

The arabluatex package

v1.11 – 2018/03/31

Robert Alessi

alessi@robertalessi.net*

CNRS UMR 8167 Paris (France)

Contents

License and disclaimer	2	4.8.2 The <i>abjad</i>	25
1 Introduction	3	4.9 Additional characters	26
1.1 arabluatex is for Lua [®] TeX	4	4.10 Arabic emphasis	26
2 The basics of arabluatex	4	5 Arabic poetry	27
2.1 Activating arabluatex	4	5.1 Example	30
2.2 Options	5	6 Special applications	31
2.2.1 Classic contrasted with modern typesetting of Arabic	6	7 Transliteration	33
2.3 Typing Arabic	7	7.1 Additional note on dmg convention .	36
2.3.1 Local options	8	7.2 Examples	38
3 Standard ArabTeX input	9	8 Buckwalter input scheme	38
3.1 Consonants	9	9 Unicode Arabic input	41
3.2 Additional characters	10	10 L^AT_EX Commands in Arabic environments	41
3.3 Vowels	11	10.1 New commands	43
3.3.1 Long vowels	11	10.2 Environments	44
3.3.2 Short vowels	11	10.2.1 Lists	45
4 arabluatex in action	12	10.3 csquotes	47
4.1 The vowels and diphthongs	12	10.4 Two-argument commands	47
4.2 Other orthographic signs	14	10.5 quran	48
4.3 Special orthographies	20	11 Future work	49
4.4 Quoting	21	12 Implementation	49
4.4.1 Quoting the <i>hamzah</i>	22	References	59
4.5 The “pipe” character ()	23	Change History	59
4.6 Putting back on broken contextual analysis rules	23	Index	61
4.7 Stretching characters: the <i>taṭwīl</i> .	25		
4.8 Digits	25		
4.8.1 Numerical figures	25		

*Institutional email: robert.alessi@cnrs.fr

List of Tables

1	ArabTeX consonants	9	5	“Quoted” <i>hamzah</i>	23
2	ArabTeX additional characters . . .	10	6	Additional Arabic codings	26
3	ArabTeX long vowels	11	7	Additional Arabic marks	32
4	ArabTeX short vowels	11	8	Buckwalter scheme	39

Abstract

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

License and disclaimer

License applicable to the software

arabluatex — Processing ArabTeX notation under Lua^ATeX.

Copyright © 2016–2018 Robert Alessi

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

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

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

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

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

This release of arabluatex consists of the following source files:

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


gpl3+

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

License applicable to this document

Copyright © 2016–2018 Robert Alessi

CC BY-SA 4.0

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

1 Introduction

In comparison to Prof. Lagally’s outstanding ArabTeX,¹ ArabLuaTeX is at present nothing more than a modest piece of software. Hopefully—if I may say so—it will eventually provide all of its valuable qualities to the LuaL^ATeX users.

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

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

¹ See <http://ctan.org/pkg/arabtex>

² Lagally (2004, p. 2).

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

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

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

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

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

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

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

2 The basics of `arabluatex`

2.1 Activating `arabluatex`

As usual put in your preamble:

```
\usepackage{arabluatex}
```

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

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

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

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

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

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

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

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

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

Other Arabic fonts may behave differently.

2.2 Options

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

`voc`

`default`

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

`fullvoc`

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

`novoc`

None of the diacritics is showed in `novoc` mode, unless otherwise specified (see “quoting” technique below section 4.4 on page 21).

`trans`

This mode transliterates the ArabTeX input into one of the accepted standards. At present, three standards are supported (see below section 7 on page 33 for more details):

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

loc *Library of Congress*;

arabica *Arabica*.

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

⁵Hosny (2015).

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

New feature
v.1.2

2.2.1 Classic contrasted with modern typesetting of Arabic

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

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

samA'uN سَمَاءُ *samā^{un}*, jA'a جَاءَ *ǧā'a*, yatasA'alUna يَتَسَاءَلُونَ *yatasā'alūna*⁷
(see on page 16 for further details).

- The euphonic *tašdīd* is generated (see on page 16).
- In *fullvoc* mode, the *sukūn* is expressed.
- In such words as ظِمًا, شَيْئًا and the like, the *hamzah* alone is not written over the letter *yā* with no diacritical points below as in ظِمًا, شَيْئًا, but over a horizontal stroke placed in the continuation of the preceding letter.

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

New feature
v1.4.4

`\SetArbEasy`
`\SetArbEasy*`
`\SetArbDflt`

Such refinements as ‘classic’ *maddah* may be discarded by the `\SetArbEasy` command, either globally in the preamble or locally at any point of the document. The difference between `\SetArbEasy` and its ‘starred’ version `\SetArbEasy*` is that the former keeps the *sukūn* that is generated by the *fullvoc* mode, while the latter further takes it away. Default ‘classic’ rules may be set back at any point of the document with the `\SetArbDflt` command. Assimilation rules laid on item b on page 17 may also be applied by the ‘starred’ version of this command `\SetArbDflt*` either in the preamble or at any point of the document.⁸ Examples follow:—

(a) `\SetArbDflt`:

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

(b) `\SetArbDflt*`:

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

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

⁸For an example, see section 5.1 on page 30.

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

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

2.3 Typing Arabic

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

```

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

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

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

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

```

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

XVI. فَوَاعِلُ.

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

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

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

like so:—

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

أَتَى صَدِيقٌ إِلَى جِحَا يَطْلُبُ مِنْهُ حِمَارَهُ لِيَرْكَبَهُ فِي سَفَرَةٍ قَصِيرَةٍ فَقَالَ لَهُ: "سَوْفَ أُعِيدُهُ إِلَيْكَ فِي الْمَسَاءِ
وَأَدْفَعُ لَكَ أُجْرَةً." فَقَالَ جِحَا: "أَنَا آسَفٌ جِدًّا أَنِّي لَا أَسْتَطِيعُ أَنْ أُحَقِّقَ لَكَ رَغْبَتَكَ فَالْحِمَارُ لَيْسَ هُنَا
الْيَوْمَ." وَقَبْلَ أَنْ يَتِمَّ جِحَا كَلَامَهُ بَدَأَ الْحِمَارُ يَنْهَقُ فِي إِصْطِلِهِ. فَقَالَ لَهُ صَدِيقُهُ: "إِنِّي أَسْمَعُ حِمَارَكَ يَا جِحَا
يَنْهَقُ." فَقَالَ لَهُ جِحَا: "غَرِيبٌ أَمْرُكَ يَا صَدِيقِي أَتَصَدِّقُ الْحِمَارَ وَتَكْذِبُنِي؟"

2.3.1 Local options

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

voc — `\arb[voc]{\Arabic text}`;

`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] \dots \end{arab}`, where $\langle mode \rangle$ may be any of `voc`, `fullvoc`, `novoc` or `trans`.

3 Standard ArabTeX input

3.1 Consonants

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

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

⁹See below section 7 on page 33.

¹⁰See below, *Rem. a*. For *ʾalif* as a consonant, see Wright (1896, i. 16 D). The *hamzah* itself is encoded <'> and may be followed by either $\langle u, a \rangle$ or $\langle i \rangle$. See below section 4.2 on page 14.

Letter	Transliteration			ArabT _E X notation
	dmg	loc	arabica	
ه	<i>h</i>	<i>h</i>	<i>h</i>	h
و	<i>w</i>	<i>w</i>	<i>w</i>	w
ي	<i>y</i>	<i>y</i>	<i>y</i>	y ¹¹
ة	<i>ah</i>	<i>ah</i>	<i>a</i>	T

Table 1: Standard ArabT_EX (consonants)

REM. *a*. Please note that in all cases of elision, the *ʿalifu ʾl-waṣli* is expressed only by the vowel that accompanies the omitted *hamzah*: ⟨*u, a, i*⟩ as in *wa-inhazama* ^{وَأَنهَزَمَ} *wa-ʾnhazama*. For more details on the definite article and the *ʿalifu ʾl-waṣli* see section 4.2 on page 17.

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

REM. *b*. The letter ي with two points below, ^{يَا} *الياء المثناة من تحته*, may also be written without diacritical points as ى. When it is used as a consonant, it is encoded aY, where a recalls the *fathah* placed above the preceding letter in vocalized Arabic, like so: qaYʾuN ^{قَايُؤُنْ} *qayʾun*, ^saYʾuN ^{سَايُؤُنْ} *sayʾun*, ^saYʾaN ^{سَايُؤُنْ} *sayʾan*.

The same result may be achieved by encoding this letter as .y, like so: qa.yʾuN ^{قَا.يُؤُنْ} *qayʾun*, ^sa.yʾuN ^{سَا.يُؤُنْ} *sayʾun*, ^sa.yʾaN ^{سَا.يُؤُنْ} *sayʾan*.

3.2 Additional characters

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

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

Table 2: Standard ArabT_EX (additional characters)

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

¹²See below section 7 on page 33.

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

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

¹⁵See footnote 14.

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

3.3 Vowels

3.3.1 Long vowels

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

Letter	Transliteration ¹⁶			ArabTeX notation
	dmg	loc	arabica	
ا	\bar{a}	\bar{a}	\bar{a}	A
و	\bar{u}	\bar{u}	\bar{u}	U
ي	\bar{i}	\bar{i}	\bar{i}	I ¹⁷
ى ¹⁸	\bar{a}	\bar{a}	\bar{a}	_A or Y
أ	\bar{a}	\bar{a}	\bar{a}	_a
و	\bar{u}	\bar{u}	\bar{u}	_u
ي	\bar{i}	\bar{i}	\bar{i}	_i

Table 3: Standard ArabTeX (long vowels)

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

REM. b. Defective writings, such as ا, *al-'alif^u 'l-mahḍūfat^u*, or defective writings of \bar{u} and \bar{i} are encoded _a_u and _i respectively, e.g. _d_alika ذَلِكَ, al-mal_a'ikaT-u 'l-ra.hm_an-u الْمَلِكَةُ الرَّحْمَنُ, .hu_dayfaT-u bn-u 'l-yamAn_i الْحَيَاءُ الْمُنَاةُ مِنْ نَحْبِهَا for *Huḍayfat^u bn^u 'l-Yamānī*, etc.

