

The arabluatex package v1.19 – 2020/03/15

Robert Alessi

alessi@robertalessi.net

Contents

License and disclaimer	3	4.3 Special orthographies	21
1 Introduction	3	The name of God 21 The	
1.1 arabluatex is for LuaL ^A T _E X	5	conjunctive الذي 21	
2 The basics of arabluatex	5	4.4 Quoting	22
2.1 Activating arabluatex	5	novoc 22 voc 23 fullvoc	
Font setup 5		23	
2.2 Options	6	4.4.1 Quoting the <i>hamzah</i>	23
2.2.1 Classic contrasted with		4.5 The ‘pipe’ character ()	24
modern typesetting of Arabic	6	4.6 Putting back on broken contextual	
2.3 Typing Arabic	8	analysis rules	24
2.3.1 Local options	9	4.7 Stretching characters: the <i>taṭwīl</i>	26
3 Standard ArabT_EX input	9	4.8 Digits	26
3.1 Consonants	9	4.8.1 Numerical figures	26
3.2 Additional characters	11	4.8.2 The <i>abjad</i>	26
3.3 Vowels	11	4.9 Additional characters	27
3.3.1 Long vowels	11	4.10 Arabic emphasis	28
3.3.2 Short vowels	12	4.10.1 Underlining words or num-	
4 arabluatex in action	13	bers	28
4.1 The vowels and diphthongs	13	5 Arabic poetry	28
Short vowels 13 Long vowels		Scaling and distortion of	
13 ‘ <i>alif maqṣūrah</i> 13		characters 31 Footnotes 31	
‘ <i>alif otiosum</i> 14 ‘ <i>alif maḥdū-</i>		Line numbering 31	
<i>fah</i> and defective <i>ū</i> , <i>ī</i> 14		5.1 Example	31
Silent و/ي 14 ‘ <i>Amr^{un}</i> , and		6 Special applications	32
the silent , 14 <i>tanwīn</i> 14		Linguistics 32 Brackets 33	
4.2 Other orthographic signs	15	Additional Arabic marks 33	
<i>tā’ marbūṭah</i> 15 <i>hamzah</i> 15		The ‘Zero width joiner’ char-	
<i>maddah</i> 17 <i>ṣaddah</i> 17 The		acter (U+200D) 35	
definite article and the ‘ <i>alif^u</i>		6.1 The Qur’ān	35
‘ <i>l-waṣlⁱ</i> 18		Caveat 36	
		7 Color	36

7.1	Tricks of the trade	37	11.2	Environments	52
	Diacritics 37 <i>tanwīn</i> 39		11.2.1	Lists	53
	<i>waṣlah</i> and <i>maddah</i> 39 <i>ṣad-</i>			رموز الكتاب 53 Caveat 55	
	<i>dah</i> 40 The definite article		11.3	csquotes	55
	and the euphonic <i>tašdid</i> 40		11.4	Two-argument special commands .	55
	<i>hamzah</i> 40			textcolor 55 reledmac 56	
8	Transliteration	41	11.5	quran	56
	Convention 41 Style 41		12	Exporting Unicode Arabic to an external	
	Font 41 Proper names 42			file	57
	Hyphenation 42 ‘Long’ pro-		12.1	Commands and environments . . .	57
	per names 42 Proper names			export global option 57	
	outside Arabic environments			Exporting running para-	
	42			graphs 57 Appending words	
8.1	Additional note on dmg convention .	43		or commands to the exter-	
	<i>īrāb</i> boundaries 44 Discar-			nal file only 58 Exporting	
	ding the <i>īrāb</i> 44 Uncertain			Arabic poetry 58	
	short vowels 45		12.2	Nested Arabic environments	58
8.2	Examples	45	12.3	Further processing of Unicode con-	
9	Buckwalter input scheme	45		verted files	59
	‘base’, ‘xml’ and ‘safe’ schemes		13	Future work	59
	46 Transliteration 47		14	Implementation	59
10	Unicode Arabic input	48	References	73	
11	L ^A T _E X Commands in Arabic environments 48		Change History	74	
	General principle 48		Index	75	
11.1	New commands	50			

List of Tables

1	ArabTeX consonants	10	6	Additional Arabic codings	27
2	ArabTeX additional characters . .	11	7	Additional Arabic marks	33
3	ArabTeX long vowels	11	8	ArabTeX diacritics for \arbcolor .	38
4	ArabTeX short vowels	12	9	Buckwalter scheme	46
5	“Quoted” <i>hamzah</i>	24			

Abstract

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

License and disclaimer

License applicable to the software

arabluatex — Processing Arab \TeX notation under Lua \LaTeX .

Copyright © 2016–2020 Robert Alessi

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

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

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

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


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

This release of arabluatex consists of the following source files:

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

License applicable to this document

Copyright © 2016–2020 Robert Alessi

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

1 Introduction

In comparison to Prof. Lagally’s outstanding Arab \TeX ,¹ ArabLua \TeX is at present nothing more than a modest piece of software. Hopefully—if I may say so—it will eventually provide all of its valuable qualities to the Lua \LaTeX users.

¹See <http://ctan.org/pkg/arabtex>

arabtex dates back to 1992. As far as I know, it was then the first and only way to typeset Arabic texts with T_EX and L^AT_EX. To achieve that, arabtex provided—and still does—an Arabic font in *Nashī* style and a macro package that defined its own input notation which was, as the author stated, “both machine, and human, readable, and suited for electronic transmission and e-mail communication”.² Even if the same can be said about Unicode, ArabT_EX ASCII input notation still surpasses Unicode input, in my opinion, when it comes to typesetting complex documents, such as scientific documents or critical editions where footnotes and other kind of annotations can be particularly abundant. It must also be said that most text editors have trouble in displaying Arabic script connected with preceding or following L^AT_EX commands: it often happens that commands seem misplaced, not to mention punctuation marks, or opening or closing braces, brackets or parentheses that are unexpectedly displayed in the wrong direction. Of course, some text editors provide ways to get around such difficulties by inserting invisible Unicode characters, such as LEFT-TO-RIGHT or RIGHT-TO-LEFT MARKS (U+200E, U+200F), RTL/LTR “embed” characters (U+202B, U+202A) and RLO/LRO “bidi-override” characters (U+202E, U+202D).³ Nonetheless, it remains that inserting all the time these invisible characters in complex documents rapidly becomes confusing and cumbersome.

The great advantage of ArabT_EX 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. ArabT_EX notation is a way of encoding Arabic language, just as T_EX “mathematics mode” is a way of processing code to display mathematics. As such, not only does it allow greater control over typographical features, but it also can be processed in several different ways: so without going into details, depending on one’s wishes, ArabT_EX 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.

²Lagally (2004, p. 2).

³Gáspár Sinai’s Yudit probably has the best Unicode support. See <http://www.yudit.org>.

1.1 arabluatex is for Lua^LA^TE^X

It goes without saying that arabluatex requires Lua^LA^TE^X. T^EX and L^AT^EX have arabtex, and X^YL^AT^EX has arabxetex. Both of them are much more advanced than arabluatex, as they can process a number of different languages,⁴ whereas arabluatex can process only Arabic for the time being. More languages will be included in future releases of arabluatex.

In comparison to arabxetex, arabluatex works in a very different way. The former relies on the **TECKit** engine which converts ArabT^EX input on the fly into Unicode Arabic script, whereas the latter passes ArabT^EX input on to a set of Lua functions. At first, L^AT^EX commands are taken care of in different ways: some, as `\emph`, `\textbf` and the like are expected to have Arabic text as arguments, while others, as `\LR`, for “left-to-right text”, are not. Then, once what is Arabic is carefully separated from what is not, it is processed by other Lua functions which rely on different sets of correspondence tables to do the actual conversion in accordance with one’s wishes. Finally, Lua returns to T^EX the converted strings—which may in turn contain some other ArabT^EX input yet to be processed—for further processing.

2 The basics of arabluatex

2.1 Activating arabluatex

arabluatex is loaded the usual way:

```
\usepackage{arabluatex}
```

The only requirement of arabluatex is Lua^LA^TE^X; it will complain if the document is compiled with another engine. That aside, arabluatex does not load packages such as polyglossia. Although it can work with polyglossia, it does not require it.

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

```
\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 font⁵ like so:—

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

⁴To 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.

⁵Hosny (2017).

REM. 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 (1896, i. 14 C–D). The placement of the *kasrah* above the consonant may be obtained by selecting the ss05 feature of the Amiri font, like so:—⁶

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

Other Arabic fonts may behave differently.

2.2 Options

arabluatex may be loaded with five global options, the first four of which are mutually exclusive and may be overridden at any point of the document (see below section 2.3.1 on page 9):

voc

Default

In this mode, which is the one selected by default, every short vowel written generates its corresponding diacritical mark: *ḍammah* (َ), *faṭḥah* (ِ) and *kasrah* (ِ). If a vowel is followed by N, viz. $\langle uN, aN, iN \rangle$, then the corresponding *tanwīn* (ُ, ِ, ِ or ِ) is generated. Finally, $\langle u, a, i \rangle$ at the commencement of a word indicate a “connective *ʿalif*” (*ʿalifu ʿl-waṣṣah*), but voc mode does not show the *waṣṣah* above the *ʿalif*; instead, the accompanying vowel may be expressed at the beginning of a sentence (أَ).

fullvoc

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

novoc

None of the diacritics is showed in novoc mode, unless otherwise specified (see “quoting” technique below section 4.4 on page 22).

trans

This mode transliterates the ArabTeX input into one of the accepted standards. At present, three standards are supported (see below section 8 on page 41 for more details):

dmg *Deutsche Morgenländische Gesellschaft*, which is selected by default;

loc *Library of Congress*;

arabica *Arabica*.

More standards will be included in future releases of arabluatex.

export

export=true|false

Default: false

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 section 12 on page 57 for more information.

2.2.1 Classic contrasted with modern typesetting of Arabic

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

⁶See the documentation of amiri, Hosny (2017, p. 6).

New feature
v.1.13

New feature
v.1.2

- ‘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} samā, jA'a جَاءَ ^a jā'a, yatasA'alUna يَتَسَاءَلُونَ ⁷ yatasā'alūna
(see on page 17 for further details).

- The euphonic *tašdīd* is generated (see on page 17).
- In *fullvoc* mode, the *sukūn* is expressed.
- In such words as ظَمًا, شَيْئًا and the like, the *hamzah* alone is not written over the letter *yā'* with no diacritical points below as in ظَمًا, شَيْئًا, but over a horizontal stroke placed in the continuation of the preceding letter.

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

New feature
v1.4.4

\SetArbEasy
\SetArbEasy*

New feature
v1.6

\SetArbDflt
\SetArbDflt*

Such refinements as ‘classic’ *maddah* may be discarded by the \SetArbEasy command, either globally in the preamble or locally at any point of the document. The difference between \SetArbEasy and its ‘starred’ version \SetArbEasy* is that the former keeps the *sukūn* that is generated by the *fullvoc* mode, while the latter further takes it away. Default ‘classic’ rules may be set back at any point of the document with the \SetArbDflt command. Assimilation rules laid on item (b) on page 18 may also be applied by the ‘starred’ version of this command \SetArbDflt* either in the preamble or at any point of the document.⁸ Examples follow:—

(a) \SetArbDflt:

- i. voc سَمَاءُ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- ii. fullvoc سَمَاءُ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- iii. trans wa-māta 'stisqā^{an} qabla 'an yutimma kitāba-hu fī nuǧūmⁱ 's-samāⁱ

(b) \SetArbDflt*:

- i. voc سَمَاءُ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- ii. fullvoc سَمَاءُ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- iii. trans wa-māta 'stisqā^{an} qabla 'ay yutimma kitāba-hu fī nuǧūmⁱ 's-samāⁱ

(c) \SetArbEasy:

- i. voc سَمَاءُ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- ii. fullvoc سَمَاءُ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- iii. trans wa-māta 'stisqā^{an} qabla 'an yutimma kitāba-hu fī nuǧūmⁱ 's-samāⁱ

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

⁸For an example, see section 5.1 on page 31.

(d) `\SetArbEasy*`:

- i. voc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- ii. fullvoc وَمَاتَ اسْتِسْقَاءَ قَبْلَ أَنْ يُتِمَّ كِتَابَهُ فِي نُجُومِ السَّمَاءِ
- iii. trans *wa-māta 'stisqā^{an} qabla 'an yutimma kitāba-hu fī nuǧūmⁱ '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 (1896, i. 1 A):— Arabic, like Hebrew and Syriac, is written and read from right to left. The letters of the alphabet (حُرُوفُ الْحِجَاءِ, حُرُوفُ الْمُجَمِّمِ, or حُرُوفُ الْمَجَائِزِ, or حُرُوفُ الْمُجَمِّمِ) are twenty-eight in number and are all consonants, though three of them are also used as vowels (see § 3).

The following example comes from Wright (1896, i. 213 C):—

```
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}
```

XVI. فَوَاعِلُ*.

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

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

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


like so:—

