

The arabluatex package

v1.5 – 2016/11/14

Robert Alessi
alessi@robertalessi.net

Contents

License and disclaimer	2	4.7.2 The <i>abjad</i>	22
1 Introduction	3	4.8 Additional characters	22
1.1 arabluatex is for Lua [®] TeX	4	4.9 Arabic emphasis	23
2 The basics of arabluatex	4	5 Special applications	23
2.1 Activating arabluatex	4	6 Transliteration	24
2.2 Options	5	6.1 Additional note on <code>dmg</code> convention	25
2.2.1 Classic contrasted with modern typesetting of Arabic	5	6.2 Examples	27
2.3 Typing Arabic	6	7 Buckwalter input scheme	27
2.3.1 Local options	8	8 Unicode Arabic input	30
3 Standard Arab[®]TeX input	8	9 L[®]TeX Commands in Arabic environments	30
3.1 Consonants	8	9.1 Environments	32
3.2 Vowels	9	9.1.1 Lists	34
3.2.1 Long vowels	9	9.2 <code>csquotes</code>	35
3.2.2 Short vowels	10	9.3 <code>reledmac</code>	36
4 arabluatex in action	11	9.4 <code>quran</code>	36
4.1 The vowels and diphthongs	11	10 Future work	37
4.2 Other orthographic signs	13	11 Implementation	37
4.3 Special orthographies	17	References	44
4.4 Quoting	19	Change History	45
4.4.1 Quoting the <i>hamzah</i>	20	Index	45
4.5 The “pipe” character (<code> </code>)	21		
4.6 Stretching characters: the <i>taṭwīl</i>	21		
4.7 Digits	22		
4.7.1 Numerical figures	22		

List of Tables

1	ArabTeX consonants	8	4	“Quoted” <i>hamzah</i>	20
2	ArabTeX long vowels	9	5	Additional Arabic codings	22
3	ArabTeX short vowels	10	6	Buckwalter scheme	28

Abstract

This package provides for Lua^LTeX 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

Copyright © 2016 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: <http://issues.robertalessi.net>

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`

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 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.

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>.

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 \LaTeX

It goes without saying that `arabluatex` requires Lua \LaTeX . \TeX and \LaTeX have `arabtex`, and X \LaTeX 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 Arab \TeX input on the fly into Unicode Arabic script, whereas the latter passes Arab \TeX input on to a set of Lua functions. At first, \LaTeX commands are taken care of in different ways: some, as `\emph`, `\textbf` and the like are expected to have Arabic text as arguments, while others, as `\LR`, for “left-to-right text”, are not. Then, once what is Arabic is carefully separated from what is not, it is processed by other Lua functions which rely on different sets of correspondence tables to do the actual conversion in accordance with one’s wishes. Finally, Lua returns to \TeX the converted strings—which may in turn contain some other Arab \TeX 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 \LaTeX ; 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.

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.

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

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 *tašdid* below the consonant, like so: *ٲ*. That is correct, as at least in the oldest manuscripts *ٲ* may stand for *ٲ* as well as *ٲ*. 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 *ʿalif*” (*ʿalifu ʿl-waṣli*), but `voc` mode does not show the *waṣlah* above the *ʿalif*; instead, the accompanying vowel may be expressed at the beginning of a sentence (◌ِ◌◌).

`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 ArabTeX input into one of the accepted standards. At present, two standards are supported (see below section 6 on page 24 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`.

2.2.1 Classic contrasted with modern typesetting of Arabic

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

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

⁵Hosny (2015).

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

New feature
v1.4.4

\SetArbEasy
\SetArbEasy*
\SetArbDflt

samA'uN سَمَاءُ ^{un} samā', jA'a جَاءَ ^a jā'a, yatasA'alUna يَتَسَاءَلُونَ ⁷ yatasā'alūna⁷
(see on page 14 for further details).

- The euphonic *tašdīd* is generated (see on page 15).
- Assimilation rules laid on item b on page 15 are applied.
- In *fullvoc* mode, the *sukūn* is expressed.

Such refinements may be discarded by the command `\SetArbEasy`, 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 command `\SetArbDflt`. Examples follow:—

(a) `\SetArbDflt`:

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

(b) `\SetArbEasy`:

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

(c) `\SetArbEasy*`:

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

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

2.3 Typing Arabic

`\arb` Once `arabluatex` is loaded, a command `\arb{⟨Arabic text⟩}` 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).

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

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، حُرُوفُ الْحِجَاءِ) 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 command `\indent` 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 جَحَا يَطْلُبُ مِنْهُ حِمَارَهُ لِيَرْكَبَهُ فِي سَفَرَةٍ قَصِيرَةٍ فَقَالَ لَهُ: "سَوْفَ أُعِيدُهُ إِلَيْكَ فِي الْمَسَاءِ
وَأَدْفَعُ لَكَ أُجْرَةً." فَقَالَ جَحَا: "أَنَا آسَفٌ جِدًّا أَنِّي لَا أَسْتَطِيعُ أَنْ أُحَقِّقَ لَكَ رَغْبَتَكَ فَالْحِمَارُ لَيْسَ هُنَا
الْيَوْمَ." وَقَبِلَ أَنْ يُتِمَّ جَحَا كَلَامَهُ بِدَأِّ الْحِمَارِ يَنْهَقُ فِي إِصْطَبَلِهِ. فَقَالَ لَهُ صَدِيقُهُ: "إِنِّي أَسْمَعُ حِمَارَكَ يَا جَحَا
يَنْهَقُ." فَقَالَ لَهُ جَحَا: "غَرِيبُ أَمْرِكَ يَا صَدِيقِي أَتَصَدِّقُ الْحِمَارَ وَتَكْذِبُنِي؟"

2.3.1 Local options

As seen above in section 2.2 on page 5, arabluatex 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:—

```

voc      - \arb[voc]{\langle Arabic text \rangle};
fullvoc  - \arb[fullvoc]{\langle Arabic text \rangle};
novoc    - \arb[novoc]{\langle Arabic text \rangle};
trans    - \arb[trans]{\langle Arabic text \rangle}.

```

The same optional arguments may be passed to the environment `arab`: one may have `\begin{arab}[\langle mode \rangle] ... \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 ⁸		ArabTeX notation
	dmg	loc	
ا ⁹	<i>a</i>	<i>a</i>	a
ب	<i>b</i>	<i>b</i>	b
ت	<i>t</i>	<i>t</i>	t

⁸See below section 6 on page 24.

⁹For ‘*alif*’ as a consonant, see Wright (1896, i. 16 D). The *hamzah* itself is encoded <’>. See below section 4.2 on page 13.

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

Table 1: Standard ArabTeX (consonants)

3.2 Vowels

3.2.1 Long vowels

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

Letter	Transliteration ¹⁰		ArabTeX notation
	dmg	loc	
ا	<i>ā</i>	<i>ā</i>	<i>A</i>
و	<i>ū</i>	<i>ū</i>	<i>U</i>
ي	<i>ī</i>	<i>ī</i>	<i>I</i>

¹⁰See below section 6 on page 24.

Letter	Transliteration		ArabTeX notation
	dmg	loc	
ى ¹¹	\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 ArabTeX (long vowels)

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

REM. *b*. Defective writings, such as ا, *al-'alif^u 'l-maḥḍūfat^u*, or defective writings of \bar{u} and \bar{i} are encoded _a_u and _i respectively, e.g. دَالِكُ *_d_alika*, اَلْمَلِكُ *al-mal_a'ika* T-u 'l-ra.hm_an-u الْمَلِكُ الرَّحْمَنُ *.hu_dayfaT-u bn-u 'l-yamAn_i* اَلْيَمَانُ حُدَيْفَةُ for *Hudayfat^u bn^u 'l-Yamānī*, etc.