REM. c. The letter ي with two points below, أَلْيَاءُ الْمُنَاةُ مِنْ نَحْبِهَا, may also be written without diacritical points as ى. When it is used as a long vowel, it is encoded iY, where i recalls the *kasrah* placed below the preceding letter in vocalized Arabic, like so: liY لِي *lī*, yam[^]siY يَمْسِي *yamṣī*.

3.3.2 Short vowels

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

Letter	Transliteration ¹⁹			ArabTeX notation
	dmg	loc	arabica	
ا	<i>a</i>	<i>a</i>	<i>a</i>	a
و	<i>u</i>	<i>u</i>	<i>u</i>	u

¹⁶See below section 7 on page 33.

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

¹⁸= *al-'alif^u 'l-maqṣūrat^u*.

¹⁹See below section 7 on page 33.

Letter	Transliteration			ArabT _E X notation
	dmg	loc	arabica	
- ا	<i>i</i>	<i>i</i>	<i>i</i>	i
- ان	<i>an</i>	<i>an</i>	<i>an</i>	aN
- un	<i>un</i>	<i>un</i>	<i>un</i>	uN
- in	<i>in</i>	<i>in</i>	<i>in</i>	iN

Table 4: Standard ArabT_EX (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.²⁰

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

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

- at the commencement of a word, to indicate that a connective *ʿalif* is needed, with the exception of the article (see below section 4.4 on page 21);
- 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 أَقِيطٌ *'aqiṭ^{uN}*.
la-hu لَهُ *la-hu*, .hujjaTuN هُجَّةٌ *ḥuğğat^{uN}*.

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

²⁰See below section 4.4 on page 21.

qAla قَالَ *qāla*, bI`a بَعَّ *bī`a*, .tUruN طُور *ṭūr^{un}*, .tInuN طِين *ṭīn^{un}*,
murU'aTuN مُرُوَّةٌ *murū`at^{un}*.

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

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

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

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

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

al-l_ah-u أَلَهُ *al-lāh^u*, 'il_ahUN إِلَهُ *ilāh^{un}*.
al-ra.hm_an-u أَلرَّحْمَنُ *ar-raḥmān^u*, l_akin لَكِنْ *lākin*, h_ahunA هَاهُنَا *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 أَلْعَاصِ *al-Āṣī*.

Silent و/ي Some words ending with اة are usually written حوة or نوة instead of لاة: see Wright (1896, i. 12 A). arabuatex preserves that particular writing; the same applies to words ending in ية for لة. Long vowels ⟨U, I⟩ shall receive no *sukūn* after a *'alif maḥḍūfah* and are discarded in **trans** mode:

.hay_aUTuN حَيوة *hayāt^{un}*, .sal_aUTuN صَلوة *ṣalāt^{un}*, mi~sk_aUTuN مَشْكورة *miš-kāt^{un}*, tawr_aITuN تَوْرِيَّة *tawrāt^{un}*.

And so also: al-rib_aIT-u أَلرِبِيَّة *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 عَمَرُوا *amr^{un}*, `amraNU عَمَرُوا *amr^{an}*, `amriNU عَمَرُوا *amrⁱⁿ*.

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

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

yatasā'alūna, murU'aTuN مُرُوءٌ, taw'amūN تَوَّعٌ taw'am^{un}, murū'at^{un}, ta'xIruN تَأْخِيرٌ ta'hīr^{un}, ta'axxara تَأَخَّرَ ta'ahhara, ji'tu-ka جِئْتُكَ ġi'tu-ka, qA'iluN قَائِلٌ qā'il^{un}, .hIna'i_diN hīna'idⁱⁿ حِينَدِ, hay'aTuN هَيْتَ hay-'at^{un}, hay'AtuN هَيْتَاتِ hay'āt^{un}.

From Wright (1896, i. 14 B):— All consonants, whatsoever, not even 'alifhəmzatum excepted, admit of being doubled and take *tašdid*. Hence we speak and write ra'AsuN رَأْسٌ ra'ās^{un}, sa'AluN سَأَلٌ sa'al^{un}, na'AjuN نَأَجٌ na'āġ^{un}.

Final hamzah: xa.ta'uN خَطَأٌ haṭa'^{un}, xa.ta'aN خَطَأٌ haṭa'^{an}, xa.ta'iN خَطِئٌ haṭa'ⁱⁿ, 'aqra'u أَقْرَأُ aqra'u, taqra'Ina تَقْرَأِينَ taqra'ina, taqra'Una تَقْرَأُونَ taqra'una, yaqra'na يَقْرَأْنَ yaqra'na, yaxba'Ani يَخْبَأَنِ yaḥba'āni, xaba'A خَبَأَ haḥba'ā, xubi'a خُبِئَ hubi'a, xubi'UA خُبِئُوا hubi'ū, jA'a جَاءَ ġā'a, ridA'uN رَدَأَ ridā'^{un}, ridA'aN رَدَأَ ridā'^{an}, jI'a جِئَ ġī'a, radI'iN رَدِئَ radī'ⁱⁿ, sU'uN سُوءٌ sū'^{un}, .daw'uN دَاوٌ daw'^{un}, qay'iN قَايَ qay'ⁱⁿ, ~sifA'I شَفَاتِي šifā'ī, man~sa'I مَنْشَى manša'ī nisa'uN نِسَاءٌ ni-sā'^{un}, nisa'u-hu نِسَاؤُهُ nisa'u-hu, nisa'i-hi نِسَائِهِ nisa'ī-hi, nisa'I نِسَائِي nisa'ī.

~say'uN سَيَئٌ say'^{un}, ~say'aN سَيَأٌ say'^{an}, ~say'iN سَيِئٌ say'ⁱⁿ, al-~say'-u أَلْسَيُّ aš-say'^u, 'a~syA'-u أَشْيَاءُ ašyā'^u, 'a~syA'-a أَشْيَاءُ ašyā'^a, .zim'aN ظِمَأٌ ḡim'^{an}, radI'aN رَدِئاً radī'^{an}.

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

Imperatives of verbs that have the *hamzah* as the first radical are other cases of *taḥfif^u 'l-hamzatⁱ:* i'sir إِيسِرْ īsir, i'_dan إِدَنْ idan, u'mul أُوْمَلْ ūmul. arabluatex also provides ways of encoding those words when the initial 'alif comes into waṣl, so as to make the 'alif waṣl fall away when preceded by وَ or فَ: wa-'sir وَأَسِرْ wa-'sir, fa-'_dan فَادَنْ fa-

'dan, fa-'ti فَاتِ fa-'ti, wa-'tamirUA وَأَتَمَرُوا wa-'tamirū; or be retained outside the imperative, as in fa-i'tazarat فَاتَّزَرْتُ fa-'tazarat, ba`da i'tilAfīN بَعْدَ اِتِّلَافٍ ba'da 'tilāfⁱⁿ.

The strange spelling of mi'at^{un}: mi'aTuN مَائَةٌ mi'at^{un}, mi'atAni مَائَتَانِ mi'atāni, mi'atayni مَائَتَيْنِ mi'atayni, mi'Una مِئُونِ mi'ūna, mi'AtuN مِئَاتٌ mi'ātun, mi'aN_A مَائِي mi'aⁿ. Of course, the 'pipe' character can be used to prevent this rule from being applied (see section 4.5 on page 23): mi'a|TuN مِئَةٍ mi'at^{un}.

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

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

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

'is'AduN إِسَادٌ 'is'ād^{un}, 'AkilUna أَكِلُونُ ākilūna, 'a'manna أَمَنَّا āmannā, al-qur'An-u الْقُرْآنُ al-qur'ān^u.
jA'a جَاءَ jā'a, yatasA'alUna يَتَسَاءَلُونَ yatasā'alūna, ridA'uN رِدَاءٌ ridā^{un}, xaba'A حَبَا ḥaba'a, yaxba'Ani يَخْبَانُ yaḥba'āni.

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

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

`allaqa عَلَّقَ `allaqa, mAdduN مَدَّ mād^{dun}, 'ammara أَمَرَ 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-ˆsams-u` الشَّمْسُ *aš-šams^u*, or `aˆs-ˆsams-u` الشَّمْسُ *aš-šams^u*.

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

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

Note the absence of *sukūn* above the passed over ن in the following examples, each of which is accompanied by a consistent transliteration: `min rabbi-hi` مِنْ رَبِّي *mir rabbi-hi*, `min layliN` مِنْ لَيْلٍ *mil laylⁱⁿ*, `'an yaqtula` أَنْ يَقْتُلَ *'ay yaqtula*.

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

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

- (c) With the letter ت after the dentals ث, د, ذ, ض, ط, ظ in certain parts of the verb: this kind of assimilation, e.g. `labittu` لَبِثْتُ for `labittu` لَبِثْتُ, will be discarded here, as it is largely condemned by the grammarians (see Wright 1896, i. 16 B–C).

The definite article and the 'alif^u 'l-waṣlⁱ At the beginning of a sentence, *ʾl* is never written, as `al-ḥamd li-l-lāh` الحمد لله; instead, to indicate that the *ʾalif* is a connective *ʾalif* (*ʾalif^u 'l-waṣlⁱ*), the *hamzah* is omitted and only its accompanying vowel is expressed:

`al-.hamd-u li-l-l-ah-i` الحمد لله *al-ḥamd^u li-l-lāhⁱ*.

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

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

`al-`, or `a<solar letter>-`

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

- (b) Definite article inside sentences is encoded

`'l-` or `'<solar letter>-`.

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

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

Particles:—

- (a) *li-*: *'alif^u 'l-waṣlⁱ* is omitted in the article أَلْ when it is preceded by the preposition لِ: li-l-rajul-i لِلرَّجُلِ *li-r-raġulⁱ*.
If the first letter of the noun be ل, then the ل of the article also falls away, but arablumatex 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 لَلْحَقِّ *la-l-ḥaqq^u*.
- (c) With the other particles, *'alif^u 'l-waṣlⁱ* is expressed: fI 'l-madIna T-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 قَالَ أَقْتُلُ *qāla 'qtul*, huwa inhazama هُوَ أَنْهَزَمَ *huwa 'nhazama*, wa-ustu`mila وَأَسْتَعْمِلَ *wa-'stu`mila*, qad-i in.sarafa قَدْ أَنْصَرَفَ *qadi 'nṣarafa*, al-iqtidAr-u الْأَقْتِدَارُ *al-iqtidār^u*, 'il_A 'l-inti qA.d-i إِلَى الْإِتِّقَاضِ *ila 'l-intiqādⁱ*, law istaqbala لَوْ اسْتَقْبَلَ *lawi 'staqbala*.

