

The arabluatex package v1.7.2 – 2017/03/15

Robert Alessi
alessi@robertalessi.net*
CNRS UMR 8167 Paris (France)

Contents

License and disclaimer	2	4.8.2 The <i>abjad</i>	23
1 Introduction	3	4.9 Additional characters	24
1.1 arabluatex is for Lua ^A T _E X	4	4.10 Arabic emphasis	25
2 The basics of arabluatex	4	5 Arabic poetry	25
2.1 Activating arabluatex	4	5.1 Example	28
2.2 Options	5	6 Special applications	29
2.2.1 Classic contrasted with modern typesetting of Arabic	6	7 Transliteration	30
2.3 Typing Arabic	7	7.1 Additional note on <code>dmg</code> convention .	31
2.3.1 Local options	8	7.2 Examples	32
3 Standard Arab^TE_X input	8	8 Buckwalter input scheme	33
3.1 Consonants	8	9 Unicode Arabic input	35
3.2 Vowels	9	10 L^AT_EX Commands in Arabic environments	36
3.2.1 Long vowels	9	10.1 Environments	38
3.2.2 Short vowels	10	10.1.1 Lists	39
4 arabluatex in action	11	10.2 csquotes	41
4.1 The vowels and diphthongs	11	10.3 reledmac	41
4.2 Other orthographic signs	13	10.4 quran	42
4.3 Special orthographies	18	11 Future work	42
4.4 Quoting	19	12 Implementation	43
4.4.1 Quoting the <i>hamzah</i>	21	References	52
4.5 The “pipe” character ()	21	Change History	52
4.6 Putting back on broken contextual analysis rules	22	Index	53
4.7 Stretching characters: the <i>taṭwīl</i> .	23		
4.8 Digits	23		
4.8.1 Numerical figures	23		

*Institutional email: robert.alessi@cnrs.fr

List of Tables

1	ArabTeX consonants	9	4	“Quoted” <i>hamzah</i>	21
2	ArabTeX long vowels	10	5	Additional Arabic codings	24
3	ArabTeX short vowels	10	6	Buckwalter scheme	34

Abstract

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

License and disclaimer

License applicable to the software

`arabluatex` — Processing ArabTeX notation under Lua[®]TeX.

Copyright © 2016–2017 Robert Alessi

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

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

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

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

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

This release of `arabluatex` consists of the following source files:

- `arabluatex.ins`
- `arabluatex.dtx`
- `arabluatex.lua`
- `arabluatex_voc.lua`
- `arabluatex_fullvoc.lua`
- `arabluatex_novoc.lua`

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

License applicable to this document

Copyright © 2016–2017 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 ArabTeX,¹ ArabLuaTeX is at present nothing more than a modest piece of software. Hopefully—if I may say so—it will eventually provide all of its valuable qualities to the LuaL^ATeX users.

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

The great advantage of ArabTeX notation is that it is immune from all these difficulties, let alone its being clear and straightforward. One also must remember that computers are designed to process code. ArabTeX notation is a way of encoding Arabic language, just as T_EX “mathematics mode” is a way of processing code to display mathematics. As such, not only does it allow greater control over typographical features, but it also can be processed in several different ways: so without going into details, depending on one’s wishes, ArabTeX input can be full

¹ See <http://ctan.org/pkg/arabtex>

² Lagally (2004, p. 2).

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

vocalized Arabic (*scriptio plena*), vocalized Arabic or non-vocalized Arabic (*scriptio defectiva*); it further can be transliterated into whichever romanization standard the user may choose.

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

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

1.1 `arabluatex` is for Lua^LA^TE^X

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

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

2 The basics of `arabluatex`

2.1 Activating `arabluatex`

As usual put in your preamble:

```
\usepackage{arabluatex}
```

The only requirement of `arabluatex` is Lua^LA^TE^X; it will complain if you try to compile your document with another engine. That aside, `arabluatex` does not load packages such as `polyglossia` or `luabidi`. It can work with `polyglossia` though, but does not require it.

⁴To date, both packages support Arabic, Maghribi, Urdu, Pashto, Sindhi, Kashmiri, Uighuric and Old Malay; in addition to these, `arabxetex` also has a Hebrew mode, including Judeo-Arabic and Yiddish.

Font setup If you wish to use your own Arabic font, you can define it before loading `arabluatex`. Assuming that `fontspec` is loaded, put this in your preamble just above the line that loads `arabluatex`:

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

where `<fontname>` is the standard name of the Arabic font you wish to use.

By default, if no Arabic font is selected, `arabluatex` will issue a warning message and attempt to load the Amiri font⁵ like so:—

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

REM. By default Amiri places the *kasrah* in combination with the *tašdīd* below the consonant, like so: `ك`. That is correct, as at least in the oldest manuscripts `ك` may stand for `ك` as well as `ك`. See Wright (1896, i.14.C–D). The placement of the *kasrah* above the consonant may be obtained by selecting the `ss05` feature of the Amiri font, like so:—⁶

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

Other Arabic fonts may behave differently.

2.2 Options

`arabluatex` may be loaded with four mutually exclusive global options, each of which may be overridden at any point of the document (see below section 2.3.1 on page 8):

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

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

novoc None of the diacritics is showed in **novoc** mode, unless otherwise specified (see “quoting” technique below section 4.4 on page 19).

trans This mode transliterates the ArabT_EX input into one of the accepted standards. At present, two standards are supported (see below section 7 on page 30 for more details):

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

loc *Library of Congress*.

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

⁵Hosny (2015).

⁶See the documentation of `amiri`, Hosny (2015, p. 5).

2.2.1 Classic contrasted with modern typesetting of Arabic

New feature
v.1.2

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

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

samA'uN سَمَاءُ ^{un}samā, jA'a جَاءَ ^ajā'a, yatasA'alUna يَتَسَاءَلُونَ ^ayatasā'alūna⁷
(see on page 14 for further details).

- The euphonic *tašdīd* is generated (see on page 15).
- In *fullvoc* mode, the *sukūn* is expressed.

New feature
v1.4.4

\SetArbEasy

Such refinements as ‘classic’ *maddah* may be discarded by the \SetArbEasy command, either globally in the preamble or locally at any point of the document. The difference between \SetArbEasy and its ‘starred’ version \SetArbEasy* is that the former keeps the *sukūn* that is generated by the *fullvoc* mode, while the latter further takes it away. Default ‘classic’ rules may be set back at any point of the document with the \SetArbDflt command. Assimilation rules laid on item b on page 15 may also be applied by the ‘starred’ version of this command \SetArbDflt* either in the preamble or at any point of the document.⁸ Examples follow:—

New feature
v1.6

\SetArbEasy*
\SetArbDflt

\SetArbDflt*

(a) \SetArbDflt:

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

(b) \SetArbDflt*:

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

(c) \SetArbEasy:

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

(d) \SetArbEasy*:

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

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

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

⁸For an example, see section 5.1 on page 28.

2.3 Typing Arabic

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

```

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

From Wright (1896, i. 1 A):— Arabic, like Hebrew and Syriac, is written and read from right to left. The letters of the alphabet (حُرُوفُ الْهِجَاءِ, حُرُوفُ الْهِجَائِيَّةِ, or حُرُوفُ الْمُعْجَمِ) are twenty-eight in number and are all consonants, though three of them are also used as vowels (see § 3).

Caveat For some reason, left-to-right paragraphs that start with Arabic words lose their indentation. For the time being, this can be circumvented by appending the `\indent` command at the commencement of such paragraphs.

The same remark applies to left-to-right list environments: when items start with Arabic words, the `\arb` command must be prefixed with `\indent`. The following example comes from Wright (1896, i. 213 C):—

```

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

XVI. فَوَاعِلُ*.

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

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

```

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

like so:—