3.2.2 Short vowels

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

Letter	Transliteration ¹²		ArabTeX 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
أ	<i>an</i>	<i>an</i>	aN
و	<i>un</i>	<i>un</i>	uN
ي	<i>in</i>	<i>in</i>	iN

Table 3: Standard ArabTeX (short vowels)

Whether Arabic texts be vocalized or not is essentially a matter of personal choice. So one may use `voc` mode and decide not to write vowels except at some particular places for disambiguation purposes, or use `novoc` mode, not write vowels—as `novoc` normally does not show them—except, again, where disambiguation is needed.¹³

¹¹ = *al-'alif^u 'l-maḥḍūrat^u*.

¹² See below section 6 on page 24.

¹³ See below section 4.4 on page 19.

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

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

- at the commencement of a word, to indicate that a connective *ʿalif* is needed, with the exception of the article (see below section 4.4 on page 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 *ḥalaqa*, ^samsuN شَمْسٌ *šams^{un}*, karImuN كَرِيمٌ *Karīm^{un}*.
 bi-hi بِهِ *bi-hi*, 'aqi.tuN أَقِطْ *ʿaḩiṭ^{un}*.
 la-hu لَهُ *la-hu*, .hujjaTuN حُجَّةٌ *ḩuġġat^{un}*.

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

qAla قَالٌ *qāla*, bI`a بَيْعٌ *bīʿa*, .tUruN طُورٌ *ṭūr^{un}*, .tInuN طِينٌ *ṭīn^{un}*,
 murU'aTuN مُرُوءَةٌ *murūʿat^{un}*.

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

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

ʿalif otiosum Said *ʿalif^u ʿl-wiqāyatⁱ*, “the guarding *ʿalif*”, after و at the end of a word, both when preceded by *ḍammah* 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*.

‘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 *lākin*, 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 نوة 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 عَمْرٍو بْنُ خَالِدٍ *Muḥammad^u bn^u Amrⁱ bnⁱ Ḥālidⁱⁿ*.
 And so also: al-rib_aUA الرِّبَا *ar-ribā*, ribaNU رِبَا *rib^{an}*.

tanwīn The marks of doubled short vowels, َ, ِ, ُ, are written ⟨uN, aN, iN⟩ respectively. arablutex deals with special cases, such as ُ taking an ِ after all consonants except ة, and *tanwīn* preceding ى as in هَدَى, 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 *maqhaⁿ*, fataNY فَتَى *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 أَوْلَاكَ *ūlā'ika*.

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

Middle hamzah: xA.ti'-Ina خَاطِنٌ *ḫāṭi^{ina}*, ru'UsuN رُؤُوسٌ *ru'ūs^{un}*, xa.tI'aTuN خَاطِنَةٌ *ḫaṭi'at^{un}*, su'ila سُئِلَ *su'ila*, 'as'ilaTuN أَسْئِلَةُ *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'axxara تَأَخَّرَ *ta'ahḫara*, ji'tu-ka جِئْتُكَ *ji'tu-ka*, qA'iluN قَائِلٌ *qā'il^{un}*.

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

Final hamzah: xa.ta'uN خَطَأٌ *ḫaṭa^{un}*, xa.ta'aN خَطَأً *ḫaṭa^{an}*, xa.ta'iN خَطَاً *ḫaṭaⁱⁿ*, 'aqra'u أَقْرَأُ *aqra'u*, taqra'Ina تَقْرَأِينَ *taqra'ina*, taqra'Una تَقْرَأُونَ *taqra'ūna*, yaqra'na يَقْرَأْنَ *yaqra'na*, yaxba'Ani يَخْبَأَنَّ *yahba'āni*,

xaba'A خَبَا *ḥaba'ā*, xubi'a خُبِيْ *ḥubī'a*, xubi'UA خُبُوا *ḥubī'ū*, jA'a جَاءَ *ǧā'a*, ridA'uN رَدَاءُ *ridā'un*, ridA'aN رَدَاءُ *ridā'an*, jI'a جِيءَ *ǧī'a*, radI'iN رَدِيءٌ *radī'in*, sU'uN سُوءٌ *sū'un*, .daw'uN ضَوْءٌ *daw'un*, qay'iN قِيءٌ *qay'in*, ^sifA'I شِفَائِي *šifā'ī*.

^say'uN شَيْءٌ *šay'un*, ^say'aN شَيْئًا *šay'an*, ^say'iN شَيْءٍ *šay'in*, al-^say'-u الشَّيْءُ *aš-šay'u*, 'a^syA'-u أَشْيَاءُ *ašyā'u*, 'a^syA'-a أَشْيَاءُ *ašyā'a*, .zim'aN ظِمًّا *ẓim'an*.

taḥfif **'l-hamzat**: if the *hamzah* has *ǧazmah* and is preceded by *'alif hamzat*, it must be changed into the letter of prolongation that is homogeneous with the preceding vowel; hence: 'a'mana آمَنَ *āmana*, 'u'minu أُمِنُوا *ūminu*, 'i'manuN إِيمَانٌ *īmān'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 *ǧazmah* (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 أَكِلٌ *ākil'un* respectively.

arabluatex otherwise determines *al-'alif* *'l-mamdūdat* by context analysis.

'is'AduN إِسَادٌ *is'ād'un*, 'AkilUna أَكِلُونُ *ākilūna*, 'a'manna آمَنَّا *āmannā*, al-qur'An-u الْقُرْآنُ *al-qur'an'u*.

jA'a جَاءَ *ǧā'a*, yatasA'alUna يَتَسَاءَلُونَ *yatasā'alūna*, ridA'uN رَدَاءُ *ridā'un*, xaba'A خَبَا *ḥaba'ā*, yaxba'Ani يَخْبَانُ *yahba'ani*.

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

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

^allaqa عَاقَ *allaqa*, mAdduN مَادٌ *mādd'un*, 'ammara أَمَّرَ *ammara*, murruN مَرَّرَ *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-^ˀsams-u اَلشَّمْسُ *aš-šams^u*, or a^ˀs-^ˀsams-u اَلشَّمْسُ *aš-šams^u*.

al-tamr-u اَلتَّمْرُ *at-tamr^u*, al-ra.hm_an-u اَلرَّحْمَنُ *ar-raḥmān^u*, al-.zulm-u اَلظُّلْمُ *aẓ-ẓulm^u*, al-lu.gaT-u اَللُّغَةُ *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 رَبِّي *mir rabbi-hi*, min layliN مِنْ لَيْلٍ *mil laylⁱⁿ*, 'an yaqtula أَنْ يَقْتُلَ *ay yaqtula*.

With *tanwīn*: kitAbuN مِيبْنُ كِتَابٌ *kitāb^{um} mubīn^{um}*.

- (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, اَ is never written, as اَلْحَمْدُ لِلَّهِ; 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 لِلَّهِ *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 is properly encoded, like so:—

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

al- , or a<solar letter>-

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

- (b) Definite article inside sentences is encoded

'l- or '<solar letter>-

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

Article: bAb-u 'l-madrasaT-i الْمَدْرَسَةُ *bāb^u 'l-madrasatⁱ*, al-maqAlaT-u 'l-'_U1_A الْمَقَالَةُ *al-maqālat^u 'l-'ūlā*, al-lu.gaT-u 'l-'arabiyyaT-u فِي صِنَاعَةِ اللُّغَةِ الْعَرَبِيَّةِ *al-luġat^u 'l-'arabiyyat^u*, fI .sinA`aT-i 'l-.tibb-i فِي صِنَاعَةِ الطِّبِّ *fī sinā`atⁱ 'l-tibbⁱ*, 'il_A 'l-intiqA.d-i إِلَى الْإِنْتِقَاضِ *ila 'l-intiqādⁱ*, fI 'l-ibtidA'-i فِي الْإِبْتِدَاءِ *fī 'l-ibtidā'ⁱ*, 'abU 'l-wazIr-i أَبُو الْوَزِيرِ *abu 'l-wazīrⁱ*, fa-lammA ra'aW 'l-najm-a رَأَوْا النَّجْمَ *fa-lammā ra'awu 'n-naġm^a*.