```

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

```

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

2.3.1 Local options

As seen above in section 2.2 on page 6, 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:—

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

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

3 Standard ArabTeX input

3.1 Consonants

Table 1 gives the ArabTeX equivalents for all of the Arabic consonants.

Letter	Transliteration ⁹			ArabTeX notation
	dmg	loc	arabica	
أ ¹⁰	'u 'a 'i	'u, 'a, 'i	'u, 'a, 'i	'u or 'a or 'i
ب	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>
ت	<i>t</i>	<i>t</i>	<i>t</i>	<i>t</i>
ث	<i>ṭ</i>	<i>th</i>	<i>ṭ</i>	<i>_t</i>
ج	<i>ǧ</i>	<i>j</i>	<i>ǧ</i>	<i>ˆg</i> or <i>j</i>
ح	<i>ḥ</i>	<i>ḥ</i>	<i>ḥ</i>	<i>.h</i>
خ	<i>ḫ</i>	<i>kh</i>	<i>ḫ</i>	<i>_h</i> or <i>x</i>
د	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>
ذ	<i>ḍ</i>	<i>dh</i>	<i>ḍ</i>	<i>_d</i>
ر	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>
ز	<i>z</i>	<i>z</i>	<i>z</i>	<i>z</i>
س	<i>s</i>	<i>s</i>	<i>s</i>	<i>s</i>
ش	<i>š</i>	<i>sh</i>	<i>š</i>	<i>ˆs</i>
ص	<i>ṣ</i>	<i>ṣ</i>	<i>ṣ</i>	<i>.s</i>
ض	<i>ḍ</i>	<i>ḍ</i>	<i>ḍ</i>	<i>.d</i>
ط	<i>ṭ</i>	<i>ṭ</i>	<i>ṭ</i>	<i>.t</i>
ظ	<i>ẓ</i>	<i>ẓ</i>	<i>ẓ</i>	<i>.z</i>
ع	<i>ʿ</i>	<i>ʿ</i>	<i>ʿ</i>	<i>`</i>
غ	<i>ǧ</i>	<i>gh</i>	<i>ǧ</i>	<i>.g</i>
ف	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>
ق	<i>q</i>	<i>q</i>	<i>q</i>	<i>q</i>
ك	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>
ل	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>
م	<i>m</i>	<i>m</i>	<i>m</i>	<i>m</i>
ن	<i>n</i>	<i>n</i>	<i>n</i>	<i>n</i>
ه	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>
و	<i>w</i>	<i>w</i>	<i>w</i>	<i>w</i>
ي	<i>y</i>	<i>y</i>	<i>y</i>	<i>y</i> ¹¹
ة	<i>ah</i>	<i>ah</i>	<i>a</i>	<i>T</i>

Table 1: Standard ArabTeX (consonants)

REM. *a*. Please note that in all cases of elision, the *'alifu 'l-waṣli* is expressed only by the vowel that accompanies the omitted *hamzah*: $\langle u, a, i \rangle$ as in *wa-inhazama* واِنْهَازَمَ *wa-'nhazama*. For more details on the definite article and the *'alifu 'l-waṣli* see section 4.2 on page 18.

⁹See below section 8 on page 41.

¹⁰See below, Rem. *a*. For *'alif* as a consonant, see Wright (1896, i. 16 D). The *hamzah* itself is encoded $\langle ' \rangle$ and may be followed by either $\langle u, a \rangle$ or $\langle i \rangle$. See below section 4.2 on page 15.

¹¹For the letter ي with no diacritical points below, see Rem. *b*. below.

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

REM. *b.* The letter ي with two points below, $\text{يَاءُ الْمُتَّاءُ مِنْ تَحْتِهَا}$, may also be written without diacritical points as ي . 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 $\text{qay}^{\text{'un}}$, ^saY'uN $\text{^say}^{\text{'un}}$, ^saY'aN $\text{^say}^{\text{'an}}$.

The same result may be achieved by encoding this letter as .y , like so: qa.y'uN $\text{qay}^{\text{'un}}$, ^sa.y'uN $\text{^say}^{\text{'un}}$, ^sa.y'aN $\text{^say}^{\text{'an}}$.

3.2 Additional characters

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

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

Table 2: Standard ArabTeX (additional characters)

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

3.3 Vowels

3.3.1 Long vowels

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

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

Table 3: Standard ArabTeX (long vowels)

¹²See below section 8 on page 41.

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

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

¹⁵See note 14.

¹⁶See below section 8 on page 41.

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

Table 3: Standard ArabTeX (long vowels)

REM. *a*. The long vowels \bar{a} , \bar{u} , \bar{i} , otherwise called *hurūf^u 'l-maddⁱ*, the letters of prolongation, involve the placing of the short vowels *a*, *u*, *i* before the letters ا, و, ي respectively. `arabluatex` does that automatically in case any from `voc`, `fullvoc` or `trans` modes is selected e.g. قَالَ *qāla*, قِيلَ *qīla*, يَقُولُ *yaqūlu*.

REM. *b*. Defective writings, such as ا, *al-'alif^u 'l-mahdūfat^u*, or defective writings of \bar{u} and \bar{i} are encoded _a _u and _i respectively, e.g. _d_alika ذَلِكَ, al-mal_a'ikaT-u 'l-ra.hm_an-u الْمَلِكُ الرَّحْمَنُ, .hu_dayfaT-u bn-u 'l-yamAn_i الْحَيَّانُ بْنُ حَظِيْفَةَ for *Hudayfat^u bn^u 'l-Yamānī*, etc.

REM. *c*. The letter ي with two points below, أَلْيَاءُ الْمُنَاةِ 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: لي iY لِي *lī*, يامسِي يَمْسِي *yamšī*.

3.3.2 Short vowels

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

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

Table 4: Standard ArabTeX (short vowels)

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

¹⁸= *al-'alif^u 'l-maqṣūrat^u*.

¹⁹See below section 8 on page 41.

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

However, it may be wise to always write the vowels, leaving to the various modes provided by `arabluatex` to take care of showing or not showing the vowels.

That said, there is no need to write the short vowels *fathah*, *ḍammah* or *kasrah* except in the following cases:—

- at the commencement of a word, to indicate that a connective *ʿalif* is needed, with the exception of the article (see below section 4.4 on page 22);
- when `arabluatex` needs to perform a contextual analysis to determine the carrier of the *hamzah*;
- in the various transliteration modes, as vowels are always expressed in romanized Arabic.

4 arabluatex in action

4.1 The vowels and diphthongs

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

`_halaqa` (or `xalaqa`) خَلَقَ *halaqa*, `^samsuN` شَمْسٌ *šams^{un}*, `karImuN`
كَرِيمٌ *Karīm^{un}*.
`bi-hi` بِه *bi-hi*, `'aqi.tuN` أَقِيْتُ *'aqiṭ^{un}*.
`la-hu` لَهُ *la-hu*, `.hujjaTuN` هُجِّجَتْ *huġġat^{un}*.

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

`qAla` قَالَ *qāla*, `bI`a` بَاعَ *bī'a*, `.tUruN` طُورَ *tūr^{un}*, `.tInuN` طِينَ *tīn^{un}*,
`murU'aTuN` مُرُوْءَةٌ *murū'at^{un}*.

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

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

²⁰See below section 4.4 on page 22.

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

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

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

al-l_ah-u اللهُ *al-lāh^u*, 'il_ahuN إِلَهٌ *'ilāh^{un}*.
 al-ra.hm_an-u الرَّحْمَنُ *ar-raḥmān^u*, l_akin لَكِنْ *lākin*, h_ahunA هَهُنَا *hāhunā*,
 .hunayn-u bn-u 'is.h_aq-a حَنِينُ 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 **ة** are usually written **ة** or **ة** instead of **ة**: see Wright (1896, i. 12 A). *arabluatex* preserves that particular writing; the same applies to words ending in **ة** for **ة**. Long vowels ⟨U, I⟩ shall receive no *sukūn* after a *'alif maḥḍūfah* and are discarded in *trans* mode:

.hay_aUTuN حَيَوة *ḥayāt^{un}*, .sal_aUTuN صَلَوة *ṣalāt^{un}*, mi^sk_aUTuN مَشْكُوة *miš-kāt^{un}*, tawr_aITuN تَوْرِيَّة *tawrāt^{un}*.
 And so also: al-rib_aIT-u الرِّبِيَّةُ *ar-ribāt^u*.

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

`amruNU عَمْرُو *'amr^{un}*, `amraNU عَمْرُوا *'amr^{an}*, `amriNU عَمْرٍو *'amrⁱⁿ*.

When the *tanwīn* falls away (Wright 1896, i. 249 B): `amr-uU bn-u mu.hammadIN عَمْرُو بْنُ مُحَمَّدٍ *Amr^u bn^u Muḥammadⁱⁿ*, mu.hammad-u bn-u `amr-iU bn-i_hAlidiN عَمْرٍو بْنُ خَالِدٍ *Muḥammad^u bn^u Amrⁱ bnⁱ Ḥālidⁱⁿ*.

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. *arabluatex* deals with special cases, such as َ taking an ِ after all consonants except ة, and *tanwīn* preceding ى as in هَدًى, which is written ⟨aN_A⟩ or ⟨aN_Y⟩:

mAluN مَال $māl^{un}$, bAbaN أَبَا $bāb^{an}$, madInaTaN مَدِينَة $madīnat^{an}$, bintiN بنت $bint^{in}$ maqhaN_A مَقْهَى $maqha^n$, fataNY فَتَى $fata^n$.
arabluatex is aware of special orthographies: $\hat{\text{say}}'uN$ شَيْءٌ say^{un} ,
 $\hat{\text{say}}'aN$ شَيْئًا say^{an} , $\hat{\text{say}}'iN$ شَيْءٍ say^{in} .

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

al-qA.dI الْقَاضِي $al-qāḍī$, qA.diyaN قَاضِيًا $qāḍiy^{an}$, qA.diNI قَاضٍ $qāḍi^n$.

4.2 Other orthographic signs

tā' marbūṭah It is written $\langle T \rangle$:

madInaTuN مَدِينَة $madīnat^{un}$, madInaTaN مَدِينَة $madīnat^{an}$, madInaTiN مَدِينَة $madīnat^{in}$.

hamzah It is written $\langle ' \rangle$, its carrier being determined by contextual analysis. In case one wishes to bypass this mechanism, he can use the “quoting” feature that is described below in section 4.4 on page 22.

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

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

hamzah followed by the long vowel ي is encoded ' _I: ' _ImAnuN إِيْمَانٌ $'īmān^{un21}$.

Middle hamzah: xA.ti'-Ina خَاطِبِينَ $hāṭi'ina$, ru'UsuN رُءُوسٌ $ru'ūs^{un}$,
xa.tI'aTuN خَاطِبَةٌ $hāṭi'at^{un}$, 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 تَوَّعَّمٌ $taw'am^{un}$,
ta'xIruN تَأْخِيرٌ $ta'hīr^{un}$, ta'axxara تَأَخَّرَ $ta'ahhara$, ji'tu-ka جِئْتُكَ $gi'tu-ka$,
qA'iluN قَائِلٌ $qā'il^{un}$, .hIna'i_diN هَيِّنَا $'id^{in}$ حَيَّنَدَ, hay'aTuN هَيْئَةً $hay-at^{un}$,
hay'AtuN هَيَّاتٌ $hay'ātun$.

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

From Wright (1896, i. 14 B):— All consonants, whatsoever, not even *ʿalif hèmzatum* excepted, admit of being doubled and take *tašdid*. Hence we speak and write ra' 'AsuN رَأْسُ *ra'ās^{un}*, sa' 'AluN سَأَلُ *sa'al^{un}*, na' 'AjuN نَأَجُ *na'āğ^{un}*.

Final *hamzah*: xa.ta'uN خَطَأُ *ḥaṭa^{un}*, xa.ta'aN خَطَأٌ *ḥaṭa'an*, xa.ta'iN خَطِئًا *ḥaṭa'in*, 'aqra'u أَقْرَأُ *'aqra'u*, taqra'Inا تَقْرَأُ *taqra'ina*, taqra'Una تَقْرَأُونَ *taqra'una*, yaqra'na يَقْرَأُ *yaqra'na*, yaxba'Anي يَخْبَأُ *yaxba'ani*, xaba'A خَبَأَ *ḥaba'a*, xubi'a خُبِئَ *ḥubi'a*, xubi'UA خُبِئُوا *ḥubi'u*, jA'a جَاءَ *ğā'a*, ridA'uN رَدَأَ *ridā^{un}*, ridA'aN رَدَأَ *ridā'an*, jI'a جِئَ *ğī'a*, radI'iN رَدِئَ *radī'in*, sU'uN سُوءُ *sū^{un}*, .daw'uN ضَوْءُ *daw^{un}*, qay'iN قَيْءُ *qay'in*, ^sifA'I شِفَائِي *šifā'ī*, man^sa'I مَنْشِئِي *manša'ī*, nisA'uN نِسَاءُ *nisā^{un}*, nisA'u-hu نِسَاءُهُ *nisā'u-hu*, nisA'i-hi نِسَائِهِ *nisā'i-hi*, nisA'I نِسَائِي *nisā'ī*. ^say'uN شَيْءُ *šay^{un}*, ^say'aN شَيْئًا *šay'an*, ^say'iN شَيْءٍ *šay'in*, al-^say'-u أَشْيَاءُ *aš-šay'u*, 'a^syA'-u أَشْيَاءُ *ašyā'u*, 'a^syA'-a أَشْيَاءُ *ašyā'a*, .zim'aN ظِمًّا *ẓim'an*, radI'aN رَدِئًا *radī'an*.

taḥfīf^u 'l-hamzat: if the *hamzah* has *ğazmah* and is preceded by *ʿalif hamzatum*, it must be changed into the letter of prolongation that is homogeneous with the preceding vowel; hence: 'a'mana أَمَنَ *āmana*, 'u'minu أَوْمِنُ *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 the following page.

Imperatives of verbs that have the *hamzah* as the first radical are other cases of *taḥfīf^u 'l-hamzat*: i'sir إِيسِرْ *īsir*, i'_dan اِدْنْ *īdan*, u'mul اُوْمُلْ *ūmul*. arabluatex also provides ways of encoding those words when the initial *ʿalif* comes into *waṣl*, so as to make the *ʿalif waṣl* fall away when preceded by وَ or فَ: wa-'sir وَأَسِرْ *wa-'sir*, fa-'_dan فَادْنْ *fa-'dan*, fa-'ti فَاتْ *fa-'ti*, wa-'tamirUA وَأَتَمِّرُوا *wa-'tamirū*; or be retained outside the imperative, as in fa-i'tazarat فَاتَزَرَّتْ *fa-'tazarat*, ba`da i'tilAfIn بَعْدَ اِتِّلَافٍ *ba`da 'tilāf'in*.

The strange spelling of *mi'at^{un}*: mi'aTuN مَائَة *mi'at^{un}*,
 mi'atAni مَائَتَانِ *mi'atāni*, mi'atayni مَائَتَيْنِ *mi'atayni*, mi'Una مِئُونِ *mi'ūna*,
 mi'AtuN مِئَاتِ *mi'ātun*, mi'aN_A مَائِي *mi'āⁿ*. Of course, the ‘pipe’ character
 can be used to prevent this rule from being applied (see section 4.5 on
 page 24): mi'a|TuN مِئَة *mi'at^{un}*.

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

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

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

'is'AduN إِسَادُ *is'ād^{un}*, 'AkilUna أَكِلُونُ *ākilūna*, 'a'manna آمَنَّا *āmannā*,
 al-qur'An-u الْقُرْآنُ *al-qur'ān^u*.
 jA'a جَاءَ *ğā'a*, yatasA'alUna يَتَسَاءَلُونَ *yatasā'alūna*, ridA'uN رِدَاءُ *ridā^{un}*,
 xaba'A خَبَا *ħaba'ā*, yaxba'Ani يَخْبَانُ *yahba'āni*.

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

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

`allaqa عَلَّقَ *allaqa*, mAdduN مَادَّ *mādd^{un}*, 'ammara أَمَّرَ *ammara*, murruN
 مُرَّرَ *murr^{un}*.

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

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

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

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

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

Note the absence of *sukūn* above the passed over ن in the following examples, each of which is accompanied by a consistent transliteration: min rabbi-hi مِنْ رَبِّي *mir rabbi-hi*, min layliN مِنْ لَيْلٍ *mil 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 section 2.2.1 on page 6 for further details. Other kinds of assimilations, including the various cases of *ʾidgām*, 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 1896, i. 16 B–C).

The definite article and the 'alif^u ʾl-waṣlⁱ At the beginning of a sentence, ٱ 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 6, *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 *'alifu 'l-waṣli* is expressed by the vowel that accompanies the omitted *hamzah*: ⟨u, a, i⟩.

Article: bAb-u 'l-madrasaT-i الْمَدْرَسَةُ *bāb*^u 'l-madrasatⁱ, al-maqA laT-u 'l-'_Ul_A الْمَقَالَةُ *al-maqālat*^u 'l-ūlā, al-lu.gaT-u 'l-`ara biyyaT-u اللُّغَةُ الْعَرَبِيَّةُ *al-luġat*^u 'l-`arabiyyat^u, fI .sinA`aT-i 'l-.tibb-i فِي صِنَاعَةِ الطِّبِّ *fī ṣinā`at*ⁱ 't-ṭibbⁱ, 'il_A 'l-intiqA.d-i إِلَى الْإِنْتِقَاضِ *ila*ⁱ 'l-intiqādⁱ, fI 'l-ibtidA'-i فِي الْإِبْتِدَاءِ *fī* 'l-ibtidā'ⁱ, 'abU 'l-wazIr-i أَبُو الْوَزِيرِ *abu* 'l-wazīrⁱ, fa-lammA ra'aW 'l-najm-a فَلَمَّا رَأَوْا النَّجْمَ *fa-lammā ra'awu* 'n-naġm^a.

Particles:—

- (a) *li-:* *'alif*^u *'l-waṣl*ⁱ is omitted in the article *لِ* when it is preceded by the preposition *لِ*: li-l-rajul-i لِلرَّجُلِ *li-r-raġul*ⁱ.
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-laylat*ⁱ.
(b) *la-:* the same applies to the affirmative particle *لَ*: la-l-.haqq-u لِلْحَقِّ *la-l-ḥaqq*^u.
(c) With the other particles, *'alif*^u *'l-waṣl*ⁱ is expressed: fI 'l-madInaT-i فِي الْمَدِينَةِ *fī* 'l-madīnatⁱ, wa-'l-rajul-u وَالرَّجُلُ *wa-*'r-raġul^u, bi-'l-qalam-i بِالْقَلَمِ *bi-*'l-qalamⁱ, bi-'l-ru`b-i بِالرُّعْبِ *bi-*'r-ru`bⁱ.

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

Other cases: 'awi ismu-hu أَوْ اسْمُهُ *'awi* 'smu-hu, zayduN ibn-u `amriNU زَيْدُ بْنُ عَمْرٍو *Zayd*^{uni} 'bn^u 'Amrⁱⁿ,²² `umar-u ibn-u 'l_-ha.t.tAb-i عُمَرُ بْنُ عَبْدِ اللَّهِ *Umar*^u 'bn^u 'l-Haṭṭābⁱ,²³ imru'-u 'l-qays-i اِمْرُؤُ الْقَيْسِ *Imru*^u 'l-Qaysⁱ, la-aymun-u 'l-l_ah-i لَا يَمُنُّ بِاللَّهِ *la-*'ymun^u 'l-lāhⁱ.

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

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

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

fI 'l-nAs-i 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-lA1-i ذُو الْأَعْلَالِ du
'l-i'lālⁱ, maqh_A 'l-'amIr-i مَقْهَى الْأَمِيرِ maqḥa 'l-'amīrⁱ.

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

fI `aynay 'l-malik-i فِي عَيْنِي الْمَلِكِ fī 'aynayi 'l-malikⁱ, ix^say 'l-qaw
m-a إِيخْشَايِ الْقَوْمِ iḥṣayī 'l-qawm^a, mu.s.tafaw 'l-l_ah-i مُصْطَفَوُ اللَّهِ muṣṭa-
fawu 'l-lāhⁱ.
ramaW 'l-.hijAraT-a رَمَوْا الْحِجَارَةَ ramawu 'l-ḥiġārat^a, fa-lamma ra'aW
'l-najm-a فَلَمَّا رَأَوْا النَّجْمَ fa-lammā ra'awu 'n-naġm^a.

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

'antumU 'l-kA_dib-Una أَنتُمْ الْكَاذِبُونَ antumu 'l-kādīb^{ūna}, ra'aytumu
'l-rajul-a رَأَيْتُمُ الرَّجُلَ ra'aytumu 'r-raġul^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. arabluatex takes care of that automatically:—

mu.hammaduN 'l-nabI مُحَمَّدُ النَّبِيِّ Muḥammad^{uni} 'n-nabī, salAmuN ud_hu1UA
سَلَامٌ أَدْخَلُوا salām^{unu} 'dḥulū, qa.sIdata-hu fI qatl-i \uc{'a}bI \uc{m}
uslimiN 'llatI yaqUlu fI-hA قَتَلَ ابْنُ مُسْلِمٍ الَّذِي يَقُولُ فِيهَا fī-hā
fī qatlⁱ 'Abī Muslimⁱⁿⁱ 'llatī yaqūlu fī-hā.

4.3 Special orthographies

The name of God The name of God, ﷲ, is compounded of the article ﷲ, and ﷲ (noted ﷲ with the defective *ʿalif*) so that it becomes ﷲ; then the *hamzah* is suppressed, its vowel being transferred to the ﷲ before it, so that there remains ﷲ (I refer to Lane, *Lexicon*, I. 83 col. 1). Finally, the first ﷲ is made quiescent and incorporated into the other, hence the *tašdīd* above it. As *arabluatex* never requires a solar letter to be written twice (see above, on page 17), the name of God is therefore encoded *al-l_ah-u* or *'l-l_ah-u*:—

al-l_ah-u ﷲ *al-lāh^u*, *yA*²⁴ *al-l_ah-u* ﷲ *yā al-lāh^u*, *'a-fa*²⁵-*al-l_ah-i* *la-ta.g`alanna* ﷲ ﷲ *'a-fa-al-lāhⁱ la-tag'alanna*, *bi-'l-l_ah-i* ﷲ *bi-'l-lāhⁱ*, *wa-'l-l_ah-i* ﷲ *wa-'l-lāhⁱ*, *bi-sm-i 'l-l_ah-i* ﷲ *bi-smⁱ 'l-lāhⁱ*, *al-.hamd-u* *li-l-l_ah-i* ﷲ *al-ḥamd^u li-l-lāhⁱ*, *li-l-l_ah-i 'l-qA'il-u* ﷲ *li-l-lāhⁱ 'l-qā'il^u*.

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

'a_hAfu *mina 'l-malik-i 'lla_dI* *ya.zlimu 'l-nAs-a* ﷲ ﷲ ﷲ ﷲ *'ahāfu mina 'l-malikⁱ 'lladī yaẓlimu 'n-nās^a*, *`udtu 'l-`say_h-a 'lla_dI* *huwa marI.duN* ﷲ ﷲ ﷲ ﷲ *'udtu 'š-šayh^a 'lladī huwa marād^{un}*, *mA 'aNA* *bi-'lla_dI qA'iluN* *la-ka `say'aN* ﷲ ﷲ ﷲ ﷲ *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* ﷲ ﷲ ﷲ ﷲ *'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* ﷲ ﷲ ﷲ ﷲ *fa-'innā nadkuru 'ṣ-ṣawt^{ayni} 'l-ladayni rawaynā-humā 'an Ḡaḥẓat^a*.

²⁴Note the “pipe” character ‘|’ here after *yA* and below after *fa* before footnote mark 25: it is needed by the *dmg* transliteration mode as in this mode any vowel at the commencement of a word preceded by a word that ends with a vowel, either short or long, is absorbed by this vowel viz. *ʿala ʾl-ṭarīqⁱ*. See section 4.5 on page 24 on the “pipe” and section 8 on page 41 on *dmg* mode.

²⁵See note 24.

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 اللّٰتِي *al-lātī*, al-lA'Ati²⁶ اللّٰءَاتِ *al-lā'āti*, al-lA'I اللّٰئِي *al-lā'ī*, and so forth.

4.4 Quoting

It is here referred to “quoting” after the `arabtex` package.²⁷ The “quoting” mechanism of `arabluatex` is designed to be very similar in effect to the one of `arabtex`.

To start with an example, suppose one types the following in `novoc` mode: عَلَّمَ علم الهيئة; is it عَلَّمَ, *he was taught the science of astronomy*, or عَلَّمَ, *he taught the science of astronomy*? In order to disambiguate this clause, it may be sensible to put a *dammah* above the first ع: عَلَّمَ علم الهيئة, which is achieved by “quoting” the vowel u, like so: `ullima, or, with no other vowel than the required u: `ullm.

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

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

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

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

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

- (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 جَاءَتْ هِنْدُ gā'at
Hind^{un}, ^sabThuN bi-man q"u.ti`at" qadamA-hu شبيه بمن قُطِعَتْ qadamā-hu.
شبيه بمن قُطِعَتْ qadamā-hu.

²⁶Note here the “pipe” character '|': as already stated on page 17, 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 section 4.5 on page 24.

²⁷See Lagally (2004, p. 22)

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

wa-"ust"u`mila ^uوَأَسْتَعْمَلُ *wa-'stu'mila*, huwa "inhazama ^uهُوَ أَنْهَزَمَ *huwa 'nhazama*, al-"intiqA.d-u ^uالْإِنْتِقَاضُ *al-intiqāḏ*^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 ^uقَالَ قَائِلٌ *qāla qā'il*^{un}, ibn-u 'abI 'u.saybi`aT-
"a ^uأَصْبَحَ ابْنُ أَبِي أُصَيْبَةَ *Ibn 'Abī 'Uṣaybi'at*^a.

- (b) *tanwīn*:—

madInaT"aN ^uمَدِينَةٌ *madīnat*^{an}, bAb"aN ^uبَابًا *bāb*^{an}, hud"aN_A ^uهُدًى *hudā*ⁿ,
^say'"iN ^uسَيِّئٌ *šay*ⁱⁿ.

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

- (a) *'alif*^u *'l-waṣl*ⁱ:—

fI "istiq.sA'-iN ^uفِي اسْتِقْصَاءٍ *fi 'stiqṣā*ⁱⁿ, wa-"istiq.sA'-uN ^uوَأَسْتِقْصَاءٌ
wa-'stiqṣā^{un}, qAla "uhrub fa-lan tuqtala ^uقَالَ أَهْرُبُ فَلَنْ تَقْتَلَ
'hrub fa-lan tuqtala.

- (b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala ^uقَالَ أَقْتُلْ فَلَنْ تَقْتَلَ *qāla 'qtul fa-lan tuq-*
tala, mA jA'at" mini imra'aTiN ^uمَا جَاءَتْ مِنِ امْرَأَةٍ *mā ḡā'at mini*
*'mra'at*ⁱⁿ, kam" qad" ma.dat" min" laylaTiN ^uكَأَمْ قَدْ مَضَتْ مِنْ لَيْلَةٍ *kam*
*qad maḏat min laylat*ⁱⁿ.

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 ^uالْجَمْرُ الصَّغِيرُ
al-ḡamr^u 'ṣ-ṣayfiyy^u 'llaḏī kāna bi-Qrānnūn^a.

4.4.1 Quoting the *hamzah*

As said above in section 4.2 on page 15, the *hamzah* is always written ⟨ ' ⟩, its carrier being determined by contextual analysis. “Quoting” that straight single quote character like so: ⟨ " ' ⟩ allows to determine the carrier of the *hamzah* freely, without any consideration for the context. Table 5 gives the equivalents for all the possible carriers the *hamzah* may take.

Letter	Transliteration ²⁸			ArabT _E X notation
	dmg	loc	arabica	
ء	’	’	’	’’
آ	’ā	’ā	’ā	A’’
أ	’	’	’	a’’
إ	’	’	’	u’’
ؤ	’	’	’	w’’
أ	’	’	’	i’’
ي	’	’	’	y’’

Table 5: “Quoted” *hamzah*

As one can see from table 5, the carrier of the *hamzah* is inferred from the letter that precedes the straight double quote (’). Of course, any “quoted” *hamzah* may take a short vowel, which is to be written *after* the ArabT_EX equivalent for the *hamzah* itself, namely (’). For example, ؤ is encoded <w’’a>, while ؤ is encoded <w’’’>. In the latter example, the second straight double quote encodes the *sukūn* in voc mode in accordance with the rule laid above on pages 22–23.

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

4.5 The ‘pipe’ character (|)

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

Aside from that usage, the “pipe” character is used to prevent almost any of the contextual analysis rules that are described above from being applied. Two examples have already been given to demonstrate how that particular mechanism works in note 24 on page 21 and in note 26 on page 22. One more example follows:—

bi-qran|nUn-a بِقِرَانُونْ *bi-Qrānnūn*^a, “in Crannon” (Thessaly, Greece).²⁹

As one can see, the “pipe” character between the two <n> prevents the necessary *tašdīd* rule (page 17) from being applied.

4.6 Putting back on broken contextual analysis rules

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

²⁸See below section 8 on page 41.

²⁹See more context on the previous page.


