَنَّا *āmannā*, al-qur'An-u الْقُرْآنُ *al-qur'ān^u*.

jA'a جَاءَ *ġā'a*, yatasA'alUna يَتَسَاءَلُونَ *yatasā'alūna*, ridA'uN رَدَاءُ *ridā^{un}*, xaba'A خَبَا *ħaba'a*, yaxba'Ani يَخْبَانِ *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:

ʿallaqa عَلَقَ *ʿallaqa*, mAdduN مَادَّ *mādd^{un}*, 'ammara أَمَّرَ *ammara*, murrN مَرَّ *murr^{un}*.

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

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

Unlike *arabtex* and *arabxetex*, *arabluatex* *never requires the solar letter to be written twice*, as it automatically generates the euphonic *tašdīd* above the letter that carries it, whether the article be written in the assimilated form or not, e.g. al-[~]sams-u الشَّمْسُ *aš-šams^u*, or a[~]s-[~]sams-u الشَّمْسُ *aš-šams^u*.

al-tamr-u أَلْتَمَرُ *at-tamr^u*, al-ra.hm_an-u الرَّحْمَنُ *ar-raḥmān^u*, al-.zulm-u الظُّلْمُ *aẓ-ẓulm^u*, al-lu.gaT-u اللَّغَةُ *al-luġat^u*.

- (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 رَبِّي *mir rabbi-hi*, min layliN مِنْ لَيْلٍ *mil laylⁱⁿ*, 'an yaqtula أَنْ يَقْتُلَ *ay yaqtula*.

With *tanwīn*: kitAbuN mubInuN كِتَابٌ مُبِينٌ *kitāb^{um} mubīn^{un}*.

REM. This particular feature must be put into operation by the `\SetArbDflt*` command explicitly. See above [sect. 2.2.1 on page 6](#) for further details. Other kinds of assimilations, including the various cases of *ʾidjām*, 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^u ʾl-waṣlⁱ At the beginning of a sentence, ʾ is never written, as اَلْحَمْدُ لِلّٰهِ; instead, to indicate that the 'alif is a connective 'alif ('alif^u ʾl-waṣlⁱ), the *hamzah* is omitted and only its accompanying vowel is expressed:

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

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

- (a) Definite article at the beginning of a sentence is encoded `al-`, or `a<solar letter>-` if one wishes to mark the assimilation—which is in no way required, as `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 *ʾalif* *ʾl-waṣli* is expressed by the vowel that accompanies the omitted *hamzah*: $\langle u, a, i \rangle$.

Article: bAb-u 'l-madrasaT-i *بَابُ الْمَدْرَسَةِ* *bāb^u 'l-madrasatⁱ*, al-maqAlaT-u 'l-'U_l_A *المَقَالَةُ الْأُولَى* *al-maqālat^u 'l-'ūlā*, al-lu.gaT-u 'l-'arabiyyaT-u *اللُّغَةُ الْعَرَبِيَّةُ* *al-luḡat^u 'l-'arabiyyat^u*, fI .sinA`aT-i 'l-.tibb-i *فِي صِنَاعَةِ الطِّبِّ* *fī ṣinā'atⁱ 't-tibbⁱ*, 'il_A 'l-intiqA.d-i *إِلَى الْإِنْتِقَاضِ* *'ila 'l-intiqādⁱ*, fI 'l-ibtidA'-i *فِي الْإِبْتِدَاءِ* *fī 'l-ibtidā'ⁱ*, 'abU 'l-wazIr-i *أَبُو الْوَزِيرِ* *'abu 'l-wazīrⁱ*, fa-lammA ra'aW 'l-najm-a *فَلَمَّا رَأَوْا النَّجْمَ* *fa-lammā ra'awu 'n-nağm^a*.

Particles:—

- (a) *li-*: *ʾalif^u 'l-waṣlⁱ* is omitted in the article *أَلْ* when it is preceded by the preposition *لِ*: *li-l-rajul-i* *لِلرَّجُلِ* *li-r-rağulⁱ*.
If the first letter of the noun be *لِ*, then the *لِ* of the article also falls away, but `arabluatex` is aware of that: *li-l-laylaT-i* *لِلَّيْلَةِ* *li-l-laylatⁱ*.
- (b) *la-*: the same applies to the affirmative particle *لِ*: *la-l-.haqq-u* *لِلْحَقِّ* *la-l-haqq^u*.
- (c) With the other particles, *ʾalif^u 'l-waṣlⁱ* is expressed: fI 'l-madInaT-i *فِي الْمَدِينَةِ* *fī 'l-madīnatⁱ*, wa-'l-rajul-u *وَالرَّجُلُ* *wa-'r-rağul^u*, bi-'l-qalam-i *بِالْقَلَمِ* *bi-'l-qalamⁱ*, bi-'l-ru`b-i *بِالرُّعْبِ* *bi-'r-ruḅⁱ*.

Perfect active, imperative, nomen actionis: qAla isma` *قَالَ أَسْمَعُ* *qāla 'sma<sup>ʿ</sup>*, qAla uqtul *قَالَ أَقْتُلْ* *qāla 'qtul*, huwa inhazama *هُوَ أَنْهَزَمَ* *huwa 'nhazama*, wa-ustu`mila *وَأَسْتَعْمَلْ* *wa-'stu'mila*, qadi in.sarafa *قَدْ أَنْصَرَفَ* *qadi 'nṣarafa*, al-iqtidAr-u *الْإِقْتِدَارُ* *al-iqtidār^u*, 'il_A 'l-intiqA.d-i *إِلَى الْإِنْتِقَاضِ* *'ila 'l-intiqādⁱ*, law istaqbala *لَوْ اسْتَقْبَلَ* *lawi 'staqbala*.

Other cases: 'awi ismu-hu *أَوْ اسْمُهُ* *'awi 'smu-hu*, zayduN ibn-u *زَيْدُ ابْنِ عَمْرٍو* *zayduN ibn-u 'amriNU*, 'umar-u ibn-u *عُمَرُ ابْنِ الْخَطَّابِ* *'umar-u ibn-u 'l-ha.t.tAb-i*, 'Umar^u *عُمَرُ* *'bn^u 'l-Ḥaṭṭābⁱ*,²⁹ imru'-u *إِمْرُؤُ الْقَيْسِ* *Imru^u 'l-Qaysⁱ*, la-aymun-u *لَا يَمُنُ* *la-'ymun^u 'l-lāhⁱ*.

²⁸ “*Zayd* is the son of *ʿAmr*”: the second noun is not in apposition to the first, but forms part of the predicate. Hence *عُمَرُ ابْنُ عَمْرٍو* and not *عُمَرُ بْنُ عَمْرٍو*, “*Zayd*, son of *ʿAmr*”.

²⁹ “*Umar* is the son of *al-Ḥaṭṭāb*” (see n. 28).

'alif^u 'l-waṣlⁱ 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 arabuatex takes it into account in some transliteration standards:—

fI 'l-nAs-i فِي النَّاسِ fi 'n-nāsⁱ, 'abU 'l-wazIr-i أَبُو الْوَزِيرِ 'abu 'l-wazīrⁱ, fI 'l-ibtidA'-i فِي الْإِبْتِدَاءِ fi 'l-ibtidā'ⁱ, _dU 'l-i`lAl-i ذُو الْأَعْلَالِ du 'l-i'lālⁱ, maqh_A 'l-'amIr-i مَقْهَى الْأَمِيرِ maqha 'l-'amīrⁱ.

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

fI `aynay 'l-malik-i فِي عَيْنِي الْمَلِكِ fi 'aynayi 'l-malikⁱ, ix^say 'l-qawm-a اخْتَبَيْ الْقَوْمَ ihšayi 'l-qawm^a, mu.s.tafaw 'l-l_ah-i مُصْطَفَوُ اللَّهِ muṣṭafawu 'l-lāhⁱ.
ramaW 'l-.hijAraT-a رَمَوْا الْحِجَارَةَ ramawu 'l-ḥiḡārat^a, fa-lammA ra'aW 'l-najm-a فَلَمَّا رَأَوْا النَّجْمَ fa-lammā ra'awu 'n-naḡm^a.

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

'antumU 'l-ka_dib-Una أَنْتُمُ الْكَادِبُونَ antumu 'l-kādib^{ūna}, ra'aytumU 'l-rajul-a رَأَيْتُمُ الرَّجُلَ ra'aytumu 'r-raḡul^a, mani 'l-ka_d_dAb-u مَنِ الْكَذَّابُ mani 'l-kaddāb^u, qatalati 'l-rUm-u قَتَلَتِ الرُّومُ qatalati 'r-Rūm^u.

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

mu.hammaduN 'l-nabI مُحَمَّدٌ النَّبِيُّ Muḥammad^{uni} 'n-nabī, salAmuN ud_hulUA سَلَامٌ أَدْخُلُوا salām^{umu} 'dhulū, qa.sIdata-hu fI qatl-i \uc{'a}bI \uc{m}uslimiN قَاتِلُ الْإِبْرَاهِيمِ الْقَاتِلِ أَبِي مُسْلِمٍ الَّتِي يَقُولُ فِيهَا 'llatI yaqUlu fI-hA قَاتِلَةُ أَبِي مُسْلِمٍ قَصِيدَتُهُ فِي قَتْلِ أَبِي مُسْلِمٍ qaṣīdata-hu fī qatlⁱ 'Abī Muslimⁱⁿⁱ 'llatī yaqūlu fī-hā.

5.3 Special orthographies

The name of God The name of God, اللَّهُ, is compounded of the article اَلْ, and إِلَٰهٌ (noted إِلَٰهٌ with the defective 'alif) so that it becomes اَلْإِلَٰهٌ; then the *hamzah* is suppressed, its vowel being transferred to the ل before it, so that there remains اَللَّهُ (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 arabuatex 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*^u, yA|³⁰ al-l_ah-u يَا *yā al-lāh*^u, 'a-fa|³¹-al-l_ah-i
 la-ta.g`alanna أَفَاللهُ لَتَعْلَنَ *'a-fa-al-lāhⁱ la-taḡ'alanna*, bi-'l-l_ah-i بِاللهِ *bi-'l-*
lāhⁱ, wa-'l-l_ah-i وَاللهِ *wa-'l-lāhⁱ*, bi-sm-i 'l-l_ah-i بِسْمِ اللهِ *bi-smⁱ 'l-lāhⁱ*,
 al-.hamd-u li-l-l_ah-i اَلْحَمْدُ لِلّٰهِ *al-ḥamd^u li-l-lāhⁱ*, li-l-l_ah-i 'l-qA'il-
 u لِلّٰهِ الْقَائِلُ *li-l-lāhⁱ 'l-qā'il^u*.

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

'a_hAfu mina 'l-malik-i 'lla_dI ya.zlimu 'l-nAs-a الَّذِي اَخَفُ مِنَ الْمَلِكِ الَّذِي يَظْلِمُ النَّاسَ
'ahāfu mina 'l-malikⁱ 'lladī yaẓlimu 'n-nās^a, `udtu 'l-`say_h-a
 'lla_dI huwa marI.duN عُدْتُ الشَّيْخَ الَّذِي هُوَ مَرِيضٌ *'udtu 'š-šayḥ^a 'lladī huwa*
marīḍ^{un}, mA 'anA bi-'lla_dI qA'iluN la-ka `say'aN مَا أَنَا بِالَّذِي قَائِلٌ لَكَ شَيْئًا
mā 'anā bi-'lladī qā'il^{un} la-ka šay^{an}.
 'ari-nA 'lla_dayni 'a.dallA-nA mina 'l-jinn-i wa-'l-'ins-i أَرِنَا الَّذِينَ
'ari-na 'lladayni 'aḍallā-nā mina 'l-ḡinnⁱ wa-'l-'insⁱ.

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 جَهْظَةَ
fa-'innā nadkuru 'š-šawt^{ayni} 'l-ladayni
rawaynā-humā 'an Ḡahẓat^a.

And also: al-la_dAni اللّٰذَانِ *al-ladāni*, al-la_dayni اللّٰذَيْنِ *al-ladayni*, al-
 latAni اللّٰتَانِ *al-latāni*, al-latayni اللّٰتَيْنِ *al-latayni*, al-lAtI اللَّاتِي *al-lātī*, al-
 lA' |Ati³² اللَّاءَاتِ *al-lā'ātī*, al-lA' I اللَّائِي *al-lā'ī*, and so forth.

5.4 Quoting

It is here referred to “quoting” after the arabt看 package.³³ The “quoting” mechanism of arabluatex is designed to be very similar in effect to the one of arabt看.

³⁰Note the “pipe” character ‘|’ here after yA and below after fa before footnote mark 31: it is needed by the dmḡ 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. ‘alq ‘t-ṭarīqⁱ. See sect. 5.5 on page 21 on the “pipe” and sect. 9 on page 37 on dmḡ mode.

³¹See 30.

³²Note 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: ٱ. The “pipe” character prevents this rule from being applied. See sect. 5.5 on page 21.