Particles:—

- (a) *li-*: *'alif^u 'l-waṣlⁱ* is omitted in the article *أل* 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-madInaT-i فِي الْمَدِينَةِ *fī 'l-madīnatⁱ*, wa-'l-rajul-u وَالرَّجُلُ *wa-'r-raġul^u*, bi-'l-qalam-i بِالْقَلَمِ *bi-'l-qalamⁱ*, bi-'l-ru`b-i بِالرُّعْبِ *bi-'r-ru`bⁱ*.

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

Other cases: 'awi ismu-hu أَوْ اسْمُهُ *'awi 'smu-hu*, zayduN ibn-u `amriNU زَيْدُ بْنُ عَمْرٍو *Zayd^{uni} 'bn^u 'Amrⁱⁿ*,¹⁴ `umar-u ibn-u 'l-_ha.t.tAb-i عُمَرُ بْنُ الْحَطَّابِ *Umar^u 'bn^u 'l-Haṭṭābⁱ*,¹⁵ imru'-u 'l-qays-i إِمْرُؤُ الْقَيْسِ *Imru^u 'l-Qaysⁱ*, la-aymun-u 'l-l_ah-i لَا يَمُنُّ اللَّهُ *la-'ymun^u 'l-lāhⁱ*.

¹⁴ “Zayd is the son of ‘Amr”: the second noun is not in apposition to the first, but forms part of the predicate. Hence زَيْدُ بْنُ عَمْرٍو and not زَيْدُ بْنُ عَمْرٍو, “Zayd, son of ‘Amr”.

¹⁵ “Umar is the son of al-Haṭṭāb” (see footnote 14).

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

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

fI `aynay 'l-malik-i فِي عَيْنِي الْمَلِكِ fī 'aynayi 'l-malikⁱ, ix^say 'l-qawm-a
إِخْشَى الْقَوْمِ ihšayi 'l-qawm^a, mu.s.tafaw 'l-l_ah-i مُصْطَفَوُ اللَّهِ muṣṭafawu
'l-lāhⁱ.
ramaW 'l-.hijAraT-a رَمَوْا الْحِجَارَةَ ramawu 'l-ḥiğārat^a, fa-lammA ra'aW 'l-najm-a
فَلَمَّا رَأَوْا النَّجْمَ fa-lammā ra'awu 'n-nağm^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 أَنْتُمْ الْكَادِبُونَ 'antumU 'l-kādib^{ūna}, ra'aytumU
'l-rajul-a رَأَيْتُمُ الرَّجُلَ ra'aytumU 'r-rağul^a, mani 'l-ka_d_dAb-u مِنْ
الْكَذَّابِ mani 'l-kaddāb^u, qatalati 'l-rUm-u قَتَلَتِ الرُّومُ qatalati 'r-Rūm^u.

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

mu.hammaduN 'l-nabI مُحَمَّدٌ النَّبِيُّ Muḥammad^{uni} 'n-nabī.

4.3 Special orthographies

The name of God The name of God, اللَّهُ, is compounded of the article اَلْ, and إِلَٰه (noted إِلَهِ with the defective *'alif*) so that it becomes إِلَٰلَهِ; then the *hamzah* is suppressed, its vowel being transferred to the ل before it, so that there remains إِلَهِ (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 اللهُ *al-lāh^u*, yA¹⁶ *al-l_ah-u* يَا اللهُ *yā al-lāh^u*, 'a-fa¹⁷-*al-l_ah-i*
la-ta.g`alanna أَفَاللهُ لَتَعْلَنَّ *'a-fa-al-lāhⁱ la-taḡ'alanna*, bi-'*l-l_ah-i*
bi-l-lāhⁱ, wa-'*l-l_ah-i* وَاللهِ *wa-l-lāhⁱ*, bi-sm-i '*l-l_ah-i* بِسْمِ
اللهِ *bi-smⁱ 'l-lāhⁱ*, al-.hamd-u li-l-l_ah-i اللهُ اَلْحَمْدُ *al-ḥamd^u li-llāhⁱ*,
li-l-l_ah-i '*l-qA'il-u* الْقَائِلُ اللهُ *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 *'alif* are encoded *'lla_dI* and *'llatI*:—

'a_hAfu mina '*l-malik-i* 'lla_dI ya.zlimu '*l-nās-a* أَخَافُ مِنَ الْمَلِكِ
'ahāfu mina 'l-malikⁱ 'lladī yaẓlimu 'n-nās^a, `udtu
'*l-~say_h-a* 'lla_dI huwa marI.duN عُدْتُ الشَّيْخَ الَّذِي هُوَ مَرِيضٌ
'*š-sayh^a 'lladī huwa marīd^{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^{ul} la-ka šay^{an}*.
'*ari-nA* 'lla_dayni '*a.dallA-nA* mina '*l-jinn-i* wa-'*l-'ins-i*
'*ari-nā 'lladayni 'adallā-nā mina 'l-ḡinnⁱ*
wa-'l-'insⁱ.

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

fa-'*innA* na_dkuru '*l-.sawt-ayni* '*l-la_dayni* rawaynA-humA `an
ja.h.zaT-a حَظَّةً عَنْ رَوَيْتَاهُمَا نَذَرْنَا نَذْرُ الصَّوْتَيْنِ الَّذِينَ رَوَيْتَاهُمَا عَنْ حَظَّةٍ *fa-'innā nadkuru 'š-sawt^{ayni}*
'*l-ladayni* rawaynā-humā `an Ḡaḥẓat^a.

And also: *al-la_dAni* الَّذِينَ *al-ladāni*, *al-la_dayni* الَّذِينَ *al-ladayni*,
al-latAni اللَّاتَيْنِ *al-latayni*, *al-latayni* اللَّاتَيْنِ *al-latayni*, *al-lAtI* اللَّاتِي *al-lātī*,
*al-lA'Ati*¹⁸ اللَّاءَاتِ *al-lā'āti*, *al-lA'I* اللَّائِي *al-lā'ī*, and so forth.

¹⁶Note the “pipe” character ‘|’ here after *yA* and below after *fa* before footnote mark 17: 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. *'ala 't-ṭariqⁱ*. See section 4.5 on page 21 on the “pipe” and section 6 on page 24 on *dmg* mode.

¹⁷See footnote 16.

¹⁸Note 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*: ٓ. The “pipe” character prevents this rule from being applied. See section 4.5 on page 21.

4.4 Quoting

It is here referred to “quoting” after the package `arabtex`.¹⁹ The “quoting” mechanism of `arabluatex` is designed to be very similar in effect to the one of `arabtex`.

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

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

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

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

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

``"ullima `ilm-a 'l-hay'aT-i علم علم الـهـيـة` `'llima 'ilma 'l-hay'ati,`
`ya.gz"UA يغزوا yaǧzū.`

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

`wa-'innA sawfa tudriku-nA 'l-manAyA muqadd"araT"aN وإنا سوف`
`wa-`innā sawfa tudriku-na 'l-manāyā muqaddaratan.`

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

`qAla isma`" قال اسمع qāla 'sma', jA'at" hinduN جاءت هند` `ǧā'at`
`Hindun, `sabIhuN bi-man q"u.ti`at" qadamA-hu شبيه بمن قُطعت`
`šabīhun bi-man quṭi'at qadamā-hu.`

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

`wa-"ust"u`mila وأستعمل wa-'stu'mila, huwa "inhazama هو أنهزم huwa`
`'nhazama, al-"intiqA.du الانتقاض al-'ntiqāḍu.`

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:—

¹⁹See Lagally (2004, p. 22)

q"Ala q"A'iluN قَالَ قَاتِلْ *qāla qā'il^{un}*, ibn-u 'abI 'u.saybi`aT-"a
 اَصْبِيْعَة Ibn^u 'Abī 'uṣaybi'at-'.
 (b) *tanwīn*:—

madInaT"aN مَدِيْنَة *madīnat^{an}*, bAb"aN بَابَا *bāb^{an}*, hud"aN_A هُدًى *hudāⁿ*,
 ʿsay'"iN شَيْءٍ *šay'ⁱⁿ*.