```

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^a النَّجْمَ...

^a A footnote which interferes with the contextual analysis.

According to the rule stated on page 20, the diphthong in *ra'aw* must be resolved into two simple vowels before the *'alif*^u *'l-wasl*ⁱ, as رَأَوْا النَّجْمَ.

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

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

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

```

1 \begin{arab}[fullvoc]
2   fa-lamma ra'aw\arbnnull{'l-na^gma}\LRfootnote{A footnote
3     which interferes with the contextual analysis.}
4   'l-na^gma...
5
6   qAla\LRfootnote{A footnote which interferes with the
7     contextual analysis.} \arbnnull{qAla}uhrub fa-lan tuqtala.
8
9   \uc{z}ayduN\arbnnull{ibnu}\LRfootnote{A footnote which
10     interferes with the contextual analysis.}
11   \arbnnull{zayduN}ibn-u \uc{'a}mriNU.\LRfootnote{See
12     \vref{fn:zayd-is-son}.}
13 \end{arab}
14
15 \begin{arab}[trans]
16   \uc{z}ayduN\arbnnull{ibnu}\LRfootnote{A footnote which
17     interferes with the contextual analysis.}
18   \arbnnull{zayduN}ibn-u \uc{'a}mriNU.\LRfootnote{See
19     \vref{fn:zayd-is-son}.}
20 \end{arab}

```

فَلَمَّا رَأَوْا^a النَّجْمَ...
قَالَ^b أَهْرَبُ فَلَنْ تُقْتَلَ.

زَيْدٌ ^dابْنُ عَمْرٍو ^e

Zayd^{unie} 'bn^u 'Amrⁱⁿ ^f

^aA footnote which interferes with the contextual analysis.

^bA footnote which interferes with the contextual analysis.

^cA footnote which interferes with the contextual analysis.

^dSee note 22 on page 19.

^eA footnote which interferes with the contextual analysis.

^fSee note 22 on page 19.

4.7 Stretching characters: the *taṭwīl*

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

.hunayn-u bn-u 'is.h--_aq-a حُنَيْنُ بْنُ إِسْحَاقَ Hunayn^u bn^u 'Ishāq^a

4.8 Digits

4.8.1 Numerical figures

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

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

4.8.2 The *abjad*

The numbers may also be expressed with letters from right to left arranged in accordance with the order of the Hebrew and Aramaic alphabets (see Wright 1896, i. 28 B–C). The *'abjad* numbers are usually distinguished from the surrounding words by a stroke placed over them.

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

`\abjad{45}` kitAbu-hu fI 'l-`AdAt-i مَ كَاتِبُهُ فِي الْعَادَاتِ 45 kitābu-hu fī 'l-`ādātⁱ.

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 section 4.10.1 on page 28 or insert the *'abjad* number in `\arb[novoc]{}`:—

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

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

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

The *'abǧad* number for the current page (27) is ٢٧.

This technique can be used to produce abjad-numbered lists as will be demonstrated on page 53.

4.9 Additional characters

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

For the time being, the following is included:—

Letter	Transliteration ³⁰			ArabTeX 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>	˘d
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
ⵍ	(((((
ⵍ)))))

Table 6: Additional Arabic codings

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

³⁰See below section 8 on page 41.

4.10 Arabic emphasis

As already seen in section 4.8.2 on page 26, the *ʿabǧad* numbers are distinguished from the surrounding words by a stroke placed over them. This technique is used to distinguish further words that are proper names or book titles.

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

`\abjad{45}: kitAbu-hu \aemph{fI 'l-`AdAt-i}` مة 45: كُتِبَ فِي الْعَادَاتِ
kitābu-hu fī 'l-ʿĀdātⁱ.

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}` مة 45: كُتِبَ فِي الْعَادَاتِ
kitābu-hu fī 'l-ʿĀdātⁱ.

4.10.1 Underlining words or numbers

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

- (a) `\aoline`, which is equivalent to `\aemph*` described above.
- (b) `\aoline*`, which is the same as `\aoline`, but better suited for *ʿabǧad* numbers.³¹
- (c) `\auline`, which can be used to underline Arabic words.

5 Arabic poetry

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

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

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

The description of the optional arguments follows:—

mode `mode=⟨mode⟩`, either `voc`, `fullvoc`, `novoc` or `trans`. The default mode is the one that is set at load time as already seen section 2.2 on page 6.

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.

³¹ See the example provided above section 4.8.2 on page 26.