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

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

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

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

fI 'l-nAs-i فِي النَّاسِ *fi 'n-nāsⁱ*, 'abU 'l-wazIr-i أَبُو الْوَزِيرِ *'abu 'l-wazīrⁱ*,
fI 'l-ibtidA'-i فِي الْإِبْتِدَاءِ *fi 'l-ibtidā'ⁱ*, _dU 'l-i`lAl-i ذُو الْأَعْلَالِ *du 'l-i`lālⁱ*,
maqH_A 'l-'amIr-i مَقْهَى الْأَمِيرِ *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* → *āu*. arablumatex detects the cases in which this rule applies:—

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

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

'antumU 'l-kA_dib-Una أَنْتُمْ الْكَاذِبُونَ *'antumU 'l-kāḏib^{ūna}*, ra'aytumU 'l-raJul-a رَأَيْتُمُ الرَّجُلَ *ra'aytumU 'r-raġul^a*, mani 'l-ka_d_dAb-u مَنِ الْكَادِدِ ابْنِ *mani 'l-kaddāb^u*, qatalati 'l-rUm-u قَتَلَتِ الرُّومُ *qatalati 'r-Rūm^u*.

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

mu.hammaduN 'l-nabI مُحَمَّدُ النَّبِيِّ *Muḥammad^{uni} 'n-nabī*, salAmuN ud_hulUA سَلَامٌ أَدْخُلُوا *salām^{umu} 'dhulū*, qa.sIdata-hu fI qatl-i قَصِيدَتُهُ فِي قَتْلِ أَبِي مُسْلِمٍ الَّتِي يَقُولُ فِيهَا *qaṣīdata-hu fī qatlⁱ*
'llatI yaqUlu fI-hA *'llatī yaqūlu fī-hā*.
'Abī Muslimⁱⁿⁱ 'llatī yaqūlu fī-hā.

4.3 Special orthographies

The name of God The name of God, ﷲ, is compounded of the article ﷲ, and ﷲ (noted ﷲ with the defective ‘*alif*’) so that it becomes ﷲ; then the *hamzah* is suppressed, its vowel being transferred to the ﷲ before it, so that there remains ﷲ (I refer to Lane, *Lexicon*, I. 83 col. 1). Finally, the first ﷲ is made quiescent and incorporated into the other, hence the *tašdīd* above it. As *arabuatex* never requires a solar letter to be written twice (see above, on page 16), the name of God is therefore encoded `al-l_ah-u` or `'l-l_ah-u`:—

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

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

`'a_hAfu` `mina` `'l-malik-i` `'lla_dI` `ya.zlimu` `'l-nAs-a` ﷲ ﷲ ﷲ ﷲ *'aḥāfu mina 'l-malikⁱ 'lladī yaẓlimu 'n-nās^a*, ``udtu` `'l-`say_h-a` `'lla_dI` `huwa.marI.duN` ﷲ ﷲ ﷲ ﷲ *'udtu 'š-šayh^a 'lladī huwa marīd^{un}*, `mA` `'anA` `bi-'lla_dI` `qA'iluN` `la-ka` ``say'aN` ﷲ ﷲ ﷲ ﷲ *mā 'anA bi-'lladī qā'il^{un} la-ka šay'^{an}*.

`'ari-nA` `'lla_dayni` `'a.dallA-nA` `mina` `'l-jinn-i` `wa-'l-'ins-i` ﷲ ﷲ ﷲ ﷲ *'ari-na 'lladayni 'adallā-nā mina 'l-ǧinnⁱ wa-'l-'insⁱ*.

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

`fa-'innA` `na_dkuru` `'l-.sawt-ayni` `'l-la_dayni` `rawaynA-humA` ``an` `ja.h.zaT-a` ﷲ ﷲ ﷲ ﷲ ﷲ ﷲ *fa-'innā nadkuru 'š-šawt^{ayni} 'l-ladayni rawaynā-humā 'an Ḡaḥẓat^a*.

²⁴Note the “pipe” character ‘|’ here after `yA` and below after `fa` before footnote mark 25: it is needed by the `dmg` transliteration mode as in this mode any vowel at the commencement of a word preceded by a word that ends with a vowel, either short or long, is absorbed by this vowel viz. ‘*alā t-tariqⁱ*’. See section 4.5 on page 23 on the “pipe” and section 7 on page 33 on `dmg` mode.

²⁵See footnote 24.

And also: al-la_dAni اللّٰذَانِ *al-ladāni*, al-la_dayni اللّٰذَيْنِ *al-ladayni*,
al-latAni اللّٰتَانِ *al-latāni*, al-latayni اللّٰتَيْنِ *al-latayni*, al-lAtI اللّٰتِي *al-*
lāti, al-lA' |Ati²⁶ اللّٰءَاتِ *al-lā'āti*, al-lA' I اللّٰتِي *al-lā'i*, and so forth.

4.4 Quoting

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

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

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

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

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

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

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

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

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

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

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

²⁶Note here the “pipe” character '|': as already stated on page 16, the sequence 'A usually encodes *ʿalif* with *hamzah* followed by *ʿalif* of prolongation, which is represented in writing *ʿalif* with *maddah*: *Ā*. The “pipe” character prevents this rule from being applied. See section 4.5 on page 23.

²⁷See Lagally (2004, p. 22)

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

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

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

- (a) Short and long vowels:—

q"Ala q"A'iluN قَال قَائِل *qāla qā'il^{un}*, ibn-u 'abI 'u.saybi`aT-"a
ابْنُ أَبِي أُصَيْبَةَ *Ibn^u 'Abi 'Uṣaybi'at^a*.

- (b) *tanwīn*:—

madInaT"aN مَدِينَة *madīnat^{an}*, bAb"aN بَابَا *bāb^{an}*, hud"aN_A هُدَى *hudāⁿ*,
سَيَّيْ *sayⁱⁿ*.

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

- (a) *'alif^u 'l-waṣl'*:—

fI "istiq.sA'-iN فِي اسْتِقْصَاءٍ *fi 'stiqṣā'ⁱⁿ*, wa-"istiq.sA'-uN وَأَسْتِقْصَاءٌ *wa-'stiqṣā'^{un}*, qAla "uhrub fa-lan tuqtala قَال أَهْرُب فَلَنْ تُقْتَلَ *qāla 'hrub fa-lan tuqtala*.

- (b) *sukūn*:—

qAla "uqtul" fa-lan tuqtala قَال أَقْتُلْ فَلَنْ تُقْتَلَ *qāla 'qtul fa-lan tuq-*
tala, 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 maḍat min laylatⁱⁿ.

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

al-jamr-u 'l-.sayfiyy-u 'lla_dI kAna bi-q"rAn" |nUn-a الْجَمْرُ الصَّيْفِيُّ *al-ḡamr^u 'ṣ-ṣayfiyy^u 'llaḍī kāna bi-Qrānnūn^a*.

4.4.1 Quoting the hamzah

As said above in section 4.2 on page 14, the *hamzah* is always written ⟨ ' ⟩, its carrier being determined by contextual analysis. “Quoting” that straight single quote character like so: ⟨ " ' ⟩ allows to determine the carrier of the *hamzah* freely, without any consideration for the context. Table 5 gives the equivalents for all the possible carriers the *hamzah* may take.

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

Table 5: “Quoted” *hamzah*

As one can see from table 5, the carrier of the *hamzah* is inferred from the letter that precedes the straight double quote (”). Of course, any “quoted” *hamzah* may take a short vowel, which is to be written *after* the ArabTeX equivalent for the *hamzah* itself, namely (’). For example, ؤ is encoded (w”’a), while ؤ is encoded (w”’”). In the latter example, the second straight double quote encodes the *sukūn* in *voc* mode in accordance with the rule laid above on pages 21–22.

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

4.5 The “pipe” character (|)

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

Aside from that usage, the “pipe” character is used to prevent almost any of the contextual analysis rules that are described above from being applied. Two examples have already been given to demonstrate how that particular mechanism works in footnote 24 on page 20 and in footnote 26 on page 21. One more example follows:—

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

As one can see, the “pipe” character between the two (n) prevents the necessary *tašdīd* rule (page 16) from being applied.

4.6 Putting back on broken contextual analysis rules

In complex documents such as critical editions where footnotes and other kind of

²⁸See below section 7 on page 33.

²⁹See more context on the preceding page.

annotations can be particularly abundant, the contextual analysis rules that are described above may be broken by L^AT_EX commands. To take an example, consider the following:—

```

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

```

This is wrong:

فَلَمَّا رَأَوْا^a النَّجْمَ...

^aA footnote that interferes with the contextual analysis.

According to the rule stated on page 19, the diphthong in *ra'aw* must be resolved into two simple vowels before the '*alif*^u *l-wasl*ⁱ', as رَأَوْا النَّجْمَ.

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

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

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

```

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

```


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

Zayd^{unie} 'bn^u Amrⁱⁿ!^f

^aA footnote that interferes with the contextual analysis.

^bA footnote that interferes with the contextual analysis.

^cA footnote that interferes with the contextual analysis.

^dSee footnote 22 on page 18.

^eA footnote that interferes with the contextual analysis.

^fSee footnote 22 on page 18.

4.7 Stretching characters: the taṭwīl

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

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

4.8 Digits

4.8.1 Numerical figures

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

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

4.8.2 The abjad

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

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

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

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

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

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

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

4.9 Additional characters

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

For the time being, the following is included:—

Letter	Transliteration ³⁰			ArabTeX notation
	dmg	loc	arabica	
ب	<i>ḅ</i>	<i>b</i>	<i>b</i>	.b
د	<i>ḍ</i>	<i>d</i>	<i>d</i>	˘d
ف	<i>f̣</i>	<i>f</i>	<i>f</i>	.f
ق	<i>q̣</i>	<i>q</i>	<i>q</i>	.q
ك	<i>ḳ</i>	<i>k</i>	<i>k</i>	.k
ن	<i>ṇ</i>	<i>n</i>	<i>n</i>	.n
ⵍ	(((((
ⵍ)))))

Table 6: Additional Arabic codings

`ʿafAman.tUs Gal.(M) .fmn.n.ts (sic) Gal.(E1)`, أفامنطوس Gal.(M) منطس (sic) Gal.(E1), *ʿafāmanṭūs Gal.(M) fmnṇṭs (sic) Gal.(E1)*.