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

To start with an example, suppose one types the following in **novoc** mode: 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 *ḍammah* above the first ع: علم الهيئة, which is achieved by “quoting” the vowel u, like so: `ullima, or, with no other vowel than the required u: `ullm.

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

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

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

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

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

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

wa-'innA sawfa tudriku-nA 'l-manAyA muqadd"araT"aN وإنا سوف تدركنا
المنايا مقدرةً wa-'innā sawfa tudriku-na 'l-manāyā muqaddarat^{an}.

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

qAla isma`" قال اسمع qāla 'sma', jA'at" hinduN جاءت هند qā'at Hind^{un},
`sabIhuN bi-man q"u.ti`at" qadamA-hu قُطعت قدماه qadama-hu شبيه بمن قطعت
man quṭi'at qadamā-hu.

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

wa-"ust"u`mila وأُستعمل wa-'stu'mila, huwa "inhazama هو أنهزم huwa
'nhazama, al-"intiqA.d-u الانتقاض al-intiqāḍ^u.

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

- (a) Short and long vowels:—

q"Ala q"A'iluN قَالَ قَائِلٌ qāla qā'il^{un}, ibn-u 'abI 'u.saybi`aT-"a إِنْ
أَبْنَى أُصْبِغَةَ Ibn^u 'Abī 'Uṣaybi'at^a.

- (b) *tanwīn*:—

madInaT"aN مَدِينَةٌ madīnat^{an}, bAb"aN أَبَا bāb^{an}, hud"aN_A هُدَى hudaⁿ,
`say'"iN شَيْءٌ šayⁱⁿ.

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

(a) *'alif^u 'l-waṣlⁱ*:—

fI "istiq.sA'-iN *وَاسْتَقَصَّاءٌ فِي* *fi 'stiqṣā'ⁱⁿ*, wa-"istiq.sA'-uN *وَاسْتَقَصَّاءٌ* *wa-'stiqṣā'^{um}*, qAla "uhrub fa-lan tuqtala *قَالَ أَهْرُبُ 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 *مَا جَاءَتْ مِنِ امْرَأَةٍ* *mā gā'at mini 'mra'atⁱⁿ*, kam" qad" ma.dat" min" laylaTiN *كَأَنَّ مَضَتْ مِنْ لَيْلَةٍ* *kam qad maḍat min laylatⁱⁿ*.

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*
al-ḡamr^u 'ṣ-ṣayfiyy^u 'llaḍī kāna bi-Qrānnūn^a.

5.4.1 Quoting the *hamzah*

As said above in [sect. 5.2 on page 14](#), the *hamzah* is always written ⟨ ' ⟩, its carrier being determined by contextual analysis. “Quoting” that straight single quote character like so: ⟨ " ' ⟩ allows to determine the carrier of the *hamzah* freely, without any consideration for the context. [Table 5](#) gives the equivalents for all the possible carriers the *hamzah* may take.

Letter	Transliteration ³⁴			ArabT _E X notation
	dmg+	loc	arabica	
ء	’	’	’	" ' "
آ	’ā	’ā	’ā	A " ' "
أ	’	’	’	a " ' "
إ	’	’	’	u " ' "
ؤ	’	’	’	w " ' "
إِ	’	’	’	i " ' "
يِ	’	’	’	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 ArabT_EX equivalent for the *hamzah* itself, namely ⟨ ' ⟩. For example, *أُ* is encoded ⟨ w " ' a ⟩, while *ؤ* is encoded ⟨ w " ' " ⟩. In the latter example, the second straight double quote encodes the *sukūn* in voc mode in accordance with the rule laid above on the previous page.

'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*.

³⁴See below [sect. 9 on page 37](#).

5.5 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. 5.4.1](#), its usage to encode the *hamzah* alone, with no carrier: |'' ء.

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

bi-qrAn|nUn-a بِقْرَانُونٌ *bi-Qrānnūn*^a, “in Crannon” (Thessaly, Greece).³⁵

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

5.6 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^AT_EX commands. To take an example, consider the following:—

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

This is wrong:

فَلَمَّا رَأَوْا النَّجْمَ...

^a A footnote which interferes with the contextual analysis.

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

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

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

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

```
1 \begin{arab}[fullvoc]
2   fa-lammA ra'aw\arbnul{'l-na^gma}\LRfootnote{A footnote
3     which interferes with the contextual analysis.}
4   'l-na^gma...
5
6   qAla\LRfootnote{A footnote which interferes with the
```

³⁵See more context on the current page.

```

7 contextual analysis.} \arbnul{qAla}uhrub fa-lan tuqtala.
8
9 \uc{z}ayduN\arbnul{ibnu}\LRfootnote{A footnote which
10 interferes with the contextual analysis.}
11 \arbnul{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\arbnul{ibnu}\LRfootnote{A footnote which
17 interferes with the contextual analysis.}
18 \arbnul{zayduN}ibn-u \uc{`a}mriNU.\LRfootnote{See
19 \vref{fn:zayd-is-son}.}
20 \end{arab}

```

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

Zayd^{unie} 'bn u 'Amrⁱⁿ.^f

^a A footnote which interferes with the contextual analysis.

^b A footnote which interferes with the contextual analysis.

^c A footnote which interferes with the contextual analysis.

^d See 28 on page 17.

^e A footnote which interferes with the contextual analysis.

^f See 28 on page 17.

5.7 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^u bn^u 'Ishāq^a*

5.8 Digits

5.8.1 Numerical figures

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

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

\SetArbNumbers \SetArbNumbers{Indian|Arabic}
New feature v1.21

Default: Indian

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.

5.8.2 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 fī 'l-`adāti.
```

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. 5.10.1 on the following page](#) or insert the *abjad* number in `\arb[novoc]{}`:—

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

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

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

The *abjad* number for the current page (24) is ك.

This technique can be used to produce *abjad*-numbered lists as will be demonstrated on [page 49](#).

5.9 Additional characters

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

For the time being, the following is included:—

³⁶Wright, see n. 7, i. 4 B–C.

Letter	Transliteration ³⁷			ArabT _E X notation
	dmg+	loc	arabica	
ب	<i>b</i>	<i>b</i>	<i>b</i>	.b
د	<i>d</i>	<i>d</i>	<i>d</i>	ˆd
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
ن	<i>n</i>	<i>n</i>	<i>n</i>	.n
ه	<i>ah</i>	<i>ah</i>	<i>a</i>	H ³⁸
ي	<i>y</i>	<i>y</i>	<i>y</i>	.y ³⁹
﴿	(	(	(	((
﴾	)	)	)	))

Table 6: Additional Arabic codings

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

5.10 Arabic emphasis

As already seen in [sect. 5.8.2 on page 23](#), the *ʾabjad* numbers are distinguished from the surrounding words by a stroke placed over them. This technique is used to distinguish further words that are proper names or book titles.

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

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

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

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

New feature v1.9.2
`\aemph*`

5.10.1 Underlining words or numbers

`\aoline` Three additional, non context-sensitive commands are provided to distinguish words or
`\aoline*` 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 *ʾabjad* numbers.⁴⁰
(c) `\auline`, which can be used to underline Arabic words.

³⁷See below [sect. 9 on page 37](#).

³⁸This letter can be used to encode the *tā' marbūtah* devoid of diacritical points as it is found in some manuscripts, with the same *tanwīn* and the same short vowels as the standard *tā' marbūtah* with two points above, e. g. al-madInaHa, madInaHaN, مَدِينَةُ، الْمَدِينَةُ.

³⁹See above [REM. b on page 9](#).

⁴⁰See the example provided above [sect. 5.8.2 on page 23](#).

6 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
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. 11 on page 44](#) 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. 13 on page 52](#) for more details.

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

⁴¹A ‘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.

`\bayt{<ṣadr>}[<tadwīr>]{<ağuz>}\`

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

- (a) If the two hemistichs be connected with one another by a prominent horizontal flexible stroke, the *taṭwīl* should be used, like so: `[--]` (see [sect. 5.7 on page 23](#)). Of course, the ending word of the *ṣadr* and the word at the commencement of the *ağuz* must have the *taṭwīl* too so that the proper shapes of the letters be selected. Consider for example the following:—

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

لَا أَرَى مَنْ عَهِدْتُ فِيهَا فَأُبَيِّ السَّيِّمَ دَهَّاءَ وَمَا يَرُدُّ الْبُكَاءُ

As one can see, *triple hyphens* have been used. In the *ṣadr*, the first hyphen triggers the rules that are related to the definite article and the *'alif^u* *l-waṣlⁱ*,⁴² while the following two select the figure of the letter *lām* connected with a following letter. In the *ağuz*, the last two hyphens select the letter *yā* connected with a preceding letter, while the first one is simply discarded in this mode, but still may appear as it should, if the `trans` mode be selected:—

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

lā 'arā man 'ahidtu fī-hā fa-'abki 'l- -yawma dalh^{an} 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}
```

غَيْرَ أَنِّي قَدْ اسْتَعِينُ عَلَى الْمَمِّ إِذَا خَفَّ بِالتَّوِيِّ النَّجَاءُ
بِزَفُوفٍ كَانَتْهَا هِقْلَةٌ أَمْ رِثَالٍ دَوِيَّةٌ سَقَفَاءُ

⁴² See [sect. 5.2 on page 16](#).

In the first line, the word أَلْهَم should be split into أَلْهَم as the first part of it belongs to the *ṣadr* and the second to the *ʿağuz*. One solution to avoid splitting this word in such a way is to write inside the *tadwīr* the part of it that belongs to either hemistich, without omitting to add a space after it. In the second line, the word أُمُّ should be split into أُمُّ , so that the only way to avoid splitting it into two parts is to write it all inside the *tadwīr*. In that case, as the word is to be placed in the middle, it has been surrounded by spaces.

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

`\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. 5.7 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 environment⁴³) in the argument of the `\bayt` command and `\footnotetext` outside the `\bayt` command.

If the footnotes are abundant in number, it is advised to load the `footnotehyper` package which `arabluatex` will then use to typeset any kind of footnote that is called from the arguments of the `\bayt` command.⁴⁴

`\bayt+` **Critical Notes** If the `ekdosis` package be loaded,⁴⁵ 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.⁴⁶

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

⁴³See [sect. 12.1 on page 46](#).

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

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

⁴⁶*Ibid.*, see “Arabic Poetry”.