New feature
v1.13

New feature
v.1.13

gutter	<code>gutter=<width></code>	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	<code>metre=<name></code>	Default: none
	If the name of the metre is expressed, it is printed after the lines and set flush left in <code>voc</code> , <code>fullvoc</code> and <code>novoc</code> modes or flush right in <code>trans</code> mode.	
delim	<code>delim=true false</code>	Default: false
<code>\SetHemistichDelim</code>	This named argument does not need a value as it defaults to <code>true</code> 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 <code>\SetHemistichDelim{<delimiter>}</code> command may be used at any point of the document to change this default setting.	
utf	<code>utf=true false</code>	Default: false
	As the preceding one, this named argument does not need a value as it defaults to <code>true</code> if it is used. If so, Unicode Arabic input is expected in the <code>arabverse</code> environment instead of <code>ASCII ArabTeX</code> or <code>Buckwalter</code> input schemes. See section 10 on page 48 for more details.	
color	<code>color=<color name></code>	Default: not set
	The color in which lines of poetry are to be rendered.	
export	<code>export=true false</code>	Default: false
	This named argument does not need a value as it defaults to <code>true</code> if it is used. If <code>export</code> is set as a global option as well (see above on page 6), all the lines will be converted to Unicode and exported to the external selected file. See below section 12 on page 57 for more details.	
<code>\bayt</code>	Inside the <code>arabverse</code> environment, each line is typeset by the <code>\bayt</code> command which takes two mandatory arguments and may accept one optional argument. ³² Additionally, every <code>\bayt</code> command <i>must</i> be followed with <code>\\</code> like so:—	

`\bayt{<ṣadr>}[<tadwīr>] {<ağuz>}\\`

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

- (a) If the two hemistichs be connected with one another by a prominent horizontal flexible stroke, the *taṭwīl* should be used, like so: `[--]` (see section 4.7 on page 26). 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:—

³²A ‘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.

```

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

```

لَا أَرَى مَنْ عَاهَدْتُ فِيهَا فَأُبْكِي الْيَوْمَ دَهْلًا وَمَا يَرُدُّ الْبُكَاءُ

As one can see, *triple hyphens* have been used. In the *ṣadr*, the first hyphen triggers the rules that are related to the definite article and the *ʿalif^u ʿl-waṣlⁱ*,³³ while the following two select the figure of the letter *lām* connected with a following letter. In the *ʿağuz*, the last two hyphens select the letter *yā* connected with a preceding letter, while the first one is simply discarded in this mode, but still may appear as it should, if the *trans* mode be selected:—

```

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

```

lā ʿarā man ʿahidtu fī-hā fa-ʿabki ʿl- -yawma dahl^{an} wa-mā yaruddu ʿl-bukāʿu

- (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 الْهَمِّ إِذَا خَفَّ بِالْثَوِيِّ النَّجَاءُ
بِزَفَوِّ كَانَهَا هِقْلَةً أُمَّ رِثَالٍ دَوِيَّةٌ سَقْفَاءُ

In the first line, the word *الْهَمِّ* should be split into *الْهَمِّم* 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,

³³See section 4.2 on page 18.

the word أَمْ should be split into أَمْ , so that the only way to avoid splitting it into two parts is to write it all inside the *tadwīr*. In that case, as the word is to be placed in the middle, it has been surrounded by spaces.

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

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

If they are little in number, each footnote may be split into pairs of `\footnote mark{}` (please mind the braces or “declare” `footnotemark` using `\MkArbBreak` to take it out of the Arabic environment³⁴) 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.³⁵

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

5.1 Example

Here follow the first lines of Imru’u ’l-Qaysi’s *Mu’allaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the *’idġām* that is fit to this declamatory poetry:—³⁶

```

1 \begin{arab}[fullvoc]
2   qĀla imru'u 'l-\uc{q}aysi fI mu`allaqati-hi:
3 \end{arab}
4
5 \begin{arabverse}[mode=fullvoc, metre={(al-.darbu 'l-_tAnI mina
6   'l-`arU.di 'l-'_U1_A mina 'l-.tawIli)}]
7   \SetArbDflt*
8   \begin{linenumbers*}

```

³⁴ See section 11.1 on page 50.

³⁵ The footnote package can also be used for the same effect. However, it must be loaded *after* `arabluatex`.

³⁶ Please note that for the time being only the assimilation rules that are laid on item (b) on page 18 are applied. See section 2.2.1 on page 6 for more information. None of the editions of the *Mu’allaqāt* that I know of feature the *’idġām* in the Arabic text, although it is often strongly marked in declamation.

```

9      \bayt{qifA nabki min _dikr_A .habIbiN wa-manzili}{bi-saq.ti
10     'l-liw_A bayna \uc{'l-d}a_hUli fa-\uc{.h}awmali}\\
11     \bayt{fa-\uc{t}U.di.ha fa-'l-\uc{m}iqrATi lam ya`fu
12     rasmu-hA}{limA nasa`gat-hA min `ganUbiN wa-`sam'ali}\\
13     \bayt{tar_A ba`ara 'l-'ar'Ami fI `ara.sATi-hA}{wa-qI`Ani-hA
14     ka-'anna-hu .habbu fulfuli}\\
15     \bayt{ka-'annI .gadATa 'l-bayni yawma ta.hammalUA}{lad_A
16     samurAti 'l-.hayyi nAqifu .han.zali}\\
17     \bayt{wuqUfaN bi-hA .sa.hbI `alayya ma.tiyya-hum}{yaqUlUna
18     lA tahlik 'asaN_A wa-ta`gammali}\\
19     \bayt{wa-'inna `sifA'I `abraTuN muharAqaTuN}{fa-hal `inda
20     rasmiN dAsiriN min mu`awwali}\\
21     \end{linenumbers*}
22     \end{arabverse}

```

قَالَ أَمْرُ الْقَيْسِ فِي مَعْلَقَتِهِ:

1 بِسَطِّ الْوَيْ بَيْنَ الدَّخُولِ لِحَوْمِ
 2 لَمَّا نَسَجْتَهَا مِنْ جَنُوبٍ وَشَتَلِ
 3 وَفَيْعَانَهَا كَانَتْ حَبُّ فَلْفُلِ
 4 لَدَى سَمَرَاتِ الْحَيِّ نَاقِفٌ حَنْظَلِ
 5 يَقُولُونَ لَا تَهْلِكْ أَسَى وَتَجَلِ
 6 فَهَلْ عِنْدَ رَسْمٍ دَاسِرٍ مِنْ مُعَوَّلِ
 قَفَا نَبِكَ مِنْ ذِكْرَى حَبِيبٍ وَمَنْزِلِ
 فَتَوْضِخٍ فَالْمُقَرَّاةِ لَمْ يَعْفُ رَسْمُهَا
 تَرَى بَعَرَ الْأَرَامِ فِي عَرَصَاتِهَا
 كَأَنِّي غَدَاةَ الْبَيْنِ يَوْمَ تَحَلَّوْا
 وَفَوْقَهَا صَحِيٌّ عَلَى مَطْبَعِهِمْ
 وَإِنْ شِفَايَ عِبْرَةٌ مَهْرَاقَةٌ

(الضَرْبُ الثَّانِي مِنَ الْعُرُوضِ الْأُولَى مِنَ الطَّوِيلِ)

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

1	qifā nabki min dīkrā ḥabīb ^{iw} wa-manzili	bi-saqī 'l-liwā bayna 'd-Daḥūli fa-Ḥawmali
2	fa-Tūḍiḥa fa-'l-Miqrāti lam ya`fu rasmu-hā	limā nasaḡat-hā min ḡanūb ^{iw} wa-šam`ali
3	tarā ba`ara 'l-'ar`āmi fī `arašāti-hā	wa-qī`āni-hā ka-'anna-hu ḥabbu fulfuli
4	ka-'annī ḡadāta 'l-bayni yawma taḡammalū	ladā samurāti 'l-ḥayyi nāqifu ḥanḡali
5	wuqū ^{um} bi-hā šaḡbī `alayya maṡiyya-hum	yaqūlūna lā tahlik `aša ^w wa-taḡammali
6	wa-'inna šifā`ī `abrat ^{um} muharāḡat ^{um}	fa-hal `inda rasm ^{im} dāsir ^{im} mim mu`awwali

(aḡ-ḡarbu 't-tānī minā 'l-'arūḡi 'l-'ūlā minā 't-tawīli)

6 Special applications

Linguistics The same horizontal stroke as the *taṭwīl* (see section 4.7 on page 26) may be encoded ⟨B⟩; ⟨BB⟩ will receive the *tašdīd*. This is useful to make linguistic annotations and comments on vowels:—

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

New feature
v1.4.3

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

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

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

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

{وَقَالَ} إِنَّ أُنِي كَانَ مِنَ الْمُقَاتِلَةِ وَكَانَتْ أُبِي مِنْ عُظَمَاءِ بُيُوتِ الزَّمَانِيَةِ.

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-raḥmāni ʾr-raḥīmi
FDF5	salam	صَلَام	ṣallā ʾLlāhu ʿalay-hi wa-sallama
FDFA	slm	ﷺ	ṣallā ʾLlāhu ʿalay-hi wa-sallama
FDFB	jalla	جَلَالًا	ġalla ġalāla-hu

Table 7: Additional Arabic marks

The mark to be inserted is determined by contextual analysis, or by an optional argument, either `r1` to have the Arabic glyph printed, or `lr` to print the tranliterated equivalent.

New feature
v1.13

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: `^^^xyz`, where xyz are lowercase hex digits (0 to 9 or a to f).

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

It must be noted that `\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 مُبَيَّنَةٍ 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

ʿIshāq (إِسْحَاقُ).³⁷ —

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

هو أبو زيد حنين بن إسحاق العبادي يُفَتِّحُ الْعَيْنَ وَخَفِيفُ الْبَاءِ.
هو أبو زيد حنين بن إسحاق العبادي يُفَتِّحُ الْعَيْنَ وَخَفِيفُ الْبَاءِ.

6.1 The Qurʾān

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

\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: ﴿١٢٣﴾—in Arabic script or inside parentheses in romanized Arabic:—

\ayah{123} ﴿١٢٣﴾ (123).

An example follows:—

³⁷\underline and \highlight are taken from the lua-ul package which is loaded by arabluatex. See Krüger (2020).

New feature
v1.15

```

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

```

مِنْ ﴿سُورَةِ الْبَقَرَةِ﴾:
 أَلَمْ يَكُنْ لَهُ الْكِتَابُ لَا رَيْبَ فِيهِ هُدًى لِلْمُتَّقِينَ ﴿٢٠﴾ الَّذِينَ يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَاةَ وَمِمَّا رَزَقْنَاهُمْ
 يُنفِقُونَ ﴿٢١﴾

min (sūraTi 'l-Baqarati):
'alif lām mīm (1) dālīka 'l-kitābu lā rayba fī-hi hudā^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.

7 Color

arabluatex 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:³⁸ —

- The “pipe” character (|, section 4.5 on page 24);
- ‘Quoting’ technique (section 4.4 on page 22), and more specifically ‘quoting the *hamzah*’ (on page 23);
- Putting back on broken contextual analysis rules (section 4.6 on page 24);
- Arabic marks (section 6 on page 33).

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

³⁸Regarding 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.

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

```

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

```

الْبَابُ الْخَامِسُ فِي طَبَقَاتِ الْأَطِبَّاءِ الَّذِينَ كَانُوا مِنْذُ زَمَانٍ جَالِينُوسَ وَقَرِيْبًا مِنْهُ. جَالِينُوسُ: وَلَنْضَعُ أَوَّلًا
كَلَامًا كُتِبَ فِي أَخْبَارِ جَالِينُوسَ وَمَا كَانَ عَلَيْهِ...
al-bābu 'l-ḥāmisu fī ṭabaqāti 'l-'aṭibbā'i 'lladīna kānū mundū zamāni
Ġālīnūsa wa-qarīb^{an} min-hu. Ġālīmūsu: wa-l-naḍa` 'awwal^{an} kalām^{an}
kulliyy^{an} fī 'ahbāri Ġālīnūsa wa-mā kāna 'alay-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` أَشْتَرَيْتُهُ بِتَمَنٍ أَجَبَكَ *ištaraytu-hu bi-tama-*
nⁱⁿ 'a'ḡaba-ka.

‘fullvoc’ mode:

`i^stara\arbcOLOR[brown]{y}tu-hu bi-_tama\arbcOLOR[red]{niN}`
`'a`\arbcOLOR[blue]{^ga}ba-ka` أَشْتَرَيْتُهُ بِتَمَنٍ أَجَبَكَ *ištaraytu-hu bi-tama-*
nⁱⁿ 'a'ḡaba-ka.

7.1 Tricks of the trade

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

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

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

Diacritic	Transliteration ³⁹			ArabTeX notation
	dmg	loc	arabica	
َ				
-	<i>a</i>	<i>a</i>	<i>a</i>	.a
ُ				
-	<i>u</i>	<i>u</i>	<i>u</i>	.u
ِ				
-	<i>i</i>	<i>i</i>	<i>i</i>	.i
◌◌◌				
-				o

Table 8: ArabTeX diacritics for `\arbcolor`

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

‘voc’ mode:

`i^staraytu-hu bi-_taman\arbcolor[red]{iN} 'a^g\arbcolor[red]{.a}ba-ka` اِشْتَرَيْتُهُ بِئِنَّ اَعْجَبَكَا *ištaraytu-hu bi-tamanⁱⁿ 'aġaba-ka.*

`i^staraytu-hu bi-_tama\arbcolor[red]{n}iN 'a\arbcolor[red]{^g}.aba-ka` اِشْتَرَيْتُهُ بِئِنَّ اَعْجَبَكَا *ištaraytu-hu bi-tamanⁱⁿ 'aġaba-ka.*

`i^staraytu-hu bi-_tama\arbcolor[red]{n}\arbcolor[blue]{iN} 'a\arbcolor[red]{^g}\arbcolor[blue]{.a}ba-ka` اِشْتَرَيْتُهُ بِئِنَّ اَعْجَبَكَا *ištaraytu-hu bi-tamanⁱⁿ 'aġaba-ka.*

‘fullvoc’ mode:

`i^staray"\arbcolor[red]{o}tu-hu bi-_taman"\arbcolor[red]{iN} 'a^g"\arbcolor[red]{.a}ba-ka` اِشْتَرَيْتُهُ بِئِنَّ اَعْجَبَكَا *ištaraytu-hu bi-tamanⁱⁿ 'aġaba-ka.*

`i^stara\arbcolor[red]{y}{o}tu-hu bi-_tama\arbcolor[red]{n}iN 'a\arbcolor[red]{^g}.aba-ka` اِشْتَرَيْتُهُ بِئِنَّ اَعْجَبَكَا *ištaraytu-hu bi-tamanⁱⁿ 'aġaba-ka.*

`i^stara\arbcolor[red]{y}\arbcolor[blue]{o}tu-hu bi-_tama\arbcolor[red]{n}\arbcolor[blue]{iN} 'a\arbcolor[red]{^g}\arbcolor[blue]{.a}ba-ka` اِشْتَرَيْتُهُ بِئِنَّ اَعْجَبَكَا *ištaraytu-hu bi-tamanⁱⁿ 'aġaba-ka.*

³⁹See below section 8 on page 41.