```

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

```

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

2.3.1 Local options

As seen above in section 2.2 on page 5, arabuatex may be loaded with four mutually exclusive global options: `voc` (which is the default option), `fullvoc`, `novoc` and `trans`. Whatever choice has been made globally, it may be overridden at any point of the document, as the `\arb` command may take any of the `voc`, `fullvoc`, `novoc` or `trans` modes as optional arguments, like so:—

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

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

3 Standard ArabTeX input

3.1 Consonants

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

Letter	Transliteration ⁹		ArabT _E X notation
	dmg	loc	
ا ¹⁰	<i>a</i>	<i>a</i>	a
ب	<i>b</i>	<i>b</i>	b
ت	<i>t</i>	<i>t</i>	t
ث	<i>ṭ</i>	<i>th</i>	_t
ج	<i>ǧ</i>	<i>j</i>	^g or j
ح	<i>ḥ</i>	<i>ḥ</i>	.h
خ	<i>ḫ</i>	<i>kh</i>	_h or x
د	<i>d</i>	<i>d</i>	d
ذ	<i>ḍ</i>	<i>dh</i>	_d
ر	<i>r</i>	<i>r</i>	r
ز	<i>z</i>	<i>z</i>	z
س	<i>s</i>	<i>s</i>	s
ش	<i>š</i>	<i>sh</i>	^s
ص	<i>ṣ</i>	<i>ṣ</i>	.s
ض	<i>ḍ</i>	<i>ḍ</i>	.d
ط	<i>ṭ</i>	<i>ṭ</i>	.t
ظ	<i>ẓ</i>	<i>ẓ</i>	.z
ع	<i>ʿ</i>	<i>ʿ</i>	`
غ	<i>ǧ</i>	<i>gh</i>	.g
ف	<i>f</i>	<i>f</i>	f
ق	<i>q</i>	<i>q</i>	q
ك	<i>k</i>	<i>k</i>	k
ل	<i>l</i>	<i>l</i>	l
م	<i>m</i>	<i>m</i>	m
ن	<i>n</i>	<i>n</i>	n
ه	<i>h</i>	<i>h</i>	h
و	<i>w</i>	<i>w</i>	w
ي	<i>y</i>	<i>y</i>	y
ة	<i>ah</i>	<i>ah</i>	T

Table 1: Standard ArabT_EX (consonants)

3.2 Vowels

3.2.1 Long vowels

Table 2 gives the ArabT_EX equivalents for the Arabic long vowels.

⁹See below section 7 on page 30.

¹⁰For *ʾalif* as a consonant, see Wright (1896, i. 16 D). The *hamzah* itself is encoded <'>. See below section 4.2 on page 13.

Letter	Transliteration ¹¹		ArabT _E X notation
	dmg	loc	
ا	\bar{a}	\bar{a}	A
و	\bar{u}	\bar{u}	U
ي	\bar{i}	\bar{i}	I
آ ¹²	$\bar{\bar{a}}$	\acute{a}	_A or Y
أ	\bar{a}	\bar{a}	_a
و	\bar{u}	\bar{u}	_u
ي	\bar{i}	\bar{i}	_i

Table 2: Standard ArabT_EX (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*, قَالَ يَقُولُ *qālu*.

REM. *b.* Defective writings, such as ا, *al-'alif^u 'l-mahḍūfat^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 اِبْنُ حَذِيفَةَ for *Hudayfat^u bn^u 'l-Yamānī*, etc.

3.2.2 Short vowels

Table 3 gives the ArabT_EX equivalents for the Arabic short vowels.

Letter	Transliteration ¹³		ArabT _E X notation
	dmg	loc	
ا	<i>a</i>	<i>a</i>	a
و	<i>u</i>	<i>u</i>	u
ي	<i>i</i>	<i>i</i>	i
ان	<i>an</i>	<i>an</i>	aN
ون	<i>un</i>	<i>un</i>	uN
ين	<i>in</i>	<i>in</i>	iN

Table 3: Standard ArabT_EX (short vowels)

¹¹ See below section 7 on page 30.

¹² = *al-'alif^u 'l-maḡṣūrat^u*.

¹³ See below section 7 on page 30.

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.¹⁴

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*, *damma* 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 19);
- when `arabluatex` needs to perform a contextual analysis to determine the carrier of the *hamzah*;
- in the various transliteration modes, as vowels are always expressed in romanized Arabic.

4 arabluatex in action

4.1 The vowels and diphthongs

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

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

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

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

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

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

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

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

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

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

¹⁴See below section 4.4 on page 19.

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

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

.hay_aUTuN حَيوة *hayāt^{un}*, .sal_aUTuN صَلوة *ṣalāt^{un}*, mi^sk_aUTuN مَشْكُوة *miškāt^{un}*,
 tawr_aITuN تَوْرِيَّة *tawrāt^{un}*.
 And so also: al-rib_aIT-u الرِّبِيَّة *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.ḥammadiN عَمْرُو بْنُ مُحَمَّدٍ *Amr^u bn^u Muḥammadⁱⁿ*, mu.ḥammad-u bn-u
 `amr-iU bn-i_hAlidiN عَمْرُو 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. arablutex 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 ⟨aNy⟩:

mAluN مَالٌ *māl^{un}*, bAbaN أَبَا *bāb^{an}*, madInaTaN مَدِينَةٌ *madīnat^{an}*, bintiN
 بِنْتُ *bintⁱⁿ* maqhaN_A مَقْهَى *maqḥaⁿ*, fataNY فَتًى *fataⁿ*.
 arablutex is aware of special orthographies: ^say'uN شَيْءٌ *šay^{un}*,
 ^say'aN شَيْئًا *šay^{an}*, ^say'iN شَيْءٍ *šayⁱⁿ*.

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

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

4.2 Other orthographic signs

tā' marbūṭah It is written ⟨*T*⟩:

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

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

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

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

hamzah followed by the long vowel ي is encoded '_I: '_ImAnuN إِيْمَانٌ *īmā-n^{un}*¹⁵.

Middle hamzah: xA.ti'-Ina خَاطِئِينَ *ḫāṭi'īna*, ru'UsuN رُؤُوسٌ *ru'ūs^{un}*, xa.tI'aTuN خَاطِئَةً *ḫāṭi'at^{un}*, su'ila سُئِلَ *su'ila*, 'as'ilaTuN 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}*, ta'xIruN تَأْخِيرٌ *ta'ḫīr^{un}*, ta'ax xara تَأَخَّرَ *ta'ahḫara*, ji'tu-ka جِئْتُكَ *gi'tu-ka*, qA'iluN قَائِلٌ *qā'il^{un}*, .hIna'i_diN هِنْدٌ *ḥīndⁱⁿ*.

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

¹⁵ For another way of encoding the initial *hamzah* followed by a long vowel, see the *taḥfīf^u 'l-hamzat* on the next page.

Final hamzah: xa.ta'uN خَطَأُ ḥaṭa^{'un}, xa.ta'aN خَطَأٌ ḥaṭa^{'an}, xa.ta'iN خَطِئَ ḥaṭa^{'in}, 'aqra'u أَقْرَأُ 'aqra^{'u}, taqra'Inا تَقْرَأُ taqra^{'ina}, taqra'Una تَقْرَأُونَ taqra^{'ūna}, yaqra'na يَقْرَأُ yaqra^{'na}, yaxba'Anي يَخْبَأُ yaḥba^{'āni}, xaba'A خَبَأَ ḥaba^{'ā}, xubi'a خُبِئَ ḥubi^{'a}, xubi'UA خُبُوا ḥubi^{'ū}, jA'a جَاءَ ḡā^{'a}, ridA'uN رَدَأَ ridā^{'un}, ridA'aN رَدَأَ ridā^{'an}, jI'a جِئَ ḡī^{'a}, radI'iN رَدِئَ radī^{'in}, sU'uN سُوءٌ sū^{'un}, .daw'uN دَاوٌ daw^{'un}, qay'iN قَيَّ qay^{'in}, ~sifa'I شَفَائِي šifā^{'i}, man~sa'I مَنْشَأِي manša^{'i}.

~say'uN سَائِي šay^{'un}, ~say'aN سَائِي šay^{'an}, ~say'iN سَائِي šay^{'in}, al-~say'-u أَشْيَاءُ aš-šay^{'u}, 'a~syA'-u أَشْيَاءُ ašyā^{'u}, 'a~syA'-a أَشْيَاءُ ašyā^{'a}, .zim'aN ظِمًّا zim^{'an}.

taḥfif^u 'l-hamzatⁱ: if the *hamzah* has *ḡazmah* and is preceded by *'alif hamzatūm*, it must be changed into the letter of prolongation that is homogeneous with the preceding vowel; hence: 'a'mana آمَنَ āmana, 'u'minu أُمِنُ ūminu, 'i'manuN إِمَانٌ imān^{un}. For other possible ways of encoding such sequences, see on page 13 (*hamzah* followed by و and ي) and the *maddah* on the current page.

maddah At the beginning of a syllable, *'alif* with *hamzah* and *fathah* (أ) followed by *'alifu 'l-maddi* (*'alif* of prolongation) or *'alif* with *hamzah* and *ḡazmah* (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 أَكَلُ ākil^{un} respectively.

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

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

jA'a جَاءَ ḡā^{'a}, yatasA'alUna يَتَسَاءَلُونَ yatasā'alūna, ridA'uN رَدَأَ ridā^{'un}, xaba'A خَبَأَ ḥaba^{'ā}, yaxba'Anي يَخْبَأُ yaḥba^{'āni}.

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

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

ʿallaqa عَلَقَ *ʿallaqa*, mAdduN مَادَّ *mādd^{um}*, 'ammara أَمَرَ *ammara*, murruN مَرَّ *murr^{un}*.

The euphonic tašdīd always follows a vowelless consonant which is passed over in pronunciation and assimilated to a following consonant. It may be found (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-ʿsams-u الشَّمْسُ *aš-šams^u*, or aʿs-ʿsams-u الشَّمْسُ *aš-šams^u*.

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

- (b) With the letters ر, ل, م, و, ي after ن with *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 رَبِّيهِ *mir rabbi-hi*, min layliN مِنْ لَيْلٍ *mil laylⁱⁿ*, 'an yaqtula أَنْ يَقْتُلَ *ay yaqtula*.

With *tanwīn*: kitAbuN مِيبُنْ *kitāb^{um} mubīn^{un}*.

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

- (c) With the letter ت after the dentals ث, د, ذ, 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 'l-waṣlⁱ At the beginning of a sentence, اَلْ is never written, as اَلْحَمْدُ لِلّٰهِ; instead, to indicate that the 'alif is a connective 'alif ('alif^u 'l-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-llāhⁱ*.