6.1 Example

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

```

1 \begin{arab}[fullvoc]
2 qāla imru'u 'l-\uc{q}aysi fī 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 fī `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 tahlik '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):—

قَالَ أَمْرُؤُ الْقَيْسِ فِي مُعَلَّقَتِهِ:

1 2 3 4 5 6	بِسْقَطِ الْوَلَى بَيْنَ الدَّخُولِ لِحَوْمِلِ لَمَّا نَسَجَتْهَا مِنْ جُنُوبٍ وَشَمَلِ وَقِيَعَانَهَا كَأَنَّهُ حَبٌّ فَلَقِلِ لَدَى سَمَرَاتِ الْحَيِّ نَاقِفٌ حَنْظَلِ يَقُولُونَ لَا تَهْلِكُ أَسَى وَتَجَلِ فَهَلْ عِنْدَ رَسْمٍ دَارِسٍ مِنْ مُعَوِّلِ	فَقَدْ نَبَكُ مِنْ ذِكْرِي حَبِيبٍ وَمَنْزِلِ فَتَوَضَّحَ فَاَلْمَقْرَأَةَ لَمْ يَعَفْ رَسْمُهَا تَرَى بَعَرَ الْأَرَامِ فِي عَرَصَاتِهَا كَأَنِّي غَدَاةَ الْبَيْنِ يَوْمَ تَحْمَلُوا وَقُوفًا بِهَا صَحِيٍّ عَلَيَّ مَطِيعِهِمْ وَإِنَّ شِفَائِي عَبْرَةَ مَهْرَاقَةٍ
---	---	--

(الضَرْبُ الثَّانِي مِنَ الْعُرُوضِ الْأُولَى مِنَ الطَّوِيلِ)

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

1 2 3 4 5 6	<i>qifā nabki min dīkrā ḥabīb^{iw} wa-manzilī</i> <i>fa-Tūḍiḥa fa-'l-Migrāti lam ya`fu rasmu-hā</i> <i>tarā ba`ara 'l-'ar`āmi fī `arašāti-hā</i> <i>ka-'annī gadāta 'l-bayni yawma taḥammalū</i> <i>wuqūf^{am} bi-hā ṣaḥbī `alayya maṭiyya-hum</i> <i>wa-'inna šifā`i `abrat<sup>um</sup> muharāqat<sup>un</sup></i>	<i>bi-siqṭi 'l-liwā bayna 'd-Daḥūli fa-Hawmalī</i> <i>limā nasaḡat-hā min ḡanūb<sup>iw</sup> wa-šam`alī</i> <i>wa-qī`āni-hā ka-'anna-hu ḥabbu fulfulī</i> <i>ladā samurāti 'l-ḥayyi nāqifu ḥanẓalī</i> <i>yaqūlūna lā tahlik `aṣa^w wa-taḡammalī</i> <i>fa-hal `inda rasm<sup>im</sup> dāris<sup>im</sup> mim mu`awwalī</i>
---	---	--

(aḍ-ḍarbu 't-tānī mina 'l-'arūḍi 'l-'ulā mina 't-tawīli)

⁴⁷Please note that for the time being only the assimilation rules that are laid on (b) on page 16 are applied. See sect. 2.2.1 on page 6 for more information. None of the editions of the *Mu'allaqat* that I know of feature the *ʿidgām* in the Arabic text, although it is often strongly marked in declamation.

\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 مَعْلَقَتِهِ:

1	بِسْقَطِ اللَّوَى بَيْنَ الدَّخُولِ حَوْمَلِ	قَفَا نَبَكٍ مِنْ ذِكْرَى حَبِيبٍ وَمَنْزِلِ
2	لَمَّا نَسَجَتْهَا مِنْ جَنْوِبٍ وَشَمَلِ	فَتَوَضَّحَ فَالْمَقْرَأَةَ لَمْ يَعْفُ رَسْمَهَا
3	وَقَبِيعَانَهَا كَأَنَّهُ حَبٌّ فَلَنْفَلِ	تَرَى بَعَرَ الْأَرَامِ فِي عَرَصَاتِهَا
4	لَدَى سَمَرَاتِ الْحَيِّ نَاقِفٌ حَنْظَلِ	كَأَنِّي غَدَاةَ الْبَيْنِ يَوْمَ تَحْمَلُوا
5	يَقُولُونَ لَا تَهْلِكُ أَسَى وَتَجْمَلِ	وَقُوفًا بِهَا صَحِيٍّ عَلَيَّ مَطِيهِمْ
6	فَهَلْ عِنْدَ رَسْمٍ دَارِسٍ مِنْ مُعَوَّلِ	وَأَنَّ شِفَائِي عَبْرَةُ مُهْرَاقَةٍ

(الضَرْبُ الثَّانِي مِنَ الْعَرُوضِ الْأُولَى مِنَ الطَّوِيلِ)

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

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

1	qifā nabki min dīkrā ḥabīb ^{iw} wa-manzilī	bi-siqṭi 'l-liwā bayna 'd-Daḥūlī fa-Ḥawmalī
2	fa-Tūḍīḥa fa-'l-Migrātī lam ya'fu rasmu-hā	limā nasaḡal-hā min ḡanūb ^{iw} wa-ṣam'ali
3	tarā ba'ara 'l-ar'āmi fī 'araṣāti-hā	wa-qī'āni-hā ka-'anna-hu ḥabbu fulfuli
4	ka-'annī ḡadāta 'l-bayni yawma taḡammalū	ladā samurātī 'l-ḡayyi nāḡifu ḡanzali
5	wuḡūf ^{un} bi-hā ṣaḡbī 'alayya maḡiyya-hum	yaḡūlūna lā tahlīk 'aṣa ^w wa-taḡammali
6	wa-'inna ṣifā'ī 'abrat ^{um} muḡarāḡat ^{un}	fa-hal 'inda rasm ⁱⁿ dāris ^{im} mim mu'awwali

(ad-ḡarbu 'l-ṭānī minā 'l-arūḍi 'l-'ulā minā 'l-ṭawīli)

7 Special applications

Linguistics The same horizontal stroke as the *taṭwīl* (see [sect. 5.7 on page 29](#)) 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 $\overset{\text{u}}{\text{---}} \overset{\text{a}}{\text{---}} \overset{\text{i}}{\text{---}} \overset{\text{un}}{\text{---}} \overset{\text{an}}{\text{---}} \overset{\text{in}}{\text{---}}$, BBu BBa BBi $\overset{\text{u}}{\text{---}} \overset{\text{a}}{\text{---}} \overset{\text{i}}{\text{---}}$, B--aN $\overset{\text{L}}{\text{---}} \overset{\text{an}}{\text{---}}$,
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: $\langle \rangle$
- square brackets: $\llbracket \rrbracket$
- angle brackets: $\langle \rangle$
- braces: $\{ \}$

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

```
1 \begin{arab}
2 \abracas{wa-qAla} 'inna 'abI kAna mina 'l-muqAtilaTi
```

```

3 wa-kAna--<--t> 'ummI min `u.zamA'i buyUti 'l-zamAzimaTi.
4 \end{arab}

```

{وَقَالَ إِنَّ أَبِي كَانَ مِنَ الْمُقَاتِلَةِ وَكَادَتْ أَبِي مِنْ عُظْمَاءِ بُيُوتِ الزَّمَانَةِ.

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

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

Codepoint	Shorthand	Glyph	Transliteration
FDFD	bismillah	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ	bi-ʾsmi ʾLlāhi ʾr-rahmāni ʾr-rahīmi
FDF5	salam	صَلِّع	ṣalla ʾLlāhu ʾalay-hi wa- sallama
FDF6	slm	سَلِّع	ṣalla ʾLlāhu ʾalay-hi wa- sallama
FDFB	jalla	جَلَّع	ḡalla ḡalāla-hu

Table 7: Additional Arabic marks

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

New feature v1.11 `\newarbmark` `\newarbmark` is also provided should one wish to define new marks in addition to the marks defined above. This command takes three arguments, like so:—

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

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

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

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

It must be noted that `\newarbmark` expects ArabT_EX 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 \newarbmark{sly}{\arbcolor[red]{~~~~06d6}}{}
3 \newarbmark{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 لِتَذْهَبُوا بِبَعْضِ مَا آتَيْتُمُوهُنَّ إِلَّا أَنْ يَأْتِيَنَّ بِفَاحِشَةٍ مُبِينَةٍ وَعَاشِرُوهُنَّ بِالْمَعْرُوفِ فَإِنْ كَرِهْتُمُوهُنَّ فَعَسَى 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. ح and س, from standing one above the other in the name of ‘Ishāq (إِسْحَاقُ)⁴⁸—

```

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}

```

هُوَ أَبُو زَيْدٍ حَنِينُ بْنُ إِسْحَاقَ الْعِيَادِيِّ يَفْتَحُ الْعَيْنَ وَيُخَفِّفُ الْبَاءَ.

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

هُوَ أَبُو زَيْدٍ حَنِينُ بْنُ إِسْحَاقَ الْعِمَادِيُّ يَفْتَحُ الْعَيْنَ وَتَخْفِيفُ الْبَاءِ.

7.1 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 arabuatex to come in the medium-term. New functions and new Arabic modes will be available as arabuatex will mature.

New feature v1.15 `\ayah` 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 \newarbmark{alifsp}{~~~~0627}{\arb[trans]{'alif} }
3 \newarbmark{lams}{~~~~0644~~~~0653}{\arb[trans]{lām} }
4 \newarbmark{mim}{~~~~0645~~~~0653}{\arb[trans]{mīm} }
5 \begin{arab}[fullvoc]
6   min ((sUraTi \uc{'l-b}aqaraTi)):
7 \end{arab}
8 \begin{arab}[fullvoc]
9   \arbmark{alifsp}\arbmark{lams}\arbmark{mim}~\ayah{1}
10  _d_alika 'l-kit_abu lā rayba fī-hi hudaN_A
11  li-l-muttaqīna~\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}
```

مِنْ سُورَةِ الْبَقَرَةِ:
 اَلَمْ يَكُنْ ذٰلِكَ الْكِتٰبُ لَا رَيْبَ فِيْهِ هٰدًى لِّلْمُتَّقِيْنَ ﴿٢٠﴾ الَّذِيْنَ يُؤْمِنُوْنَ بِالْغَيْبِ وَيُقِيْمُوْنَ الصَّلٰوةَ وَمِمَّا رَزَقْنٰهُمْ يُنْفِقُوْنَ ﴿٢١﴾

min (sūraTi 'l-Baqarati):

'alif lām mīm (1) dālīka 'l-kitābu lā rayba fī-hi huda^l li-l-muttaqīna (2) 'lladīna yu'minūna bi-'l-gaybi wa-yuqīmūna 'ṣ-ṣalāta wa-mimmā razaqnā-hum yunfiqūna (3)

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

8 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:⁴⁹—

⁴⁹Regarding 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.

- (a) The “pipe” character (|, [sect. 5.5 on page 21](#));
- (b) ‘Quoting’ technique ([sect. 5.4 on page 19](#)), and more specifically ‘quoting the *hamzah*’ (on page 21);
- (c) Putting back on broken contextual analysis rules ([sect. 5.6 on page 22](#));
- (d) Arabic marks ([sect. 7 on page 31](#)).

`\arbcolor` `\arbcolor` takes the text to be colored into $\langle color \rangle$ as an argument:—

`\arbcolor[\langle color \rangle]{\langle Arabic text \rangle}`

```

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 kulliyya 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 kulliyya fI 'a_hbAri \uc{~gAlInUsa} wa-mA kAna
13  `alay-hi...
14 \end{arab}

```

الْبَابُ الْخَامِسُ فِي طَبَقَاتِ الْأَطِبَّاءِ الَّذِينَ كَانُوا مِنْذُ زَمَانِ جَالِينُوسَ وَقَرِيبًا مِنْهُ. جَالِينُوسُ: وَلَنْضَعُ أَوَّلًا كَلَامًا
كُلِّيًّا فِي أَخْبَارِ جَالِينُوسَ وَمَا كَانَ عَلَيْهِ...

al-bābu 'l-hāmisu fī tabaqāti 'l-ʾaṭibbāʾi ʾlladīna kānū mundu zamāni Ġālīnūsa wa-qarīb^{an} min-hu. Ġālīnūsu: wa-l-naḍaʿ ʾawwal^{an} kalām^{an} kulliyy^{an} fī ʾahbāri Ġālīnūsa 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. `arabluatex` takes care of selecting the appropriate shape of the letters while coloring them:—

‘voc’ mode:

`i^stara\arbcolor[brown]{y}tu-hu bi-_tama\arbcolor[red]{niN}`
 'a`\arbcolor[blue]{~ga}ba-ka أَعْجَبَكُ بَيْتَنِي إِشْتَرَيْتُهُ *ištara^ytu-hu bi-ṭamaⁿin*
 ʾa^ḡaba-ka.

‘fullvoc’ mode:

`i^stara\arbcolor[brown]{y}tu-hu bi-_tama\arbcolor[red]{niN}`
 'a`\arbcolor[blue]{~ga}ba-ka أَعْجَبَكُ بَيْتَنِي إِشْتَرَيْتُهُ *ištara^ytu-hu bi-ṭamaⁿin*
 ʾa^ḡaba-ka.

8.1 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.

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

Table 8: ArabTeX diacritics for \arbcolor

‘voc’ mode:

i-ṣtaraytu-hu bi-tama\arbcolor[red]{n}\arbcolor[blue]{iN}
'a'\arbcolor[red]{g}\arbcolor[blue]{.a}ba-ka أَعْجَبَكَ بَيْنَ أَشْرَيْتَهُ
iṣtaraytu-hu bi-tamanⁱⁿ 'a'ḡaba-ka.

‘fullvoc’ mode:

iːstara\arbcolor[red]{y}otu-hu bi-tama\arbcolor[red]{n}iN
'a'\arbcolor[red]{g}.aba-ka اَسْتَرَيْتُهُ بِئِنَّ اَعْجَبَكْ *ištara^ytu-hu bi-tamanⁿ*
'a^ʕ*gaba-ka.*

iṣṭara\arbcolor[red]{y"}\arbcolor[blue]{o}tu-hu
bi_ṭama\arbcolor[red]{n"}\arbcolor[blue]{iN} 'a`\arbcolor[red]{
{`g"}\arb color[blue]{.a}ba-ka أَجَبَكَ إِشْرَيتُهُ iṣṭara^ytu-hu bi-ṭama^{nⁱⁿ}
'a^qaba-ka.

⁵⁰See below [sect. 9](#) on page 37.

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

```
mu`allim\arbnul{m}aN} مَعْلَمٌ mu'alliman,
istisqA'\arbnul{A'}aN} اِسْتِسْقَاءٌ istisqā'an,
`say'\arbnul{ay'}aN} شَيْءٌ šay'an,
`gAmi`aT|\arbnul{T}aN} جَامِعَةٌ ḡāmi'atan.
```

REM. Note that in the last example (*ḡāmi'at^{an}*), the ‘pipe’ character has been inserted before `\arbnul`. Otherwise, the `dmg` mode of the transliteration rules would have interpreted the *tā' marbūṭah* as *final* (e.g. *h* instead of the expected *t*).⁵¹

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

```
1 \newarbnul{Y}{~~~~0649}{ }
2 \arbnul{hud\arbnul{red}{aN\arbnul{A}}\arbnul{Y}}
3 \arbnul{trans}{hud\arbnul{red}{aN\arbnul{A}}\arbnul{Y}}
```

هُدًى hudāⁿ

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

```
wa-\arbnul{red}{\arbnul{wa}i}stisqA'uN} اِسْتِسْقَاءٌ wa-'stisqā'un52.
fi "al".i-\arbnul{red}{\arbnul{l-i}btidA'i}
فِي اَلْاِبْتِدَاءِ fi 'li-'btidā'.
\arbnul{red}{'a'\arbnul{k}}kulu} أَكُلُ akulu,
\arbnul{red}{'A'\arbnul{k}}kiluN} أَكِلُ akilun.
```

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

```
1 \newarbnul{alifmaddahred}{~~~~0627\arbnul{red}{~~~~0653}}%
2 {\arbnul{trans}{\arbnul{red}{'a'\arbnul{k}}}}
3 \arbnul{\arbnul{alifmaddahred}kulu}
4 \arbnul{trans}{\arbnul{alifmaddahred}kulu}.
```

أَكُلُ akulu.