4.10 Arabic emphasis

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

`\aemph` One may use the `\aemph{<Arabic text>}` command to use the same technique to

³⁰See below section 7 on page 33.

emphasize words, like so:—

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

New feature
v1.9.2

`\aemph*`

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

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

45: مة: كُتَابُهُ فِي الْعَادَاتِ
kitābu-hu fi 'l-`Ādātⁱ.

5 Arabic poetry

New feature
v1.6

`arabverse`

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

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

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

The description of the optional arguments follows:—

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

`width` `width=<length>` Default: 0.3\linewidth

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

`gutter` `gutter=<width>` Default: 0.15 x (hemistich width)

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

`metre` `metre=<name>` Default: none

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

`delim` `delim=true|false` Default: false

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

`\SetHemistichDelim`

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

As the preceding one, this named argument does not need a value as it defaults to `true` if it is used. If so, Unicode Arabic input is expected in the `arabverse` environment instead of ASCII ArabTeX or Buckwalter input schemes. See section 9 on page 41 for more details.

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

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

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

- (a) If the two hemistichs be connected with one another by a prominent horizontal flexible stroke, the *taṭwīl* should be used, like so: `[--]` (see section 4.7 on page 25). Of course, the ending word of the *ṣadr* and the word at the commencement of the *ağuz* must have the *taṭwīl* too so that the proper shapes of the letters be selected. Consider for example the following:—

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

لَا أَرَى مَنْ عَهِدْتُ فِيهَا فَأُبْكِي الْيَوْمَ دَهْلًا وَمَا يَرُدُّ الْبُكَاءُ

As one can see, *triple hyphens* have been used. In the *ṣadr*, the first hyphen triggers the rules that are related to the definite article and the *ʿalif^u* *ʾl-waṣlⁱ*,³¹ while the following two select the figure of the letter *lām* connected with a following letter. In the *ağuz*, the last two hyphens select the letter *yā* connected with a preceding letter, while the first one is simply discarded in this mode, but still may appear as it should, if the `trans` mode be selected:—

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

lā ʾarā man ʾahidtu fī-hā fa-ʾabki ʾl- -yawma dahl^{an} wa-mā yaruddu ʾl-bukāʾu

³¹See section 4.2 on page 17

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

```

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

```

غَيْرَ أَنِّي قَدْ اسْتَعِينُ عَلَى الْهَمِّ إِذَا خَفَّ بِالْثَوِيِّ النَّجَاءُ
 بِزَفْوْفٍ كَانَهَا هِقْلَةً أُمُّ رِثَالٍ دَوِيَّةً سَقْفَاءُ

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

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

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

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

If the footnotes are abundant in number, it is advised to load the `footnotehyper` package which `arabluatex` will then use to typeset any kind of footnote that is called from the arguments of the `\bayt` command.³²

³²The `footnote` package can also be used for the same effect. However, it must be loaded *after* `arabluatex`.

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

5.1 Example

Here follow the first lines of Imru'ū 'l-Qaysi's *Mu'allaqah*. In this example, `\SetArbDflt*` has been selected so as to mark the *ʿidgām* that is fit to this declamatory poetry:—³³

```

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

```

قَالَ أَمْرُؤُ الْقَيْسِ فِي مَعْلَقَتِهِ:

1	بَسَقَطَ اللَّوَى بَيْنَ الدَّخُولِ وَالْخَوَالِ	قَفَا نَبِكُ مِنْ ذِكْرِي حَبِيبٍ وَمَنْزِلِ
2	لَمَّا نَسَجْتَهَا مِنْ جَنُوبٍ وَشَمَالِ	فَتَوَضَّحَ فَأَلْمَقَرَّةَ لَمْ يَعْفُ رَسْمُهَا
3	وَقِيْعَانِهَا كَأَنَّهُ حَبُّ فَلْفُلِ	تَرَى بَعْرَ الْأَرَامِ فِي عَرَصَاتِهَا
4	لَدَى سُرَرَاتِ الْحَيِّ نَاقِفُ حَنْظَلِ	كَأَنِّي غَدَاةَ الْبَيْنِ يَوْمَ تَحْمَلُوا
5	يُقُولُونَ لَا تَهْلِكُ أَسَى وَتَجْمَلِ	وَقُوفًا بِهَا صَحْبِي عَلَيَّ مَطْمَعِمْ

³³Please note that for the time being only the assimilation rules that are laid on item **b** on page 17 are applied. See section 2.2.1 on page 6 for more information. None of the editions of the *Mu'allaqāt* that I know of feature the *ʿidgām* in the Arabic text, although it is often strongly marked in declamation.

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

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

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

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

6 Special applications

Linguistics The same horizontal stroke as the *taṭwīl* (see section 4.7 on page 25) may be encoded $\langle B \rangle$; $\langle BB \rangle$ will receive the *tašdīd*. This is useful to make linguistic annotations and comments on vowels:—

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

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. *arabuatex* will automatically fit those symbols to the direction of the text. For the time being, the following symbols are supported:

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

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

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

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

New feature
v1.11

`\arbmark`

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

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

Codepoint	Shorthand	Glyph	Transliteration
FDFD	bismillah	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ	<i>bi-'smi 'Llāhi 'r-rahmāni 'r-rahīmi</i>
FDF5	salam	صَلَام	<i>ṣalla 'Llāhu 'alay-hi wa-sallama</i>
FDF6	slm	سَلَام	<i>ṣalla 'Llāhu 'alay-hi wa-sallama</i>
FDFB	jalla	جَلَالَا	<i>ǧalla ǧalāla-hu</i>

Table 7: Additional Arabic marks

New feature
v1.11

`\newarbmark`

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

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

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

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

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

It must be noted that `\newarbmark` expects Arab \TeX input scheme inside `\arb[trans]{}` to the exclusion of buckwalter input scheme.

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

```

1 \SetArbDflt*
2 \newarbmark{sly}{\textcolor{red}{~~~~06d6}}{}
3 \newarbmark{jim}{\textcolor{red}{~~~~06da}}{}
4 \begin{arab}

```



```

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

```

سُورَةُ النَّسَاءِ، ١٩:

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

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

7 Transliteration

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

This mode transliterates the ArabTeX input into one of the accepted standards.

As said above on page 5, three standards are supported at present:

dmg *Deutsche Morgenländische Gesellschaft*, which was adopted by the International Convention of Orientalist Scholars in Rome in 1935.³⁴ **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,³⁵

³⁴See Brockelmann et al. (1935).

³⁵See <http://www.loc.gov/catdir/cpsd/roman.html> for the source document concerning Arabic language.

New feature
v1.8

arabica *Journal of Arabic and Islamic Studies/Revue d'études arabes et islamiques*: this standard is most widely used by scholars in the field of Arabic studies.³⁶
More standards will be included in future releases of arabluatex.

`\SetTranslitConvention`

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

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

`\SetTranslitStyle`

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

New feature
v1.4

`\SetTranslitFont`

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

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

`\uc`

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

`\uc{.hunayn-u}` `bn-u` `\uc{'is.h_aq-a}` حُنَيْنُ بْنُ إِسْحَاقَ *Hunayn^u bn^u*
Ishāq^a, `\uc{`u_tm_an-u}` عُثْمَانُ *Utmān^u*, .daraba `\uc{zayd-u}` `bn-u`
`\uc{_h_alidiN}` `\uc{sa`d-a}` `bn-a` `\uc{`awf-i}` `bn-i` `\uc{`abd-i}`
`\uc{'l-l_ah-i}` ضَرْبَ زَيْدُ بْنُ خَلْدٍ سَعْدُ بْنُ عَوْفِ بْنِ عَبْدِ اللَّهِ *daraba Zayd^u bn^u*
Hālidiⁱⁿ Sa`d^a bn^a Awfⁱ bnⁱ Abdⁱ 'Llāhⁱ.

³⁶See http://www.brill.nl/files/brill.nl/specific/authors_instructions/ARAB.pdf.

However, `\uc` must be used cautiously in some very particular cases, for the closing brace of its argument may prevent a rule from being applied. To take an example, as seen above on page 19, the transliteration of مُحَمَّدُ 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 مُحَمَّد like so: `\uc{mu.hammaduN}` is wrong, because the closing brace would prevent `arabluatex` from detecting the sequence `<-uN>` immediately followed by `<'l->`. Fortunately, this can be circumvented in a straightforward way by inserting only part of the noun in the argument of `\uc` vz. up to the first letter that is to be uppercased, like so: `\uc{m}u.hammaduN`.

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

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

New feature
v1.10

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

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

New feature
v1.10

`\prname` **Proper names outside Arabic environments** Transliterated proper names inserted in paragraphs of English text should be printed in the same typeface as the surrounding text. `\prname{<Arabic proper name>}` is provided to that effect:³⁷—

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

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

³⁷ Just as `\uc`, `\prname` is also able to process proper names consisting of several subsequent words.

The following example shows how `\prname` can be used in conjunction with the `nameauth` package to have Arabic proper names printed first in full then in partial forms:³⁸—

```

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

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

7.1 Additional note on `dmg` convention

According to Brockelmann et al. (1935, p. 6), Arabic *ʿrāb* may be rendered into `dmg` in three different ways:

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

`\arbup` By default, `arabluatex` applies rule **b**. Once delimited by a set of Lua functions, *ʿrāb* is passed as an argument on to a `\arbup` command which is set to `\textsuperscript`.

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

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

```

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

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

³⁸See the documentation of `nameauth` for more details: <https://ctan.org/pkg/nameauth>

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.

’i’rāb boundaries Every declinable noun (*mu’rab*) 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`allimN ^{مُعَلِّمٌ} *mu`allim^{un}*, kA'inuN ^{كَائِنٌ} *kā'in^{un}*, kA'inAtuN ^{كَائِنَاتٌ} *kā'in^{ātun}*, \uc{`amraNU} ^{عَمَرُوا} *Amr^{an}*, fataN_A ^{فَتَى} *fataⁿ*, qA.diNI ^{قَاضٍ} *qāḍiⁿ*.