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

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

Article: bAb-u 'l-madrasaT-i الْمَدْرَسَةُ بِأَبْ *bāb^u 'l-madrasatⁱ*, al-maqA laT-u 'l-'_U1_A الْمَقَالَةُ الْأُولَى *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 إِلَى الْأَيْتِقَاضِ *ilā 'l-intiqāḍⁱ*, fI 'l-ibtidā'-i فِي الْإِبْتِدَاءِ *fi '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-waṣli* is omitted in the article أَلْ when it is preceded by the preposition لِ: li-l-rajul-i لِلرَّجُلِ *li-r-raġulⁱ*.
If the first letter of the noun be ل, then the ل of the article also falls away, but `arabluatex` is aware of that: li-l-laylaT-i لِلَّيْلَةِ *li-l-laylatⁱ*.
- (b) *la-*: the same applies to the affirmative particle لَ: la-l-.haqq-u لِلْحَقِّ *la-l-ḥaqq^u*.
- (c) With the other particles, *'alif^u 'l-waṣli* is expressed: fI 'l-madIna T-i فِي الْمَدِينَةِ *fi '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 قَالَ أَقْتُلُ *qāla 'qtul*, huwa inhazama هُوَ أَنْهَزَمَ *huwa 'nhazama*, wa-ustu`mila وَأَسْتَعْمِلُ *wa-'stu`mila*, qad-i in.sarafa قَدْ أَنْصَرَفَ *qadi 'nṣarafa*, al-iqtidAr-u الْأَقْتِدَارُ *al-iqtidār^u*, 'il_A 'l-inti qA.d-i إِلَى الْأَيْتِقَاضِ *ilā 'l-intiqāḍⁱ*, law istaqbala لَوْ اسْتَقْبَلَ *lawi 'staqbala*.

Other cases: 'awi ismu-hu أو اسمه 'awi 'smu-hu, zayduN ibn-u `amriNU
 زَيْدُ بْنُ عَمْرٍو Zayd^{uni} 'bn^u 'Amrⁱⁿ,¹⁶ `umar-u ibn-u 'l-_ha.t.tAb-i عُمَرُ
 ابْنُ الْخَطَّابِ Umar^u 'bn^u 'l-Haṭṭābⁱ,¹⁷ imru'-u 'l-qays-i اِمْرُؤُ الْقَيْسِ Imru^u
 'l-Qaysⁱ, la-aymun-u 'l-l_ah-i لَا يُؤْمِنُ اللَّهُ la-'ymun^u 'l-lāhⁱ.

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

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

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

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

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

'antumU 'l-kA_dib-Una اَنْتُمْ الْكَاذِبُونَ 'antumU 'l-kāḏib^{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 *ḍammah* which may be taken by the nouns having *tanwīn* although it is explicit in pronunciation and must appear in some transliteration standards. arabluatex takes care of that automatically:—

¹⁶ “Zayd is the son of ‘Amr”: the second noun is not in apposition to the first, but forms part of the predicate. Hence زَيْدُ بْنُ عَمْرٍو and not عَمْرٍو بْنُ زَيْدٍ, “Zayd, son of ‘Amr”.

¹⁷ “Umar is the son of al-Haṭṭāb” (see footnote 16).

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

4.3 Special orthographies

The name of God The name of God, اَللّٰهُ, is compounded of the article اَلْ, and اِلَٰه (noted اِله with the defective *ʿalif*) so that it becomes اِلَٰلَٰه; then the *hamzah* is suppressed, its vowel being transferred to the ل before it, so that there remains اَللّٰهُ (I refer to Lane, *Lexicon*, I. 83 col. 1). Finally, the first ل is made quiescent and incorporated into the other, hence the *tašdīd* above it. As *arabluatex* never requires a solar letter to be written twice (see above, on page 15), the name of God is therefore encoded al-l_ah-u or 'l-l_ah-u:—

al-l_ah-u اَللّٰهُ *al-lāh^u*, yA¹⁸ al-l_ah-u يَا اَللّٰهُ *yā al-lāh^u*, 'a-fa¹⁹-al-l_ah-i la-ta.g`alanna أَفَاللّٰهُ لَتَعْلَنَ *'a-fa-al-lāhⁱ la-tag'alanna*, bi-'l-l_ah-i بِاللّٰهِ *bi-'l-lāhⁱ*, wa-'l-l_ah-i وَاللّٰهِ *wa-'l-lāhⁱ*, bi-sm-i 'l-l_ah-i بِسْمِ اللّٰهِ *bi-smⁱ 'l-lāhⁱ*, al-.hamd-u li-l-l_ah-i اَلْحَمْدُ لِلّٰهِ *al-ḥamd^u li-llāhⁱ*, li-l-l_ah-i 'l-qA'il-u اَللّٰهُ الْقَائِلُ *li-llāhⁱ 'l-qā'il^u*.

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

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

¹⁸Note the “pipe” character ‘|’ here after yA and below after fa before footnote mark 19: 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 21 on the “pipe” and section 7 on page 30 on *dmg* mode.

¹⁹See footnote 18.

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 جَهْطَةً رَوَيْنَاهُمَا عَنِ الصَّوْتَيْنِ الَّذِينَ نَذْكُرُ فَإِنَّا نَذْكُرُ الصَّوْتَيْنِ الَّذِينَ رَوَيْنَاهُمَا عَنِ جَهْطَةٍ
'l-la_dayni rawaynā-humā `an Ġahẓat^a.

And also: al-la_dAni اللّٰذَانِ al-ladāni, al-la_dayni اللّٰذَيْنِ al-ladayni,
al-latAni اللّٰتَانِ al-latāni, al-latayni اللّٰتَيْنِ al-latayni, al-lAtI اللّٰتِي al-lātī,
al-lA' |Ati²⁰ اللّٰءَاتِ al-lā'ātī, al-lA' I اللّٰئِي al-lā'ī, and so forth.

4.4 Quoting

It is here referred to “quoting” after the arabtex package.²¹ The “quoting” mechanism of arabluatex is designed to be very similar in effect to the one of arabtex.

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

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

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

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

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

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

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

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

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

²⁰Note here the “pipe” character '|': as already stated on page 14, 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 21.

²¹See Lagally (2004, p. 22)

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

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

wa-"ust"u`mila وَأَسْتَعْمَل *wa-'stu'mila*, huwa "inhazama هو أَنْهَزَم *huwa*
'nhazama, al-"intiqA.d-u الِإِنْتِقَاضُ *al-intiqāḍ^u*.

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

- (a) Short and long vowels:—

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

- (b) *tanwīn*:—

madInaT"aN مَدِينَةٌ *madīnat^{an}*, bAb"aN أَبَا *bāb^{an}*, hud"aN_A هُدًى *hudāⁿ*,
 شَيْءٌ *ṣay' "iN 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':—

fI "istisqA'-a أَتَسْقَاءُ فِي *'stisqā'a*, wa-"istisqA'-u وَأَسْتَسْقَاءُ *wa-'s-*
tisqā'u, qAla "uhrub fa-lan tuqtala أَهْرُبُ فَلَنْ تُقْتَلَ *qāla 'hrub fa-*
lan tuqtala.

- (b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala أَقْتُلْ فَلَنْ تُقْتَلَ *qāla 'qtul fa-lan*
tuqtala, mA jA'at" mini imra'aTIN مِمَّا جَاءَتْ مِنْ امْرَأَةٍ *mā gā'at mini*
 'mra'atⁱⁿ, kam" qad" ma.dat" min" laylaTiN كَمْ قَدْ مَضَتْ مِنْ لَيْلَةٍ *kam*
qad maḍat min laylatⁱⁿ.

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

al-jamr-u 'l-.sayfiyy-u 'lla_dI kAna bi-q"rAn" |nUn-a أَلْجَمْرُ الصِّفِيُّ
 الَّذِي كَانَ يَقْرَأُونُ *al-ğamr^u 'ṣ-ṣayfiyy^u 'lladī kāna bi-Qrānnūn^a*.

4.4.1 Quoting the hamzah

As said above in section 4.2 on page 13, the *hamzah* is always written ⟨'⟩, its carrier being determined by contextual analysis. “Quoting” that straight single quote character like so: ⟨"'"⟩ allows to determine the carrier of the *hamzah* freely, without any consideration for the context. Table 4 gives the equivalents for all the possible carriers the *hamzah* may take.

Letter	Transliteration ²²		ArabTeX notation
	dmg	loc	
ء	’	’	"'
آ	ā	’ā	A"'
أ	’	’	a"'
إ	’	’	u"'
ؤ	’	’	w"'
إِ	’	’	i"'
يِ	’	’	y"'

Table 4: “Quoted” *hamzah*

As one can see from table 4, the carrier of the *hamzah* is inferred from the letter that precedes the straight double quote ⟨"⟩. Of course, any “quoted” *hamzah* may take a short vowel, which is to be written *after* the ArabTeX equivalent for the *hamzah* itself, namely ⟨'⟩. For example, ؤ is encoded ⟨w" 'a⟩, while ؤُ 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 19–20.

'a`dA'ukum أَعدَّاءُكُمْ 'a`dā'ukum, 'a`dA|"'"'ukum أَعدَّاءُكُمْ 'a`dā'ukum, 'a`dA'ikum أَعدَّاءُكُمْ 'a`dā'ikum, 'a`dA|"'"'ikum أَعدَّاءُكُمْ 'a`dā'ikum.

4.5 The “pipe” character (|)

In the terminology of ArabTeX, the “pipe” character ‘|’ is referred to as the “invisible consonant”. Hence, as already seen above in section 4.4.1, its usage to encode the *hamzah* alone, with no carrier: |"'" ء.

Aside from that usage, the “pipe” character is used to prevent almost any of the contextual analysis rules that are described above from being applied. Two examples have already been given to demonstrate how that particular mechanism works in footnote 18 on page 18 and in footnote 20 on page 19. One more example follows:—

bi-qrAn|nUn-a بِقِرَآنُونْ^a bi-Qrānnūn^a, “in Crannon” (Thessaly, Greece).²³

²²See below section 7 on page 30.

²³See more context on page 20.

As one can see, the “pipe” character between the two $\langle n \rangle$ prevents the necessary *tašdīd* rule (page 14) from being applied.

4.6 Putting back on broken contextual analysis rules

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