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

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

```
mu`allim\arbcOLOR[red]{\arbnul{m}aN} مَعْلَمٌ mu'alliman,
istisqA'\arbcOLOR[red]{\arbnul{A'}aN} اِسْتِسْقَاءٌ istisqā'an,
ˆsay'\arbcOLOR[red]{\arbnul{ay'}aN} شَيْءٌ šay'an,
ˆgAmi`aT|\arbcOLOR[red]{\arbnul{T}aN} جَامِعَةٌ ġāmi'atan.
```

REM. Note that in the last example (*ġāmi'at^{an}*), the ‘pipe’ character has been inserted before `\arbcOLOR`. Otherwise, the `dmg` mode of the transliteration rules would have interpreted the *tā'* *marbūṭah* as *final* (e.g. *h* instead of the expected *t*).⁴⁰

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

```
1 \newarbmArk{Y}{~~~~0649}{ }
2 \arb{hud\arbcOLOR[red]{aN\arbnul{ _A}}\arbmArk{Y}}
3 \arb[trans]{hud\arbcOLOR[red]{aN\arbnul{ _A}}\arbmArk{Y}}
```

هُدًى *hudāⁿ*

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

```
wa-\arbcOLOR[red]{\arbnul{wa}i}stisqA'uN اِسْتِسْقَاءٌ wa-'stisqā'un41.
fi "al".i-\arbcOLOR[red]{\arbnul{'l-}i}btidA'i فِي اِلْبِتْدَاءِ
fi 'li-'btidā'i.
\arbcOLOR[red]{'a'\arbnul{k}}kulu اَكُلُ ākulu,
\arbcOLOR[red]{'A'\arbnul{k}}kiluN اَكِلُ ākilun.
```

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

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

⁴⁰See also on page 44 “Discarding the *ʿirāb*” for more information.

⁴¹To the knowledge of the writer, the *waṣlah* alone is not part of the Arabic Unicode block.

أَكُلُ *ākulu*.

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

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

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

المَلْبُونِ *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 فِ النَّاسِ *fi 'n-nāsi*, the initial *ʿalif^w l-waṣlⁱ* 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}\arbookmark{lnn}Asi}
4 \arb[trans]{fI\arbnul{al-}
5 \arbcolor[red]{\arbnul{al-}a}\arbookmark{lnn}Asi}.
```

فِ 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 24—:

yatasA\arbnul{'a}\arbcolor[red]{|''}.alUna يَتَسَاءَلُونَ *yatasā'a-lūna*, ^say\arbcolor[red]{|''}\arbnul{'a}n شَيْئًا *šay^{an}*, ^say\arbcolor[red]{|''}iN شَيْءٍ *šayⁱⁿ*, \arbcolor[red]{a''}.as\arbcolor[red]{y''}.ilaTuN أَسْئَلَةً *as'ilat^{un}*.

8 Transliteration

It may be more appropriate to speak of “romanization” than “transliteration” of Arabic. As seen above in section 2.2 on pages 6–9, 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 6, three standards are supported at present:

dmg *Deutsche Morgenländische Gesellschaft*, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.⁴² **dmg** transliteration convention is selected by default;

loc *Library of Congress*: this standard is part of a large set of standards for romanization of non-roman scripts adopted by the American Library Association and the Library of Congress;⁴³

arabica *Journal of Arabic and Islamic Studies/Revue d'études arabes et islamiques*: this standard is most widely used by scholars in the field of Arabic studies.⁴⁴

More standards will be included in future releases of arabluatex.

New feature
v1.8

`\SetTranslitConvention`

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

⁴²See Brockelmann et al. (1935).

⁴³See <http://www.loc.gov/catdir/cpsd/roman.html> for the source document concerning Arabic language.

⁴⁴See http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf.

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

`\uc{.hunayn-u} bn-u \uc{'is.h_aq-a}` حُنَيْنُ بْنُ إِسْحَقَ *Hunayn^u bn^u*
`'Ishāqa, \uc{`u_tm_an-u}` عَثْمَنُ *Utmān^u*, .daraba `\uc{zayd-u} bn-u`
`\uc{h_alidiN} \uc{sa`d-a} bn-a \uc{`awf-i} bn-i \uc{`abd-i}`
`\uc{'l-l_ah-i}` ضَرَبَ زَيْدُ بْنُ خَلْدٍ سَعْدَ بْنَ عَوْفٍ بْنَ عَبْدِ اللَّهِ *daraba Zayd^u bn^u*
Hālidiⁱⁿ Sa`d^a bn^a Awfⁱ bnⁱ Abdⁱ 'Llāhⁱ.

However, `\uc` must be used cautiously in some very particular cases, for the closing brace of its argument may prevent a rule from being applied. To take an example, as seen above on page 20, 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` vz. up to the first letter that is to be uppercased, like so: `\uc{m}u.hammaduN`.

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

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

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

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

\prname Proper names outside Arabic environments Transliterated proper names inserted in paragraphs of English text should be printed in the same typeface as the surrounding text. `\prname{⟨Arabic proper name⟩}` is provided to that effect:⁴⁵—

⁴⁵Just as `\uc`, `\prname` is also able to process proper names consisting of several subsequent words.

```

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

```

From Wright (1896, i. 23 C):— If the name following إِبْنُ be that of the mother or the grandfather, the أ is retained; as $\text{عِيسَى ابْنُ مَرْيَمَ}$, “Jesus the son of Mary”; $\text{عَمَّارُ ابْنُ مَنْصُورٍ}$, “Ammār the (grand)son of Manṣūr”.

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:⁴⁶ —

```

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. `arabluatex` also provides `\prname*` which only renders in upright roman style already transliterated proper names without applying any further processing. It is mostly used internally and applied to proper names exported in Unicode to an external selected file.⁴⁷

8.1 Additional note on `dmg` convention

According to Brockelmann et al. (1935, p. 6), Arabic $\text{ʾi}ʿrāb$ may be rendered into `dmg` in three different ways:

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

`\arbup`

By default, `arabluatex` applies rule (b). Once delimited by a set of Lua functions, $\text{ʾi}ʿrāb$ is passed as an argument on to a `\arbup` command which is set to `\textsuperscript`.

`\NoArbUp`

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

`\ArbUpDflt`

⁴⁶See the documentation of `nameauth` for more details: <https://ctan.org/pkg/nameauth>

⁴⁷See below section 12 on page 57 for more details.

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 *ʿirāb* is displayed. To take one example, here is how Arabic *ʿirāb* may be rendered as subscript text:—

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

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

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

ʿirāb boundaries Every declinable noun (*muʿrab*) may be declined either with or without *tanwīn*, viz. *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`allimūN مَعْلِمٌ *mu'allim^{un}*, kA'inuN كَائِنٌ *kā'in^{un}*, kA'inAtuN كَائِنَاتٌ *kā'in^{atun}*, \uc{`amraNU} عَمْرُوْا *Amr^{an}*, fataN_A فَتًى *fataⁿ*, qA.diNI قَاضٍ *qādiⁿ*.

ḡayr munṣarīf: al-mu`allim-u اَلْمُعَلِّمُ *al-mu'allim^u*, kitAb-Ani كِتَابَانِ *kitāb^{ani}*, ra`sa'-Ani رِشَانٍ *raša'ani*, sAriq-Una سَارِقُونَ *sāriq^{una}*, qA.d-Una قَاضُونَ *qāḍ^{una}*, al-.zulm-Atu اَلْظُلُمَاتُ *aẓ-ẓulm^{atu}*.

REM. a. As the *tanwīn* is passed over in pronunciation when it is followed by the letters ج, ل, ي, و, م (see item (b) on page 18), 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 مَرَقًا *kāna ḡaniyy^{an} lākinna-hu labisa ḡubbat^{an} mumazzaq^{an} 'aydu-hā*.

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

mana`a 'l-nAs-a kAffaT-aN min mu_hA.tabati-hi 'a.had-uN bi-sayyidi-nA مَنَّ النَّاسُ كَافَّةً مِنْ مَخَاطِبَتِهِ أَحَدٌ بِسَيِّدِنَا *mana'a 'n-nās^a kāffat^{an} min muḡāḡabatⁱ-hi 'aḡad^{um} bi-sayyidi-nā*.

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

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

REM. It may so happen, as in the absence of the article before the annexed word, that `arabluatex` be unable to determine which of the above two cases the word ending with `tā'` *marbūṭah* falls into. The ‘pipe’ character (see section 4.5 on page 24) may be appended to that word to indicate that what follows is in the construct state: `\uc{r}isAlaT fI tartIb qirA'aT| kutub \uc{g}AlInUs Risālah fī tartīb qirā'at kutub Ġālīmūs`.

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

8.2 Examples

Here follows in transliteration the story of Ġuḥā and his donkey (جُحَا وَحَامُهُ). See the code on page 9:—

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

‘loc’ standard: `atā ṣadīqun ilā Juḥā yaṭlubu min-hu ḥimāra-hu li-yarkaba-hu fī safratīn qaṣīratīn fa-qāla la-hu: “sawfa u’idu-hu ilay-ka fī al-masā’i wa-adfa’u la-ka uḡratan.”` `fa-qāla Juḥā: “anā āsifun ġiddan annī lā astaṭī’u an uḥaqqiqa la-ka raġbata-ka fa-al-ḥimāru laysa hunā al-yawma.”` `wa-qabla an yutimma Juḥā kalāma-hu bada’a al-ḥimāru yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: “innī asma’u ḥimāra-ka yā Juḥā yanhaqu.”` `fa-qāla la-hu Juḥā: “ġharībun amru-ka yā ṣadīqī a-tuṣaddiqu al-ḥimāra wa-tukaddhība-nī?”`

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

9 Buckwalter input scheme

Even though `arabluatex` is primarily designed to process the `ArabTeX` notation, it

`\SetInputScheme`

can also process the Buckwalter input scheme to a large extent.⁴⁸ The Buckwalter scheme is actually processed in two steps, as it is first converted into ArabTeX. Then, once this is accomplished, the ArabTeX scheme is processed through the above described functions. In this way, the Buckwalter input scheme can make the most of the `arabluatex` special features that are presented in section 2.2 on page 6.

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 (2010, pp. 25–26).⁴⁹ However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode ا and ؤ, must be replaced with square brackets viz. [and] respectively.

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

Table 9 gives the Buckwalter equivalents that are currently used by `arabluatex`. The additional characters that are defined in table 6 on page 27 are also available.

Letter	Transliteration ⁵⁰			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

Table 9: Buckwalter scheme

⁴⁸See <http://www.qamus.org/transliteration.htm>

⁴⁹I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in `arabluatex`.

⁵⁰See section 8 on page 41.

Letter	Transliteration			Buckwalter notation	
	dmg	loc	arabica	base/xml	safe
ظ	<i>ẓ</i>	<i>ẓ</i>	<i>ẓ</i>	Z	Z
ع	<i>ʿ</i>	<i>ʿ</i>	<i>ʿ</i>	E	E
غ	<i>ġ</i>	<i>gh</i>	<i>ġ</i>	g	g
ف	<i>f</i>	<i>f</i>	<i>f</i>	f	f
ق	<i>q</i>	<i>q</i>	<i>q</i>	q	q
ك	<i>k</i>	<i>k</i>	<i>k</i>	k	k
ل	<i>l</i>	<i>l</i>	<i>l</i>	l	l
م	<i>m</i>	<i>m</i>	<i>m</i>	m	m
ن	<i>n</i>	<i>n</i>	<i>n</i>	n	n
ه	<i>h</i>	<i>h</i>	<i>h</i>	h	h
و	<i>w</i>	<i>w</i>	<i>w</i>	w	w
ي	<i>y</i>	<i>y</i>	<i>y</i>	y	y
ى	<i>ā</i>	<i>á</i>	<i>ā</i>	Y	Y
ة	<i>ah</i>	<i>ah</i>	<i>a</i>	p	p
ء	’	’	’	’	C
آ	<i>’ā</i>	<i>’ā</i>	<i>’ā</i>		M
أ	’	’	’	>	O
ؤ	’	’	’	&	W
إ	’	’	’	<	I
ئ	’	’	’]	Q
ـ	—	—	—	~	~
ـ	,	,	—	[L
ـ	<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
ـ	<i>ā</i>	<i>ā</i>	<i>ā</i>	`	e
ـ (taṭwīl)	—	—	—	—	—

Table 9: Buckwalter scheme

Transliteration The Buckwalter notation can also be transliterated into any accepted romanization standard of Arabic. See above section 8 on page 41 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, الطِّبِّ بِصِنَاعَةِ *bi-ṣinā‘at*ⁱ ‘*t-tibb*ⁱ.
 wa-Al-l~ehi وَاللَّهِ *wa-‘l-lāh*ⁱ, Al-Hamdu li-l~ehi اَلْحَمْدُ لِلَّهِ *al-ḥamd*^u *li-llāh*ⁱ.

Similarly, it is not advisable to use | and [(‘base’ and ‘xml’ schemes) or M and L (‘safe’ scheme) to encode the ‘*alif*^u’ ‘*l-mamdūdāt*ⁱ’ and the ‘*alif*^u’ ‘*l-waṣl*ⁱ’ for such signs are supposed to be generated by arabluatex internal functions. Besides, as they do not *per se* convey any morphological information on what they are derived from, they cannot be transliterated accurately. To take one example, <ilY Al-LntiqāADi gives اِلَى الْاِنْتِقَاضِ as expected, but only <ilY Al-intiqāDi can be transliterated as ‘*ila*’ ‘*l-intiqāḍi*’ with the correct vowel ⟨i⟩ in place of the ‘*alif*^u’ ‘*l-waṣl*ⁱ’.

10 Unicode Arabic input

As said above in section 9 on page 45 about the Buckwalter input scheme, even though arabluatex is primarily designed to process the ArabTeX notation, it also accepts Unicode Arabic input. It should be noted that arabluatex does in no way interfere with Unicode Arabic input: none of the `voc`, `fullvoc`, `novoc` or `trans` options will have any effect on plain Unicode Arabic for the time being.

That said, there are two ways of inserting Unicode Arabic:

- \txarb (a) The \txarb{<Unicode Arabic>} command for inserting Unicode Arabic text in paragraphs;
- txarab (b) The txarab environment for inserting running paragraphs of Arabic text, like so:—

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

11 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}}` مَهْ كِتَابُهُ فِي الْعَادَاتِ⁵¹
`kitābu-hu fi 'l-‘Ādāt’.`
`\arb{\abjad{45} \rlframebox[1in][s]{kitAbu-hu fi 'l-‘AdAti}}`

مَهْ⁵² كِتَابُهُ فِي الْعَادَاتِ

The same applies to footnotes:—

```

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

```

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

^aالْمُقَاتِلَةُ: الْمُقَاتِلِينَ.
^bالزَّمَازِمَةُ: 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

```

⁵¹This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.

⁵²`\rlframefox` has been adapted from `\framebox` for insertions of right-to-left text.