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

(a) *ʿalif^u* *ʿl-waṣlⁱ*:—

fI "istisqA'-a اَسْتِسْقَاءُ فِي *stisqā'^a*, wa-"istisqA'-u اَسْتِسْقَاءُ *wa-'stisqā'^u*,
 qAla "uhrub fa-lan tuqtala قَاتِلْ اَهْرُبْ فَلَنْ تُقْتَلَ *qāla 'hrub fa-lan tuqtala*.

(b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala قَاتِلْ اَقْتُلْ فَلَنْ تُقْتَلَ *qāla 'qtul fa-lan tuqtala*, mA jA'at" mini imra'aTiN مَا جَاءَتْ مِنْ امْرَأَةٍ *mā ḡā'at mini 'mra'atⁱⁿ*, kam" qad" ma.dat" min" laylaTiN كَمْ قَدْ مَضَتْ مِنْ لَيْلَةٍ *kam qad maḡat 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 اَلْجَمْرُ الصَّفِيُّ اَلَّذِي كَانَ يَقْرَأُونَ *al-ḡamr^u 's-sayfiyy^u 'llaḡī 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 ²⁰		ArabTeX notation
	dmg	loc	
ء	ʿ	ʿ	"'"
أ	ʾā	ʾā	A"'"
إ	ʿ	ʿ	a"'"

²⁰See below section 6 on page 24.

Letter	Transliteration		ArabTeX notation
	dmg	loc	
أ	’	’	u"'
و	’	’	w"'
إ	’	’	i"'
ئ	’	’	y"'

Table 4: “Quoted” *hamzah*

As one can see from table 4 on page 20, 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, *أُ* is encoded ⟨w" 'a⟩, while *وُ* is encoded ⟨w" 'u⟩. In the latter example, the second straight double quote encodes the *sukūn* in voc mode in accordance with the rule laid above on pages 19–20.

'a`dA'ukum أَعْدَاؤُكُمْ 'a`dA|"'"ukum أَعْدَاءُكُمْ 'a`dA'ikun
أَعْدَائِكُمْ 'a`dA|"'"ikun أَعْدَاءُكُمْ 'a`dA'ikun.

4.5 The “pipe” character (|)

In the terminology of ArabTeX, the “pipe” character ‘|’ is referred to as the “invisible consonant”. Hence, as already seen above in section 4.4.1 on page 20, 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 16 on page 18 and in footnote 18 on page 18. One more example follows:—

bi-qran|nUn-a بَقْرَانُونٌ *bi-Qrānnūn^a*, “in Crannon” (Thessaly, Greece).²¹

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

4.6 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*

²¹See more context on page 20.

4.7 Digits

4.7.1 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

4.7.2 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 *’abġad* numbers are usually distinguished from the surrounding words by a stroke placed over them.

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

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

REM. *a*. As can be seen in the above given example, `arabluatex` expresses the *’abġad* numbers in Roman numerals if it finds the command `\abjad` 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 4.9 on the next page:—

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

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

4.8 Additional characters

In the manuscripts, the unpointed letters, *al-ḥurūfu ’l-muḥmalatu*, 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 ²²		ArabT _E X notation
	dmg	loc	
ب	<i>ḅ</i>	<i>b</i>	.b

²²See below section 6 on page 24.

Letter	Transliteration		ArabT _E X notation
	dmg	loc	
د	<i>ḍ</i>	<i>d</i>	<code>ˆd</code>
ف	<i>f̣</i>	<i>f</i>	<code>.f</code>
ق	<i>q̣</i>	<i>q</i>	<code>.q</code>
ك	<i>ḳ</i>	<i>k</i>	<code>.k</code>
ن	<i>ṇ</i>	<i>n</i>	<code>.n</code>
⸮	<i>(</i>	<i>(</i>	<code>((</code>
⸭	<i>)</i>	<i>)</i>	<code>))</code>

Table 5: Additional Arabic codings

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

4.9 Arabic emphasis

As already seen in section 4.7.2 on page 22, 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 command `\aemph{<Arabic text>}` to use the same technique to emphasize words, like so:—

`\abjad{45}: kitAbu-hu \aemph{fi l-ʿAdāt-i}` 45: مة: كُتَابُهُ فِي الْعَادَاتِ
kitābu-hu fi l-ʿĀdātⁱ.

5 Special applications

Linguistics The same horizontal stroke as the *taṭwīl* (see section 4.6 on page 21) may be encoded ``; `<BB>` will receive the *tašdīd*. This is useful to make linguistic annotations and comments on vowels:—

Bu Ba Bi BuN BaN BiN ----- *u a i un an in*, BBu BBa BBi ----- *u a i*, B--aN
C - *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: `()`
- square brackets: `[]`
- angle brackets: `<>`
- braces: `{ }`