```

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

```

This is wrong:

^aA footnote that interferes with the contextual analysis.

فَلَمَّا رَأَوْا^a النَّجْمَ...

According to the rule stated on page 17, the diphthong in *ra'aw* must be resolved into two simple vowels before the *'alif^a l-waṣlⁱ*, as رَأَوْا النَّجْمَ.

`\arbnul`

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

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

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

```

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

```

```

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

```

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

Zayd^{unie} 'bn^u 'Amrⁱⁿ.^f

- ^aA footnote that interferes with the contextual analysis.
^bA footnote that interferes with the contextual analysis.
^cA footnote that interferes with the contextual analysis.
^dSee footnote 16 on page 17.
^eA footnote that interferes with the contextual analysis.
^fSee footnote 16 on page 17.

4.7 Stretching characters: the taṭwīl

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

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

4.8 Digits

4.8.1 Numerical figures

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

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

4.8.2 The abjad

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

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

New feature
v.1.1

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

REM. *a.* As can be seen in the above given example, `arabluatex` expresses the *abjad* numbers in Roman numerals if it finds the `\abjad` command in any of the transliteration modes.

REM. *b.* `\abjad` may also be found outside Arabic environments. In that case, `arabluatex` does not print the stroke as a distinctive mark over the number for it is not surrounded by other Arabic words. In case one nonetheless wishes to print the stroke, he can use the `\aemph` command that is described below in section 4.10 on the next page:—

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

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

4.9 Additional characters

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

For the time being, the following is included:—

Letter	Transliteration ²⁴		ArabTeX notation
	dmg	loc	
ب	<i>b</i>	<i>b</i>	.b
د	<i>d</i>	<i>d</i>	˘d
ف	<i>f</i>	<i>f</i>	.f
ق	<i>q</i>	<i>q</i>	.q
ك	<i>k</i>	<i>k</i>	.k
ن	<i>n</i>	<i>n</i>	.n
⸰	((((
⸱))))

Table 5: Additional Arabic codings

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

²⁴See below section 7 on page 30.

4.10 Arabic emphasis

As already seen in section 4.8.2 on page 23, the *ʿabjad* numbers are distinguished from the surrounding words by a stroke placed over them. This technique is used to distinguish further words that are proper names or book titles.

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

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

5 Arabic poetry

New feature
v1.6

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

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

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

The description of the optional arguments follows:—

`mode` `mode=<mode>`, either `voc`, `fullvoc`, `novoc` or `trans`. The default mode is the one that is set at load time as already seen section 2.2 on page 5.

`width` `width=<length>` Default: `0.3\linewidth`
The default width of each hemistich that the verse consists of. It may be expressed in any accepted unit of measurement, such as 4cm or 2in. However, one must keep in mind that the total length of the two hemistichs added to the one of the gutter that separates them must not exceed the length of the base line, unless one wishes to have the hemistichs distributed on subsequent lines.

`gutter` `gutter=<width>` Default: `0.15 x (hemistich width)`
The gutter consists of the blank space that is between the two hemistichs. By default, it is commensurate with the width of the hemistich, but it may be expressed in any accepted unit of measurement as well.

`metre` `metre=<name>` Default: `none`
If the name of the metre is expressed, it is printed after the lines and set flush left in `voc`, `fullvoc` and `novoc` modes or flush right in `trans` mode.

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

`\SetHemistichDelim`

`utf` `utf=true|false` Default: `false`

As the preceding one, this named argument does not need a value as it defaults to `true` if it is used. If so, unicode Arabic input is expected in the `arabverse` environment instead of ASCII ArabTeX or Buckwalter input schemes. See section 9 on page 35 for more details.

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

```
\bayt{<ṣadr>}[<tadwīr>] {<ağuz>}\\
```

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

- (a) If the two hemistichs be connected with one another by a prominent horizontal flexible stroke, the *taṭwīl* should be used, like so: `[--]` (see section 4.7 on page 23). Of course, the ending word of the *ṣadr* and the word at the commencement of the *ağuz* must have the *taṭwīl* too so that the proper shapes of the letters be selected. Consider for example the following:—

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

لَا أَرَى مِنْ عَهْدُتُ فِيهَا فَأُبْكِي الْيَوْمَ دَهْلًا وَمَا يَرُدُّ الْبُكَاءُ

As one can see, *triple hyphens* have been used. In the *ṣadr*, the first hyphen triggers the rules that are related to the definite article and the *ʿalif^u* *ʿl-waṣlⁱ*,²⁵ while the following two select the figure of the letter *lām* connected with a following letter. In the *ağuz*, the last two hyphens select the letter *yā* connected with a preceding letter, while the first one is simply discarded in this mode, but still may appear as it should, if the `trans` mode be selected:—

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

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

²⁵See section 4.2 on page 15

- (b) In some other cases, it may seem difficult, if not fairly impossible, to split a given word into two parts. This happens mostly because of the *šaddah*. Consider for example the following:—

```

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

```

غَيْرَ أَنِّي قَدْ اسْتَعِينُ عَلَى الْهَمِّ إِذَا خَفَّ بِالْتَّوَيِّ النَّجَاءُ
يَرْفُوفٍ كَانَهَا هِقْلَةً أَمْ رِثَالٍ دَوِيَّةً سَقْفَاءُ

In the first line, the word *أَلْهَمَّ* should be split into *أَلْهَمَّ* as the first part of it belongs to the *šadr* and the second to the *‘ağuz*. One solution to avoid splitting this word in such a way is to write inside the *tadwīr* the part of it that belongs to either hemistich, without omitting to add a space after it. In the second line, the word *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.

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

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

If they are little in number, each footnote may be split into pairs of `\footnote mark{}` (please mind the braces) in the argument of the `\bayt` command and `\footnotetext` outside the `\bayt` command.

If the footnotes are abundant in number, it is advised to load the `footnotehyper` package which `arabluatex` will then use to typeset any kind of footnote that is called from the arguments of the `\bayt` command.²⁶

²⁶ The `footnote` package may also be used for the same effect. However, it must be loaded *after* `arabluatex`.

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

5.1 Example

Here follow the first lines of Imru'u 'l-Qaysi's *Mu'allaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the *idgām* that is fit to this declamatory poetry:—²⁷

```

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

```

قَالَ أَمْرُؤُ الْقَيْسِ فِي مَعْلَقَتِهِ:

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

²⁷ Please note that for the time being only the assimilation rules that are laid on item **b** on page 15 are applied. See section 2.2.1 on page 6 for more information. None of the editions of the *Mu'allaqāt* that I know of feature the *idgām* in the Arabic text, although it is often strongly marked in declamation.

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

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

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

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

6 Special applications

Linguistics The same horizontal stroke as the *taṭwīl* (see section 4.7 on page 23) may be encoded $\langle B \rangle$; $\langle BB \rangle$ will receive the *tašdīd*. This is useful to make linguistic annotations and comments on vowels:—

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

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

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

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

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

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

7 Transliteration

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

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

dmg *Deutsche Morgenländische Gesellschaft*, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.²⁸ **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.²⁹

More standards will be included in future releases of arabluatex.

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