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

REM. a. As the *tanwīn* is passed over in pronunciation when it is followed by the letters ر, ل, م, و, ي (see item b on page 17), 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 ^{كَانَ غَنِيًّا لَكِنَّهُ لَبِسَ ثِيَابًا} *gubbaTaN mumazzaqaN 'aydu-hA* ^{مُزَقًّا} *aydu-hā*.
^{أَيُّدِهَا} *kāna ḡaniyy^{an} lākinna-hu labisa ḡubbat^{an} mumazzaq^{an} 'aydu-hā*.

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

mana`a 'l-nAs-a kAffaT-aN min mu_hA.tabati-hi 'a.had-uN bi-sayyidi-nA ^{مَنْعَ النَّاسِ كُلِّهِمْ} *mana'a 'n-nās^a kāffat^{an} min muḥāṭabati-hi 'aḥad^{un} bi-sayyidi-nā*.
^{بِمَنْعِهِ أَحَدٌ}

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

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

REM. It may so happen, as in the absence of the article before the annexed word, that *arabluatex* be unable to determine which of the above two cases the word ending with *tā’ marbūṭah* falls into. The ‘pipe’ character (see section 4.5 on page 23) may be appended to that word to indicate that what follows is in the construct state: \uc{r}isAlaT fI tartIb qirA'aT| kutub \uc{g}AlInUs *Risālah fī tartīb qirā’at kutub Ġālīnūs*.

Uncertain short vowels In some printed books, it may happen that more than one short vowel be placed on a consonant in cases where the vocalization is uncertain or ambiguous, like so: ^{فَعِلَ} *fa`uaila* ^{فَعِلَ} *fa`/u,a,i/la*.

7.2 Examples

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

‘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 āsīf^{un} ḡidd^{an} annī lā astatī’u an uḥaqqīqa la-ka raġbata-ka fa-l-ḥimār^u laysa huna l-yawm^a.” wa-qabla an yutimma Ġuḥā kalāma-hu bada’a l-ḥimār^u yanhaqu fī iṣṭabli-hi. fa-qāla la-hu ṣadīqu-hu: “innī asma’u ḥimāra-ka yā Ġuḥā yanhaqu.” fa-qāla la-hu Ġuḥā: “ġarīb^{un} amru-ka yā ṣadīqī a-tuṣaddiqu l-ḥimār^a wa-tukaddība-nī?”*

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

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

8 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.³⁹ 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 `\SetInputScheme{<scheme>}` command, where `<scheme>` may be either `arabtex` or `buckwalter`. This command is also accepted in the preamble should one wish to set the input scheme globally, like so:—

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

³⁹See <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).⁴⁰ 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 8 gives the Buckwalter equivalents that are currently used by `arabluatex`. The additional characters that are defined in table 6 on page 26 are also available.

Letter	Transliteration ⁴¹			Buckwalter notation	
	dmg	loc	arabica	base/xml	safe
ا	<i>a</i>	<i>a</i>	<i>a</i>	A	A
ب	<i>b</i>	<i>b</i>	<i>b</i>	b	b
ت	<i>t</i>	<i>t</i>	<i>t</i>	t	t
ث	<i>ṭ</i>	<i>th</i>	<i>ṭ</i>	v	v
ج	<i>ǧ</i>	<i>j</i>	<i>ǧ</i>	j	j
ح	<i>h</i>	<i>h</i>	<i>h</i>	H	H
خ	<i>ḫ</i>	<i>kh</i>	<i>ḫ</i>	x	x
د	<i>d</i>	<i>d</i>	<i>d</i>	d	d
ذ	<i>ḍ</i>	<i>dh</i>	<i>ḍ</i>	*	V
ر	<i>r</i>	<i>r</i>	<i>r</i>	r	r
ز	<i>z</i>	<i>z</i>	<i>z</i>	z	z
س	<i>s</i>	<i>s</i>	<i>s</i>	s	s
ش	<i>š</i>	<i>sh</i>	<i>š</i>	\$	c
ص	<i>ṣ</i>	<i>ṣ</i>	<i>ṣ</i>	S	S
ض	<i>ḍ</i>	<i>ḍ</i>	<i>ḍ</i>	D	D
ط	<i>ṭ</i>	<i>ṭ</i>	<i>ṭ</i>	T	T
ظ	<i>ẓ</i>	<i>ẓ</i>	<i>ẓ</i>	Z	Z
ع				E	E
غ	<i>ǧ</i>	<i>gh</i>	<i>ǧ</i>	g	g
ف	<i>f</i>	<i>f</i>	<i>f</i>	f	f
ق	<i>q</i>	<i>q</i>	<i>q</i>	q	q
ك	<i>k</i>	<i>k</i>	<i>k</i>	k	k
ل	<i>l</i>	<i>l</i>	<i>l</i>	l	l
م	<i>m</i>	<i>m</i>	<i>m</i>	m	m
ن	<i>n</i>	<i>n</i>	<i>n</i>	n	n
ه	<i>h</i>	<i>h</i>	<i>h</i>	h	h
و	<i>w</i>	<i>w</i>	<i>w</i>	w	w
ي	<i>y</i>	<i>y</i>	<i>y</i>	y	y
ى	<i>ā</i>	<i>á</i>	<i>ā</i>	Y	Y
ة	<i>ah</i>	<i>ah</i>	<i>a</i>	p	p

⁴⁰I am grateful to Graeme Andrews who suggested that the ‘safe’ scheme be included in `arabluatex`.

⁴¹See section 7 on page 33.

Letter	Transliteration			Buckwalter notation	
	dmg	loc	arabica	base/xml	safe
ع	’	’	’	’	C
آ	’ā	’ā	’ā		M
أ	’	’	’	>	O
ؤ	’	’	’	&	W
إ	’	’	’	<	I
ئ	’	’	’]	Q
ـ	—	—	—	~	~
ـ	’	’	—	[L
ا	<i>a</i>	<i>a</i>	<i>a</i>	a	a
ـ	<i>u</i>	<i>u</i>	<i>u</i>	u	u
ـ	<i>i</i>	<i>i</i>	<i>i</i>	i	i
ان	<i>an</i>	<i>an</i>	<i>an</i>	F	F
ـ	<i>un</i>	<i>un</i>	<i>un</i>	N	N
ـ	<i>in</i>	<i>in</i>	<i>in</i>	K	K
ـ	—	—	—	o	o
اَ	<i>ā</i>	<i>ā</i>	<i>ā</i>	˘	e
ـ (taṭwīl)	—	—	—	—	—

Table 8: Buckwalter scheme

Transliteration The Buckwalter notation can also be transliterated into any accepted romanization standard of Arabic. See above section 7 on page 33 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-tibbⁱ*.
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-intiqādi* with the correct vowel ⟨i⟩ in place of the *'alif^u 'l-waṣlⁱ*.

9 Unicode Arabic input

As said above in section 8 on page 38 about the Buckwalter input scheme, even though `arabluatex` is primarily designed to process the `ArabTeX` notation, it also accepts Unicode Arabic input. It should be noted that `arabluatex` does in no way interfere with Unicode Arabic input: none of the `voc`, `fullvoc`, `novoc` or `trans` options will have any effect on plain Unicode Arabic for the time being.

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

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

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

10 L^AT_EX Commands in Arabic environments

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

`\abjad{45} kitAbu-hu \emph{fI 'l-\uc{`AdAt-i}}` مَهْ كِتَابُهُ فِي الْعَادَاتِ 45
kitābu-hu fi 'l-Ādātⁱ.⁴²

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.}.
```

⁴²This 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 الزَّمَازِمَةِ^b.

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

Some commands, however, do not expect running text in their arguments, or one may wish to insert English text 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 `LATEX` command that would otherwise be rejected by `arabluatex`, such as commands the argument of which is expected to be a dimension or a unit of measurement.

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

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

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

`\FixArbFtnmk` command. However, for more control over the layout of footnotes marks, it is advisable to use the `scrextend` package.⁴³

`\LRmarginpar` The `\LRmarginpar` command does for marginal notes the same as `\LRfootnote` does for footnotes. Of course, it is supposed to be used in Arabic environments. Note that `\marginpar` also works in Arabic environments, but it acts as any other single-argument command inserted in Arabic environments. The general principle laid on page 41 applies.

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

```

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

```

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

10.1 New commands

New feature
v1.9

In some particular cases, it may be useful to define new commands to be inserted in Arabic environments. From the general principle laid on page 41, it follows that any command that is found inside an Arabic environment is assumed to have Arabic text in its argument which `arabluatex` will process as such before passing it on to the command itself for any further processing. As a result of this feature, such a command as:

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

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

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

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

⁴³See <http://ctan.org/pkg/koma-script>; read the documentation of KOMA-script for details about the `\deffootnotemark` and `\deffootnote` commands.

For example, here follows a way to define a new command `\fvred` to distinguish words with a different color and always print them in fully vocalized Arabic:—⁴⁴

```

1 \MkArbBreak{fvred}
2 \newcommand{\fvred}[1]{\textcolor{red}{\arb[fullvoc]{#1}}}
3 \begin{arab}[voc]
4 _tumma "intalaga _dU 'l-qarn-ayni 'il_A 'ummaT-iN 'u_hr_A fI
5 \fvred{((ma.tli`-i 'l-^sams-i))} wa-lA binA'-a la-hum
6 yu'amminu-hum mina 'l-^sams-i.
7 \end{arab}

```

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

10.2 Environments

New feature
v1.5

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

```

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

```

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

```

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

```

⁴⁴See below section 10.4 on page 47 for more information on two-argument commands.