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 المَعْلُونِ, *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:—

⁵¹See also on page 41 “Discarding the *ṣ̣rāb*” for more information.

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

```

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

```

المعلون *al-mu`allimūna*.

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

From the examples provided above, in fI 'l-nAsi في النَّاسِ *fi 'n-nāsi*, the initial *ʿalif al-waṣl*ⁱ can be rendered in red like so: `\arbcOLOR[red]{\arbnul{al-}a}` **أ**. 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}\arbmark{lnn}Asi}
4 \arb[trans]{fI\arbnul{al-}
5 \arbcOLOR[red]{\arbnul{al-}a}\arbmark{lnn}Asi}.

```

في النَّاسِ *fi 'n-nāsi*.

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]{|''}.alUna يَتَسَاءُلُونَ *yatasā'alūna*,
^{an}say\arbcOLOR[red]{|''}\arbnul{'a}N شَيْئًا *šay'an*, ⁱⁿsay\arbcOLOR[red]{|''}iN شَيْءٍ *šay'in*, \arbcOLOR[red]{a''}.as\arbcOLOR[red]{y''}.ilaTuN
 أَسْئَلَةُ *'as'ilat^{un}*.

9 Transliteration

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

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

dmg *Deutsche Morgenländische Gesellschaft*, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.⁵³ **dmg** transliteration convention is selected by default;

dmg+ This is the same as above, with the difference that the hamza is always printed, even in words preceded by the definite article, as in *al-'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;⁵⁴

New feature v1.8 arabica *Journal of Arabic and Islamic Studies/Revue d'études arabes et islamiques*: this

⁵³See Brockelmann et al., “Die Transliteration der arabischen Schrift”.

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

standard is most widely used by scholars in the field of Arabic studies.⁵⁵ More standards will be included in future releases of arabluatex.

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

```
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:—

```
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 إِسْحَقَ *Hunayn^u bn^u 'Ishāq^a,*
`\uc{`u_tm_an-u}` عُمَيْيْنُ *Utmān<sup>u</sup>, .daraba \uc{zayd-u} bn-u \uc{`h_alidiN}*
`\uc{sa`d-a} bn-a \uc{`awf-i} bn-i \uc{`abd-i} \uc{'l-l_ah-i}` ضَرَبَ
`\uc{`zayd-u} bn-u \uc{`hālidiN}` *daraba Zayd^u bn^u Hālidiⁿ Sa'd^a bn^a Awfⁱ bnⁱ*
`\uc{`abd-i} \uc{'l-l_ah-i}` *Abdiⁱ '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^{uni} '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`.

⁵⁵See http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf.

Hyphenation In case transliterated Arabic words break the T_EX hyphenation algorithm, one may use the `\-` command to insert discretionary hyphens. This command will be discarded in all of the Arabic modes of `arabxatex`, but will be processed by any of the transliteration modes:—

```
\uc{'abU} \uc{bakriN} \uc{mu\-.ham\-.madu} bnu \uc{za\-.ka \-.riy\-.
yA'a} \uc{'l-rAziyyu} أَبُو بَكْرٍ مُحَمَّدُ بْنُ زَكْرِيَّا 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-`ibAdiyyu}}
Abū Zaydin Ḥunaynu bnu Ishāqa 'l-Ibādiyyu.
```

New feature v1.10 **Proper names outside Arabic environments** Transliterated proper names inserted in paragraphs of English text should be printed in the same typeface as the surrounding text. `\prname{Arabic proper name}` is provided to that effect:⁵⁶—

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

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

^aWright, see n. 7, i. 23 C.

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

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

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

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

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

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

⁵⁸See below [sect. 13](#) on page 52 for more details.

9.1 Additional note on dmḡ convention

New feature v1.3 According to Brockelmann et al.,⁵⁹ Arabic *ʾiṣṣāb* may be rendered into **dmḡ** in three different ways:

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

\arbup By default, **arabluatex** applies rule (b). Once delimited by a set of Lua functions, *ʾiṣṣāb* 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 *ʾiṣṣāb* is displayed. To take one example, here is how Arabic *ʾiṣṣāb* may be rendered as subscript text:—

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

Arabic **dmḡ** transliteration for رَأَيْتُ جَامِعًا مُهْدَمَةً مُذْنَنَةً: *ra'aytu ḡami_{an} muhaddamat_{an} mi'danatu-hu*.

As shown in the above example, #1 is the token that is replaced with the actual *tanwīn* in the formatting directives of the **\SetArbUp** command.

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

munṣarīf: mu`allimuN مُعَلِّمٌ *mu'allim<sup>un</sup>*, kA'inuN كَائِنٌ *kā'in<sup>un</sup>*, kA'inAtuN كَائِنَاتٌ *kā'in<sup>ātun</sup>*, \uc{`amraNU} عَمَرُوا *Amr^{an}*, fataN_A فَتَى *fataⁿ*, qA.diNI قَاضٍ *qāḏiⁿ*.

ḡayr munṣarīf: al-mu`allim-u الْمُعَلِّمُ *al-mu'allim<sup>u</sup>*, kitAb-Ani كِتَابَانِ *kitāb<sup>āni</sup>*, ra`sa'-Ani رَسَائِنِ *raša^{āni}*, sAriq-Una سَارِقُونَ *sāriq^{āna}*, qA.d-Una قَاضُونَ *qāḏ^{āna}*, al-.zulm-Atu الظُّلُمَاتُ *aḏ-ḡulm^{ātu}*.

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

kAna .ganiyyaN l_akinna-hu labisa ḡubbaTaN mumazzaqaN 'aydu-hA كَانَ غَنِيًّا لَكِنَّهُ لَيْسَ جَبَّةً *kāna ḡaniyy^{an} lākinna-hu labisa ḡubbat^{an} mumazzaq^{an} 'aydu-hā*.

⁵⁹Carl 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/dmḡtransliteration.pdf, 6.

REM. *b* Although the hyphen before the *tanwīn* is optional as *arabluatex* always parses nouns with such termination, it may also be used to mark better the inflectional endings:—

mana`a 'l-nAs-a kAffaT-aN min mu\_hA.tabati-hi 'a.had-uN bi-sayyidi-nA مَنَّعَ النَّاسَ كُلَّهُ مِنْ مَخَاطِبَةِ أَحَدٍ بِسَيِّدِنَا mana`a 'n-nās^a kāffat^{an} min muḥāṭabatī-hi `aḥad^{un} bi-sayyidi-nā.

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

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

REM. It may so happen, as in the absence of the article before the annexed word, that *arabluatex* be unable to determine which of the above two cases the word ending with *tāʾ marbūṭah* falls into. The ‘pipe’ character (see sect. 5.5 on page 21) may be appended to that word to indicate that what follows is in the construct state: \uc{r}isAlaT fI tartIb qirA'aT| kutub \uc{~g}AlInUs *Risālah fī tartīb qirāʾat kutub Ġālīnūs*.

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

9.2 Examples

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

‘dmg+’ standard: `atā ṣadīq<sup>un</sup> `ilā Ġuḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratⁱⁿ qaṣīratⁱⁿ fa-qāla la-hu: “sawfa u`īdu-hu `ilay-ka fī l-masā`i wa-`adfa`u la-ka uḡrat<sup>an</sup>.” fa-qāla Ġuḥā: “anā `āsif^{un} ḡidd^{an} `annī lā `astatī`u `an uḥaqqiqa la-ka raḡbata-ka fa-l-ḥimār^u laysa hunā l-yawm^a.” wa-qabla `an yutimma Ġuḥā kalāma-hu bada`a l-ḥimār^u yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: “innī `asma`u ḥimāra-ka yā Ġuḥā yanhaqu.” fa-qāla la-hu Ġuḥā: “ḡarīb^{un} `amru-ka yā ṣadīqī `a-tuṣaddiqu l-ḥimār^a wa-tukaddiba-nī?”

‘loc’ standard: atā ṣadīqun ilā Juḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratīn qaṣīratīn fa-qāla la-hu: “sawfa u`īdu-hu ilay-ka fī al-masā`i wa-`adfa`u la-ka ujratan.” fa-qāla Juḥā: “anā `āsifun 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-tukadhdhiba-nī?”

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

10 Buckwalter input scheme

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

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

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

‘base’, ‘xml’ and ‘safe’ schemes `arabluatex` can use any of the so-called Buckwalter ‘base’, ‘xml’ or ‘safe’ schemes as they are described in Habash.^{61,62} However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode ا and ء, must be replaced with square brackets viz. [and] respectively.

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

[Table 9](#) gives the Buckwalter equivalents that are currently used by `arabluatex`. The additional characters that are defined in [table 6 on page 24](#) are also available.

Letter	Transliteration ⁶³			Buckwalter notation	
	dmg+	loc	arabica	base/xml	safe
ا	<i>a</i>	<i>a</i>	<i>a</i>	A	A
ب	<i>b</i>	<i>b</i>	<i>b</i>	b	b
ت	<i>t</i>	<i>t</i>	<i>t</i>	t	t
ث	<i>ṭ</i>	<i>th</i>	<i>ṭ</i>	v	v
ج	<i>ǧ</i>	<i>j</i>	<i>ǧ</i>	j	j
ح	<i>ḥ</i>	<i>ḥ</i>	<i>ḥ</i>	H	H
خ	<i>ḫ</i>	<i>kh</i>	<i>ḫ</i>	x	x
د	<i>d</i>	<i>d</i>	<i>d</i>	d	d
ذ	<i>ḏ</i>	<i>dh</i>	<i>ḏ</i>	*	V
ر	<i>r</i>	<i>r</i>	<i>r</i>	r	r
ز	<i>z</i>	<i>z</i>	<i>z</i>	z	z
س	<i>s</i>	<i>s</i>	<i>s</i>	s	s
ش	<i>š</i>	<i>sh</i>	<i>š</i>	\$	c
ص	<i>ṣ</i>	<i>ṣ</i>	<i>ṣ</i>	S	S
ض	<i>ḍ</i>	<i>ḍ</i>	<i>ḍ</i>	D	D
ط	<i>ṭ</i>	<i>ṭ</i>	<i>ṭ</i>	T	T
ظ	<i>ẓ</i>	<i>ẓ</i>	<i>ẓ</i>	Z	Z

Table 9: Buckwalter input scheme

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

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

⁶²I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in `arabluatex`.

⁶³See [sect. 9 on page 37](#).

Letter	Transliteration			Buckwalter notation	
	dmg+	loc	arabica	base/xml	safe
ع	‘	‘	‘	E	E
غ	<i>ġ</i>	<i>gh</i>	<i>ġ</i>	g	g
ف	<i>f</i>	<i>f</i>	<i>f</i>	f	f
ق	<i>q</i>	<i>q</i>	<i>q</i>	q	q
ك	<i>k</i>	<i>k</i>	<i>k</i>	k	k
ل	<i>l</i>	<i>l</i>	<i>l</i>	l	l
م	<i>m</i>	<i>m</i>	<i>m</i>	m	m
ن	<i>n</i>	<i>n</i>	<i>n</i>	n	n
ه	<i>h</i>	<i>h</i>	<i>h</i>	h	h
و	<i>w</i>	<i>w</i>	<i>w</i>	w	w
ي	<i>y</i>	<i>y</i>	<i>y</i>	y	y
آ	<i>ā</i>	<i>ā</i>	<i>ā</i>	Y	Y
أ	<i>ah</i>	<i>ah</i>	<i>a</i>	p	p
ء	’	’	’	’	C
أ	<i>ā</i>	<i>ā</i>	<i>ā</i>		M
أ	’	’	’	>	O
ؤ	’	’	’	&	W
إ	’	’	’	<	I
ئ	’	’	’	]	Q
ـ	—	—	—	~	~
أ	’	’	—	[	L
ا	<i>a</i>	<i>a</i>	<i>a</i>	a	a
ـ	<i>u</i>	<i>u</i>	<i>u</i>	u	u
ـ	<i>i</i>	<i>i</i>	<i>i</i>	i	i
ـ	<i>an</i>	<i>an</i>	<i>an</i>	F	F
ـ	<i>un</i>	<i>un</i>	<i>un</i>	N	N
ـ	<i>in</i>	<i>in</i>	<i>in</i>	K	K
ـ	—	—	—	o	o
ا	<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. 9 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*^u, Al-camsu الشَّمْسُ *aš-šams*^u, bi-SinaAEapi Al-T-ib~i, بِصِنَاعَةِ الطِّبِّ *bi-ṣinā‘atⁱ ṭ-ṭibbⁱ*.

wa-Al-l~ehi وَاللهِ wa-'l-lāhⁱ, Al-Hamdu li-l~ehi الْحَمْدُ لِلَّهِ al-ḥamd^u li-llāhⁱ.

Similary, it is not advisable to use | and [('base' and 'xml' schemes) or M and L ('safe' scheme) to encode the 'alif^u 'l-mamdūdātⁱ and the 'alif^u 'l-waṣlⁱ for such signs are supposed to be generated by arabuatex internal functions. Besides, as they do not *per se* convey any morphological information on what they are derived from, they cannot be transliterated accurately. To take one example, <ily Al-LntiqāADi gives إِلَى الْإِنْتِقَاضِ as expected, but only <ily Al-intiqADi can be transliterated as 'ilā 'l-intiqāḍī with the correct vowel ⟨i⟩ in place of the 'alif^u 'l-waṣlⁱ.

11 Unicode Arabic input

New feature v1.5 As said above in [sect. 10 on page 42](#) about the Buckwalter input scheme, even though arabuatex is primarily designed to process the ArabTeX notation, it also accepts Unicode Arabic input. It should be noted that arabuatex 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}
```