`\abracces` 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 command `\abracces`:—

```

1 \begin{arab}
2   \abracces{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 أَيْ كَانَ مِنَ الْمُقَاتِلَةِ وَكَانَ حَتَّ أُمِّي مِنْ عُظَمَاءِ بُيُوتِ الزَّمَاذِمَةِ.

6 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 1935.²³ **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.²⁴

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 command `\SetTranslitConvention{<mode>}`, 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 command `\SetTranslitStyle{<style>}`, where `<style>` may be any font shape selection command, eg. `\upshape`, `\itshape`, `\slshape`, and so forth.

²³See Brockelmann et al. (1935).

²⁴See <http://www.loc.gov/catdir/cpsd/roman.html> for the source document concerning Arabic language.

New feature
v1.4

`\SetTranslitFont`

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

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

`\cap` **Proper names** Proper names or book titles that must have their first letters uppercased may be passed as arguments to the command `\cap{word}`. `\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*
Ishāq^a, `\cap{'u-tm-an-u}` عُثْمَانُ *Uṭmān^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}` هَالِدُ بْنُ عَبْدِ اللَّهِ عَوْفُ بْنُ سَعْدٍ خَلْدُ بْنُ زَيْدٍ ضَرَبَ *daraba Zayd^u bn^u*
Hālidⁱⁿ Sa'd^a bn^a 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` viz. 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-*
hammad^u bn^u Zakariyyā^a 'r-Rāzī.

6.1 Additional note on dmḡ convention

New feature
v1.3

According to Brockelmann et al. (1935, p. 6), Arabic *ʾirāb* may be rendered into dmḡ 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, *ʾirāb* 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 مُهْدَمَةً مِثْلَ تَنْوِينِ: *ra'aytu ḡami`am muhaddamat_{am} 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 (*muʾrab*) 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`allimuN مَعْلِمٌ *mu'allim^{un}*, kA'inuN كَائِنٌ *kā'in^{un}*, kA'inAtuN 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 *qāḍi^{āna}*, al-.zulm-Atu الظُّلْمَاتُ *aẓ-ẓulm^{ātu}*.

REM. *a*. As the *tanwīn* is passed over in pronunciation when it is followed by the letters ر, ل, و, م, ي (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^{al} lākinna-hu labisa ḡubbat^{am} 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 مَنَّعَ النَّاسَ كَافَّةً مِّنْ
مَّنَعِ النَّاسَ كَافَّةً مِّنْ mana`a 'n-nās^a kāffat^{am} mim muḥāṭabati-hi 'aḥad^{un} bi-sayyidi-nā.

6.2 Examples

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

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

7 Buckwalter input scheme

Even though *arabluatex* is primarily designed to process the ArabTeX notation, it can also process the Buckwalter input scheme to a large extent.²⁵ The Buckwalter scheme is actually processed in two steps, as it is first converted into ArabTeX. Then, once this is accomplished, the ArabTeX scheme is processed through the above described functions. In this way, the Buckwalter input scheme can make the most of the *arabluatex* special features that are presented in section 2.2 on page 5.

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

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

²⁵See <http://www.qamus.org/transliteration.htm>

‘base’, ‘xml’ and ‘safe’ schemes `arabluatex` can use any of the so-called Buckwalter ‘base’, ‘xml’ or ‘safe’ schemes as they are described in Habash (2010, pp. 25–26).²⁶ However, the following limitation apply to the ‘base’ and ‘xml’ schemes: the braces { and }, which are used to encode ا and ئ, 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 `arabluatex`. The additional characters that are defined in table 5 on page 22 are also available.

Letter	Transliteration ²⁷		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>h</i>	<i>h</i>	H	H
خ	<i>ḥ</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>‘</i>	<i>‘</i>	E	E
ف	<i>ǧ</i>	<i>gh</i>	g	g
ق	<i>f</i>	<i>f</i>	f	f
ك	<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

²⁶I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in `arabluatex`.

²⁷See section 6 on page 24.

Letter	Transliteration		Buckwalter notation	
	dmg	loc	base/xml	safe
ء	’	’	’	C
آ	ā	’ā		M
أ	’	’	>	O
ؤ	’	’	&	W
إ	’	’	<	I
ئ	’	’]	Q
ـ	—	—	~	~
ا	,	,	[L
أ				
ـ	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>
ـ	<i>u</i>	<i>u</i>	<i>u</i>	<i>u</i>
ـ	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>
ـ	<i>an</i>	<i>an</i>	F	F
ـ	<i>un</i>	<i>un</i>	N	N
ـ	<i>in</i>	<i>in</i>	K	K
ـ	—	—	o	o
ا	ā	ā	˘	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 6 on page 24 for more information. However, it should be pointed out again that only accurate coding produces accurate transliteration. It is therefore at the very least highly advisable to use the hyphen for tying the definite article and the inseparable particles (viz. prepositions, adverbs and conjunctions) to words, like so:—

Al-EaAlamu الْعَالَمُ *al-ālam^u*, Al-camsu الشَّمْسُ *aš-šams^u*, bi-SinaAEapi
Al-T~ib~i, بِصِنَاعَةِ الطِّبِّ *bi-ṣinā‘atⁱ ’t-ṭibbⁱ*.
wa-Al-l~ehi وَاللَّهِ *wa-’l-lāhⁱ*, Al-Hamdu li-l~ehi اَلْحَمْدُ لِلَّهِ *al-ḥamd^u*
li-llāhⁱ.

Similarly, 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-LntiqAADI` gives اِلَى الْاِنْتِقَاضِ as expected, but only `<ilY Al-intiqADI` can be transliterated as *ila 'l-intiqāḍi* with the correct vowel ⟨i⟩ in place of the *'alif^u 'l-waṣlⁱ*.

8 Unicode Arabic input

As said above in section 7 on page 27 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 options `voc`, `fullvoc`, `novoc` or `trans` 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}
```

9 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}}` مَهْ كَاتِبُهُ فِي الْعَادَاتِ 45
kitābu-hu fi 'l-Ādātⁱ.²⁸

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.}.
```

²⁸This is odd in Arabic script, but using such features as `\emph` or `\textbf` is a matter of personal taste.

8 \end{arab}

إِنَّ أَبِي كَانَ مِنَ الْمُقَاتِلَةِ^a, وَكَانَتْ أُمِّي مِنْ عُظَمَاءِ بُيُوتِ الزَّمَاظِمَةِ^b.

^aالْمُقَاتِلَةُ: الْمُقَاتِلِينَ.
^bالزَّمَاظِمَةُ: طَائِفَةٌ مِنَ الْفُرْسِ.

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

`\LR{<arg>}` is designed to typeset its argument from left to right. It may be used in an Arabic environment, either `\arb{<Arabic text>}` or `\begin{arab} <Arabic text> \end{arab}`, for short insertions of left-to-right text, or to insert any L^AT_EX command that would otherwise be rejected by `arabluatex`, such as commands the argument of which is expected to be a dimension or a unit of measurement.

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

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 أَتَلَقَ ذُو الْقَرْنَيْنِ إِلَى أُمَّةٍ أُخْرَى فِي ﴿مَطْلَعِ الشَّمْسِ﴾ وَلَا بِنَاءَ لَهُمْ يَوْمَئِذٍ مِنَ الشَّمْسِ.

`\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^a ابْنُ عَمْرٍو

^a “*zayd* is the son of *‘amr*”: the second noun is not in apposition to the first, but forms part of the predicate...

When footnotes are typeset from right to left, it may happen that the numbers of the footnotes that are at the bottom of the page be typeset in the wrong direction. For example, instead of an expected number 18, one may get 81. `arabluatex` is not responsible for that, but should it happen, it may be necessary to redefine in the preamble the `LATEX` macro `\thefootnote` like so:—

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

`\FixArbFtnmk` Another solution is to put in the preamble, below the line that loads `arabluatex`, the command `\FixArbFtnmk`. However, for more control over the layout of footnotes marks, it is advisable to use the package `scrextend`.²⁹

`\LRmarginpar` The command `\LRmarginpar` 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 30 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}
```

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

9.1 Environments

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

²⁹See <http://ctan.org/pkg/koma-script>; read the documentation of KOMA-script for details about the `\deffootnotemark` and `\deffootnote` commands.


```

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

```

كَانَ أَبُو الْهَذِيلِ أَهْدَى إِلَى مُوسَى دَجَاجَةً. وَكَانَتْ دَجَاجَتُهُ الَّتِي أَهْدَاهَا دُونَ مَا كَانَ يَتَّخِذُ لِمُوسَى. وَلَكِنَّهُ
يَكْرَهُهُ وَيُحْسِنُ خُلُقَهُ أَظْهَرَ التَّعَجُّبِ مِنْ سَمْنِهَا وَطَيْبِ لَحْمِهَا. وَكَانَ أَبُو الْهَذِيلِ يُعْرِفُ بِالْأَمْسَاكِ الشَّدِيدِ.
فَقَالَ: «وَكَيْفَ رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ الدَّجَاجَةَ؟» قَالَ: «كَانَتْ عَجَبًا مِنْ الْعَجَبِ!» فَيَقُولُ:

« وَتَدْرِي مَا جَنَسُهَا؟ وَتَدْرِي مَا سَنُهَا؟ فَإِنَّ الدَّجَاجَةَ إِذَا تَطَيَّبَ بِالْجَنَسِ وَالسِّنِّ. وَتَدْرِي بِأَيِّ شَيْءٍ كُنَّا نُسَمِّنُهَا؟ وَفِي أَيِّ
مَكَانٍ كُنَّا نَعْلِفُهَا؟ »

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

9.1.1 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       `arabiyuN.
6     \item[s] max.tU.tu 'l-maktabaTi 'l-'ahliyyaTi bi-\cap{bArIs} 2859
7       `arabiyuN.
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 ملي طهران ٥٢١.