```

14      bi-'ayyi ^say'iN kunna nusamminu-hA? wa-fI 'ayyi makAniN
      kunna
15      na`lifu-hA?
16      \end{quoting}
17      fa-lA yazAlu fI h_a_dA wa-'l-'A_haru ya.d.haku .da.hkaN
      na`rifu-hu
18      na.hnu wa-lA ya`rifu-hu \uc{'abU} \uc{'l-hu_dayli}.
19      \end{arab}

```

كَانَ أَبُو الْهَذِيلِ أَهْدَى إِلَى مُوسَى دَجَاجَةً. وَكَانَتْ دَجَاجَتُهُ الَّتِي أَهْدَاهَا دُونَ مَا كَانَ يَتَّخِذُ لِمُوسَى. وَلَكِنَّهُ بِكَرَمِهِ وَبِحُسْنِ خُلُقِهِ أَظْهَرَ التَّعَجُّبَ مِنْ سَمْنِهَا وَطَيِّبِ رَحْمَتِهَا. وَكَانَ <أَبُو الْهَذِيلِ> يَعْرِفُ بِالْأَمْسَاكِ الشَّدِيدِ. فَقَالَ: «وَكَيْفَ رَأَيْتَ يَا أَبَا عِمْرَانَ تِلْكَ الدَّجَاجَةَ؟» قَالَ: «كَانَتْ عَجَبًا مِنَ الْعَجَبِ!» فَيَقُولُ:

« وَتَدْرِي مَا جَنَسُهَا؟ وَتَدْرِي مَا سَمْنُهَا؟ فَإِنَّ الدَّجَاجَةَ إِذَا تَطَيَّبُ بِالْجَنَسِ وَالسَّيِّئِ. وَتَدْرِي بِأَيِّ شَيْءٍ كُنَّا نُسَمِّنُهَا؟ وَفِي أَيِّ مَكَانٍ كُنَّا نَعْلِفُهَا؟ »

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

10.2.1 Lists

Lists environments are also accepted inside the arab environment. One may either use any of the three standard list environments, viz. `itemize`, `enumerate` and `description` or use 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 can be used like so:—

```

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

```

رموز الكتاب

ب مخطوط المكتبة الأهلية بباريس ٢٨٦٠ عربي.

س مخطوط المكتبة الأهلية بباريس ٢٨٥٩ عربيّ.
م مخطوط مجلس شورى على طهران ٥٢١.

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

```

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

```

القانون في الطب
- الفن الأول في حدّ الطبّ وموضوعاته من الأمور الطبيعيّة ويشتمل على ستّة تعاليم
- التعلّيم الأول [وهو فصلان]
- الفصل الأول

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

```

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

```

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

10.3 csquotes

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

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

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

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

فَقَالَ لَهُ جُحَا: "غَرِيبٌ أَمْرُكَ يَا صَدِيقِي أَتَصَدِّقُ الْحِمَارَ وَتَكْذِبُنِي؟"

REM. Do not forget to set back the quoting style to its initial state once the Arabic environment is closed. See the last line in the code above.

10.4 Two-argument commands

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

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

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

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

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

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

10.5 quran

`arabluatex` is compatible with the `quran` package so that both can be used in conjunction with one another for typesetting the *Qur'ān*. As `quran` draws the text of the *Qur'ān* from a Unicode encoded database, its commands have to be passed as arguments to the `\txarb` command for short insertions in left-to-right paragraphs, or inserted inside the `txarab` environment for typesetting running paragraphs of *Qur'ānic* text (see above section 9 on page 41 for more details). Please note that `arabluatex` takes care of formatting the Arabic: therefore, it is recommended to load the `quran` package with the `nopar` option, after `arabluatex` itself has been loaded, like so:—

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

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

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


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

11 Future work

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

- (a) Short-term:
 - i. TEI xml support: `arabluatex` will interoperate with TEI xml through new global and local options that will output Arabic in a TEI xml compliant file in addition to the usual PDF output: see on page 4.
- (b) Medium-term:
 - i. More languages: the list of supported languages will eventually be the same as `arabtex`: see footnote 4 on page 4.
 - ii. Formulate propositions for extending the `ArabTeX` notation and the transliteration tables. Include them in `arabluatex`. See section 4.9 on page 26.

12 Implementation

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

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{arabluatex}%
3 [2018/03/31 v1.11 An ArabTeX-like interface for LuaLaTeX]
4 \RequirePackage{ifluatex}
```

`arabluatex` requires `LuaLaTeX` 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 \RequirePackage{xkeyval}
13 \DeclareOptionX{voc}{\def\al@mode{voc}}
14 \DeclareOptionX{fullvoc}{\def\al@mode{fullvoc}}
15 \DeclareOptionX{novoc}{\def\al@mode{novoc}}
```

```

16 \DeclareOptionX{trans}{\def\al@mode{trans}}
17 \ExecuteOptionsX{voc}
18 \ProcessOptionsX\relax
19 \def\al@mode@voc{voc}
20 \def\al@mode@fullvoc{fullvoc}
21 \def\al@mode@novoc{novoc}
22 \def\al@mode@trans{trans}

```

Packages that are required by arabluatex:

```

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

```

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

```

31 \newbool{al@rlmode}

```

Here begins the real work: load arabluatex.lua:

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

46 \DeclareDocumentCommand{\MkArbBreak}{m}{%
47   \luadirect{mkarbbreak(\luastringN{#1})}}
```

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

`\aemph*` The ‘starred’ version of this command always puts the stroke over its argument.

```

48 \AtBeginDocument{\ifdef{\aemph}%
49   {\RenewDocumentCommand{\aemph}{s m}{%
50     \IfBooleanTF{#1}{%
51       $\overline{\text{#2}}$}
52     {\luadirect{tex.sprint(aemph(\luastringN{#2}))}}}}
53   {\NewDocumentCommand{\aemph}{s m}{%
54     \IfBooleanTF{#1}{%
55       $\overline{\text{#2}}$}
56     {\luadirect{tex.sprint(aemph(\luastringN{#2}))}}}}}
```

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

```

57 \def\al@input@scheme{arabtex}
58 \NewDocumentCommand{\SetInputScheme}{m}{\def\al@input@scheme{#1}}
```

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

`\SetArbDflt*` As of v1.6, `arabluatex` does not apply any more the assimilation rules that are laid on item [b](#) on page [17](#); a new starred version `\SetArbDflt*` is now available to the user should he wish to apply them.

```

59 \def\al@arb@rules{dflt}
60 \NewDocumentCommand{\SetArbEasy}{s}{%
61   \IfBooleanTF{#1}
62   {\def\al@arb@rules{easynosukun}}
63   {\def\al@arb@rules{easy}}}
64 \NewDocumentCommand{\SetArbDflt}{s}{%
65   \IfBooleanTF{#1}
66   {\def\al@arb@rules{idgham}}
67   {\def\al@arb@rules{dflt}}}
```

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

```

68 \def\al@trans@font{\rmfamily}%
69 \NewDocumentCommand{\SetTranslitFont}{m}{\def\al@trans@font{#1}}

```

`\SetTranslitStyle` By default any transliterated Arabic text is printed in italics. This can be changed either globally in the preamble or at any point of the document:

```

70 \def\al@trans@style{\itshape}%
71 \NewDocumentCommand{\SetTranslitStyle}{m}{\def\al@trans@style{#1}}

```

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

```

72 \def\al@trans@convention{dmg}
73 \NewDocumentCommand{\SetTranslitConvention}{m}{\def\al@trans@convention{#1}}

```

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

```

74 \NewDocumentCommand{\al@arbup@dflt}{m}{\textsuperscript{\thinspace#1}}%
75 \NewDocumentCommand{\al@arbup}{m}{\al@arbup@dflt{#1}}
76 \NewDocumentCommand{\arbup}{m}{\al@arbup{#1}}
77 \NewDocumentCommand{\ArbUpDflt}{}{\let\al@arbup=\al@arbup@dflt}
78 \NewDocumentCommand{\NoArbUp}{}{\RenewDocumentCommand{\al@arbup}{m}{##1}}
79 \NewDocumentCommand{\SetArbUp}{m}{\RenewDocumentCommand{\al@arbup}{m}{#1}}

```

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

```

80 \NewDocumentCommand{\uc}{m}%
81 {\luairect{tex.sprint(uc(\luastringN{#1}))}}

```

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

```

82 \let\Uc\uc

```

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

```

83 \NewDocumentCommand{\prname}{m}{%
84   \bgroup\SetTranslitStyle{\relax}\arb[trans]{\uc{#1}}\egroup}

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

\txtrans \txtrans is used internally by several Lua functions to insert transliterated Arabic
text. Therefore, it is not documented.
85 \NewDocumentCommand{\txarb}{+m}{\bgroup\textdir
86   TRT\arabicfont#1\egroup}
87 \NewDocumentCommand{\txtrans}{+m}{\bgroup\textdir
88   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.
89 \NewDocumentEnvironment{txarab}{}{%
90   \par%
91   \booltrue{al@rlmode}%
92   \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.
93 \NewDocumentCommand{\arb}{0{\al@mode} +m}%
94 {\edef\@tempa{#1}%
95   \ifx\@tempa\al@mode@voc%
96     \bgroup\textdir TRT\arabicfont%
97     \luadirect{tex.sprint(processvoc(\luastringN{#2},
98       \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
99   \else%
100    \ifx\@tempa\al@mode@fullvoc%
101      \bgroup\textdir TRT\arabicfont%
102      \luadirect{tex.sprint(processfullvoc(\luastringN{#2},
103        \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
104    \else%
105      \ifx\@tempa\al@mode@novoc%
106        \bgroup\textdir TRT\arabicfont%
107        \luadirect{tex.sprint(processnovoc(\luastringN{#2},
108          \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
109      \else%
110        \ifx\@tempa\al@mode@trans%
111          \bgroup\textdir TLT\al@trans@style%
112          \luadirect{tex.sprint(processstrans(\luastringN{#2},
113            \luastring0{\al@trans@convention},
114            \luastring0{\al@arb@rules},
115            \luastring0{\al@input@scheme}))}\egroup%
116        \else%
117          \fi\fi\fi\fi}

```

`\arbmark` `\arbmark` takes one argument from a list of defined elements. This command is coded in Lua.

```
118 \NewDocumentCommand{\arbmark}{m}{%
119   \bgroup%
120   \SetInputScheme{arabtex}%
121   \luadirect{tex.sprint(processarbmarks(\luastringN{#1}))}%
122   \egroup}
```

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

```
123 \NewDocumentCommand{\newarbmark}{m m m}{%
124   \luadirect{newarbmark(\luastringN{#1}, \luastringN{#2}, \luastringN{#3})}}
```