12 L^AT_EX Commands in Arabic environments

General principle L^AT_EX commands are accepted in Arabic environments. The general principle which applies is that 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}}` 45 مَهْ كِتَابُهُ فِي الْعَادَاتِ
*kitābu-hu fī 'l-Ādātⁱ.*⁶⁴

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

45 مَهْ كِتَابُهُ فِي الْعَادَاتِ⁶⁵

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
```

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

⁶⁵`\rlframefox` has been adapted from `\framebox` for insertions of right-to-left text.

```

7      'l-furs-i.}.
8      \end{arab}

```

إِنَّ أَبِي كَانَ مِنَ الْمُقَاتِلَةِ^a، وَكَانَتْ أُمِّي مِنْ عُظَمَاءِ بُيُوتِ الزَّمَاةِ^b.

^aالْمُقَاتِلَةِ: الْمُقَاتِلِينَ.
^bالزَّمَاةِ: 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^AT_EX command that would otherwise be rejected by `arabluatex`, such as commands the argument of which is expected to be a dimension or a unit of measurement.

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

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

```

1      \begin{arab}[fullvoc]
2          \uc{z}ayd-uN\arbnnull{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} \arbnnull{zayduN}ibn-u \uc{`a}mr-iNU
7      \end{arab}

```

زَيْدٌ^a ابْنُ عَمْرٍو

^a“*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^AT_EX macro `\thefootnote` like so:—

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

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

`\LRmarginpar` The `\LRmarginpar[<left>]{<right>}` command does for marginal notes the same as

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

`\LRfootnote` does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that `\marginpar` also works in Arabic environments, but it acts as any other single-argument command inserted in Arabic environments. The general principle laid on page 44 applies.

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

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

12.1 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 اتَّخَذَ 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) [...] (no argument)
- (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.

12.2 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_akinna-hu bi-karami-hi
6   wa-bi-.husni xuluqi-hi 'a.zhara 'l-ta'ajjuba min simani-hA
7   wa-.tIbi la.hmi-hA. wa-kAna <\uc{'abU} \uc{'l-hu_dayli}>
8   yu`rafu 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 الْهَذِيلِ أَهْدَى إِلَى مُوسَى دَجَاجَةً. وَكَانَتْ دَجَاجَتُهُ الَّتِي أَهْدَاهَا دُونَ مَا كَانَ يُخَذُّ لِمُوسَى. وَلَكِنَّهُ بِكَرَمِهِ وَبِحُسْنِ خُلُقِهِ أَظْهَرَ التَّعَجُّبَ مِنْ سَمْنِهَا وَطِيبِ لَحْمِهَا. وَكَانَ <أَبُو الْهَذِيلِ> يَعْرِفُ بِالْإِمْسَاكِ الشَّدِيدِ. فَقَالَ: "وَكَيْفَ رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ الدَّجَاجَةَ؟" قَالَ: "كَانَتْ عَجَبًا مِنَ الْعَجَبِ!" فَيَقُولُ:

"وَتَدْرِي مَا جِنْسُهَا؟ وَتَدْرِي مَا سَمْتُهَا؟ فَإِنَّ الدَّجَاجَةَ إِذَا تَطِيبُ بِالْجِنْسِ وَالسِّنِّ. وَتَدْرِي بِأَيِّ شَيْءٍ نَكَّأُ نُسَمُّهَا؟ وَفِي أَيِّ مَكَانٍ نَكَّأُ نَعْلِفُهَا؟"

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

12.2.1 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 شورای ملی 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.sIAni]
9 \begin{itemize}
10 \item \textbf{al-fa.slu 'l-'awwalu}
11 \end{itemize}
12 \end{itemize}

```

```

13 \end{itemize}
14 \end{arab}

```

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

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

```

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

```

```

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:^a— 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.

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

يَا أَفْعَالٌ	وَتَفَاعَلٌ	أَفْعَلٌ
يَبْ أَفْعَوْلٌ	زَ أَفْعَلٌ	بَفْعَلٌ
يَجْ أَفْعَوْلٌ	حَ أَفْعَلٌ	جَفَاعَلٌ
يَدْ أَفْعَلٌ	طَ أَفْعَلٌ	دَ أَفْعَلٌ
يَهْ أَفْعَلٌ	يَ اسْتَفْعَلٌ	هَ تَفْعَلٌ

^aWright, 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.

12.3 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 جُحَا: “غَرِيبٌ أَمْرُكَ يَا صَدِيقِي أَتَصَدِّقُ الْحِمَارَ وَتُكَذِّبُنِي؟”

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.

12.4 Two-argument special commands

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

```

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.ammadiN \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.ammadiN \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 الْإِمَامُ الصَّدْرُ الْكَبِيرُ الْعَالِمُ الْفَاضِلُ مَهْدَبُ الدِّينِ أَبُو مُحَمَّدٍ عَبْدُ الرَّحِيمِ بْنِ
عَلِيٍّ بْنِ حَامِدٍ وَيُعرفُ بِالذَّخْوَرِ.

*Muhaddabu 'd-Dīni 'Abdu 'r-Raḥīmi bnu 'Aliyyⁱⁿ huwa sayḥu-na 'l-'imāmu 'ṣ-
ṣadru 'l-kabīru 'l-'ālimu 'l-fādilu Muhaddabu 'd-Dīni 'Abū Muḥammadⁱⁿ 'Abdu
'r-Raḥīmi bnu 'Aliyyi bni Ḥāmidⁱⁿ wa-yu`rafu bi-'d-Daḥwari.*

reledmac The two-argument command `\edtext{<lemma>}{<commands>}` is supported inside `\begin{arab} ... \end{arab}`.⁶⁹ As an example, one may get arabuatex 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

```

12.5 quran

arabuatex 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. 11 on page 44](#) for more details). Please note that arabuatex takes care of formatting the Arabic: therefore, it is recommended to load the quran package with the `nopar` option, after arabuatex itself has been loaded, like so:—

⁶⁸arabuatex provides its own `\arbcolor` command which is able to render syllables or diacritics in colors. See [sect. 8 on page 33](#).

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

```

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

```

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

```

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﴾

13 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 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 Lua⁷⁰TeX to be run several times as will be explained below.

13.1 Commands and environments

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

⁷⁰See above on page 5 for more information.

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

`\ArbOutFile` **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.

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

Please note that inside `arabverse` environments `\bayt` is replaced with `\bayt*`.⁷⁵

13.2 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 `<jobname>` is the name of the original `.tex` source file, and `<suffix>` the suffix appended to the file that is to be opened and compiled again.

13.3 Further processing of Unicode converted files

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

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

⁷¹See above [sect. 11](#) on page 44.

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

⁷³See above on page 46.

⁷⁴See above on page 39.

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

⁷⁶See <http://pandoc.org/>

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

```

14 Future work

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

- (a) Short-term:
 - i. TEI `xml` support: `arabluatex` will interoperate with TEI `xml` through new global and local options that will output Arabic in a TEI `xml` compliant file in addition to the usual PDF output: see on page 4.
- (b) Medium-term:
 - i. More languages: the list of supported languages will eventually be the same as `arabtex`: see 5 on page 4.
 - ii. Formulate propositions for extending the Arab $\TeX$ notation and the transliteration tables. Include them in `arabluatex`. See sect. 5.9 on page 24.

15 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 Lua $\TeX$ 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 T_EXlive:

```

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\rmfamily#1\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   {\luairect{arabluatex.mkarbbreak(\luastringN{#2}, "out")}}
53   {\luairect{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 alway puts the stroke over its argument. As of v1.19, `arabluatex` uses `lua-ul` to render the strokes, thus allowing line breaks and manual hyphenation for transliterated Arabic.

`\aoline` `\aoline` and `\aoline` derive from `\newunderlinetype` provided by the `lua-ul` package
`\aoline*` 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\@aoline{\@ifstar\@aoline\@aoline}
57 \def\@aoline#1{\ensuremath{\overline{\mbox{#1}}}}
58 \def\@aoline#1{\@aoverLine#1}
59 \newunderlinetype\@aunderLine{\leaders\vrule height -.65ex depth .75ex}
60 \def\@aoline#1{\@aunderLine#1}
61 \AtBeginDocument{\ifdef{\aemph}%
62   {\RenewDocumentCommand{\aemph}{s m}{%
63     \IfBooleanTF{#1}{%
64       \luairect{tex.sprint(arabluatex.aemph(\luastringN{#2},
65         "over"))}}
66       {\luairect{tex.sprint(arabluatex.aemph(\luastringN{#2},
67         "dflt"))}}}
68   {\NewDocumentCommand{\aemph}{s m}{%
69     \IfBooleanTF{#1}{%
70       \luairect{tex.sprint(arabluatex.aemph(\luastringN{#2},
71         "over"))}}
72       {\luairect{tex.sprint(arabluatex.aemph(\luastringN{#2},
73         "dflt"))}}}
73 }
```

`\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_{TEX} 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*, *‘alif mamdūdah*
`\SetArbEasy*` and *sukūn* depending on the modes which are selected, either `voc`, `fullvoc` or `trans`. Such
`\SetArbDflt` 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 16; 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}}}

```

`\SetArab` This command activates the Maghribi writing convention. `\SetArab` can be used at any point of the document to set or revert to classical Arabic. Both commands must be issued outside Arabic commands or environments.

`\SetMaghribi`

```

87 \def\al@arb@variant{dflt}
88 \NewDocumentCommand{\SetMaghribi}{-}{%
89   \def\al@arb@variant{maghribi}%
90 }
91 \NewDocumentCommand{\SetArab}{-}{%
92   \def\al@arb@variant{dflt}%
93 }

```

`\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.

```

94 \def\al@trans@font{\rmfamily}%
95 \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:

```

96 \def\al@trans@style{\itshape}%
97 \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.

```

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

```

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

```

99 \def\al@trans@convention{dmg}
100 \NewDocumentCommand{\SetTranslitConvention}{m}{%
101   \def\al@trans@convention{#1}}

```

`\arbut` By default, `\arbut` is set to `\textsuperscript`. This is how the *tanwīn* that takes place

`\NoArbUp` at the end of a word should be displayed in `dmg` mode. `\NoArbUp` may be used either in the

`\ArbUpDflt` preamble or at any point of the document in case one wishes to have the *tanwīn* on the line.

`\SetArbUp` The default rule can be set back with `\ArbUpDflt` at any point of the document. Finally `\SetArbUp` can be used to customize the way *tanwīn* is displayed: this command takes the formatting directives as argument, like so: `\SetArbUp{<code>}`.

```

102 \NewDocumentCommand{\al@arbut@dflt}{m}{\textsuperscript{#1}}%
103 \NewDocumentCommand{\al@arbut}{m}{\al@arbut@dflt{#1}}

```

```

104 \NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
105 \NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt}
106 \NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\al@arbup}{m}{#1}}
107 \NewDocumentCommand{\SetArbUp}{m}{%
108   \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.

```

109 \NewDocumentCommand{\uc}{m}%
110   {\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.

```

111 \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.

```

112 \NewDocumentCommand{\prname}{s m}{%
113   \bgroup\SetTranslitStyle{\relax}%
114   \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.

```

115 \NewDocumentCommand{\txarb}{+m}{%
116   \ifvmode\leavevmode\fi%
117   \bgroup\textdir TRT\arabicfont#1\egroup}
118 \NewDocumentCommand{\txtrans}{+m}{%
119   \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.