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

```

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

```

القانون في الطبّ

- الْفَنُّ الْأَوَّلُ فِي حَدِّ الطِّبِّ وَمَوْضُوعَاتِهِ مِنَ الْأُمُورِ الطَّبِيعِيَّةِ وَيَشْتَمِلُ عَلَى سِتَّةِ تَعَالِيمٍ
 - التَّلْعِيمُ الْأَوَّلُ [وَهُوَ فَصْلَانِ]
 - الْفَصْلُ الْأَوَّلُ

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

9.2 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: ”غَرِيبُ أَمْرِكَ يَا صَدِيقِي أَتَصَدِّقُ الْجَمَارَ 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.

9.3 reledmac

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

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

9.4 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 8 on page 30 for more details). Please note that `arabluatex` takes care of formatting the Arabic: therefore, it is recommended to load the `quran` package with the option `nopar`, 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﴾

10 Future work

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

- (a) Short-term:
 - i. Support for typesetting Arabic poetry.
 - ii. The *Qurʾān*: support for typesetting the *Qurʾān*.
 - iii. 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 3.
- (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 Arab \TeX notation and the transliteration tables. Include them in `arabluatex`. See section 4.8 on page 22.

11 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 [2016/11/14 v1.5 An ArabTeX-like interface for LuaLaTeX]
4 \RequirePackage{ifluatex}
```

`arabluatex` requires Lua \LaTeX 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}

```

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

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

Here begins the real work: load arabluatex.lua:

```
30 \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.

```
31 \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:

```

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

```

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

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

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

```
37 \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:

```

38 \AtBeginDocument{\ifdef{\LR}%
39 {\RenewDocumentCommand{\LR}{m}{\bgroup\textdir TLT\rmfamily#1\egroup}}
40 {\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.

```

41 \AtBeginDocument{\ifdef{\RL}%
42 {\RenewDocumentCommand{\RL}{m}{\bgroup\textdir TRT\rmfamily#1\egroup}}
43 {\NewDocumentCommand{\RL}{m}{\bgroup\textdir TRT#1\rmfamily\egroup}}}

```

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

```

44 \AtBeginDocument{\ifdef{\aemph}%
45 {\RenewDocumentCommand{\aemph}{m}{\overline{\text{#1}}}}
46 {\NewDocumentCommand{\aemph}{m}{\overline{\text{#1}}}}}

```

<code>\SetInputScheme</code>	<p>arabluatex is designed for processing Arab_T_E_X input notation. <code>\SetInputScheme</code> 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’.</p> <pre> 47 \def\al@input@scheme{arabtex} 48 \NewDocumentCommand{\SetInputScheme}{m}{\def\al@input@scheme{#1}} </pre>
<code>\SetArbEasy</code> <code>\SetArbEasy*</code> <code>\SetArbDflt</code>	<p>By default, arabluatex applies complex rules to generate euphonic <i>tašdīd</i>, <i>ʾalif mamdūdah</i> and <i>sukūn</i> depending on the modes which are selected, either <code>voc</code>, <code>fullvoc</code> or <code>trans</code>. Such refinements can be discarded with <code>\SetArbEasy</code>, either globally in the preamble or at any point of the document. Note that <code>\SetArbEasy</code> keeps the <i>sukūn</i> that is generated, while the starred version <code>\SetArbEasy*</code> takes it away. Default complex rules can be set back at any point of the document with <code>\SetArbDflt</code>.</p> <pre> 49 \def\al@arb@rules{dflt} 50 \NewDocumentCommand{\SetArbEasy}{s}{% 51 \IfBooleanTF{#1} 52 {\def\al@arb@rules{easynosukun}} 53 {\def\al@arb@rules{easy}}} 54 \NewDocumentCommand{\SetArbDflt}{}{\def\al@arb@rules{dflt}} </pre>
<code>\SetTranslitFont</code>	<p>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 <code>fontspec</code> package.</p> <pre> 55 \def\al@trans@font{\rmfamily}% 56 \NewDocumentCommand{\SetTranslitFont}{m}{\def\al@trans@font{#1}} </pre>
<code>\SetTranslitStyle</code>	<p>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:</p> <pre> 57 \def\al@trans@style{\itshape}% 58 \NewDocumentCommand{\SetTranslitStyle}{m}{\def\al@trans@style{#1}} </pre>
<code>\SetTranslitConvention</code>	<p><code>\SetTranslitConvention{⟨convention⟩}</code> may be used to change the transliteration convention, which is <code>dmg</code> by default:</p> <pre> 59 \def\al@trans@convention{dmg} 60 \NewDocumentCommand{\SetTranslitConvention}{m}{\def\al@trans@convention{#1}} </pre>
<code>\arbup</code> <code>\NoArbUp</code> <code>\ArbUpDflt</code> <code>\SetArbUp</code>	<p>By default, <code>\arbup</code> is set to <code>\textsuperscript</code>. This is how the <i>tanwīn</i> that takes place at the end of a word should be displayed in <code>dmg</code> mode. <code>\NoArbUp</code> may be used either in the preamble or at any point of the document in case one wishes to have the <i>tanwīn</i> on the line. The default rule can be set back with <code>\ArbUpDflt</code> at any point of the document. Finally <code>\SetArbUp</code> may be used to customize the way <i>tanwīn</i> is displayed: this command takes the formatting directives as argument, like so: <code>\SetArbUp{⟨code⟩}</code>.</p> <pre> 61 \NewDocumentCommand{\al@arbup@dflt}{m}{\thinspace#1}% 62 \NewDocumentCommand{\al@arbup}{m}{\al@arbup@dflt{#1}} 63 \NewDocumentCommand{\arbup}{m}{\al@arbup{#1}} 64 \NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt} </pre>

```

65 \NewDocumentCommand{\NoArbUp}{-}{\RenewDocumentCommand{\al@arbup}{m}{##1}}
66 \NewDocumentCommand{\SetArbUp}{m}{\RenewDocumentCommand{\al@arbup}{m}{#1}}

\cap Proper Arabic names or book titles should be passed to the command \cap so that
they have their first letters uppercased. \cap is actually coded in Lua.
67 \DeclareDocumentCommand{\cap}{m}%
68 {\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.
69 \DeclareDocumentCommand{\txarb}{+m}{\bgroup\textdir
70 TRT\arabicfont#1\egroup}
71 \DeclareDocumentCommand{\txtrans}{+m}{\bgroup\textdir
72 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.
73 \NewDocumentEnvironment{txarab}{-}{%
74 \par%
75 \booltrue{al@rlmode}%
76 \pdir 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.
77 \DeclareDocumentCommand{\arb}{O{\al@mode} +m}%
78 {\edef\@tempa{#1}%
79 \ifx\@tempa\al@mode@voc%
80 \bgroup\textdir TRT\arabicfont%
81 \luairect{tex.sprint(processvoc(\luastringN{#2},
82 \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
83 \else%
84 \ifx\@tempa\al@mode@fullvoc%
85 \bgroup\textdir TRT\arabicfont%
86 \luairect{tex.sprint(processfullvoc(\luastringN{#2},
87 \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
88 \else%
89 \ifx\@tempa\al@mode@novoc%
90 \bgroup\textdir TRT\arabicfont%
91 \luairect{tex.sprint(processnovoc(\luastringN{#2},
92 \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
93 \else%
94 \ifx\@tempa\al@mode@trans%
95 \bgroup\textdir TLT\al@trans@style%
96 \luairect{tex.sprint(processtrans(\luastringN{#2},
97 \luastring0{\al@trans@convention},

```



```

98   \luastring0{\al@arb@rules},
99   \luastring0{\al@input@scheme}}))\egroup%
100 \else%
101 \fi\fi\fi\fi}