```
7 \end{arab}
```

زَيْدٌ^{٥٣} 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.⁵³

`\LRmarginpar`

The `\LRmarginpar[⟨left⟩]{⟨right⟩}` command does for marginal notes the same as `\LRfootnote` does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that `\marginpar` also works in Arabic environments, but it acts as any other single-argument command inserted in Arabic environments. The general principle laid on page 48 applies.

`\setRL`

`\setLR`

`\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 لِبُزْرَجُمِهَرِ بْنِ الْبُخْتِكَانِ
قَالَ بَرْزَوِيهِ بْنُ أَزْهَرَ، رَأْسُ أَطِبَّاءِ فَارِسَ...

11.1 New commands

In some particular cases, it may be useful to define new commands to be inserted in Arabic environments. From the general principle laid on page 48, it follows that any command that is found inside an Arabic environment is assumed to have Arabic

⁵³See <http://ctan.org/pkg/koma-script>; read the documentation of KOMA-script for details about the `\deffootnotemark` and `\deffootnote` commands.

text in its argument which `arabluatex` will process as such before passing it on to the command itself for any further processing. As a result of this feature, such a command as:

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

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

`\MkArbBreak`

The `\MkArbBreak{<csv list of commands>}` command 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 أَتَلَقَ ذُو الْقَرْنَيْنِ إِلَى أُمَّةٍ أُخْرَى فِي مَطْلَعِ الشَّمْسِ وَلَا بِنَاءَ لَهُمْ يُؤْمِنُهُم مِّنَ الشَّمْسِ.

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

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

`\MkArbBreak*`

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

New feature
v1.12

New feature
v1.12

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

11.2 Environments

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 مُوَيْسَ دَجَاجَةٍ. وَكَانَتْ دَجَاجَتُهُ الَّتِي أَهْدَاهَا دُونَ مَا كَانَ يَتَخَذُ لِمُوَيْسَ. وَلَكِنَّهُ
بِكَرَمِهِ وَبِحُسْنِ خُلُقِهِ أَظْهَرَ التَّعَجُّبَ مِنْ سَمَنِهَا وَطَيْبِ لَحْمِهَا. وَكَانَ أَبُو الْهَذِيلِ يَعْرِفُ بِالْأَمْسَاكِ الشَّدِيدِ.
فَقَالَ: "وَكَيْفَ رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ الدَّجَاجَةَ؟" قَالَ: "كَانَتْ عَجَبًا مِنَ الْعَجَبِ!" فَيَقُولُ:

"وَتَدْرِي مَا جَنَسَهَا؟ وَتَدْرِي مَا سَنَاهَا؟ فَإِنَّ الدَّجَاجَةَ إِذَا تَطَيَّبُ بِالْجَنَسِ وَالسِّنِّ. وَتَدْرِي بِأَيِّ شَيْءٍ كُنَّا نُسَمِّيهَا؟ وَفِي أَيِّ
مَكَانٍ كُنَّا نَعْلِفُهَا؟"

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

11.2.1 Lists

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

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

```

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

```

رموز الكتاب
 ب مخطوط المكتبة الأهلية بباريس ٢٨٦٠ عربي.
 س مخطوط المكتبة الأهلية بباريس ٢٨٥٩ عربي.
 م مخطوط مجلس شوراى ملى طهران ٥٢١.

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

```

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

```

القانون في الطب
 - الفَنُّ الأوَّلُ فِي حَدِّ الطِّبِّ وَمَوْضُوعَاتِهِ مِنَ الْأُمُورِ الطَّبِيعِيَّةِ وَشَتَّى عَلَى سِتَّةِ تَعَالِيمٍ
 - التَّعْلِيمُ الأوَّلُ [وَهُوَ فَصْلَانِ]
 - الْفَصْلُ الأوَّلُ

As a third example, abjad-numbered lists can be typeset in conjunction with the `enumitem` package,⁵⁴ 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 tafaA`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 (1896, i. 29 B–C):— 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.

أَفْعَالٌ	وَتَفَاعِلٌ	أَفْعَلٌ
يَبْ أَفْعَوْلٌ	زَ أَفْعَلٌ	بَفْعَلٌ
يَجْ أَفْعَوْلٌ	حَ أَفْعَلٌ	جَفَاعِلٌ
يَدْ أَفْعِنَلٌ	طَ أَفْعَلٌ	دَ أَفْعَلٌ
يَهْ أَفْعِنَلِي	يَ اسْتَفْعَلٌ	هَ تَفْعَلٌ

⁵⁴See the documentation of `enumitem` for more details: <https://ctan.org/pkg/enumitem>

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

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

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

11.3 csquotes

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

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

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

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

فَقَالَ لَهُ جَا: ”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.

11.4 Two-argument special commands

textcolor The two-argument command `\textcolor{color}{Arabic text}` is supported inside `\begin{arab} ... \end{arab}`. One simple example follows:⁵⁵—

```
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
```

⁵⁵`arabxetex` provides its own `\arbcOLOR` command which is able to render syllables or diacritics in colors. See section 7 on page 36.

```

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

```

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

Muhaddabu 'd-Dīni 'Abdu 'r-Raḥīmi bnu 'Aliyyⁱⁿ huwa šayḥu-na 'l-'imāmu
'š-sadru 'l-kabīru 'l-'ālimu 'l-fāḍilu 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}`.⁵⁶ As an example, one may get arablualatex and reledmac to work together like so:—

```

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

```

11.5 quran

arablualatex is compatible with the quran package so that both can be used in conjunction with one another for typesetting the *Qur'ān*. As quran draws the text of the *Qur'ān* from a Unicode encoded database, its commands have to be passed as arguments to the `\txarb` command for short insertions in left-to-right paragraphs, or inserted inside the `txarab` environment for typesetting running paragraphs of *Qur'ānic* text (see above section 10 on page 48 for more details). Please note that arablualatex takes care of formatting the Arabic: therefore, it is recommended to load the quran package with the `nopar` option, after arablualatex itself has been loaded, like so:—

⁵⁶`\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﴾

12 Exporting Unicode Arabic to an external file

New feature
v.1.13

arabluatex is able to produce a duplicate of the original .tex source file in which all arabtex or buckwalter strings will have been replaced with Unicode equivalents, either in Arabic script or in any accepted standard of transliteration. Exporting ASCII strings to Unicode while preserving the exact selected global or local options is a fairly complex operation which may require Lua^LTeX to be run several times as will be explained below.

12.1 Commands and environments

export global option First, arabluatex must be loaded with the export global option enabled,⁵⁷ 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 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:—

⁵⁷See above on page 6 for more information.

```

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:—

- | | |
|---|---|
| <p>(a) Unicode Arabic text, either in Arabic script or in transliteration, is inserted as argument of <code>\txarb</code>⁵⁸ or <code>\txtrans</code>⁵⁹ accordingly.</p> <p>(b) Additionally, Arabic paragraphs may receive <code>\arbpardir</code>, which arabluatex uses to determine the direction of Arabic paragraphs to be set by default, or either <code>\setRL</code> or <code>\setLR</code> depending on what may have been set locally.⁶⁰</p> <p>(c) Proper names are inserted as arguments of <code>\prname*</code>.⁶¹</p> | <p><code>\arbpardir</code></p> <p><code>\prname*</code></p> |
|---|---|
- Appending words or commands to the external file only** `\ArbOutFile[<newline>]{<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 29 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*`.⁶²

12.2 Nested Arabic environments

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

```

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

```

Where *<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.

⁵⁸See above section 10 on page 48.

⁵⁹`\txtrans` is used internally by several Lua functions to format transliterated Arabic. Therefore, it is not documented.

⁶⁰See above on page 50.

⁶¹See above on page 43.

⁶²See above note 32 on page 29 for more information.

12.3 Further processing of Unicode converted files

Unicode files can be further processed by document converters such as John McFarlane’s `pandoc`⁶³. To take here one simple example, here is how `file_out.tex` can be converted from Lua^AT_EX into Open Document format (`.odt`):—

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

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

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

13 Future work

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

- (a) Short-term:
 - 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 note 4 on page 5.
 - ii. Formulate propositions for extending the Arab^AT_EX notation and the transliteration tables. Include them in `arabluatex`. See section 4.9 on page 27.

14 Implementation

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

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

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

⁶³See <http://pandoc.org/>

```
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   \texdir TRT}}
```

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

```
39 \AtBeginDocument{\def\setLR{\boolfalse{al@rlmode}\pdir 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\rmfamily#1\egroup}}
43   {\NewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\rmfamily#1\egroup}}}
```

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

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

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

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

```
47 \NewDocumentCommand{\MkArbBreak}{s m}{%
48   \IfBooleanTF{#1}
49   {\luairect{arabluatex.mkarbbreak(\luastringN{#2}, "out")}}
50   {\luairect{arabluatex.mkarbbreak(\luastringN{#2}, "dflt")}}
51 }
```

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

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

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

```
52 \newunderlinetype\@aoverLine{\leaders\vrule height 3ex depth -2.9ex}
53 \def\@aoline{\@ifstar\@aoline\@aoline}
54 \def\@aoline#1{\ensuremath{\overline{\mbox{#1}}}}
55 \def\@aoline#1{\@aoverLine#1}
56 \newunderlinetype\@aunderLine{\leaders\vrule height -.65ex depth .75ex}
57 \def\@aoline#1{\@aunderLine#1}
58 \AtBeginDocument{\ifdef{\aemph}%
59   {\RenewDocumentCommand{\aemph}{s m}{%
60     \IfBooleanTF{#1}{%
61       \luairect{tex.sprint(arabluatex.aemph(\luastringN{#2},
62         "over"))}}}
```

```

63     {\luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
64         "dflt"))}}}
65 {\NewDocumentCommand{\aemph}{s m}{%
66     \IfBooleanTF{#1}{%
67         \luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
68             "over"))}}
69     {\luadirect{tex.sprint(arabluatex.aemph(\luastringN{#2},
70         "dflt"))}}}

```

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

```

71 \NewDocumentCommand{\arbcolor}{o m}{%
72     \IfNoValueTF{#1}{#2}{\textcolor{#1}{#2}}}

```

`\SetInputScheme` arabluatex is designed for processing ArabTeX input notation. `\SetInputScheme` may be used in the preamble or at any point of the document should the user wish to use a different notation such as the ‘Buckwalter scheme’.

```

73 \def\al@input@scheme{arabtex}
74 \NewDocumentCommand{\SetInputScheme}{m}{\def\al@input@scheme{#1}}

```

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

`\SetArbDflt*` As of v1.6, arabluatex does not applies any more the assimilation rules that are laid on item (b) on page 18; a new starred version `\SetArbDflt*` is now available to the user should he wish to apply them.

```

75 \def\al@arb@rules{dflt}
76 \NewDocumentCommand{\SetArbEasy}{s}{%
77     \IfBooleanTF{#1}
78     {\def\al@arb@rules{easynosukun}}
79     {\def\al@arb@rules{easy}}}
80 \NewDocumentCommand{\SetArbDflt}{s}{%
81     \IfBooleanTF{#1}
82     {\def\al@arb@rules{idgham}}
83     {\def\al@arb@rules{dflt}}}

```

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

```

84 \def\al@trans@font{\rmfamily}%
85 \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:

```

86 \def\al@trans@style{\itshape}%
87 \NewDocumentCommand{\SetTranslitStyle}{m}{\def\al@trans@style{#1}}

```

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

```

88 \def\al@trans@convention{dmg}
89 \NewDocumentCommand{\SetTranslitConvention}{m}{%
90   \def\al@trans@convention{#1}}

```

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

```

91 \NewDocumentCommand{\al@arbup@dflt}{m}{\textsuperscript{#1}}%
92 \NewDocumentCommand{\al@arbup}{m}{\al@arbup@dflt{#1}}
93 \NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
94 \NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt}
95 \NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\al@arbup}{m}{##1}}
96 \NewDocumentCommand{\SetArbUp}{m}{%
97   \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.

```

98 \NewDocumentCommand{\uc}{m}%
99   {\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.

```

100 \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.

```

101 \NewDocumentCommand{\prname}{s m}{%
102   \bgroup\SetTranslitStyle{\relax}%
103   \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.

```
104 \NewDocumentCommand{\txarb}{+m}{%
105   \ifvmode\leavevmode\fi%
106   \bgroup\textdir TRT\arabicfont#1\egroup}
107 \NewDocumentCommand{\txtrans}{+m}{%
108   \bgroup\textdir TLT\al@trans@font\al@trans@style#1\egroup}
```

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

```
109 \NewDocumentEnvironment{txarab}{}{%
110   \par%
111   \booltrue{al@rlmode}%
112   \pardir TRT\textdir TRT\arabicfont}{\par}
```

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

```
113 \NewDocumentEnvironment{txarabtr}{}{%
114   \par%
115   \pardir TLT\textdir TLT%
116   \al@trans@font\al@trans@style}{\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.

```
117 \NewDocumentCommand{\arb}{0{\al@mode} +m}%
118 {\edef\@tempa{#1}%
119   \ifx\@tempa\al@mode@voc%
120     \ifvmode\leavevmode\fi%
121     \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
122     \luadirect{tex.sprint(arabluatex.processvoc(\luastringN{#2},
123       \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
124   \else%
125     \ifx\@tempa\al@mode@fullvoc%
126       \ifvmode\leavevmode\fi%
127       \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
128       \luadirect{tex.sprint(arabluatex.processfullvoc(\luastringN{#2},
129         \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
130     \else%
131       \ifx\@tempa\al@mode@novoc%
132         \ifvmode\leavevmode\fi%
133         \bgroup\booltrue{al@rlmode}\textdir TRT\arabicfont%
134         \luadirect{tex.sprint(arabluatex.processnovoc(\luastringN{#2},
135           \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
136       \else%
137         \ifx\@tempa\al@mode@trans%
138           \bgroup\textdir TLT\al@trans@font\al@trans@style%
139           \luadirect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
140             \luastring0{\al@trans@convention},
141             \luastring0{\al@arb@rules},
```



```

142 \luastring0{\al@input@scheme}}))\egroup%
143 \else%
144 \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.

```

145 \NewDocumentCommand{\arbmark}{0{} m}{%
146 \bgroup%
147 \SetInputScheme{arabtex}%
148 \luadirect{tex.sprint(arabluatex.processarbmarks(\luastringN{#2},
149 \luastringN{#1}))}%
150 \egroup}

```

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

```

151 \NewDocumentCommand{\newarbmark}{m m m}{%
152 \luadirect{arabluatex.newarbmark(\luastringN{#1}, \luastringN{#2},
153 \luastringN{#3})}%

```

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

```

154 \NewDocumentEnvironment{arab}{!0{\al@mode} +b}%
155 {\par\edef\@tempa{#1}%
156 \ifx\@tempa\al@mode@voc%
157 \booltrue{al@rlmode}%
158 \bgroup\pardir TRT\textdir TRT\arabicfont%
159 \luadirect{tex.sprint(arabluatex.processvoc(\luastringN{#2},
160 \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
161 \else%
162 \ifx\@tempa\al@mode@fullvoc%
163 \booltrue{al@rlmode}%
164 \bgroup\pardir TRT\textdir TRT\arabicfont%
165 \luadirect{tex.sprint(arabluatex.processfullvoc(\luastringN{#2},
166 \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
167 \else%
168 \ifx\@tempa\al@mode@novoc%
169 \booltrue{al@rlmode}%
170 \bgroup\pardir TRT\textdir TRT\arabicfont%
171 \luadirect{tex.sprint(arabluatex.processnovoc(\luastringN{#2},
172 \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
173 \else%
174 \ifx\@tempa\al@mode@trans%
175 \bgroup\pardir TLT\textdir TLT\al@trans@font\al@trans@style%
176 \luadirect{tex.sprint(arabluatex.processtrans(\luastringN{#2},
177 \luastring0{\al@trans@convention},
178 \luastring0{\al@arb@rules},

```

```

179 \luastring0{\al@input@scheme}}))\egroup%
180 \else \fi\fi\fi\fi{\par}

```

arabverse 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.