```
1 \usepackage{arabluatex}
2 \SetTranslitConvention{loc}
```

\SetTranslitStyle **Style** Any transliterated Arabic text is printed in italics by default. This also can be changed either globally in the preamble or locally at any point of the document by the `\SetTranslitStyle{<style>}` command, where *<style>* may be any font shape selection command, eg. `\upshape`, `\itshape`, `\slshape`, and so forth.

New feature
v1.4

\SetTranslitFont

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

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

\cap **Proper names** Proper names or book titles that must have their first letters up-

²⁸See Brockelmann et al. (1935).

²⁹See <http://www.loc.gov/catdir/cpsd/roman.html> for the source document concerning Arabic language.

percased may be passed as arguments to the `\cap{⟨word⟩}` command. `\cap` is a clever command, for it will give the definite article *al-* in lower case in all positions. Moreover, if the initial letter, apart from the article, cannot be uppercased, viz. ‘ or ‘, the letter next to it will be uppercased:—

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

However, `\cap` must be used cautiously in some very particular cases, for the closing brace of its argument may prevent a rule from being applied. To take an example, as seen above on page 17, the transliteration of مُحَمَّدٌ النَّبِيُّ must be *Muḥammad^{uni} 'n-nabī*, as nouns having the *tanwīn* take a *kasrah* in pronunciation before ‘*alifu* ‘*l-waṣli*. In that case, encoding مُحَمَّدٌ like so: `\cap{mu.hammaduN}` is wrong, because the closing brace would prevent `arabluatex` from detecting the sequence `⟨-uN⟩` immediately followed by `⟨'-l⟩`. Fortunately, this can be circumvented in a straightforward way by inserting only part of the noun in the argument of `\cap` vz. up to the first letter that is to be uppercased, like so: `\cap{m}u.hammaduN`.

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

`\cap{'abU} \cap{bakriN} \cap{mu\-.ham\--madu} bnu \cap{za\--ka`
`\-riy\--yA'a} \cap{'l-rAzI}` أَبُو بَكْرٍ مُحَمَّدُ بْنُ زَكْرِيَّا الرَّازِي *Abū Bakrⁱⁿ Mu-*
`hammadu bnu Zakariyyāa 'r-Rāzī.`

7.1 Additional note on dṃg convention

According to Brockelmann et al. (1935, p. 6), Arabic ‘*rāb* may be rendered into *dṃg* 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, ‘*rāb* is passed as an argument on to a `\arbup` command which is set to `\textsuperscript`.

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

`\SetArbUp` Finally, `\SetArbUp{⟨formatting directives⟩}` may be used to customize the way

ʾirāb is displayed. To take one example, here is how Arabic *ʾirāb* 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 مِثْلَهُ: *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.

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

munṣarif: mu`allim^N مُعَلِّمٌ *mu'allim^{un}*, kA'in^N كَائِنٌ *kā'in^{un}*, kA'inAt^N كَائِنَاتٌ *kā'in^{ātun}*, \cap{`amraNU} عَمَرُوا *Amr^{an}*, fataN_A فَتَى *fataⁿ*, qA.diNI قَاضٍ *qāḍiⁿ*.

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

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

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

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

mana`a 'l-nAs-a kAffaT-aN min mu_hA.tabati-hi 'a.had-uN bi-sayyidi-nA مَنَّ النَّاسُ كُلُّهُ مِنْ مَنْعِ النَّاسِ *mana'a 'n-nās^a kaffat^{an} min muḥāṭabati-hi 'aḥad^{un} bi-sayyidi-nā*.

7.2 Examples

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

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

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

8 Buckwalter input scheme

Even though arabxatex is primarily designed to process the ArabTeX notation, it can also process the Buckwalter input scheme to a large extent.³⁰ 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 arabxatex special features that are presented in section 2.2 on page 5.

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

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

‘base’, ‘xml’ and ‘safe’ schemes arabxatex can use any of the so-called Buckwalter ‘base’, ‘xml’ or ‘safe’ schemes as they are described in Habash (2010, pp. 25–26).³¹ However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode ʾ and ʿ, must be replaced with square brackets viz. [and] respectively.

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

Table 6 gives the Buckwalter equivalents that are currently used by arabxatex. The additional characters that are defined in table 5 on page 24 are also available.

³⁰ See <http://www.qamus.org/transliteration.htm>

³¹ I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in arabxatex.

Letter	Transliteration ³²		Buckwalter notation	
	dmg	loc	base/xml	safe
ا	<i>a</i>	<i>a</i>	A	A
ب	<i>b</i>	<i>b</i>	b	b
ت	<i>t</i>	<i>t</i>	t	t
ث	<i>ṭ</i>	<i>th</i>	v	v
ج	<i>ǧ</i>	<i>j</i>	j	j
ح	<i>ḥ</i>	<i>ḥ</i>	H	H
خ	<i>ḫ</i>	<i>kh</i>	x	x
د	<i>d</i>	<i>d</i>	d	d
ذ	<i>ḍ</i>	<i>dh</i>	*	V
ر	<i>r</i>	<i>r</i>	r	r
ز	<i>z</i>	<i>z</i>	z	z
س	<i>s</i>	<i>s</i>	s	s
ش	<i>š</i>	<i>sh</i>	\$	c
ص	<i>ṣ</i>	<i>ṣ</i>	S	S
ض	<i>ḍ</i>	<i>ḍ</i>	D	D
ط	<i>ṭ</i>	<i>ṭ</i>	T	T
ظ	<i>ẓ</i>	<i>ẓ</i>	Z	Z
ع	<i>ʿ</i>	<i>ʿ</i>	E	E
غ	<i>ǧ</i>	<i>gh</i>	g	g
ف	<i>f</i>	<i>f</i>	f	f
ق	<i>q</i>	<i>q</i>	q	q
ك	<i>k</i>	<i>k</i>	k	k
ل	<i>l</i>	<i>l</i>	l	l
م	<i>m</i>	<i>m</i>	m	m
ن	<i>n</i>	<i>n</i>	n	n
ه	<i>h</i>	<i>h</i>	h	h
و	<i>w</i>	<i>w</i>	w	w
ي	<i>y</i>	<i>y</i>	y	y
ى	<i>ā</i>	<i>á</i>	Y	Y
ة	<i>ah</i>	<i>ah</i>	p	p
ء	◌	◌	◌	C
آ	◌ <i>ā</i>	◌ <i>ā</i>		M
أ	◌	◌	>	O
ؤ	◌	◌	&	W
إ	◌	◌	<	I
ئ	◌	◌]	Q
ـ	—	—	~	~
ـِ	◌	◌	[L
ـَ	<i>a</i>	<i>a</i>	a	a

³²See section 7 on page 30.

Letter	Transliteration		Buckwalter notation	
	dmg	loc	base/xml	safe
ا	<i>u</i>	<i>u</i>	u	u
ـ	<i>i</i>	<i>i</i>	i	i
ان	<i>an</i>	<i>an</i>	F	F
انـ	<i>un</i>	<i>un</i>	N	N
انـ	<i>in</i>	<i>in</i>	K	K
ـ	—	—	o	o
اَ	<i>ā</i>	<i>ā</i>	ˆ	e
ـ (taṭwīl)	—	—	-	-

Table 6: Buckwalter scheme

Transliteration The Buckwalter notation can also be transliterated into any accepted romanization standard of Arabic. See above section 7 on page 30 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‘atⁱ ‘t-ṭibbⁱ*.
wa-Al-l~ehi وَاللَّهِ *wa-‘l-lāhⁱ*, Al-Hamdu li-l~ehi اَلْحَمْدُ لِلَّهِ *al-ḥamd^u*
i.

Similary, it is not advisable to use | and [(‘base’ and ‘xml’ schemes) or M and L (‘safe’ scheme) to encode the ‘*alif^u*’ *‘l-mamdūdātⁱ* and the ‘*alif^u*’ *‘l-waṣlⁱ* for such signs are supposed to be generated by `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-LntiqADi` gives اِلَى الْاِنْتِقَاضِ as expected, but only `<ilY Al-intiqADi` can be transliterated as *‘ilā ‘l-intiqāḍi* with the correct vowel ⟨*i*⟩ in place of the ‘*alif^u*’ *‘l-waṣlⁱ*.

9 Unicode Arabic input

As said above in section 8 on page 33 about the Buckwalter input scheme, even though `arabluatex` is primarily designed to process the `ArabTeX` notation, it also accepts unicode Arabic input. It should be noted that `arabluatex` does in no way interfere with unicode Arabic input: none of the `voc`, `fullvoc`, `novoc` or `trans` options will have any effect on plain unicode Arabic for the time being.

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

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

```

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

```

10 L^AT_EX Commands in Arabic environments

General principle L^AT_EX commands are accepted in Arabic environments. The general principle which applies is that single-argument commands (`\command{<arg>}`) such as `\emph{<text>}`, `\textbf{<text>}` and the like, are assumed to have Arabic text as their arguments:—

`\abjad{45}` kitAbu-hu `\emph{fi 'l-\cap{`AdAt-i}}` العَادَاتِ فِي كِتَابِهِ 45
*kitābu-hu fi 'l-‘Ādāt*³³

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 الْفُرسِ.

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

`\LR{<arg>}` is designed to typeset its argument from left to right. It may be used

³³This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.

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` `\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.

For example, to distinguish words with a different color, one may proceed like so:—

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

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

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

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

زَيْدٌ ^aابْنُ 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.³⁴

³⁴See <http://ctan.org/pkg/koma-script>; read the documentation of KOMA-script for details about the `\deffootnotemark` and `\deffootnote` commands.

`\LRmarginpar` The `\LRmarginpar` command does for marginal notes the same as `\LRfootnote` does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that `\marginpar` also works in Arabic environments, but it acts as any other single-argument command inserted in Arabic environments. The general principle laid on page 36 applies.

`\setRL` `\setLR` `\setLR` and `\setLR` may be used to change the direction of paragraphs, either form left to right or from right to left. As an example, an easy way to typeset a right-to-left sectional title follows:—

```

1 \setRL
2 \section*{\arb{barzawayhi li-buzurjumihra bn-i 'l-buxtikAni}}
3 \setLR
4 \begin{arab}
5 qAla barzawayhi bn-u 'azhar-a, ra's-u 'a.tibbA'-i fAris-a...
6 \end{arab}

```

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

10.1 Environments

New feature
v1.5

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

```

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

```

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

```

1 \setquotestyle{arabic}
2 \begin{arab}[fullvoc]
3   kAna \cap{'abU} \cap{'l-hu_dayli} 'ahd_A 'il_A \cap{muwaysiN}
4   dajAjaTaN. wa-kAnat dajAjatu-hu 'llatI 'ahdA-hA dUna mA kAna
5   yuttaxa_du li-\cap{muwaysiN}. wa-l_akinna-hu bi-karami-hi
6   wa-bi-.husni xuluqi-hi 'a.zhara 'l-ta'ajjuba min simani-hA
7   wa-.tIbi la.hmi-hA. wa-kAna <\cap{'abU} \cap{'l-hu_dayli}>
8   yu`rafu
9   bi-'l-'imsAki 'l-^sadIdi. fa-qAla: \enquote{wa-kayfa ra'ayta yA

```