```

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

```

102 \NewEnviron{arab}[1][\al@mode]%
103 {\par\edef\@tempa{#1}%
104   \ifx\@tempa\al@mode@voc%
105     \booltrue{al@rlmode}%
106     \bgroup\pdir TRT\textdir TRT\arabicfont%
107     \luadirect{tex.sprint(processvoc(\luastring0{\BODY},
108       \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
109     \else%
110       \ifx\@tempa\al@mode@fullvoc%
111         \booltrue{al@rlmode}%
112         \bgroup\pdir TRT\textdir TRT\arabicfont%
113         \luadirect{tex.sprint(processfullvoc(\luastring0{\BODY},
114           \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
115         \else%
116           \ifx\@tempa\al@mode@novoc%
117             \booltrue{al@rlmode}%
118             \bgroup\pdir TRT\textdir TRT\arabicfont%
119             \luadirect{tex.sprint(processnovoc(\luastring0{\BODY},
120               \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}}))\egroup%
121             \else%
122               \ifx\@tempa\al@mode@trans%
123                 \bgroup\pdir TLT\textdir TLT\al@trans@style%
124                 \luadirect{tex.sprint(processtrans(\luastring0{\BODY},
125                   \luastring0{\al@trans@convention},
126                   \luastring0{\al@arb@rules},
127                   \luastring0{\al@input@scheme}}))\egroup%
128                 \else \fi\fi\fi\fi}\par]

```

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

```

129 \AtBeginDocument{%
130   \ifdefined\abjad%
131     \RenewDocumentCommand{\abjad}{m}%
132     {\luadirect{tex.sprint(abjadify(#1))}}%
133   \else%
134     \NewDocumentCommand{\abjad}{m}%
135     {\luadirect{tex.sprint(abjadify(#1))}}
136 \fi}

```

`\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`.

```

137 \NewDocumentCommand{\abraces}{+m}{%
138   \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.

```

139 \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.

```

140 \DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pdir
141   TLT\LR{\footnote{#1}}\egroup}
142 \DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pdir
143   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.

```

144 \NewDocumentCommand{\FixArbFtnmk}{}{%
145   \@ifpackageloaded{scrextend}%
146   {\AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%
147   {\RequirePackage{scrextend}
148   \AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%

```

That is it. Say goodbye before leaving.

Patches

```

149 \NeedsTeXFormat{LaTeX2e}
150 \ProvidesPackage{arabluatex-patch}%
151 [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:

```

152 \def\list#1#2{%
153   \ifnum \@listdepth >5\relax
154     \@toodeep
155   \else
156     \global\advance\@listdepth\@ne
157   \fi
158   \rightmargin\z@
159   \listparindent\z@

```

```

160 \itemindent\z@
161 \csname @list\romannumeral\the\@listdepth\endcsname
162 \def\@itemlabel{#1}%
163 \let\makelabel\@mklab
164 \@nmbrrlistfalse
165 #2\relax
166 \@trivlist
167 \parskip\parsep
168 \parindent\listparindent
169 \advance\linewidth -\rightmargin
170 \advance\linewidth -\leftmargin
patch begins:
171 \ifbool{al@rlmode}{\advance\@totalleftmargin \rightmargin}%
172 {\advance\@totalleftmargin \leftmargin}
patch ends.
173 \parshape \@ne \@totalleftmargin \linewidth
174 \ignorespaces}
175 \def\@item[#1]{%
176 \if@noparitem
177 \@donoparitem
178 \else
179 \if@inlabel
180 \indent \par
181 \fi
182 \ifhmode
183 \unskip\unskip \par
184 \fi
185 \if@newlist
186 \if@nobreak
187 \@nbitem
188 \else
189 \addpenalty\@beginparpenalty
190 \addvspace\@topsep
191 \addvspace{-\parskip}%
192 \fi
193 \else
194 \addpenalty\@itempenalty
195 \addvspace\itemsep
196 \fi
197 \global\@inlabeltrue
198 \fi
199 \everypar{%
200 \@minipagefalse
201 \global\@newlistfalse
202 \if@inlabel
203 \global\@inlabelfalse
204 {\setbox\z@\lastbox
205 \ifvoid\z@
206 \kern-\itemindent

```

```

207     \fi}%
208     \box\@labels
209     \penalty\z@
210   \fi
211   \if@nobreak
212     \@nobreakfalse
213     \clubpenalty \@M
214   \else
215     \clubpenalty \@clubpenalty
216     \everypar{}\%
217   \fi}%
218   \if@noitemarg
219     \@noitemargfalse
220     \if@nmbrowse
221       \refstepcounter\@listctr
222     \fi
223   \fi
patch begins:
224   \ifbool{al@rlmode}{\sRLbox\@tempboxa{\makelabel{#1}}}{%
225     \sbox\@tempboxa{\makelabel{#1}}}%
226   \ifbool{al@rlmode}{\global\setbox\@labels\hbox dir TRT}{%
227     {\global\setbox\@labels\hbox}{%
patch ends.
228     \unhbox\@labels
229     \hskip \itemindent
230     \hskip -\labelwidth
231     \hskip -\labelsep
232     \ifdim \wd\@tempboxa >\labelwidth
233       \box\@tempboxa
234     \else
235       \hbox to\labelwidth {\unhbox\@tempboxa}%
236     \fi
237     \hskip \labelsep}%
238   \ignorespaces}