```

120 \NewDocumentEnvironment{txarab}{}{%
121   \par%
122   \booltrue{al@rlmode}%
123   \pdir 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.

```

124 \NewDocumentEnvironment{txarabtr}{}{%
125   \par%
126   \pdir TLT\textdir TLT%
127   \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.

```

128 \NewDocumentCommand{\arb}{0{\al@mode} +m}{%
129 {\edef\@tempa{#1}%
130  \ifx\@tempa\al@mode@voc%
131  \ifvmode\leavevmode\fi%
132  \bgroup\booltrue{al@rlmode}\texdir TRT\arabicfont%
133  \luadirect{tex.sprint(arabluatex.processvoc(
134    \luastringN{#2},
135    \luastringO{\al@arb@rules},
136    \luastringO{\al@arb@variant},
137    \luastringO{\al@input@scheme}
138    )}\egroup%
139  \else%
140  \ifx\@tempa\al@mode@fullvoc%
141  \ifvmode\leavevmode\fi%
142  \bgroup\booltrue{al@rlmode}\texdir TRT\arabicfont%
143  \luadirect{tex.sprint(arabluatex.processfullvoc(
144    \luastringN{#2},
145    \luastringO{\al@arb@rules},
146    \luastringO{\al@arb@variant},
147    \luastringO{\al@input@scheme}
148    )}\egroup%
149  \else%
150  \ifx\@tempa\al@mode@novoc%
151  \ifvmode\leavevmode\fi%
152  \bgroup\booltrue{al@rlmode}\texdir TRT\arabicfont%
153  \luadirect{tex.sprint(arabluatex.processnovoc(
154    \luastringN{#2},
155    \luastringO{\al@arb@rules},
156    \luastringO{\al@arb@variant},
157    \luastringO{\al@input@scheme}
158    )}\egroup%
159  \else%
160  \ifx\@tempa\al@mode@trans%
161  \bgroup\texdir TLT%
162  \luadirect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
163    \luastringO{\al@trans@convention},
164    \luastringO{\al@arb@rules},
165    \luastringO{\al@input@scheme}))}\egroup%
166  \else%
167  \fi\fi\fi\fi}

```

`\arbmark` `\arbmark[<rl>]{<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.

```

168 \NewDocumentCommand{\arbmark}{0{ } m}{%
169  \bgroup%
170  \SetInputScheme{arabtex}%
171  \luadirect{tex.sprint(arabluatex.processarbmarks(\luastringN{#2},
172    \luastringN{#1}))}%
173  \egroup}

```

`\newarbookmark` `\newarbookmark` 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.

```

174 \NewDocumentCommand{\newarbookmark}{m m m}{%
175   \luairect{arabluatex.newarbookmark(\luastringN{#1}, \luastringN{#2},
176     \luastringN{#3})}}

```

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

```

177 \NewDocumentEnvironment{arab}{!O{\al@mode} +b}%
178 {\par\edef\@tempa{#1}%
179   \ifx\@tempa\al@mode@voc%
180     \booltrue{al@rlmode}%
181     \bgroup\pdir TRT\textdir TRT\arabicfont%
182     \luairect{tex.sprint(arabluatex.processvoc(
183       \luastringN{#2},
184       \luastringO{\al@arb@rules},
185       \luastringO{\al@arb@variant},
186       \luastringO{\al@input@scheme}
187     ))}\egroup%
188   \else%
189     \ifx\@tempa\al@mode@fullvoc%
190       \booltrue{al@rlmode}%
191       \bgroup\pdir TRT\textdir TRT\arabicfont%
192       \luairect{tex.sprint(arabluatex.processfullvoc(
193         \luastringN{#2},
194         \luastringO{\al@arb@rules},
195         \luastringO{\al@arb@variant},
196         \luastringO{\al@input@scheme}
197       ))}\egroup%
198     \else%
199       \ifx\@tempa\al@mode@novoc%
200         \booltrue{al@rlmode}%
201         \bgroup\pdir TRT\textdir TRT\arabicfont%
202         \luairect{tex.sprint(arabluatex.processnovoc(
203           \luastringN{#2},
204           \luastringO{\al@arb@rules},
205           \luastringO{\al@arb@variant},
206           \luastringO{\al@input@scheme}
207         ))}\egroup%
208       \else%
209         \ifx\@tempa\al@mode@trans%
210           \bgroup\pdir TLT\textdir TLT%
211           \luairect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
212             \luastringO{\al@trans@convention},
213             \luastringO{\al@arb@rules},
214             \luastringO{\al@input@scheme}))}\egroup%
215         \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.

```

216 \newlength{\al@bayt@width}
217 \newlength{\al@gutter@width}
218 \newlength{\al@verse@twidth}

```

```

219 \setlength{\al@bayt@width}{.3\textwidth}
220 \setlength{\al@gutter@width}{.15\al@bayt@width}
221 \define@key[al]{verse}{width}{\setlength{\al@bayt@width}{#1}}
222 \define@key[al]{verse}{gutter}{\setlength{\al@gutter@width}{#1}}
223 \define@key[al]{verse}{metre}{\def\al@verse@metre@value{\arb{#1}}}
224 \define@key[al]{verse}{color}[]{\color{#1}}
225 \define@boolkey[al]{verse}{utf}[true]{}
226 \define@boolkey[al]{verse}{delim}[true]{}
227 \define@boolkey[al]{verse}{export}[true]{}
228 \define@choicekey[al]{verse}{mode}{fullvoc, voc, novoc,
229   trans}{\def\al@mode{#1}}
230 \presetkeys[al]{verse}{utf=false, delim=false}{}

```

Then follows the environment itself:

```

231 \NewDocumentEnvironment{arabverse}{!O{}}%
232 {\bgroup\setkeys[al]{verse}[width, gutter, color, utf, delim,
233   metre]{#1}%
234   \if@pkg@export
235     \ifal@verse@export
236       \ArbOutFile{\begin{arabverse}}%
237       % \ifx\al@mode\al@mode@trans%
238       % \luadirect{arabluatex.tooutfile(\luastringN{[#1]})}%
239       % \else%
240       \IfSubStr[1]{#1}{utf}%
241       {\luadirect{arabluatex.tooutfile(\luastringN{[#1]})}}%
242       {\luadirect{arabluatex.tooutfile(\luastringN{[#1, utf]})}}%
243       % \fi
244     \else
245     \fi
246   \else
247   \fi
248   \egroup
249   \centering\noindent\bgroup\setkeys[al]{verse}[metre]{#1}%
250   % \ifx\al@mode\al@mode@trans%
251   % \ifal@verse@utf\setRL\else\setLR\fi%
252   % \else\setRL\fi%
253   \ifal@verse@utf
254     \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi
255   \else
256     \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi
257   \fi
258   \addtolength{\al@verse@twidth}{2\al@bayt@width}%
259   \addtolength{\al@verse@twidth}{\al@gutter@width}%
260   \arab@v@export{#1}
261 }%
262 {\endarab@v@export
263   \setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
264     export]{#1}%
265   \ifdefined\al@verse@metre@value\hfill\al@verse@metre@value\fi
266 \egroup
267 \bgroup\setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
268   metre]{#1}%
269 \if@pkg@export\ifal@verse@export
270   \ArbOutFile{\end{arabverse}}%
271 \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 *ʿağuz*. That two subsequent hemistichs should be connected with one another is technically named *tadwīr*. In some of these cases, the hemistichs may be connected by a prominent horizontal flexible stroke which is drawn by the `\al@verse@stroke` command.

`\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.

```

272 \newif\ifal@warp@bayt
273 \al@warp@bayttrue
274 \NewDocumentCommand{\StretchBayt}{0{true}}{
275   \edef\oarg@true{true}
276   \edef\oarg@false{false}
277   \edef\@tempa{#1}
278   \ifx\@tempa\oarg@true\al@warp@bayttrue
279   \else
280   \ifx\@tempa\oarg@false\al@warp@baytfalse
281   \else
282   \PackageError{arabluatex}{\string\StretchBayt\space must be
283     either 'true' or 'false'}}{}
284   \fi
285   \fi
286 }
287 \NewDocumentCommand{\arb@utf}{m}{%
288   \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
289 \def\al@hemistich@delim{*}
290 \NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
291 \def\al@verse@stroke{\leavevmode\xleaders\hbox{\arb{--}}\hfill\kern0pt}
292 \providebool{ekd@state}
293 \NewDocumentCommand{\bayt}{t+ s m o m}{%
294   \IfBooleanTF{#1}{%
295     \ifekd@state
296       \leavevmode
297       \stepcounter{ekd@lab}%
298       \zlabel{ekd:\theekd@lab}%
299       \luadirect{ekdosis.storeabspg(
300         \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}{%
301       \add@apparatus
302       \fi
303     }{\relax}%
304     \IfBooleanTF{#2}{\relax}{\relax}%
305     \ifdefined\savenotes\savenotes\else\fi%
306     \edef\al@tatweel{--}%
307     \ifal@warp@bayt%
308       \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
309     \else%
310       \makebox[\al@bayt@width][s]{\arb@utf{#3}}%
311     \fi%
312     \IfNoValueTF{#4}{%

```

```

313 \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
314 \else%
315 \hspace{\al@gutter@width}%
316 \fi
317 }{%
318 \edef\@tempa{#4}%
319 \ifx\@tempa\al@tatweel%
320 \ifx\al@mode\al@mode@trans%
321 \hspace{\al@gutter@width}%
322 \else%
323 \makebox[\al@gutter@width][s]{\al@verse@stroke}%
324 \fi%
325 \else%
326 \ifx\al@mode\al@mode@trans%
327 \ifal@warp@bayt%
328 \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#4}}%
329 \else%
330 \makebox[\al@gutter@width][s]{\arb@utf{#4}}%
331 \fi%
332 \else%
333 \makebox[\al@gutter@width][s]{\arb@utf{#4}}%
334 \fi\fi}%
335 \ifal@warp@bayt%
336 \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#5}}%
337 \else%
338 \makebox[\al@bayt@width][s]{\arb@utf{#5}}%
339 \fi%
340 \ifdefined\spewnotes\spewnotes\else\fi%
341 }

```

`\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`.

```

342 \NewDocumentCommand{\SetDefaultIndex}{m}{
343 \edef\@tempa{#1}
344 \ifx\@tempa\empty
345 \def\al@default@index{\jobname}
346 \else
347 \def\al@default@index{#1}
348 \fi
349 }

350 \def\al@index@mode{\al@mode}
351 \NewDocumentCommand{\SetIndexMode}{m}{
352 \def\al@index@mode{#1}
353 }

354 \define@cmdkeys[al]{index}[alind@]{index,root,form,pipe}
355 \NewDocumentCommand{\arind}{o m}{%
356 \IfNoValueTF{#1}{%
357 \ifdefined\al@default@index%
358 \csname index\endcsname[\al@default@index]{#2}%
359 \else%
360 \csname index\endcsname{#2}%
361 \fi%
362 }{%
363 \bgroup

```

```

364 \setkeys[al]{index}{#1}%
365 \def\al@one{%
366   \ifdefined\alind@root
367     \ifnum\alind@root < 10%
368       !\LR{0\alind@root}%
369     \else
370       !\LR{\alind@root}%
371     \fi
372   \else
373     !\LR{01}%
374   \fi}%
375 \def\al@two{%
376   \ifdefined\alind@form @\arb[\al@index@mode]{\alind@form}\else\fi}%
377 \ifdefined\alind@index%
378   \ifdefined\alind@pipe
379     \csname index\endcsname[\alind@index]{#2\al@one\al@two|\alind@pipe}%
380   \else
381     \csname index\endcsname[\alind@index]{#2\al@one\al@two}%
382   \fi
383 \else%
384   \ifdefined\al@default@index%
385     \ifdefined\alind@pipe
386       \csname index\endcsname[\al@default@index]{%
387         #2\al@one\al@two|\alind@pipe}%
388     \else
389       \csname index\endcsname[\al@default@index]{%
390         #2\al@one\al@two}%
391     \fi
392   \else%
393     \ifdefined\alind@pipe
394       \csname index\endcsname{#2\al@one\al@two|\alind@pipe}%
395     \else
396       \csname index\endcsname{#2\al@one\al@two}%
397     \fi
398   \fi%
399 \fi%
400 \egroup}}

```

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

```

401 \AtBeginDocument{%
402   \ifdefined\abjad%
403     \RenewDocumentCommand{\abjad}{m}%
404     {\ifbool{al@rlmode}%
405       {\aoline*{%
406         \luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}
407       {\luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}
408   \else%
409     \NewDocumentCommand{\abjad}{m}%
410     {\ifbool{al@rlmode}%
411       {\aoline*{%
412         \luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}
413       {\luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}
414     \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.

```

415 \NewDocumentCommand{\ayah}{m}{%
416   \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.

```

417 \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`.

```

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

```

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

```

420 \DeclareDocumentCommand{\LRmarginpar}{o m}{%
421   \IfNoValueTF{#1}
422   {\marginpar{\textdir TLT #2}}
423   {\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.