```

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

```

Then follows the environment itself:

```

196 \NewDocumentEnvironment{arabverse}{!0{}}%
197 {\bgroup\setkeys[al]{verse}[width, gutter, color, utf, delim,
198   metre]{#1}%
199   \if@pkg@export\ifal@verse@export%
200   \ArbOutFile{\begin{arabverse}}%
201   % \ifx\al@mode\al@mode@trans%
202   % \luadirect{arabluatex.tooutfile(\luastringN{[#1]})}%
203   % \else%
204   % \IfSubStr[1]{#1}{utf}%
205   % {\luadirect{arabluatex.tooutfile(\luastringN{[#1]})}}%
206   % {\luadirect{arabluatex.tooutfile(\luastringN{[#1, utf]})}}%
207   % \fi
208   \else\fi\else\fi\egroup%
209   \par\centering\noindent\bgroup\setkeys[al]{verse}[metre]{#1}%
210   % \ifx\al@mode\al@mode@trans%
211   % \ifal@verse@utf\setRL\else\setLR\fi%
212   % \else\setRL\fi%
213   \ifal@verse@utf%
214   \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi%
215   \else%
216   \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi%
217   \fi%
218   \arab@v@export[#1]
219 }%
220 {\endarab@v@export
221   \hfill\setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
222   export]{#1}%

```

```

223 \egroup\par%
224 \bgroup\setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
225 metre]{#1}%
226 \if@pkg@export\ifal@verse@export%
227 \ArbOutFile{\end{arabverse}}
228 \else\fi\else\fi\egroup}

```

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

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

```

229 \NewDocumentCommand{\arb@utf}{m}{%
230 \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
231 \def\al@hemistich@delim{*}
232 \NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
233 \def\al@verse@stroke{\leavevmode\xleaders\hbox{\arb{--}}\hfill\kern0pt}
234 \NewDocumentCommand{\bayt}{s m o m}{%
235 \IfBooleanTF{#1}{\relax}{\relax}%
236 \ifdefined\savenotes\savenotes\else\fi%
237 \edef\al@tatweel{--}%
238 \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#2}}%
239 \IfNoValueTF{#3}{%
240 \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
241 \else%
242 \hspace{\al@gutter@width}%
243 \fi
244 }{%
245 \edef\@tempa{#3}%
246 \ifx\@tempa\al@tatweel%
247 \ifx\al@mode\al@mode@trans%
248 \hspace{\al@gutter@width}%
249 \else%
250 \makebox[\al@gutter@width][s]{\al@verse@stroke}%
251 \fi%
252 \else%
253 \ifx\al@mode\al@mode@trans%
254 \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#3}}%
255 \else%
256 \makebox[\al@gutter@width][s]{\arb@utf{#3}}%
257 \fi\fi}%
258 \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#4}}%
259 \ifdefined\spewnotes\spewnotes\else\fi%
260 }

```

`\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`.

```

261 \NewDocumentCommand{\SetDefaultIndex}{m}{
262   \edef\@tempa{#1}
263   \ifx\@tempa\empty
264     \def\al@default@index{\jobname}
265   \else
266     \def\al@default@index{#1}
267   \fi
268 }

269 \def\al@index@mode{\al@mode}
270 \NewDocumentCommand{\SetIndexMode}{m}{
271   \def\al@index@mode{#1}
272 }

273 \define@cmdkeys[al]{index}[alind@]{index,root,form}
274 \NewDocumentCommand{\arind}{o m}{%
275   \IfNoValueTF{#1}{%
276     \ifdefined\al@default@index%
277       \csname index\endcsname[\al@default@index]{#2}%
278     \else%
279       \csname index\endcsname{#2}%
280     \fi%
281   }{%
282     \bgroup
283     \setkeys[al]{index}{#1}%
284     \def\al@one{%
285       \ifdefined\alind@root!\LR{\alind@root}\else!\LR{1}\fi%
286     }
287     \def\al@two{%
288       \ifdefined\alind@form @\arb[\al@index@mode]{\alind@form}\else\fi%
289     }
290     \ifdefined\alind@index%
291       \csname index\endcsname[\alind@index]{#2\al@one\al@two}%
292     \else%
293       \ifdefined\al@default@index%
294         \csname index\endcsname[\al@default@index]{#2\al@one\al@two}%
295       \else%
296         \csname index\endcsname{#2\al@one\al@two}%
297       \fi%
298     \fi%
299   }
\egroup}}
```

`\abjad` `\abjad{⟨number⟩}` expresses its argument in Arabic letters in accordance with the *abjad* arrangement of the alphabet. `⟨number⟩` must be between 1 and 1999. It is now coded in Lua so that `polyglossia` is no longer needed. See `arabluatex.lua` for more information.

```

298 \AtBeginDocument{%
299   \ifdefined\abjad%
```

```

300 \RenewDocumentCommand{\abjad}{m}%
301 {\ifbool{al@rlmode}%
302   {\aoline*{%
303     \luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}
304   {\luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}
305 \else%
306 \NewDocumentCommand{\abjad}{m}%
307 {\ifbool{al@rlmode}%
308   {\aoline*{%
309     \luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}
310   {\luadirect{tex.sprint(arabluatex.abjadify(\luastring{#1}))}}}
311 \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.
312 \NewDocumentCommand{\ayah}{m}{%
313   \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.
314 \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.
315 \NewDocumentCommand{\abraces}{+m}{%
316   \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.
317 \DeclareDocumentCommand{\LRmarginpar}{o m}{%
318   \IfNoValueTF{#1}
319   {\marginpar{\textdir TLT #2}}
320   {\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.
321 \DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pdir
322   TLT\textdir TLT\footnote{#1}\egroup}
323 \DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pdir
324   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.
325 \NewDocumentCommand{\FixArbFtnmk}{}{%

```

```

326 \@ifpackageloaded{scrextend}%
327 {\AtBeginDocument{%
328     \deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%
329 {\RequirePackage{scrextend}
330  \AtBeginDocument{%
331     \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>}`.

```

332 \NewDocumentCommand{\SetArbOutSuffix}{m}{
333   \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.

```

334 \NewDocumentCommand{\ArbOutFile}{s O{no} +m}{%
335   \if@pkg@export%
336   \IfBooleanTF{#1}{%
337     #3\luadirect{arabluatex.tooutfile(\luastringN{#3}, "#2")}}{%
338     \luadirect{arabluatex.tooutfile(\luastringN{#3}, "#2")}}%
339   \else\IfBooleanTF{#1}{#3}{\fi}

```

arabexport 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.

```

340 \NewDocumentEnvironment{arabexport}{+b}{%
341   \if@pkg@export%
342   \par
343   #1
344   \luadirect{arabluatex.doexport("yes")}
345   \luadirect{tex.sprint(arabluatex.arbtoutf(\luastringN{#1}))}
346   \luadirect{arabluatex.doexport("no")}
347   \else\par#1\fi
348 }{\par}

```

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

```

349 \NewDocumentEnvironment{arab@v@export}{0{ } +b}{%
350   \setkeys[al]{verse}[width, gutter, color, utf, delim, mode,
351   metre]{#1}
352   \if@pkg@export\ifal@verse@export%

```

```

353 \par
354 #2
355 \luadirect{arabluatex.doexport("arabverse")}
356 \luadirect{tex.sprint(arabluatex.arbtoutf(\luastringN{#2}))}
357 \luadirect{arabluatex.doexport("no")}
358 \else\par#2\fi\else\par#2\fi
359 }\par}

```

`\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.

```

360 \NewDocumentCommand{\arbpardir}{}{%
361 \ifx\al@mode\al@mode@trans\setLR\else\setRL\fi}

```

Errors and Warnings

```

362 \newcommand{\al@warning}[1]{\PackageWarning{arabluatex}{#1}}
363 \newcommand{\al@error}[2]{\PackageError{arabluatex}{#1}{#2}}
364 \newcommand{\al@wrong@nesting}{\al@error{%
365 (RL/LR)\string\footnote\space is not allowed\MessageBreak inside
366 \string\RL{} and \string\RL{} commands}{%
367 Get rid of the surrounding \string\RL{} or \string\LR{} command.}}
368 \newcommand{\al@wrong@mark}{\al@warning{%
369 Unknown Arabic mark in \string\arbmark{}. Replaced
370 with\MessageBreak <??>. Please check your code}}

```

That is it. Say goodbye before leaving.

Patches

```

371 \NeedsTeXFormat{LaTeX2e}
372 \ProvidesPackage{arabluatex-patch}%
373 [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:

```

374 \def\list#1#2{%
375 \ifnum \@listdepth >5\relax
376 \toodeep
377 \else
378 \global\advance\@listdepth\@ne
379 \fi
380 \rightmargin\z@
381 \listparindent\z@
382 \itemindent\z@
383 \csname @list\romannumeral\the\@listdepth\endcsname
384 \def\@itemlabel{#1}%
385 \let\makelabel\mklab

```

```

386 \@nmbrlistfalse
387 #2\relax
388 \@trivlist
389 \parskip\parsep
390 \parindent\listparindent
391 \advance\linewidth -\rightmargin
392 \advance\linewidth -\leftmargin

patch begins:
393 \ifbool{al@rlmode}{\advance\@totalleftmargin \rightmargin}%
394 {\advance\@totalleftmargin \leftmargin}

patch ends.
395 \parshape \@ne \@totalleftmargin \linewidth
396 \ignorespaces}
397 \def\@item[#1]{%
398 \if@noperitem
399 \@donoperitem
400 \else
401 \if@inlabel
402 \indent \par
403 \fi
404 \ifhmode
405 \unskip\unskip \par
406 \fi
407 \if@newlist
408 \if@nobreak
409 \@nbitem
410 \else
411 \addpenalty\@beginparpenalty
412 \addvspace\@topsep
413 \addvspace{-\parskip}%
414 \fi
415 \else
416 \addpenalty\@itempenalty
417 \addvspace\itemsep
418 \fi
419 \global\@inlabeltrue
420 \fi
421 \everypar{%
422 \@minipagefalse
423 \global\@newlistfalse
424 \if@inlabel
425 \global\@inlabelfalse
426 {\setbox\z@\lastbox
427 \ifvoid\z@
428 \kern-\itemindent
429 \fi}%
430 \box\@labels
431 \penalty\z@
432 \fi

```



```

433 \if@nobreak
434 \@nobreakfalse
435 \clubpenalty \@M
436 \else
437 \clubpenalty \@clubpenalty
438 \everypar{}%
439 \fi}%
440 \if@noitemarg
441 \@noitemargfalse
442 \if@nmbrrlist
443 \refstepcounter\@listctr
444 \fi
445 \fi

patch begins:
446 \ifbool{al@rlmode}{\sRLbox\@tempboxa{\makelabel{#1}}}{%
447 \sbox\@tempboxa{\makelabel{#1}}}%
448 \ifbool{al@rlmode}{\global\setbox\@labels\hbox dir TRT}%
449 {\global\setbox\@labels\hbox}{%

patch ends.
450 \unhbox\@labels
451 \hskip \itemindent
452 \hskip -\labelwidth
453 \hskip -\labelsep
454 \ifdim \wd\@tempboxa >\labelwidth
455 \box\@tempboxa
456 \else
457 \hbox to\labelwidth {\unhbox\@tempboxa}%
458 \fi
459 \hskip \labelsep}%
460 \ignorespaces}

This is adapted from Vafa Khalighi's bidi package. Thanks to him.
461 \long\def\sRLbox#1#2{\setbox#1\hbox dir TRT{%
462 \color@setgroup#2\color@endgroup}}

```

References

- Brockelmann, Carl et al. (1935). “Die Transliteration der arabischen Schrift in ihrer Anwendung auf die Hauptliteratursprachen der islamischen Welt”. In: *Denkschrift dem 19. internationalen Orientalistenkongreß in Rom vorgelegt von der Transkriptionskommission der Deutschen Morgenländischen Gesellschaft*. In collab. with Ph. S. van Ronkel and Otto Spies. Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockhaus. URL: http://www.naher-osten.uni-muenchen.de/studium_lehre/werkzeugkasten/dmgtransliteration.pdf.
- Habash, Nizar Y. (2010). *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies 10. Toronto: Morgan & Claypool Publishers.
- Hosny, Khaled (Dec. 13, 2017). *Amiri*. URL: <http://www.amirifont.org/>.

- DIN 31 635 (July 2011). *Information and Documentation - Romanization of the Arabic Alphabet for Arabic, Ottoman-Turkish, Persian, Kurdish, Urdu and Pushto*. URL: <http://www.din.de>.
- Krüger, Marcel (Mar. 12, 2020). *The Lua-ul package. Underlining for LuaLaTeX*. Version 0.0.1. URL: <http://www.ctan.org/pkg/lua-ul>.
- Lagally, Klaus (Nov. 3, 2004). *ArabTeX. Typesetting Arabic and Hebrew*. User Manual Version 4.00. Version 4.00. URL: <http://mirrors.ctan.org/language/arabic/arabtex/doc/html/arabtex.htm>.
- Lane, Edward William (1863–1893). *An Arabic-English lexicon*. 8 vols. London – Edinburgh: Williams and Norgate.
- Wright, W. LL.D (1896). *A Grammar of the Arabic Language*. Rev. by W. Robertson Smith and M. J. de Goeje. With a forew. by Pierre Cachia. 3rd ed. 2 vols. Beirut: Librairie du Liban.

Change History

v1.0.	General: Initial release	1	v1.4.4.	\SetArbEasy*: this starred version discards the <i>sukūn</i> in addition to what is already discarded by \SetArbEasy.	62
v1.0.1.	General: Minor update of the documentation	1			
v1.1.	\abjad: New and more flexible \abjad command.	68	v1.5.	General: Compatibility with the quran package	56
v1.2.	\SetArbEasy: New \SetArbEasy/\SetArbDflt for ‘modern’ or ‘classic’ Arabic styles.	62		Environments may be nested inside the arab environment . .	52
v1.3.	\arbup: <i>ʾirāb</i> is now written as superscript text in dmg mode by default.	63		txarab : New txarab environment for typesetting running paragraphs in Unicode Arabic . .	64
v1.4.	\SetInputScheme: \SetInputScheme can be used to process other input schemes such as ‘Buckwalter’	62	v1.6.	arabverse : New environment arabverse for typesetting Arabic poetry	66
	\SetTranslitFont: For selecting a specific font for transliterated texts	62		\bayt: New macro \bayt for typesetting each verse inside the arabverse environment	67
v1.4.3.	\abraces: New \abraces command which expresses its argument between braces.	69		\SetArbDflt*: This starred version applies the assimilation rules in addition to what \SetArbDflt already does.	62
				\SetHemistichDelim: New \SetHemistichDelim command for changing the default delimiter between hemistichs	67

v1.7.	\arbnul1: New \arbnul1 command for putting back on any contextual analysis rule broken by other commands.	69	\MkArbBreak*: ‘starred’ version which closes Arabic environments before processing declared commands.	61	
v1.8.	General: arabica transliteration standard is now supported	41	v1.13.	arabexport: Processes and print its argument in the current file and exports it in full Unicode in the external selected .tex file. . .	70
v1.8.5.	General: Six additional Persian characters are now available . . .	11	arabverse: New options color and export to arabverse environment.	66	
v1.9.	\MkArbBreak: New \MkArbBreak command for inserting user-defined macros in Arabic environments	61	\arbmark: New optional argument: either rl or lr	65	
v1.9.2.	\aemph*: Starred version which always puts the stroke over its argument	61	\ArbOutFile: Silently exports its argument in the selected external file.	70	
v1.10.	General: \uc supersedes \cap	42	\arbpardir: Sets the direction of Arabic paragraphs once they are converted to Unicode.	71	
	\prname: New command for typesetting Arabic proper names in transliteration	63	\prname*: Renders proper names already converted to Unicode in upright roman style	63	
v1.11.	\arbmark: New command for inserting additional marks in Arabic environments	65	\SetArbOutSuffix: Sets a suffix to be appended to the filename of the external Unicode file.	70	
	\newarbmark: Allows defining additional sets of Arabic marks	65	v1.15.	\ayah: Prints End of Ayah sign . . .	69
v1.12.	General: \abjad can now process L ^A T _E X counters	27	v1.16.	\aemph: Now uses ulem	61
	\arbcolor: Standard color command for Arabic environments	62	v1.18.	\arind: New command \arind for building indexes	67
			v1.19.	\aemph: Now uses lua-ul	61
				\auline: Non context-sensitive command to underline Arabic words is provided	61

Index

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

Symbols	\@M	435	\@aoverLine	52, 55	
\@@aoline	53, 55	\@aoline	53, 54	\@aunderLine	56, 57