```

9      \cap{'abA} \cap{'imrAna} tilka 'l-dajAjaTa?} qAla:
10     \enquote{kAnat `ajabaN mina 'l-`ajabi!} fa-yaqUlu:
11     \begin{quoting}[begintext=», endtext=«]
12         wa-tadrI mA jinsu-hA? wa-tadrI mA sinnu-hA? fa-'inna
13         'l-dajAjaTa 'inna-mA ta.tIbu bi-'l-jinsi wa-'l-sinni. wa-tadrI
14         bi-'ayyi `say'iN kunna nusamminu-hA? wa-fI 'ayyi makAniN kunna
15         na`lifu-hA?
16     \end{quoting}
17     fa-lA yazAlu fI h_a_dA wa-'l-'A_haru ya.d.haku .da.hkaN
18     na`rifu-hu
19     na.hnu wa-lA ya`rifu-hu \cap{'abU} \cap{'l-hu_dayli}.
\end{arab}

```

كَانَ أَبُو الْهَدَيْلِ أَهْدَى إِلَى مُوسَى دَجَاجَةً. وَكَانَتْ دَجَاجَتُهُ الَّتِي أَهْدَاهَا دُونَ مَا كَانَ يَتَّخِذُ لِمُوسَى. وَلَكِنَّهُ يَكْرَهُهُ وَيُحْسِنُ خَلْقَهُ أَظْهَرَ التَّعَجُّبِ مِنْ سَمْنِهَا وَطِيبِ لَحْمِهَا. وَكَانَ أَبُو الْهَدَيْلِ يُعْرِفُ بِالْإِمْسَاكِ الشَّدِيدِ. فَقَالَ: "وَكَيْفَ رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ الدَّجَاجَةَ؟" قَالَ: "كَانَتْ عَجَبًا مِنَ الْعَجَبِ!" فَيَقُولُ:

« وَتَدْرِي مَا جَنَسُهَا؟ وَتَدْرِي مَا سِنُّهَا؟ فَإِنَّ الدَّجَاجَةَ إِذَا تَطِيبَ بِالْجَنَسِ وَالسِّنِّ. وَتَدْرِي بِأَيِّ شَيْءٍ كُنَّا نُسَمِّيْهَا؟ وَفِي أَيِّ مَكَانٍ كُنَّا نَعْلِفُهَا؟ »

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

10.1.1 Lists

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

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

```

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

```

رموز الكتاب

ب مخطوط المكتبة الأهلية بباريس ٢٨٦٠ عربيّ.

س مخطوط المكتبة الأهلية بباريس ٢٨٥٩ عربيّ.

م مخطوط مجلس شورى على 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 الطبّ

- الفنّ الأوّل في حدّ الطبّ وموضوعاته من الأمور الطّبيعيّة ويشتمل على ستّة تعاليم

- التّعليم الأوّل [وهو فصلان]

- الفصل الأوّل

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

```
1 \usepackage[french]{babel}
2 \frenchbsetup{StandardLayout=true}
```


This option will prevent babel-french from interfering with the layout of the document. Then the package paralist may be used to make the lists ‘compact’ as babel-french do.

10.2 csquotes

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

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

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

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

فَقَالَ لَهُ جُحَا: “غَرِيبٌ أَمْرُكَ يَا صَدِيقِي أَتَصَدِّقُ 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.

10.3 reledmac

The two-arguments command \edtext{\<lemma>}{\<commands>} is supported inside \begin{arab} ... \end{arab}. As an example, one may get arabuatex and reledmac to work together like so:—

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

10.4 quran

arabluatex is compatible with the quran package so that both can be used in conjunction with one another for typesetting the *Qur'ān*. As quran draws the text of the *Qur'ān* from a unicode encoded database, its commands have to be passed as arguments to the `\txarb` command for short insertions in left-to-right paragraphs, or inserted inside the `txarab` environment for typesetting running paragraphs of *Qur'ānic* text (see above section 9 on page 35 for more details). Please note that arabluatex takes care of formatting the Arabic: therefore, it is recommended to load the quran package with the `nopar` option, after arabluatex itself has been loaded, like so:—

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

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

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

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ ﴿١﴾ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ ﴿٢﴾ الرَّحْمَنِ الرَّحِيمِ ﴿٣﴾ مَالِكِ يَوْمِ الدِّينِ ﴿٤﴾
إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ ﴿٥﴾ اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ ﴿٦﴾ صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ
عَلَيْهِمْ وَلَا الضَّالِّينَ ﴿٧﴾

11 Future work

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

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

12 Implementation

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

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{arabluatex}%
3 [2017/03/15 v1.7.2 An ArabTeX-like interface for LuaLaTeX]
4 \RequirePackage{ifluatex}
```

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

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

Declare the global options, and define them:

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

Packages that are required by arabluatex:

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

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

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

Here begins the real work: load arabluatex.lua:

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

This is needed by the current versions of polyglossia and luabidi. luabidi provides a `\Footnote` command. Use it as well if it is loaded.

```
33 \luadirect{tex.enableprimitives("luatex",tex.extraprimitives("omega"))}
```

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

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

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

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

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

```
39 \AtBeginDocument{\def\setLR{\boolfalse{al@rlmode}\pardir TLT\textdir TLT}}
```

`\LR` This command typesets its argument from left to right. As `\LR` may be already defined, we need to redefine for it to suit our purpose:

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

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

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

`\aemph` Arabic emphasis. Needs to be redefined as well.

```
46 \AtBeginDocument{\ifdef{\aemph}}%
47 {\RenewDocumentCommand{\aemph}{m}{\overline{\text{#1}}}}
48 {\NewDocumentCommand{\aemph}{m}{\overline{\text{#1}}}}
```

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

```
49 \def\al@input@scheme{arabtex}
50 \NewDocumentCommand{\SetInputScheme}{m}{\def\al@input@scheme{#1}}
```

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

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

```
51 \def\al@arb@rules{dflt}
```

```

52 \NewDocumentCommand{\SetArbEasy}{s}{%
53   \IfBooleanTF{#1}
54   {\def\al@arb@rules{easynosukun}}
55   {\def\al@arb@rules{easy}}}
56 \NewDocumentCommand{\SetArbDflt}{s}{%
57   \IfBooleanTF{#1}
58   {\def\al@arb@rules{idgham}}
59   {\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.

```

60 \def\al@trans@font{\rmfamily}%
61 \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:

```

62 \def\al@trans@style{\itshape}%
63 \NewDocumentCommand{\SetTranslitStyle}{m}{\def\al@trans@style{#1}}

```

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

```

64 \def\al@trans@convention{dmg}
65 \NewDocumentCommand{\SetTranslitConvention}{m}{\def\al@trans@convention{#1}}

```

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

```

66 \NewDocumentCommand{\al@arbup@dflt}{m}{\textsuperscript{\thinspace#1}}%
67 \NewDocumentCommand{\al@arbup}{m}{\al@arbup@dflt{#1}}
68 \NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
69 \NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt}
70 \NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\al@arbup}{m}{##1}}
71 \NewDocumentCommand{\SetArbUp}{m}{\RenewDocumentCommand{\al@arbup}{m}{#1}}

```

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

```

72 \DeclareDocumentCommand{\cap}{m}%
73   {\luairect{tex.sprint(cap(\luastringN{#1}))}}

```

\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.

```

74 \DeclareDocumentCommand{\txarb}{+m}{\bgroup\textdir
75   TRT\arabicfont#1\egroup}
76 \DeclareDocumentCommand{\txtrans}{+m}{\bgroup\textdir
77   TLT\al@trans@font#1\egroup}

```

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

```

78 \NewDocumentEnvironment{txarab}{}{
79   \par%
80   \booltrue{al@rlmode}%
81   \pardir TRT\textdir TRT\arabicfont}{\par}

```

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

```

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

```

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

```

107 \NewEnviron{arab}[1][\al@mode]%
108 {\par\edef\@tempa{#1}%
109   \ifx\@tempa\al@mode@voc%
110     \booltrue{al@rlmode}%
111     \bgroup\pardir TRT\textdir TRT\arabicfont%

```

```

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

```

arabverse The arabverse environment may receive different options: mode, width, gutter, metre, utf and delim; all of them are defined here just before the arabverse environment:

```

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

```

Then follows the environment itself:

```

147 \NewDocumentEnvironment{arabverse}{0{}}%
148 {\par\centering\noindent\bgroup\setkeys{al}{verse}[metre]{#1}%
149   \ifx\al@mode\al@mode@trans%
150   \ifal@verse@utf\setRL\else\setLR\fi%
151   \else\setRL\fi}%
152 {\hfill\setkeys{al}{verse}[width,gutter,utf,mode]{#1}\egroup}

```

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

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

```

153 \NewDocumentCommand{\arb@utf}{m}{%
154   \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
155 \def\al@hemistich@delim{*}
156 \NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
157 \def\al@verse@stroke{\leavevmode\xleaders\hbox{\arb{--}}\hfill\kern0pt}
158 \NewDocumentCommand{\bayt}{m o m}{%
159   \ifdefined\savenotes\savenotes\else\fi%
160   \edef\al@tatweel{--}%
161   \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#1}}%
162   \IfNoValueTF{#2}{%
163     \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
164     \else%
165       \hspace{\al@gutter@width}%
166     \fi
167   }{%
168     \edef\@tempa{#2}%
169     \ifx\@tempa\al@tatweel%
170       \ifx\al@mode\al@mode@trans%
171         \hspace{\al@gutter@width}%
172       \else%
173         \makebox[\al@gutter@width][s]{\al@verse@stroke}%
174       \fi%
175     \else%
176       \ifx\al@mode\al@mode@trans%
177         \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#2}}%
178       \else%
179         \makebox[\al@gutter@width][s]{\arb@utf{#2}}%
180       \fi\fi}%
181   \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
182   \ifdefined\spewnotes\spewnotes\else\fi%
183 }
```