```

424 \DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pdir
425   TLT\textdir TLT\footnote{#1}\egroup}
426 \DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pdir
427   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.

```

428 \NewDocumentCommand{\FixArbFtnmk}{}{%
429   \@ifpackageloaded{scrextend}%
430   {\AtBeginDocument{%
431     \deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%
432   {\RequirePackage{scrextend}
433     \AtBeginDocument{%
434       \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⟩}`.

```

435 \NewDocumentCommand{\SetArbOutSuffix}{m}{
436   \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.

```

437 \NewDocumentCommand{\ArbOutFile}{s O{no} +m}{%
438   \if@pkg@export%
439   \IfBooleanTF{#1}{%
440     #3\luadirect{arabluatex.tooutfile(\luastringN{#3}, "#2")}}{%
441     \luadirect{arabluatex.tooutfile(\luastringN{#3}, "#2")}}%
442   \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.

```

443 \NewDocumentEnvironment{arabexport}{+b}{%
444   \if@pkg@export%
445   \par
446   #1
447   \luadirect{arabluatex.doexport("yes")}
448   \luadirect{tex.sprint(arabluatex.arbtoutf(\luastringN{#1}))}
449   \luadirect{arabluatex.doexport("no")}
450   \else\par#1\fi
451 }{\par}

```

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

```

452 \NewDocumentEnvironment{arab@v@export}{0{} +b}{%
453   \setkeys{al}{verse}[width, gutter, color, utf, delim, mode,
454   metre]{#1}
455   \if@pkg@export
456     \ifal@verse@export
457       \ifekd@state
458         \begin{ekdverse}[width=\al@verse@twidth]
459           #2
460         \end{ekdverse}
461       \else
462         #2
463       \fi
464       \luadirect{arabluatex.doexport("arabverse")}%
465       \luadirect{tex.sprint(arabluatex.arbtoutf(\luastringN{#2}))}%
466       \luadirect{arabluatex.doexport("no")}%
467     \else
468       \ifekd@state
469         \begin{ekdverse}[width=\al@verse@twidth]
470           #2
471         \end{ekdverse}
472       \else
473         #2
474       \fi
475     \fi
476   \else
477     \ifekd@state
478       \begin{ekdverse}[width=\al@verse@twidth]
479         #2
480       \end{ekdverse}
481     \else

```

```

482     #2
483     \fi
484 \fi
485 \}{}

```

`\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.

```

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

```

Errors and Warnings

```

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

```

That is it. Say goodbye before leaving.

Patches

```

497 \NeedsTeXFormat{LaTeX2e}
498 \ProvidesPackage{arabluatex-patch}%
499 [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:

```

500 \def\list#1#2{%
501   \ifnum \@listdepth >5\relax
502     \@toodeep
503   \else
504     \global\advance\@listdepth\@ne
505   \fi
506   \rightmargin\z@
507   \listparindent\z@
508   \itemindent\z@
509   \csname @list\romannumeral\the\@listdepth\endcsname
510   \def\@itemlabel{#1}%
511   \let\makelabel\mklab
512   \nmbrlistfalse
513   #2\relax
514   \@trivlist
515   \parskip\parsep
516   \parindent\listparindent
517   \advance\linewidth -\rightmargin
518   \advance\linewidth -\leftmargin

```

patch begins:

```

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

```

patch ends.

```

521 \parshape \@ne \@totalleftmargin \linewidth
522 \ignorespaces}
523 \def\@item[#1]{%
524 \if@noprimer
525 \@donoprimer
526 \else
527 \if@inlabel
528 \indent \par
529 \fi
530 \ifhmode
531 \unskip\unskip \par
532 \fi
533 \if@newlist
534 \if@noprimer
535 \@noprimer
536 \else
537 \addpenalty\@beginparpenalty
538 \addvspace\@topsep
539 \addvspace{-\parskip}%
540 \fi
541 \else
542 \addpenalty\@itempenalty
543 \addvspace\itemsep
544 \fi
545 \global\@inlabeltrue
546 \fi
547 \everypar{%
548 \@minipagefalse
549 \global\@newlistfalse
550 \if@inlabel
551 \global\@inlabelfalse
552 {\setbox\z@\lastbox
553 \ifvoid\z@
554 \kern-\itemindent
555 \fi}%
556 \box\@labels
557 \penalty\z@
558 \fi
559 \if@noprimer
560 \@noprimerfalse
561 \clubpenalty \@M
562 \else
563 \clubpenalty \@clubpenalty
564 \everypar{}%
565 \fi}%
566 \if@noitemarg
567 \@noitemargfalse
568 \if@noprimer
569 \refstepcounter\@listctr
570 \fi
571 \fi

```

patch begins:

```

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

```

patch ends.

```

576 \unhbox\@labels
577 \hskip \itemindent
578 \hskip -\labelwidth
579 \hskip -\labelsep
580 \ifdim \wd\@tempboxa >\labelwidth
581 \box\@tempboxa
582 \else
583 \hbox to\labelwidth {\unhbox\@tempboxa}%
584 \fi
585 \hskip \labelsep}%
586 \ignorespaces}

```

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

```

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

```

16 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.
- 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.d in.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).

17 Change History

v1.0.	General: Initial release	1	\SetHemistichDelim: New		
			\SetHemistichDelim command for		
			changing the default delimiter		
			between hemistichs	62	
v1.0.1.	General: Minor update of the				
	documentation	1			
v1.1.	\abjad: New and more flexible \abjad				
	command.	64			
v1.2.	\SetArbEasy: New				
	\SetArbEasy/\SetArbDflt for				
	‘modern’ or ‘classic’ Arabic styles. .	56			
v1.3.	\arbut: <i>ʾrāb</i> is now written as				
	superscript text in dng mode by				
	default.	57			
v1.4.	\SetInputScheme: \SetInputScheme				
	can be used to process other input				
	schemes such as ‘Buckwalter’	56			
	\SetTranslitFont: For selecting a				
	specific font for transliterated texts	57			
v1.4.3.	\abraces: New \abraces command				
	which expresses its argument				
	between braces.	65			
v1.4.4.	\SetArbEasy*: this starred version				
	discards the <i>sukūn</i> in addition to				
	what is already discarded by				
	\SetArbEasy.	56			
v1.5.	General: Compatibility with the quran				
	package	51			
	Environments may be nested inside				
	the arab environment	47			
	txarab: New txarab environment for				
	typesetting running paragraphs in				
	Unicode Arabic	58			
v1.6.	arabverse: New environment				
	arabverse for typesetting Arabic				
	poetry	60			
	\bayt: New macro \bayt for typesetting				
	each verse inside the arabverse				
	environment	61			
	\SetArbDflt*: This starred version				
	applies the assimilation rules in				
	addition to what \SetArbDflt				
	already does.	57			
			\SetHemistichDelim: New		
			\SetHemistichDelim command for		
			changing the default delimiter		
			between hemistichs	62	
			v1.7.		
			\arbnul: New \arbnul command for		
			putting back on any contextual		
			analysis rule broken by other		
			commands.	65	
			v1.8.		
			General: arabica transliteration		
			standard is now supported	37	
			v1.8.5.		
			General: Six additional Persian		
			characters are now available	9	
			v1.9.		
			\MkArbBreak: New \MkArbBreak		
			command for inserting user-defined		
			macros in Arabic environments . . .	55	
			v1.9.2.		
			\aemph*: Starred version which always		
			puts the stroke over its argument .	56	
			v1.10.		
			General: \uc supersedes \cap	39	
			\prname: New command for typesetting		
			Arabic proper names in		
			transliteration	58	
			v1.11.		
			\arbmark: New command for inserting		
			additional marks in Arabic		
			environments	59	
			\newarbmark: Allows defining additional		
			sets of Arabic marks	59	
			v1.12.		
			General: \abjad can now process L ^A T _E X		
			counters	24	
			\arbcolor: Standard color command for		
			Arabic environments	56	
			\MkArbBreak*: ‘starred’ version which		
			closes Arabic environments before		
			processing declared commands. . .	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.	66	
			arabverse: New options color and		
			export to arabverse environment. .	60	
			\arbmark: New optional argument:		
			either rl or lr	59	
			\ArbOutFile: Silently exports its		
			argument in the selected external file.	65	

<code>\arbparidir</code> : Sets the direction of Arabic paragraphs once they are converted to Unicode.	67	v1.19.	<code>\aemph</code> : Now uses lua-ul	56
<code>\prname*</code> : Renders proper names already converted to Unicode in upright roman style	58	v1.20.	<code>\auline</code> : Non context-sensitive command to underline Arabic words is provided	56
<code>\SetArbOutSuffix</code> : Sets a suffix to be appended to the filename of the external Unicode file.	65	v1.21.	<code>\StretchBayt</code> : Optionally removes stretching from lines of poetry ...	62
v1.15.			<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	64		<code>\SetArbNumbers</code> : selects Indian or Arabic numbers	55
v1.16.			<code>\SetMaghribi</code> : Maghribi writing convention	57
<code>\aemph</code> : Now uses ulem	56			
v1.18.				
<code>\arind</code> : New command <code>\arind</code> for building indexes	63			

18 Index

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

Symbols		
<code>\@@aoline</code>	56, 58	<code>\@tempboxa</code>
<code>\@M</code>	561	572, 573, 580, 581, 583
<code>\@aoline</code>	56, 57	<code>\@toodeep</code>
<code>\@aoverLine</code>	55, 58	502
<code>\@aunderLine</code>	59, 60	<code>\@topsep</code>
<code>\@beginparpenalty</code> ...	537	538
<code>\@clubpenalty</code>	563	<code>\@totalleftmargin</code> ...
<code>\@donoparitem</code>	525	519, 520, 521
<code>\@ifpackageloaded</code> ...	429	<code>\@trivlist</code>
<code>\@ifstar</code>	56	514
<code>\@inlabelfalse</code>	551	<code>_</code>
<code>\@inlabeltrue</code>	545	34
<code>\@item</code>	523	
<code>\@itemlabel</code>	510	
<code>\@itempenalty</code>	542	
<code>\@labels</code>	556, 574, 575, 576	
<code>\@listctr</code>	569	
<code>\@listdepth</code>	501, 504, 509	
<code>\@minipagefalse</code>	548	
<code>\@mklab</code>	511	
<code>\@nbitem</code>	535	
<code>\@ne</code>	504, 521	
<code>\@newlistfalse</code>	549	
<code>\@nmbrlistfalse</code>	512	
<code>\@nbreakfalse</code>	560	
<code>\@noitemargfalse</code> ...	567	
<code>\@tempa</code>	129, 130, 140, 150, 160, 178, 179, 189, 199, 209, 277, 278, 280, 318, 319, 343, 344	
		A
		<code>\abjad</code>
		24, 401
		<code>\abraces</code>
		30, 418
		<code>\add@apparatus</code>
		301
		<code>\addpenalty</code> ...
		537, 542
		<code>\addtolength</code> ...
		258, 259
		<code>\addvspace</code> .
		538, 539, 543
		<code>\adjustbox</code> .
		308, 328, 336
		<code>\advance</code>
		504, 517, 518, 519, 520
		<code>\aemph</code>
		25, 55
		<code>\aemph*</code>
		25, 55
		<code>\al@arb@rules</code>
		... 78, 81, 82, 85,
		86, 135, 145, 155,
		164, 184, 194, 204, 213
		<code>\al@arb@variant</code>
		.. 87, 89, 92, 136,
		146, 156, 185, 195, 205
		<code>\al@arbup</code>
		103, 104, 105, 106, 108
		<code>\al@arbup@dflt</code> 102, 103, 105
		<code>\al@bayt@width</code>
		216, 219, 220, 221,
		258, 308, 310, 336, 338
		<code>\al@default@index</code> ...
		345, 347,
		357, 358, 384, 386, 389
		<code>\al@error</code>
		489, 490
		<code>\al@gutter@width</code> ...
		217, 220,
		222, 259, 313, 315,
		321, 323, 328, 330, 333
		<code>\al@chemistich@delim</code> .
		289, 290, 313
		<code>\al@index@mode</code>
		350, 352, 376
		<code>\al@input@scheme</code> ..
		76,
		77, 137, 147, 157,
		165, 186, 196, 206, 214
		<code>\al@mode</code>
		4, 5, 6, 7, 128, 177,
		229, 237, 250, 254,
		256, 320, 326, 350, 487
		<code>\al@mode@fullvoc</code>
		17, 140, 189
		<code>\al@mode@novoc</code> 18, 150, 199
		<code>\al@mode@trans</code>
		19,
		160, 209, 237, 250,
		254, 256, 320, 326, 487
		<code>\al@mode@voc</code> ..
		16, 130, 179
		<code>\al@one</code>
		365, 379,
		381, 387, 390, 394, 396