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

```
125 \NewEnviron{arab}[1][\al@mode]%
126 {\par\edef\@tempa{#1}%
127   \ifx\@tempa\al@mode@voc%
128     \booltrue{al@rlmode}%
129     \bgroup\pdir TRT\textdir TRT\arabicfont%
130     \luadirect{tex.sprint(processvoc(\luastring0{\BODY},
131       \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
132   \else%
133     \ifx\@tempa\al@mode@fullvoc%
134       \booltrue{al@rlmode}%
135       \bgroup\pdir TRT\textdir TRT\arabicfont%
136       \luadirect{tex.sprint(processfullvoc(\luastring0{\BODY},
137         \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
138     \else%
139       \ifx\@tempa\al@mode@novoc%
140         \booltrue{al@rlmode}%
141         \bgroup\pdir TRT\textdir TRT\arabicfont%
142         \luadirect{tex.sprint(processnovoc(\luastring0{\BODY},
143           \luastring0{\al@arb@rules}, \luastring0{\al@input@scheme}))}\egroup%
144       \else%
145         \ifx\@tempa\al@mode@trans%
146           \bgroup\pdir TLT\textdir TLT\al@trans@style%
147           \luadirect{tex.sprint(processtrans(\luastring0{\BODY},
148             \luastring0{\al@trans@convention},
149             \luastring0{\al@arb@rules},
150             \luastring0{\al@input@scheme}))}\egroup%
151         \else \fi\fi\fi\fi}{\par}
```

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

```
152 \newlength{\al@bayt@width}
```

```

153 \newlength{\al@gutter@width}
154 \setlength{\al@bayt@width}{.3\textwidth}
155 \setlength{\al@gutter@width}{.15\al@bayt@width}
156 \define@key[al]{verse}{width}{\setlength{\al@bayt@width}{#1}}
157 \define@key[al]{verse}{gutter}{\setlength{\al@gutter@width}{#1}}
158 \define@key[al]{verse}{metre}{\arb{#1}}
159 \define@boolkey[al]{verse}{utf}[true]{}
160 \define@boolkey[al]{verse}{delim}[true]{}
161 \define@choicekey[al]{verse}{mode}{fullvoc, voc, novoc,
162   trans}{\def\al@mode{#1}}
163 \presetkeys[al]{verse}{metre={}, utf=false,
164   delim=false}{}

```

Then follows the environment itself:

```

165 \NewDocumentEnvironment{arabverse}{0{}}%
166 {\par\centering\noindent\bgroup\setkeys[al]{verse}[metre]{#1}%
167   \ifx\al@mode\al@mode@trans%
168   \ifal@verse@utf\setRL\else\setLR\fi%
169   \else\setRL\fi}%
170 {\hfill\setkeys[al]{verse}[width,gutter,utf,mode]{#1}\egroup}

```

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

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

```

171 \NewDocumentCommand{\arb@utf}{m}{%
172   \ifal@verse@utf\txarb{#1}\else\arb{#1}\fi}
173 \def\al@hemistich@delim{*}
174 \NewDocumentCommand{\SetHemistichDelim}{m}{\def\al@hemistich@delim{#1}}
175 \def\al@verse@stroke{\leavevmode\xleaders\hbox{\arb{--}}\hfill\kern0pt}
176 \NewDocumentCommand{\bayt}{m o m}{%
177   \ifdefined\savenotes\savenotes\else\fi%
178   \edef\al@tatweel{--}%
179   \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#1}}%
180   \IfNoValueTF{#2}{%
181     \ifal@verse@delim\makebox[\al@gutter@width][c]{\al@hemistich@delim}%
182     \else%
183     \hspace{\al@gutter@width}%
184     \fi
185   }{%
186     \edef\@tempa{#2}%
187     \ifx\@tempa\al@tatweel%
188     \ifx\al@mode\al@mode@trans%
189     \hspace{\al@gutter@width}%
190     \else%

```

```

191 \makebox[\al@gutter@width][s]{\al@verse@stroke}%
192 \fi%
193 \else%
194 \ifx\al@mode\al@mode@trans%
195 \adjustbox{width=\al@gutter@width, height=\Height}{\arb@utf{#2}}%
196 \else%
197 \makebox[\al@gutter@width][s]{\arb@utf{#2}}%
198 \fi\fi}%
199 \adjustbox{width=\al@bayt@width, height=\Height}{\arb@utf{#3}}%
200 \ifdefined\spewnotes\spewnotes\else\fi%
201 }

```

`\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.

```

202 \AtBeginDocument{%
203 \ifdefined\abjad%
204 \RenewDocumentCommand{\abjad}{m}%
205 {\luairect{tex.sprint(abjadify(#1))}}%
206 \else%
207 \NewDocumentCommand{\abjad}{m}%
208 {\luairect{tex.sprint(abjadify(#1))}}
209 \fi}

```

`\arbnul1` The `\arbnul1` command does nothing by itself. It is processed only if it is found in Arabic context so as to put back on contextual analysis in case it has been broken by other commands.

```

210 \NewDocumentCommand{\arbnul1}{m}{\relax}

```

`\abraces` `\abraces{<Arabic text>}` puts its argument between braces. This macro is written in Lua and is dependent on the current value of `tex.textdir`.

```

211 \NewDocumentCommand{\abraces}{+m}{%
212 \luairect{tex.sprint(abraces(\luastringN{#1}))}}

```

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

```

213 \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.

```

214 \DeclareDocumentCommand{\LRfootnote}{m}{\bgroup\pdir
215 TLT\textdir TLT\footnote{#1}\egroup}
216 \DeclareDocumentCommand{\RLfootnote}{m}{\bgroup\pdir
217 TRT\textdir TRT\footnote{#1}\egroup}

```


`\FixArbFtnmk` In the preamble, just below `\usepackage{arabluatex}`, `\FixArbFtnmk` may be of some help in case the footnote numbers at the bottom of the page are printed in the wrong direction. This quick fix uses and loads `scrextend` if it is not already loaded.

```

218 \NewDocumentCommand{\FixArbFtnmk}{-}{%
219   \@ifpackageloaded{scrextend}%
220   {\AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}%
221   {\RequirePackage{scrextend}
222    \AtBeginDocument{\deffootnote{2em}{1.6em}{\LR{\thefootnotemark}.\enskip}}}}

```

That is it. Say goodbye before leaving.

Patches

```

223 \NeedsTeXFormat{LaTeX2e}
224 \ProvidesPackage{arabluatex-patch}%
225 [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:

```

226 \def\list#1#2{%
227   \ifnum \@listdepth >5\relax
228     \@toodeep
229   \else
230     \global\advance\@listdepth\@ne
231   \fi
232   \rightmargin\z@
233   \listparindent\z@
234   \itemindent\z@
235   \csname @list\romannumeral\the\@listdepth\endcsname
236   \def\@itemlabel{#1}%
237   \let\makelabel\@mklab
238   \@nmbrlistfalse
239   #2\relax
240   \@trivlist
241   \parskip\parsep
242   \parindent\listparindent
243   \advance\linewidth -\rightmargin
244   \advance\linewidth -\leftmargin

```

patch begins:

```

245   \ifbool{al@rlmode}{\advance\@totalleftmargin \rightmargin}%
246   {\advance\@totalleftmargin \leftmargin}

```

patch ends.

```

247   \parshape \@ne \@totalleftmargin \linewidth
248   \ignorespaces}
249 \def\@item[#1]{%

```

```

250 \if@noparitem
251   \@donoparitem
252 \else
253   \if@inlabel
254     \indent \par
255   \fi
256   \ifhmode
257     \unskip\unskip \par
258   \fi
259   \if@newlist
260     \if@nobreak
261       \@nbitem
262     \else
263       \addpenalty\@beginparpenalty
264       \addvspace\@topsep
265       \addvspace{-\parskip}%
266     \fi
267   \else
268     \addpenalty\@itempenalty
269     \addvspace\itemsep
270   \fi
271   \global\@inlabeltrue
272 \fi
273 \everypar{%
274   \@minipagefalse
275   \global\@newlistfalse
276   \if@inlabel
277     \global\@inlabelfalse
278     {\setbox\z@\lastbox
279     \ifvoid\z@
280       \kern-\itemindent
281     \fi}%
282   \box\@labels
283   \penalty\z@
284 \fi
285 \if@nobreak
286   \@nobreakfalse
287   \clubpenalty \@M
288 \else
289   \clubpenalty \@clubpenalty
290   \everypar{}%
291 \fi}%
292 \if@noitemarg
293   \@noitemargfalse
294   \if@nmbrlist
295     \refstepcounter\@listctr
296   \fi
297 \fi

```

patch begins:

```

298 \ifbool{al@rlmode}{\sRLbox\@tempboxa{\makelabel{#1}}}{%
299 \sbox\@tempboxa{\makelabel{#1}}}%
300 \ifbool{al@rlmode}{\global\setbox\@labels\hbox dir TRT}%
301 {\global\setbox\@labels\hbox}{%

```

patch ends.

```

302 \unhbox\@labels
303 \hskip \itemindent
304 \hskip -\labelwidth
305 \hskip -\labelsep
306 \ifdim \wd\@tempboxa >\labelwidth
307 \box\@tempboxa
308 \else
309 \hbox to\labelwidth {\unhbox\@tempboxa}%
310 \fi
311 \hskip \labelsep}%
312 \ignorespaces}

```

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

```

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

```

References

- Brockelmann, Carl et al. (1935). “Die Transliteration der arabischen Schrift in ihrer Anwendung auf die Hauptliteratursprachen der islamischen Welt”. In: *Denkschrift dem 19. internationalen Orientalistenkongreß in Rom vorgelegt von der Transkriptionskommission der Deutschen Morgenländischen Gesellschaft*. In collab. with Ph. S. van Ronkel and Otto Spies. Leipzig: Deutsche Morgenländische Gesellschaft, in Kommission bei F. A. Brockhaus. URL: http://www.naher-osten.uni-muenchen.de/studium_lehre/werkzeugkasten/dmgtransliteration.pdf.
- Habash, Nizar Y. (2010). *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies 10. Toronto: Morgan & Claypool Publishers.
- Hosny, Khaled (Sept. 20, 2015). *Amiri*. URL: <http://www.amirifont.org/>.
- DIN 31 635 (July 2011). *Information and Documentation - Romanization of the Arabic Alphabet for Arabic, Ottoman-Turkish, Persian, Kurdish, Urdu and Pushto*. URL: <http://www.din.de>.
- Lagally, Klaus (Nov. 3, 2004). *ArabTeX. Typesetting Arabic and Hebrew*. User Manual Version 4.00. Version 4.00. URL: <http://mirrors.ctan.org/language/arabic/arabtex/doc/html/arabtex.htm>.
- Lane, Edward William (1863–1893). *An Arabic-English lexicon*. 8 vols. London – Edinburgh: Williams and Norgate.
- Wright, W. LL.D (1896). *A Grammar of the Arabic Language*. Rev. by W. Robertson Smith and M. J. de Goeje. With a forew. by Pierre Cachia. 3rd ed. 2 vols. Beirut: Librairie du Liban.