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

```

184 \AtBeginDocument{%
185   \ifdefined\abjad%
186     \RenewDocumentCommand{\abjad}{m}%
187     {\luairect{tex.sprint(abjadify(#1))}}%
```



```

188 \else%
189 \NewDocumentCommand{\abjad}{m}%
190 {\luadirect{tex.sprint(abjadify(#1))}}
191 \fi}

\arbnul1 The \arbnul1 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.
192 \NewDocumentCommand{\arbnul1}{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.
193 \NewDocumentCommand{\abraces}{+m}{%
194 \luadirect{tex.sprint(abraces(luastringN{#1}))}}

\LRmarginpar \LRmarginpar is supposed to be inserted in an Arabic environment. It typsets his
argument in a marginal note from left to right.
195 \DeclareDocumentCommand{\LRmarginpar}{m}{\marginpar{\textdir TLT #1}}

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

\RLfootnote while \RLfootnote typesets its argument left-to-right.
196 \DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pardir
197 TLT\LR{\footnote{#1}}\egroup}
198 \DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pardir
199 TRT\LR{\footnote{#1}}\egroup}

\FixArbFtnmk In the preamble, just below \usepackage{arabluatex}, \FixArbFtnmk may be of
some help in case the footnote numbers at the bottom of the page are printed in the
wrong direction. This quick fix uses and loads scrextend if it is not already loaded.
200 \NewDocumentCommand{\FixArbFtnmk}{}{%
201 \@ifpackageloaded{scrextend}%
202 {\AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%
203 {\RequirePackage{scrextend}
204 \AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}}

```

That is it. Say goodbye before leaving.

Patches

```

205 \NeedsTeXFormat{LaTeX2e}
206 \ProvidesPackage{arabluatex-patch}%
207 [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:

```

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

patch begins:
227   \ifbool{al@rlmode}{\advance\@totalleftmargin \rightmargin}%
228   {\advance\@totalleftmargin \leftmargin}

patch ends.
229   \parshape \@ne \@totalleftmargin \linewidth
230   \ignorespaces}
231 \def\@item[#1]{%
232   \if@nparitem
233     \@donoparitem
234   \else
235     \if@inlabel
236       \indent \par
237     \fi
238     \ifhmode
239       \unskip\unskip \par
240     \fi
241     \if@newlist
242       \if@nobreak
243         \@nbitem
244       \else
245         \addpenalty\@beginparpenalty
246         \addvspace\@topsep
247         \addvspace{-\parskip}%
248       \fi
249     \else
250       \addpenalty\@itempenalty
251       \addvspace\itemsep

```

```

252 \fi
253 \global\@inlabeltrue
254 \fi
255 \everypar{%
256 \@minipagefalse
257 \global\@newlistfalse
258 \if@inlabel
259 \global\@inlabelfalse
260 {\setbox\z@\lastbox
261 \ifvoid\z@
262 \kern-\itemindent
263 \fi}%
264 \box\@labels
265 \penalty\z@
266 \fi
267 \if@nobreak
268 \@nobreakfalse
269 \clubpenalty \@M
270 \else
271 \clubpenalty \@clubpenalty
272 \everypar{}}%
273 \fi}%
274 \if@noitemarg
275 \@noitemargfalse
276 \if@nmbrlist
277 \refstepcounter\@listctr
278 \fi
279 \fi

```

patch begins:

```

280 \ifbool{al@rlmode}{\sRLbox\@tempboxa{\makelabel{#1}}}{%
281 \sbox\@tempboxa{\makelabel{#1}}}%
282 \ifbool{al@rlmode}{\global\setbox\@labels\hbox dir TRT}{%
283 {\global\setbox\@labels\hbox}{%

```

patch ends.

```

284 \unhbox\@labels
285 \hskip \itemindent
286 \hskip -\labelwidth
287 \hskip -\labelsep
288 \ifdim \wd\@tempboxa >\labelwidth
289 \box\@tempboxa
290 \else
291 \hbox to\labelwidth {\unhbox\@tempboxa}%
292 \fi
293 \hskip \labelsep}%
294 \ignorespaces}

```

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

```

295 \long\def\sRLbox#1#2{\setbox#1\hbox dir TRT{%
296 \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*. In col- lab. with Ph. S. van Ronkel and Otto Spies. Deutschen Morgenländischen Gesellschaft. Leipzig: Deutschen Morgenländischen Gesellschaft, in Kommis- sion bei F. A. Brockhaus. URL: http://www.naher-osten.uni-muenchen.de/studium_lehre/werkzeugkasten/dmgtransliteration.pdf.
- Habash, Nizar Y. (2010). *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies 10. Toronto: Morgan & Claypool Publishers.
- Hosny, Khaled (2015). *Amiri*. URL: <http://www.amirifont.org/>.
- Lagally, Klaus (2004). *ArabTeX. Typesetting Arabic and Hebrew*. User Manual Version 4.00. Version 4.00. URL: <http://mirrors.ctan.org/language/arabic/arabtex/doc/html/arabtex.htm>.
- Lane, Edward William (1863–1893). *An Arabic-English lexicon*. 8 vols. London – Edinburgh: Williams and Norgate.
- Wright, W. LL.D (1896). *A Grammar of the Arabic Language*. Rev. by W. Robert- son 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	\SetInputScheme: \SetInputScheme may be used to process other input schemes such as ‘Buckwalter’	44
v1.0.1	General: Minor update of the documentation	1		\SetTranslitFont: For selecting a specific font for transliterated texts	45
v1.1	\abjad: New and more flexible \abjad command.	48	v1.4.3	\abraces: New \abraces command which expresses its argument between braces.	49
v1.2	\SetArbEasy: New \SetArbEasy/\SetArbDflt for ‘modern’ or ‘classic’ Arabic styles.	44	v1.4.4	\SetArbEasy*: this starred version discards the <i>sukūn</i> in addition to what is already discarded by \SetArbEasy.	44
v1.3	\arbup: ʾrāb is now written as superscript text in dmg mode by default.	45	v1.5	General: Compatibility with the quran package	42

Environments may be nested inside the <code>arab</code> environment . .	38	<code>\SetArbDflt*</code> : This starred version applies the assimilation rules in addition to what <code>\SetArbDflt</code> already does.	44
<code>txarab</code> : New <code>txarab</code> environment for typesetting running paragraphs in unicode Arabic .	46	<code>\SetHemistichDelim</code> : New <code>\SetHemistichDelim</code> command for changing the default delimiter between hemistichs .	48
v1.6 <code>arabverse</code> : New environment <code>arabverse</code> for typesetting Arabic poetry	47	v1.7 <code>\arbnull</code> : New <code>\arbnull</code> command for putting back on any contextual analysis rule broken by other commands.	48
<code>\bayt</code> : New macro <code>\bayt</code> for typesetting each verse inside the <code>arabverse</code> environment	47		

Index

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

Symbols		
<code>\@M</code>	269	<code>\@tempboxa</code> 280, 281, 288, 289, 291
<code>\@beginparpenalty</code> .	245	<code>\@toodeep</code> 210
<code>\@clubpenalty</code>	271	<code>\@topsep</code> 246
<code>\@donoparitem</code>	233	<code>\@totalleftmargin</code> 227, 228, 229
<code>\@ifpackageloaded</code> .	201	<code>\@trivlist</code> 222
<code>\@inlabelfalse</code>	259	
<code>\@inlabeltrue</code>	253	
<code>\@item</code>	231	<code>\@</code> 35
<code>\@itemlabel</code>	218	
<code>\@itempenalty</code>	250	
<code>\@labels</code>		A
264, 282, 283, 284		<code>\abjad</code> 23, <u>184</u>
<code>\@listctr</code>	277	<code>\abraces</code> 29, <u>193</u>
<code>\@listdepth</code> 209, 212, 217		<code>\addpenalty</code> . . 245, <u>250</u>
<code>\@minipagefalse</code> . . .	256	<code>\addvspace</code> 246, 247, 251
<code>\@mklab</code>	219	<code>\adjustbox</code> 161, 177, 181
<code>\@nbitem</code>	243	<code>\advance</code> 212, 225, 226, 227, 228
<code>\@ne</code>	212, 229	<code>\aemph</code> 25, <u>46</u>
<code>\@newlistfalse</code>	257	<code>\al@arb@rules</code> . . . 51, 54, 55, 58, 59,
<code>\@nmbrlistfalse</code> . . .	220	87, 92, 97, 103, 113, 119, 125, 131
<code>\@nbreakfalse</code>	268	<code>\al@arbut</code>
<code>\@noitemargfalse</code> . .	275	. 67, 68, 69, 70, 71
<code>\@tempa</code>	83, 84, 89, 94, 99, 108, 109, 115, 121, 127, 168, 169	<code>\al@arbut@dfmt</code> 66, 67, 69
		<code>\al@bayt@width</code> 134, 136, 137, 138, 161, 181
		<code>\al@gutter@width</code> 135, 137, 139, 163, 165, 171, 173, 177, 179
		<code>\al@hemistich@delim</code> 155, 156, 163
		<code>\al@input@scheme</code> 49, 50, 87, 92, 97, 104, 113, 119, 125, 132
		<code>\al@mode</code> 12, 13, 14, 15, 82, 107, 144, 149, 170, 176
		<code>\al@mode@fullvoc</code> 19, 89, 115
		<code>\al@mode@novoc</code> 20, 94, 121
		<code>\al@mode@trans</code> 21, 99, 127, 149, 170, 176
		<code>\al@mode@voc</code> 18, 84, 109
		<code>\al@tatweel</code> . . 160, 169
		<code>\al@trans@convention</code> . . 64, 65, 102, 130