`\al@tatweel` 306, 319
`\al@trans@convention` .
 99, 101, 163, 212
`\al@trans@font` . 94, 95, 98
`\al@trans@style` 96, 97, 98
`\al@two` 375, 379,
 381, 387, 390, 394, 396
`\al@verse@metre@value`
 223, 265
`\al@verse@stroke` 291, 323
`\al@verse@twidth` . 218,
 258, 259, 458, 469, 478
`\al@warning` 488, 494
`\al@warp@baytfalse` .. 280
`\al@warp@bayttrue` 273, 278
`\al@wrong@mark` 494
`\al@wrong@nesting` .. 490
Alessi, Robert 28
`\alind@form` 376
`\alind@index` . 377, 379, 381
`\alind@pipe` 378,
 379, 385, 387, 393, 394
`\alind@root`
 .. 366, 367, 368, 370
`\alttrfont` 98, 119, 127
Amiri 5
amiri (package) 5
`\aoline` ... 25, 55, 405, 411
`\aoline*` 25, 55
arab (env.) 7, 177
`\arab@v@export` 260
`\arab@v@export (env.)` .. 452
ArabTeX 3, 19
arabexport (env.) .. 52, 443
Arabic-English Lexicon,
 An 18
`\arabicfont` 32,
 34, 36, 117, 123, 132,
 142, 152, 181, 191, 201
arabluatex (package) ...
 2–8, 10–13,
 15–19, 23–26, 28,
 30, 32–34, 38–42,
 44–47, 50–58, 65–67
arabtex (package)
 3, 4, 16, 19, 54
arabulatex (package) 16
arabverse (env.) .. 25, 216
arabxetex (package) ... 4, 16
`\arb` 7, 114,
 128, 223, 288, 291, 376
`\arb@utf` . 287, 308, 310,
 328, 330, 333, 336, 338
`\arbcOLOR` 34, 74
`\arbmark` 31, 168, 495
`\arbnulL` 22, 417
`\ArbOutFile` 53, 236, 270, 437
`\ArbOutFile*` 53, 437
`\arbpardir` 53, 486
`\arbpup` 40, 102
`\ArbUpDflt` 40, 102
`\arind` 342
`\AtEndDocument` 12
`\auline` 25, 55
`\ayah` 33, 415

B

babel (package) 50
babel-french (package) .. 50
`\bayt` 26, 272
`\bayt+` 28
`\begin` . 236, 458, 469, 478
bidi (package) 69
`\boolfalse` 39
`\booltrue` . 37, 122, 132,
 142, 152, 180, 190, 200
`\box` 556, 581
Brockelmann, Carl 10, 37, 40

C

`\centering` 249
`\clubpenalty` 561, 563
`\color` 224
color (option) 26
`\color@endgroup` 588
`\color@setgroup` 588
csquotes (package) 50

D

`\DeclareOptionX` . 4, 5, 6, 7
`\deffootnote` 431, 434
`\define@boolkey`
 8, 225, 226, 227
`\define@choicekey` ... 228
`\define@cmdkeys` 354
`\define@key` 221, 222, 223, 224
`\delim (option)` 26

E

`\empty` 344
`\end` 270, 460, 471, 480
`\endarab@v@export` ... 262
`\ensuremath` 57
enumitem (package) .. 48–50
environments:
 arab 7, 177
 arab@v@export 452
 arabexport 52, 443
arabverse 25, 216
txarab 44, 120
txarabtr 124
`\everypar` 547, 564
`\ExecuteOptionsX` 14
export (option) ... 5, 26, 52

F

`\FixArbFtnmk` 45, 428
fontspec (package) . 4, 38, 57
`\footnote` ... 425, 427, 491
footnote (package) 28
footnotehyper (package) .. 28
fullvoc (option) 5, 8

G

`\global` 504,
 545, 549, 551, 574, 575
Grammar of the Arabic
 Language, A 5–8,
 13–18, 23, 24, 39, 50
gutter (option) 26

H

Habash, Nizar Y. 42
`\hbox` 291, 574, 575, 583, 587
`\Height` 308, 328, 336
`\hfill` 265, 291
Hosny, Khaled 5
`\hskip` .. 577, 578, 579, 585
`\hspace` 315, 321

I

`\if@inlabel` 527, 550
`\if@newlist` 533
`\if@nmbrrlist` 568
`\if@nbreak` 534, 559
`\if@noitemarg` 566
`\if@noparitem` 524
`\if@pkg@export` 9,
 234, 269, 438, 444, 455
`\ifal@verse@delim` ... 313
`\ifal@verse@export` ..
 235, 269, 456
`\ifal@verse@utf`
 251, 253, 288
`\ifal@warp@bayt`
 ... 272, 307, 327, 335
`\ifbool`
 404, 410, 519, 572, 574
`\IfBooleanTF`
 51, 63, 69, 80, 84,
 114, 294, 304, 439, 442
`\ifdim` 580
`\ifekd@state`
 .. 295, 457, 468, 477

<code>\ifhmode</code>	530	211, 238, 241, 242,	metre	26
<code>\IfNoValueTF</code>		299, 406, 407, 412,	mode	26
.....	75, 312, 356, 421	413, 416, 419, 436,	novoc	5, 8
<code>\ifnum</code>	367, 501	440, 441, 447, 448,	trans	5, 8
<code>\IfSubStr</code>	240	449, 464, 465, 466	utf	26
<code>\ifvmode</code> ..	116, 131, 141, 151	luaotfload (package)	voc	5, 8
<code>\ifvoid</code>	553	38	width	26
<code>\ignorespaces</code> ...	522, 586	<code>\luastring</code>	<code>\overline</code>	57
<code>\indent</code>	528	300, 406, 407, 412, 413		
<i>Information and Documen-</i>		<code>\luastringN</code>		
<i>tation - Romaniza-</i>		... 45, 52, 53, 64,	P	
<i>tion of the Arabic</i>		66, 70, 72, 110, 134,	<code>\PackageInfo</code>	33
<i>Alphabet for Arabic,</i>		144, 154, 162, 171,	<code>\par</code>	121, 123,
<i>Ottoman-Turkish,</i>		172, 175, 176, 183,	125, 127, 178, 215,	
<i>Persian, Kurdish,</i>		193, 203, 211, 238,	445, 450, 451, 528, 531	
<i>Urdu and Pushto</i> .	10	241, 242, 416, 419,	paralist (package) ...	48, 50
<i>Introduction to Arabic Nat-</i>		436, 440, 441, 448, 465	<code>\pardir</code>	
<i>ural Language Pro-</i>		<code>\luastringO</code> 135, 136, 137,	37, 39, 123, 126, 181,	
<i>cessing</i>	42	145, 146, 147, 155,	191, 201, 210, 424, 426	
<code>\itemindent</code>	508, 554, 577	156, 157, 163, 164,	<code>\parindent</code>	516
<code>\itemsep</code>	543	165, 184, 185, 186,	<code>\parsep</code>	515
<code>\itshape</code>	96	194, 195, 196, 204,	<code>\parshape</code>	521
		205, 206, 212, 213, 214	<code>\parskip</code>	515, 539
			<code>\penalty</code>	557
J		M	polyglossia (package) ..	4, 64
<code>\jobname</code>	345	<code>\makebox</code>	<code>\presetkeys</code>	230
		310,	<code>\prname</code>	39, 112
K		313, 323, 330, 333, 338	<code>\prname*</code>	39, 53, 112
<code>\kern</code>	291, 554	<code>\makelabel</code> ..	<code>\ProcessOptionsX</code>	15
KOMA-script (package) ..	45	511, 572, 573	<code>\providebool</code>	30, 292
Krüger, Marcel	32	<code>\marginpar</code>	<code>\ProvidesPackage</code> ...	498
		422, 423		
L		<code>\mbox</code>		
<code>\labelsep</code>	579, 585	57	Q	
<code>\labelwidth</code>	578, 580, 583	metre (option)	quoting (package)	47
Lagally, Klaus	3, 19	26	quran (package)	51
Lane, Edward William ...	18	<code>\MkArbBreak</code>		
<code>\lastbox</code>	552	11, 46, 50	R	
<code>\leaders</code>	55, 59	<code>\MkArbBreak*</code>	<code>\refstepcounter</code>	569
<code>\leavevmode</code>	116,	47, 50	reledmac (package)	51
131, 141, 151, 291, 296		mode (option)	<code>\RequireLuaTeX</code>	2
<code>\leftmargin</code>	518, 520	26	<code>\RequirePackage</code>	1,
lineno (package)	28		3, 20, 21, 22, 23, 24,	
<code>\linewidth</code> ..	517, 518, 521	N	25, 26, 27, 28, 29, 432	
<code>\list</code>	500	nameauth (package)	<code>\reset@font</code>	42, 43
<code>\listparindent</code> ..	507, 516	39	<code>\rightmargin</code> .	506, 517, 519
<code>\long</code>	587	<code>\newarbookmark</code>	<code>\RL</code>	45, 47, 492, 493
<code>\LR</code>	41, 45, 368,	<code>\NewDocumentEnvironment</code>	<code>\RLfootnote</code>	45, 424
370, 373, 431, 434, 493			<code>\rmfamily</code>	48, 49, 94
<code>\LRfootnote</code>	45, 424	120,	<code>\romannumeral</code>	509
<code>\LRmarginpar</code>	45, 420	124, 177, 231, 443, 452		
lua-ul (package)	32, 56	<code>\newfontfamily</code>	S	
<code>\luadirect</code>	10, 12,	36	<code>\savenotes</code>	305
31, 45, 52, 53, 64,		<code>\newlength</code> ...	<code>\sbox</code>	573
66, 70, 72, 110, 133,		216, 217, 218	scrextend (package) ..	45, 65
143, 153, 162, 171,		<code>\newunderlinetype</code> .	<code>\SetArab</code>	11, 87
175, 182, 192, 202,		55, 59	<code>\SetArbDflt</code>	6, 78
		<code>\NoArbUp</code>		
		40, 102		
		<code>\noindent</code>		
		249		
		novoc (option)		
		5, 8		
		O		
		<code>\oarg@false</code>		
		276, 280		
		<code>\oarg@true</code>		
		275, 278		
		options:		
		color		
		26		
		delim		
		26		
		export		
		5, 26, 52		
		fullvoc		
		5, 8		
		gutter		
		26		

<code>\SetArbDflt*</code>	6, 78	<code>\spewnotes</code>	340	<code>\txtrans</code>	114 , 115
<code>\SetArbEasy</code>	6, 78	<code>\sRLbox</code>	572, 587	U	
<code>\SetArbEasy*</code>	6, 78	<code>\stepcounter</code>	297	<code>\Uc</code>	111
<code>\SetArbNumbers</code>	23 , 44	<code>\StretchBayt</code>	28, 272 , 274 , 282	<code>\uc</code>	38 , 109 , 111 , 114
<code>\SetArbOutSuffix</code>	52, 435	T		<code>\unhbox</code>	576, 583
<code>\SetArbUp</code>	40, 102	<code>\textcolor</code>	75	<code>\unskip</code>	531
<code>\setbox</code>	552, 574, 575, 587	<code>\textdir</code> 38, 40, 42, 43,		<code>utf</code> (option)	26
<code>\SetDefaultIndex</code>	342	48, 49, 117 , 119 , 123 ,		V	
<code>\SetHemistichDelim</code>	26, 272 , 290	126, 132 , 142 , 152 ,		<code>voc</code> (option)	5, 8
<code>\SetIndexMode</code>	351	161, 181 , 191 , 201,		<code>\vrule</code>	55, 59
<code>\SetInputScheme</code> 42, 76 , 170		210, 422 , 423 , 425 , 427		W	
<code>\setkeys</code>	232,	<code>\textsuperscript</code>	102	<code>\wd</code>	580
249, 263 , 267 , 364 , 453		<code>\textwidth</code>	219	<code>width</code> (option)	26
<code>\setlength</code> 219 , 220 , 221 , 222		<code>\the</code>	509	<code>Wright, W. LL.D</code>	5–8,
<code>\setLR</code>	39 ,	<i>The Ekdosis package</i>	28	13–18, 23 , 24 , 39 , 50	
46, 251 , 254 , 256 , 487		<i>The Lua-ul package</i>	32	X	
<code>\SetMaghribi</code>	12, 87	<code>\theekd@lab</code>	298, 300	<code>xcolor</code> (package)	33
<code>\setRL</code>	37, 46,	<code>\thefootnotemark</code> 431, 434		<code>\xleaders</code>	291
251, 252 , 254 , 256 , 487		<code>trans</code> (option)	5, 8	Z	
<code>\SetTranslitConvention</code>		<i>Transliteration der arabis-</i>		<code>\z@</code>	506,
.	38, 99	<i>chen Schrift, Die</i>		507, 508, 552, 553, 557	
<code>\SetTranslitFont</code>	38, 94		10, 37, 40	<code>\zlabel</code>	298
<code>\SetTranslitStyle</code>	38, 96 , 113	<code>txarab</code> (env.)	44, 120	<code>\zref@extract</code>	300
<code>\space</code>	282, 491	<code>txarab</code> (environment)	51		
		<code>txarabtr</code> (env.)	124		
		<code>\txarb</code>	44, 115 , 288		