```

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

```

239 \long\def\sRLbox#1#2{\setbox#1\hbox dir TRT{%
240   \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 Kom-mission 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. Robertson Smith and M. J. de Goeje. With a forew. by Pierre Cachia. 3rd ed. 2 vols. Beirut: Librairie du Liban.

Change History

v1.0		<code>\SetTranslitFont</code> : For selecting a specific font for transliterated texts	1	39
v1.0.1	General: Initial release		1	
	General: Minor update of the documentation		1	
v1.1	<code>\abjad</code> : New and more flexible <code>\abjad</code> command.		41	
v1.2	<code>\SetArbEasy</code> : New <code>\SetArbEasy/\SetArbDflt</code> for ‘modern’ or ‘classic’ Arabic styles.		39	
v1.3	<code>\arbup</code> : <i>ʾrāb</i> is now written as superscript text in <code>dmg</code> mode by default.		39	
v1.4	<code>\SetInputScheme</code> : <code>\SetInputScheme</code> may be used to process other input schemes such as ‘Buckwalter’		39	
		<code>\SetTranslitFont</code> : For selecting a specific font for transliterated texts		39
		<code>\abracas</code> : New command <code>\abracas</code> which expresses its argument between braces.		42
		<code>\SetArbEasy*</code> : this starred version discards the <i>sukūn</i> in addition to what is already discarded by <code>\SetArbEasy</code>		39
		<code>txarab</code> : New <code>txarab</code> environment for typesetting running paragraphs in unicode Arabic		40
		General: Compatibility with the <code>quran</code> package		36
		Environments may be nested inside the <code>arab</code> environment		32

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>	213
	45

<code>\@beginparpenalty</code> . 189	<code>\al@arbup@dflt</code>	<code>\booltrue</code> 36,
<code>\@clubpenalty</code> 215 61, 62, 64	75, 105, 111, 117
<code>\@donoparitem</code> 177	<code>\al@input@scheme</code> . .	<code>\box</code> 208, 233
<code>\@ifpackageloaded</code> . 145 47, 48,	Brockelmann, Carl . 24, 25
<code>\@inlabelfalse</code> 203	82, 87, 92, 99,	
<code>\@inlabeltrue</code> 197	108, 114, 120, 127	C
<code>\@item</code> 175	<code>\al@mode</code> 12,	<code>\cap</code> 25, 67
<code>\@itemlabel</code> 162	13, 14, 15, 77, 102	<code>\clubpenalty</code> . . 213, 215
<code>\@itempenalty</code> 194	<code>\al@mode@fullvoc</code> . .	<code>\color@endgroup</code> . . . 240
<code>\@labels</code> 19, 84, 110	<code>\color@setgroup</code> . . . 240
208, 226, 227, 228	<code>\al@mode@novoc</code>	<code>csquotes</code> (package) . . . 35
<code>\@listctr</code> 221 20, 89, 116	
<code>\@listdepth</code> 153, 156, 161	<code>\al@mode@trans</code>	D
<code>\@minipagefalse</code> . . . 200 21, 94, 122	<code>\DeclareOption</code>
<code>\@mklab</code> 163	<code>\al@mode@voc</code> 18, 79, 104 12, 13, 14, 15
<code>\@nbitem</code> 187	<code>\al@trans@convention</code>	<code>\deffootnote</code> . . 146, 148
<code>\@ne</code> 156, 173	. . . 59, 60, 97, 125	E
<code>\@newlistfalse</code> 201	<code>\al@trans@font</code>	environments:
<code>\@nmbrrlistfalse</code> . . . 164 55, 56, 72	<code>arab</code> 7, 102
<code>\@nobreakfalse</code> 212	<code>\al@trans@style</code>	<code>txarab</code> 30, 73
<code>\@noitemargfalse</code> . . 219	. . . 57, 58, 95, 123	<code>\everypar</code> 199, 216
<code>\@tempa</code> 78, 79,	<i>Amiri</i> 5	<code>\ExecuteOptions</code> 16
84, 89, 94, 103,	<i>amiri</i> (package) 5	
104, 110, 116, 122	<i>arab</i> (environment) 7, 102	F
<code>\@tempboxa</code> 224,	<i>ArabTeX</i> 3, 19	<code>\FixArbFtnmk</code> . . . 32, 144
225, 232, 233, 235	<i>Arabic-English Lexicon</i> ,	<code>fontspec</code> (package) . . .
<code>\@toodeep</code> 154	<i>An</i> 17 4, 25, 39
<code>\@topsep</code> 190	<code>\arabicfont</code>	<code>\footnote</code> 141, 143
<code>\@totalleftmargin</code> 32, 33, 35,	<code>fullvoc</code> (option) 5, 8
. . . . 171, 172, 173	70, 76, 80, 85,	
<code>\@trivlist</code> 166	90, 106, 112, 118	G
	<i>arabluatex</i> (package) .	<code>\global</code> . . . 156, 197,
 2, 4–6, 8,	201, 203, 226, 227
	10–12, 14–19,	<i>Grammar of the Arabic</i>
	22–32, 35–37, 39	<i>Language, A</i> . .
	<i>arabtex</i> (package) . . .	5–8, 12–15, 17, 22
	. . . 3, 4, 15, 19, 37	H
A	<i>arabulatex</i> (package) . . 15	Habash, Nizar Y. 28
<code>\abjad</code> 22, 129	<i>arabxetex</i> (package) . 4, 15	<code>\hbox</code> 226, 227, 235, 239
<code>\abraces</code> 24, 137	<code>\arb</code> 6, 77	Hosny, Khaled 5
<code>\addpenalty</code> . . 189, 194	<code>\arbup</code> 26, 61	<code>\hskip</code> 229, 230, 231, 237
<code>\addvspace</code> 190, 191, 195	<code>\ArbUpDflt</code> 26, 61	
<code>\advance</code> 156,		I
169, 170, 171, 172		<code>\if@inlabel</code> . . 179, 202
<code>\aemph</code> 23, 44	B	<code>\if@newlist</code> 185
<code>\al@arb@rules</code>	<i>babel</i> (package) 35	<code>\if@nmbrrlist</code> 220
49, 52, 53, 54,	<i>babel-french</i> (package) . 35	<code>\if@nobreak</code> . . 186, 211
82, 87, 92, 98,	<i>bidi</i> (package) 44	<code>\if@noitemarg</code> 218
108, 114, 120, 126	<code>\BODY</code> 107, 113, 119, 124	<code>\if@noparitem</code> 176
<code>\al@arbup</code>	<code>\boolfalse</code> 37	
. 62, 63, 64, 65, 66		

`\ifbool` .. 171, 224, 226
`\IfBooleanTF` 51
`\ifdim` 232
`\ifhmode` 182
`\ifluatex` 5
`\ifnum` 153
`\ifvoid` 205
`\ignorespaces` . 174, 238
`\indent` 180
Introduction to Arabic
Natural Lan-
guage Process-
ing 28
`\itemindent` 160, 206, 229
`\itemsep` 195
`\itshape` 57

K
`\kern` 206
KOMA-script (package) 32

L
`\labelsep` 231, 237
`\labelwidth` 230, 232, 235
Lagally, Klaus 3, 19
Lane, Edward William . 17
`\lastbox` 204
`\leftmargin` .. 170, 172
`\linewidth` 169, 170, 173
`\list` 152
`\listparindent` 159, 168
`\long` 239
`\LR` 31, 38,
141, 143, 146, 148
`\LRfootnote` ... 31, 140
`\LRmarginpar` ... 32, 139
luabidi (package) ... 4, 38
`\luadirect` 30, 31, 68,
81, 86, 91, 96,
107, 113, 119,
124, 132, 135, 138
luaotload (package) ... 25
`\luastringN` 68,
81, 86, 91, 96, 138
`\luastringO` . 82, 87,
92, 97, 98, 99,
107, 108, 113,
114, 119, 120,
124, 125, 126, 127

M
`\makelabel` 163, 224, 225
`\marginpar` 139

N
`\newbool` 29
`\NewDocumentEnvironment`
..... 73
`\NewEnviron` 102
`\newfontfamily` 35
`\NoArbUp` 26, 61
novoc (option) 5, 8

O
options:
fullvoc 5, 8
novoc 5, 8
trans 5, 8
voc 5, 8
`\overline` 45, 46

P
`\par` 74, 76,
103, 128, 180, 183
paralist (package) . 34, 35
`\pardir` 36, 37,
76, 106, 112,
118, 123, 140, 142
`\parindent` 168
`\parsep` 167
`\parshape` 173
`\parskip` 167, 191
`\penalty` 209
polyglossia (package) .
..... 4, 38, 41
`\ProcessOptions` 17
`\ProvidesPackage` 2, 150

Q
quoting (package) 33
quran (package) 36

R
`\refstepcounter` ... 221
reledmac (package) ... 36
`\RequirePackage` ...
. 4, 22, 23, 24,
25, 26, 27, 28, 147
`\rightmargin`
..... 158, 169, 171

`\RL` 31, 41
`\RLfootnote` ... 31, 140
`\rmfamily`
. 39, 40, 42, 43, 55
`\romannumeral` 161

S
`\sbox` 225
scrextend (package) 32, 42
`\SetArbDflt` 6, 49
`\SetArbEasy` 6, 49
`\SetArbEasy*` 6, 49
`\SetArbUp` 26, 61
`\setbox`
204, 226, 227, 239
`\SetInputScheme` . 27, 47
`\setLR` 32, 37
`\setRL` 32, 36
`\SetTranslitConvention`
..... 24, 59
`\SetTranslitFont` 25, 55
`\SetTranslitStyle` 24, 57
`\sRLbox` 224, 239

T
`\text` 45, 46
`\textdir` 36, 37,
39, 40, 42, 43,
69, 71, 76, 80,
85, 90, 95, 106,
112, 118, 123, 139
`\textsuperscript` ... 61
`\the` 161
`\thefootnotemark` ..
..... 146, 148
`\thinspace` 61
trans (option) 5, 8
Transliteration der ara-
bischen Schrift,
Die 24, 25
txarab (environment) .. 36
txarab (environment)
..... 30, 73
`\txarb` 30, 69
`\txtrans` 69

U
`\unhbox` 228, 235
`\unskip` 183

V	W	Z
	<code>\wd</code> 232	
	Wright, W. LL.D . . .	<code>\z0</code> 158, 159,
<code>voc</code> (option) 5, 8	5–8, 12–15, 17, 22	160, 204, 205, 209