<code>\al@trans@font</code>	<code>\centering</code> 148	I
. 60, 61, 77	<code>\clubpenalty</code> . . 269, 271	<code>\if@inlabel</code> . . 235, 258
<code>\al@trans@style</code>	<code>\color@endgroup</code> . . 296	<code>\if@newlist</code> 241
. . 62, 63, 100, 128	<code>\color@setgroup</code> . . 296	<code>\if@nmbrlist</code> 276
<code>\al@verse@stroke</code>	<code>csquotes</code> (package) . . . 41	<code>\if@nobreak</code> . . 242, 267
. 157, 173	D	<code>\if@noitemarg</code> 274
<i>Amiri</i> 5	<code>\DeclareOption</code>	<code>\if@noparitem</code> 232
<i>amiri</i> (package) 5 12, 13, 14, 15	<code>\ifal@verse@delim</code> . 163
<i>arab</i> (environment) 7, 107	<code>\deffootnote</code> . . 202, 204	<code>\ifal@verse@utf</code> 150, 154
<i>ArabTeX</i> 3, 19	<code>\define@boolkey</code> 141, 142	<code>\ifbool</code> . . 227, 280, 282
<i>Arabic-English Lexicon</i> ,	<code>\define@choicekey</code> . 143	<code>\IfBooleanTF</code> 53, 57
<i>An</i> 18	<code>\define@key</code> 138, 139, 140	<code>\ifdim</code> 288
<code>\arabicfont</code>	<code>\delim</code> (option) 25	<code>\ifhmode</code> 238
. . . 34, 35, 37,	E	<code>\ifluatex</code> 5
75, 81, 85, 90,	environments:	<code>\IfNoValueTF</code> 162
95, 111, 117, 123	<i>arab</i> 7, 107	<code>\ifnum</code> 209
<i>arabluatex</i> (package) 2,	<i>arabverse</i> . . 25, 134	<code>\ifvoid</code> 261
4–8, 10–12,	<i>txarab</i> 36, 78	<code>\ignorespaces</code> . 230, 294
14–19, 24,	<code>\everypar</code> 255, 272	<code>\indent</code> 236
25, 27, 29–33,	<code>\ExecuteOptions</code> 16	<i>Introduction to Arabic</i>
35–37, 40–44	F	<i>Natural Lan-</i>
<i>arabtex</i> (package)	<code>\FixArbFtnmk</code> . . . 37, 200	<i>guage Process-</i>
. . . 3, 4, 15, 19, 42	fontspec (package)	<i>ing</i> 33
<i>arabulatex</i> (package) . . 16 5, 30, 45	<code>\itemindent</code> 216, 262, 285
<i>arabverse</i> (environ-	footnote (package) 27	<code>\itemsep</code> 251
ment) . . . 25, 134	<code>\footnote</code> 197, 199	<code>\itshape</code> 62
<i>arabxetex</i> (package) . 4, 15	footnotehyper (package) 27	K
<code>\arb</code> 7, 82, 140, 154, 157	fullvoc (option) 5, 8	<code>\kern</code> 157, 262
<code>\arb@utf</code> 153,	G	KOMA-script (package) 37
161, 177, 179, 181	<code>\global</code> . . . 212, 253,	L
<code>\arbnul</code> 22, 192	257, 259, 282, 283	<code>\labelsep</code> 287, 293
<code>\arbup</code> 31, 66	<i>Grammar of the Arabic</i>	<code>\labelwidth</code> 286, 288, 291
<code>\ArbUpDflt</code> 31, 66	<i>Language, A</i>	Lagally, Klaus 3, 19
B 5–7, 9,	Lane, Edward William . 18
<i>babel</i> (package) 40	12–15, 17, 23, 24	<code>\lastbox</code> 260
<i>babel-french</i> (package)	gutter (option) 25	<code>\leavevmode</code> 157
. 40, 41	H	<code>\leftmargin</code> . . 226, 228
<code>\baselineskip</code> 145	Habash, Nizar Y. 33	lineno (package) 28
<code>\bayt</code> 26, 153	<code>\hbox</code> 157,	<code>\linewidth</code> 225, 226, 229
<i>bidi</i> (package) 51	282, 283, 291, 295	<code>\list</code> 208
<code>\BODY</code> 112, 118, 124, 129	<code>\Height</code> . . 161, 177, 181	<code>\listparindent</code> 215, 224
<code>\boolfalse</code> 39	<code>\hfill</code> 152, 157	<code>\long</code> 295
<code>\booltrue</code> 38,	Hosny, Khaled 5	<code>\LR</code> 36, 40, 145,
80, 110, 116, 122	<code>\hskip</code> 285, 286, 287, 293	197, 199, 202, 204
<code>\box</code> 264, 289	<code>\hspace</code> 165, 171	<code>\LRfootnote</code> . . . 37, 196
Brockelmann, Carl . 30, 31		<code>\LRmarginpar</code> . . . 38, 195
C		<i>luabidi</i> (package) 4, 43, 44
<code>\cap</code> 30, 72		

<code>\luadirect</code> 32, 33, 73, 86, 91, 96, 101, 112, 118, 124, 129, 187, 190, 194	<code>\paddir</code> 38, 39, 81, 111, 117, 123, 128, 196, 198	<code>\SetTranslitConvention</code> 30, 64
<code>luaotfload (package)</code> 30	<code>\parindent</code> 224	<code>\SetTranslitFont</code> 30, 60
<code>\luastringN</code> 73, 86, 91, 96, 101, 194	<code>\parsep</code> 223	<code>\SetTranslitStyle</code> 30, 62
<code>\luastringO</code> 87, 92, 97, 102, 103, 104, 112, 113, 118, 119, 124, 125, 129, 130, 131, 132	<code>\parshape</code> 229	<code>\spewnotes</code> 182
	<code>\parskip</code> 223, 247	<code>\sRLbox</code> 280, 295
	<code>\penalty</code> 265	
	<code>polyglossia (package)</code> 4, 43, 48	T
	<code>\presetkeys</code> 145	<code>\text</code> 47, 48
	<code>\ProcessOptions</code> 17	<code>\textdir</code> 38, 39, 41, 42, 44, 45, 74, 76, 81, 85, 90, 95, 100, 111, 117, 123, 128, 195
	<code>\ProvidesPackage</code> 2, 206	<code>\textsuperscript</code> 66
		<code>\textwidth</code> 136
M	Q	<code>\the</code> 217
<code>\makebox</code> 163, 173, 179	<code>quoting (package)</code> 38	<code>\thefootnotemark</code> 202, 204
<code>\makelabel</code> 219, 280, 281	<code>quran (package)</code> 42	<code>\thinspace</code> 66
<code>\marginpar</code> 195		<code>trans (option)</code> 5, 8
<code>metre (option)</code> 25	R	<i>Transliteration der arabischen Schrift, Die</i> 30, 31
<code>mode (option)</code> 25	<code>\refstepcounter</code> 277	<code>txarab (environment)</code> 42
	<code>reledmac (package)</code> 41	<code>txarab (environment)</code> 36, 78
	<code>\RequirePackage</code> 4, 22, 23, 24, 25, 26, 27, 28, 29, 30, 203	<code>\txarb</code> 36, 74, 154
N	<code>\rightmargin</code> 214, 225, 227	<code>\txtrans</code> 74
<code>\newbool</code> 31	<code>RL</code> 37, 43	
<code>\NewDocumentEnvironment</code> 78, 147	<code>RLfootnote</code> 37, 196	U
<code>\NewEnviron</code> 107	<code>\rmfamily</code> 41, 42, 44, 45, 60	<code>\unhbox</code> 284, 291
<code>\newfontfamily</code> 37	<code>\romannumeral</code> 217	<code>\unskip</code> 239
<code>\newlength</code> 134, 135		<code>utf (option)</code> 25
<code>\NoArbUp</code> 31, 66	S	V
<code>\noindent</code> 148	<code>\savenotes</code> 159	<code>voc (option)</code> 5, 8
<code>novoc (option)</code> 5, 8	<code>\sbox</code> 281	<code>\vskip</code> 145
O	<code>scrextend (package)</code> 37, 49	
<code>options:</code>	<code>\SetArbDflt</code> 6, 51	W
<code>delim</code> 25	<code>\SetArbDflt*</code> 6, 51	<code>\wd</code> 288
<code>fullvoc</code> 5, 8	<code>\SetArbEasy</code> 6, 51	<code>width (option)</code> 25
<code>gutter</code> 25	<code>\SetArbEasy*</code> 6, 51	<code>Wright, W. LL.D</code> 5–7, 9, 12–15, 17, 23, 24
<code>metre</code> 25	<code>\SetArbUp</code> 31, 66	
<code>mode</code> 25	<code>\setbox</code> 260, 282, 283, 295	X
<code>novoc</code> 5, 8	<code>\SetHemistichDelim</code> 25, 153, 156	<code>\xleaders</code> 157
<code>trans</code> 5, 8	<code>\SetInputScheme</code> 33, 49	
<code>utf</code> 25	<code>\setkeys</code> 148, 152	Z
<code>voc</code> 5, 8	<code>\setlength</code> 136, 137, 138, 139	<code>\z@</code> 214, 215, 216, 260, 261, 265
<code>width</code> 25	<code>\setLR</code> 38, 39, 150	
<code>\overline</code> 47, 48	<code>\setRL</code> 38, 38, 150, 151	
P		
<code>\par</code> 79, 81, 108, 133, 148, 236, 239		
<code>paralist (package)</code> 39, 41		