<code>\@beginparpenalty</code> . 411	<code>\al@arb@rules</code> 75, 78,	<code>\al@verse@stroke</code> ..
<code>\@clubpenalty</code> 437	79, 82, 83, 123, 233, 250
<code>\@donoparitem</code> 399	129, 135, 141,	<code>\al@warning</code> .. 362, 368
<code>\@ifpackageloaded</code> . 326	160, 166, 172, 178	<code>\al@wrong@mark</code> 368
<code>\@ifstar</code> 53	<code>\al@arbup</code>	<code>\al@wrong@nesting</code> . 364
<code>\@inlabelfalse</code> 425	. 92, 93, 94, 95, 97	<code>\alind@form</code> 287
<code>\@inlabeltrue</code> 419	<code>\al@arbup@dflt</code> 91, 92, 94	<code>\alind@index</code> .. 288, 289
<code>\@item</code> 397	<code>\al@bayt@width</code>	<code>\alind@root</code> 285
<code>\@itemlabel</code> 384 181, 183,	<i>Amiri</i> 5, 6
<code>\@itempenalty</code> 416	184, 185, 238, 258	amiri (package) 6
<code>\@labels</code>	<code>\al@default@index</code> .	<code>\aoline</code> 28, <u>52</u> , 302, 308
430, 448, 449, 450 264, 266,	<code>\aoline*</code> 28, <u>52</u>
<code>\@listctr</code> 443	276, 277, 291, 292	arab (environment) 8, <u>154</u>
<code>\@listdepth</code> 375, 378, 383	<code>\al@error</code> 363, 364	<code>\arab@v@export</code> 218
<code>\@minipagefalse</code> ... 422	<code>\al@gutter@width</code> ..	arab@v@export (environ-
<code>\@mklab</code> 385 182, 184,	ment) <u>349</u>
<code>\@nbitem</code> 409	186, 240, 242,	<i>ArabTeX</i> 4, 22
<code>\@ne</code> 378, 395	248, 250, 254, 256	arabexport (environ-
<code>\@newlistfalse</code> 423	<code>\al@chemistich@delim</code>	ment) ... 57, <u>340</u>
<code>\@nmbrrlistfalse</code> ... 386 231, 232, 240	<i>Arabic-English Lexicon,</i>
<code>\@nobreakfalse</code> 434	<code>\al@index@mode</code>	<i>An</i> 21
<code>\@noitemargfalse</code> .. 441 269, 271, 287	<code>\arabicfont</code>
<code>\@tempa</code> 118,	<code>\al@input@scheme</code> ..	32, 34, 36, 106,
119, 125, 131,	... 73, 74, 123,	112, 121, 127,
137, 155, 156,	129, 135, 142,	133, 158, 164, 170
162, 168, 174,	160, 166, 172, 179	arabluatex (package) .
245, 246, 262, 263	<code>\al@mode</code> 4, 5, 6, 7, 117, 2–6, 8, 9,
<code>\@tempboxa</code> 446,	154, 193, 201,	11–22, 26–29,
447, 454, 455, 457	210, 214, 216,	31, 33, 35–37,
<code>\@toodeep</code> 376	247, 253, 269, 361	41–46, 48–51,
<code>\@topsep</code> 412	<code>\al@mode@fullvoc</code> ..	55–63, 70, 71
<code>\@totalleftmargin</code> 17, 125, 162	arabtex (package) ...
.... 393, 394, 395	<code>\al@mode@novoc</code> 4, 5, 18, 22, 59
<code>\@trivlist</code> 388 18, 131, 168	arabulatex (package) .. 18
\sqcup 34	<code>\al@mode@trans</code>	arabverse (environ-
	.. 19, 137, 174,	ment) ... 28, <u>181</u>
	201, 210, 214,	arabxetex (package) . 5, 18
	216, 247, 253, 361	<code>\arb</code> 8, 103, <u>117</u> ,
	<code>\al@mode@voc</code> 16, 119, 156	187, 230, 233, 287
	<code>\al@one</code> 284, 289, 292, 294	<code>\arb@utf</code> 229,
	<code>\al@tatweel</code> .. 237, 246	238, 254, 256, 258
	<code>\al@trans@convention</code>	<code>\arbcOLOR</code> 36, <u>71</u>
	.. 88, 90, 140, 177	<code>\arbmark</code> .. 33, <u>145</u> , 369
	<code>\al@trans@font</code> 84, 85,	<code>\arbnulL</code> 25, <u>314</u>
	108, 116, 138, 175	<code>\ArbOutFile</code>
	<code>\al@trans@style</code> 58, 200, 227, <u>334</u>
 86, 87,	<code>\ArbOutFile*</code> ... 58, <u>334</u>
	108, 116, 138, 175	<code>\arbpardir</code> 58, <u>360</u>
<code>\aemph</code> 28, <u>52</u>	<code>\al@two</code> 286, 289, 292, 294	<code>\arbup</code> 43, <u>91</u>
<code>\aemph*</code> 28, <u>52</u>		<code>\ArbUpDflt</code> 43, <u>91</u>

A

<code>\abjad</code> 26, <u>298</u>
<code>\abraces</code> 33, <u>315</u>
<code>\addpenalty</code> .. 411, 416
<code>\addvspace</code> 412, 413, 417
<code>\adjustbox</code> 238, 254, 258
<code>\advance</code> 378,
391, 392, 393, 394
<code>\aemph</code> 28, <u>52</u>
<code>\aemph*</code> 28, <u>52</u>

<code>\arind</code>	261	<code>arabverse</code> ..	28, 181	<code>\ifbool</code>	301,
<code>\AtEndDocument</code>	12	<code>txarab</code>	48, 109		307, 393, 446, 448
<code>\auline</code>	28, 52	<code>txarabtr</code>	113	<code>\IfBooleanTF</code>	48,
<code>\ayah</code>	35, 312	<code>\everypar</code>	421, 438		60, 66, 77, 81,
B		<code>\ExecuteOptionsX</code> ...	14		103, 235, 336, 339
<code>babel</code> (package)	55	<code>export</code> (option) .	6, 29, 57	<code>\ifdim</code>	454
<code>babel-french</code> (package) .	55	F		<code>\ifhmode</code>	404
<code>\bayt</code>	29, 229	<code>\FixArbFtnmk</code> ...	50, 325	<code>\IfNoValueTF</code>	
<code>\begin</code>	200	<code>fontspec</code> (package) 5,	41, 62		72, 239, 275, 318
<code>bidi</code> (package)	73	<code>\footnote</code> 322,	324, 365	<code>\ifnum</code>	375
<code>\boolfalse</code>	39	<code>footnote</code> (package)	31	<code>\IfSubStr</code>	204
<code>\booltrue</code>	37,	<code>footnotehyper</code> (package) 31		<code>\ifvmode</code>	
	111, 121, 127,	<code>fullvoc</code> (option)	6, 9		105, 120, 126, 132
	133, 157, 163, 169	G		<code>\ifvoid</code>	427
<code>\box</code>	430, 455	<code>\global</code> ...	378, 419,	<code>\ignorespaces</code> .	396, 460
<code>Brockelmann, Carl</code> ...			423, 425, 448, 449	<code>\indent</code>	402
 11, 41, 43	<i>Grammar of the Arabic</i>		<i>Information and Docu-</i>	
C		<i>Language, A</i> ..		<i>mentation - Ro-</i>	
<code>\centering</code>	209	 6–8,	<i>manization of</i>	
<code>\clubpenalty</code> ..	435, 437		10, 14, 16–18,	<i>the Arabic Al-</i>	
<code>\color</code>	188		20, 26, 27, 43, 54	<i>phabet for Ara-</i>	
<code>color</code> (option)	29	<code>gutter</code> (option)	29	<i>bic, Ottoman-</i>	
<code>\color@endgroup</code> ...	462	H		<i>Turkish, Persian,</i>	
<code>\color@setgroup</code> ...	462	<code>Habash, Nizar Y.</code>	46	<i>Kurdish, Urdu</i>	
<code>csquotes</code> (package) ...	55	<code>\hbox</code>	233,	<i>and Pushto</i> ... 11	
D			448, 449, 457, 461	<i>Introduction to Arabic</i>	
<code>\DeclareOptionX</code> ...		<code>\height</code> ..	238, 254, 258	<i>Natural Lan-</i>	
 4, 5, 6, 7	<code>\hfill</code>	221, 233	<i>guage Processing</i> 46	
<code>\deffootnote</code> ..	328, 331	<code>Hosny, Khaled</code>	5, 6	<code>\itemindent</code> 382,	428, 451
<code>\define@boolkey</code> ...		<code>\hskip</code> 451,	452, 453, 459	<code>\itemsep</code>	417
	.. 8, 189, 190, 191	<code>\hspace</code>	242, 248	<code>\itshape</code>	86
<code>\define@choicekey</code> .	192	I		J	
<code>\define@cmdkeys</code> ...	273	<code>\if@inlabel</code> ..	401, 424	<code>\jobname</code>	264
<code>\define@key</code>		<code>\if@newlist</code>	407	K	
	185, 186, 187, 188	<code>\if@nmbrlist</code>	442	<code>\kern</code>	233, 428
<code>delim</code> (option)	29	<code>\if@nobreak</code> ..	408, 433	<code>KOMA-script</code> (package)	50
E		<code>\if@noitemarg</code>	440	<code>Krüger, Marcel</code>	35
<code>\empty</code>	263	<code>\if@nparitem</code>	398	L	
<code>\end</code>	227	<code>\if@pkg@export</code> 9,	199,	<code>\labelsep</code>	453, 459
<code>\endarab@v@export</code> .	220		226, 335, 341, 352	<code>\labelwidth</code> 452,	454, 457
<code>\ensuremath</code>	54	<code>\ifal@verse@delim</code> .	240	<code>Lagally, Klaus</code>	4, 22
<code>enumitem</code> (package)	53–55	<code>\ifal@verse@export</code>		<code>Lane, Edward William</code> .	21
<code>environments:</code>		 199, 226, 352	<code>\lastbox</code>	426
<code>arab</code>	8, 154	<code>\ifal@verse@utf</code> ...		<code>\leaders</code>	52, 56
<code>arab@v@export</code> ..	349	 211, 213, 230	<code>\leavevmode</code> ...	105,
<code>arabexport</code> ..	57, 340				120, 126, 132, 233
				<code>\leftmargin</code> ..	392, 394

lineno (package)	31		
\linewidth	391, 392, 395		
\list	374		
\listparindent	381, 390		
\long	461		
\LR	41, 49, 285, 328, 331, 367		
\LRfootnote . . .	49, 321		
\LRmarginpar . . .	50, 317		
lua-ul (package) . .	35, 61		
\luadirect . .	10, 12, 31, 49, 50, 61, 63, 67, 69, 99, 122, 128, 134, 139, 148, 152, 159, 165, 171, 176, 202, 205, 206, 303, 304, 309, 310, 313, 316, 333, 337, 338, 344, 345, 346, 355, 356, 357		
luaotfload (package) . . .	41		
\luastring	303, 304, 309, 310		
\luastringN	49, 50, 61, 63, 67, 69, 99, 122, 128, 134, 139, 148, 149, 152, 153, 159, 165, 171, 176, 202, 205, 206, 313, 316, 333, 337, 338, 345, 356		
\luastringO	123, 129, 135, 140, 141, 142, 160, 166, 172, 177, 178, 179		
M			
\makebox .	240, 250, 256		
\makelabel	385, 446, 447		
\marginpar . . .	319, 320		
\mbox	54		
metre (option)	29		
\MkArbBreak .	11, 47, 51		
\MkArbBreak*	47, 51		
mode (option)	28		
N			
nameauth (package) . . .	43		
\newarbmark . . .	33, 151		
\NewDocumentEnvironment	109, 113, 154, 196, 340, 349		
\newfontfamily	36		
\newlength . . .	181, 182		
\newunderlinetype	52, 56		
\NoArbUp	43, 91		
\noindent	209		
novoc (option)	6, 9		
O			
options:			
color	29		
delim	29		
export	6, 29, 57		
fullvoc	6, 9		
gutter	29		
metre	29		
mode	28		
novoc	6, 9		
trans	6, 9		
utf	29		
voc	6, 9		
width	28		
\overline	54		
P			
\PackageInfo	33		
\par . .	110, 112, 114, 116, 155, 180, 209, 223, 342, 347, 348, 353, 358, 359, 402, 405		
paralist (package) .	53, 55		
\pardir .	37, 39, 112, 115, 158, 164, 170, 175, 321, 323		
\parindent	390		
\parsep	389		
\parshape	395		
\parskip	389, 413		
\penalty	431		
polyglossia (package)	5, 68		
\presetkeys	194		
\prname	42, 101		
\prname* . . .	43, 58, 101		
\ProcessOptionsX . . .	15		
\providebool	30		
\ProvidesPackage . .	372		
Q			
quoting (package)	52		
quran (package)	56		
R			
\refstepcounter . . .	443		
reledmac (package) . . .	56		
\RequireLuaTeX	2		
\RequirePackage	1, 3, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 329		
\rightmargin	380, 391, 393		
\RL . . .	44, 49, 366, 367		
\RLfootnote . . .	49, 321		
\rmfamily	42, 43, 45, 46, 84		
\romannumeral	383		
S			
\savenotes	236		
\sbox	447		
scrxextend (package)	50, 69		
\SetArbDflt	7, 75		
\SetArbDflt*	7, 75		
\SetArbEasy	7, 75		
\SetArbEasy*	7, 75		
\SetArbOutSuffix	57, 332		
\SetArbUp	44, 91		
\setbox	426, 448, 449, 461		
\SetDefaultIndex . .	261		
\SetHemistichDelim	29, 229, 232		
\SetIndexMode	270		
\SetInputScheme	46, 73, 147		
\setkeys . .	197, 209, 221, 224, 283, 350		
\setlength	183, 184, 185, 186		
\setLR	39, 50, 211, 214, 216, 361		
\setRL . .	37, 50, 211, 212, 214, 216, 361		
\SetTranslitConvention	41, 88		

<code>\SetTranslitFont</code> 41, 84	<code>\thefootnotemark</code> ..	V
<code>\SetTranslitStyle</code> 328 , 331	<code>voc</code> (option) 6, 9
..... 41, 86 , 102	<code>trans</code> (option) 6, 9	<code>\vrule</code> 52 , 56
<code>\space</code> 365	<i>Transliteration der ara-</i>	
<code>\spewnotes</code> 259	<i>bischen Schrift,</i>	W
<code>\sRLbox</code> 446 , 461	<i>Die</i> ... 11, 41, 43	<code>\wd</code> 454
T	<code>txarab</code> (environment)	<code>width</code> (option) 28
<code>\textcolor</code> 72 48 , 109	Wright, W. LL.D 6–8,
<code>\textdir</code> 38 , 40 ,	<code>txarab</code> (environment) .. 56	10, 14, 16–18,
42 , 43 , 45 , 46 ,	<code>txarabtr</code> (environment)	20, 26, 27, 43, 54
106 , 108 , 112 , 113	
115 , 121 , 127 ,	<code>\txarb</code> 48 , 104 , 230	X
133 , 138 , 158 ,	<code>\txtrans</code> 103 , 104	<code>xcolor</code> (package) 36
164 , 170 , 175 ,	U	<code>\xleaders</code> 233
319 , 320 , 322 , 324	<code>\Uc</code> 100	Z
<code>\textsuperscript</code> ... 91	<code>\uc</code> ... 42 , 98 , 100 , 103	<code>\z@</code> 380 , 381 ,
<code>\textwidth</code> 183	<code>\unhbox</code> 450 , 457	382 , 426 , 427 , 431
<code>\the</code> 383	<code>\unskip</code> 405	
<i>The Lua-ul package</i> ... 35	<code>utf</code> (option) 29	