Change History

v1.0.	General: Initial release	1	\bayt: New macro \bayt for typesetting each verse inside the arabverse environment	55
v1.0.1.	General: Minor update of the documentation	1	\SetArbDflt*: This starred version applies the assimilation rules in addition to what \SetArbDflt already does.	51
v1.1.	\abjad: New and more flexible \abjad command.	56	\SetHemistichDelim: New \SetHemistichDelim command for changing the default delimiter between hemistichs	55
v1.2.	\SetArbEasy: New \SetArbEasy/\SetArbDflt for ‘modern’ or ‘classic’ Arabic styles.	51	v1.7.	\arbnnull: New \arbnnull command for putting back on any contextual analysis rule broken by other commands.
v1.3.	\arbut: <i>ʾrāb</i> is now written as superscript text in dmg mode by default.	52	v1.8.	General: arabica transliteration standard is now supported . . .
v1.4.	\SetInputScheme: \SetInputScheme can be used to process other input schemes such as ‘Buckwalter’	51	v1.8.5.	General: Six additional Persian characters are now available . .
	\SetTranslitFont: For selecting a specific font for transliterated texts	51	v1.9.	\MkArbBreak: New \MkArbBreak command for inserting user-defined macros in Arabic environments
v1.4.3.	\abraces: New \abraces command which expresses its argument between braces.	56	v1.9.2.	\aemph*: Starred version which always puts the stroke over its argument
v1.4.4.	\SetArbEasy*: this starred version discards the <i>sukūn</i> in addition to what is already discarded by \SetArbEasy.	51	v1.10.	General: \uc supersedes \cap
v1.5.	General: Compatibility with the quran package	48	\prname: New command for typesetting Arabic proper names in transliteration	52
	Environments may be nested inside the arab environment . .	44	v1.11.	\arbmark: New command for inserting additional marks in Arabic environments
	txarab: New txarab environment for typesetting running paragraphs in Unicode Arabic .	53	\newarbmark: Allows defining additional sets of Arabic marks	54
v1.6.	arabverse: New environment arabverse for typesetting Arabic poetry	54		

Index

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

Symbols		
\@M	287	\adjustbox 179, 195, 199
\@beginparpenalty .	263	\advance 230, 243, 244, 245, 246
\@clubpenalty	289	\aemph 26, <u>48</u>
\@donoparitem	251	\aemph* 27, <u>48</u>
\@ifpackageloaded .	219	\al@arb@rules 59, 62, 63, 66, 67, 98, 103, 108, 114, 131, 137, 143, 149
\@inlabelfalse	277	\al@arbup 75, 76, 77, 78, 79
\@inlabeltrue	271	\al@arbup@dflt 74, 75, 77
\@item	249	\al@bayt@width 152, 154, 155, 156, 179, 199
\@itemlabel	236	\al@gutter@width 153, 155, 157, 181, 183, 189, 191, 195, 197
\@itempenalty	268	\al@hemistich@delim 173, 174, 181
\@labels	282, 300, 301, 302	\al@input@scheme 57, 58, 98, 103, 108, 115, 131, 137, 143, 150
\@listctr	295	\al@mode 13, 14, 15, 16, 93, 125, 162, 167, 188, 194
\@listdepth 227, 230, 235		\al@mode@fullvoc 20, 100, 133
\@minipagefalse	274	\al@mode@novoc 21, 105, 139
\@mklab	237	\al@mode@trans 22, 110, 145, 167, 188, 194
\@nbitem	261	\al@mode@voc 19, 95, 127
\@ne	230, 247	\al@tatweel 178, 187
\@newlistfalse	275	\al@trans@convention 72, 73, 113, 148
\@nmbrlistfalse	238	\al@trans@font 68, 69, 88
\@nbreakfalse	286	\al@trans@style 70, 71, 111, 146
\@noitemargfalse	293	\al@verse@stroke 175, 191
\@tempa 94, 95, 100, 105, 110, 126, 127, 133, 139, 145, 186, 187		<i>Amiri</i> 5
\@tempboxa 298, 299, 306, 307, 309		amiri (package) 5
\@toodeep	228	arab (environment) 8, <u>125</u>
\@topsep	264	<i>ArabTeX</i> 3, 21
\@totalleftmargin 245, 246, 247		<i>Arabic-English Lexicon</i> , <i>An</i> 20
\@trivlist	240	\arabicfont 34, 35, 37, 86, 92, 96, 101, 106, 129, 135, 141
_	35	arabluatex (package) 2, 4–8, 11–21, 26, 27, 29, 31, 34–43, 46, 48–52
A		arabtex (package) 3, 4, 17, 21, 49
\abjad	25, <u>202</u>	arabulatex (package) 17
\abraces	31, <u>211</u>	arabverse (environ- ment) 27, <u>152</u>
\addpenalty	263, 268	arabxetex (package) 4, 17
\addvspace 264, 265, 269		\arb 7, 84, <u>93</u> , 158, 172, 175
		\arb@utf 171, 179, 195, 197, 199
		\arbmark 32, <u>118</u>
		\arbnnull 24, <u>210</u>
		\arbup 36, <u>74</u>
		\ArbUpDflt 36, <u>74</u>
		B
		babel (package) 46
		babel-french (package) 46
		\bayt 28, <u>171</u>
		bidi (package) 59
		\BODY 130, 136, 142, 147
		\boolfalse 39
		\booltrue 38, 91, 128, 134, 140
		\box 282, 307

Brockelmann, Carl ...	\Height .. 179, 195, 199	\labelwidth 304, 306, 309
..... 10, 33, 36	\hfill 170, 175	Lagally, Klaus 3, 21
C	Hosny, Khaled 5	Lane, Edward William . 20
\centering 166	\hskip 303, 304, 305, 311	\lastbox 278
\clubpenalty .. 287, 289	\hspace 183, 189	\leavevmode 175
\color@endgroup ... 314	I	\leftmargin .. 244, 246
\color@setgroup ... 314	\if@inlabel .. 253, 276	lineno (package) 30
csquotes (package) ... 47	\if@newlist 259	\linewidth 243, 244, 247
D	\if@nmbulist 294	\list 226
\DeclareOptionX ...	\if@nobreak .. 260, 285	\listparindent 233, 242
.... 13, 14, 15, 16	\if@noitemarg 292	\long 313
\deffootnote .. 220, 222	\if@noparitem 250	\LR ... 40, 42, 220, 222
\define@boolkey 159, 160	\ifal@verse@delim . 181	\LRfootnote ... 42, 214
\define@choicekey . 161	\ifal@verse@utf 168, 172	\LRmarginpar ... 43, 213
\define@key 156, 157, 158	\ifbool .. 245, 298, 300	luabidi (package) ... 4, 50
delim (option) 27	\IfBooleanTF 47, 52, 56, 81,	\luadirect .. 32, 33,
E 50, 54, 61, 65	47, 52, 56, 81,
environments:	\ifdim 306	97, 102, 107,
arab 8, 125	\ifhmode 256	112, 121, 124,
arabverse .. 27, 152	\ifluatex 5	130, 136, 142,
txarab 41, 89	\IfNoValueTF 180	147, 205, 208, 212
\everypar 273, 290	\ifnum 227	luaotfload (package) ... 34
\ExecuteOptionsX ... 17	\ifvoid 279	\luastringN 47, 52, 56, 81,
F	\ignorespaces . 248, 312	97, 102, 107,
\FixArbFtnmk ... 42, 218	\indent 254	112, 121, 124, 212
fontspec (package) ...	<i>Information and Docu-</i>	\luastringO 98, 103, 108,
..... 5, 34, 52	<i>mentation - Ro-</i>	113, 114, 115,
\footnote 215, 217	<i>manization of</i>	130, 131, 136,
footnote (package) 29	<i>the Arabic Al-</i>	137, 142, 143,
footnotehyper (package) 29	<i>phabet for Ara-</i>	147, 148, 149, 150
fullvoc (option) 5, 9	<i>bic, Ottoman-</i>	
G	<i>Turkish, Per-</i>	M
\global ... 230, 271,	<i>sian, Kur-</i>	\makebox . 181, 191, 197
275, 277, 300, 301	<i>dish, Urdu and</i>	\makelabel 237, 298, 299
<i>Grammar of the Ara-</i>	<i>Pushto</i> 10	\marginpar 213
<i>bic Language,</i>	<i>Introduction to Arabic</i>	metre (option) 27
<i>A</i> 5–7,	<i>Natural Lan-</i>	\MkArbBreak 43, 46
9, 13, 15–17,	<i>guage Process-</i>	mode (option) 27
19, 25, 26, 35	<i>ing</i> 39	
gutter (option) 27	\itemindent 234, 280, 303	N
H	\itemsep 269	nameauth (package) ... 36
Habash, Nizar Y. 39	\itshape 70	\newarbmark ... 32, 123
\hbox 175,	K	\newbool 31
300, 301, 309, 313	\kern 175, 280	\NewDocumentEnvironment
	KOMA-script (package) 43 89, 165
	L	\NewEnviron 125
	\labelsep 305, 311	\newfontfamily 37
		\newlength ... 152, 153

<code>\NoArbUp</code>	36, 74	<code>\RequirePackage</code>	101, 106, 111, 129, 135, 141, 146, 213, 215, 217
<code>\noindent</code>	166	4, 12,
<code>novoc</code> (option)	5, 9	23, 24, 25, 26, 27, 28, 29, 30, 221	
O			
options:			
<code>delim</code>	27	<code>\rightmargin</code>	232, 243 , 245
<code>fullvoc</code>	5, 9	<code>\RL</code>	42, 43
<code>gutter</code>	27	<code>\RLfootnote</code> . . .	42, 214
<code>metre</code>	27	<code>\rmfamily</code>	41, 42, 44, 45, 68
<code>mode</code>	27	<code>\romannumeral</code>	235
<code>novoc</code>	5, 9	S	
<code>trans</code>	5, 9	<code>\savenotes</code>	177
<code>utf</code>	27	<code>\sbox</code>	299
<code>voc</code>	5, 8	<code>scrextend</code> (package) .	43, 57
<code>width</code>	27	<code>\SetArbDflt</code>	6, 59
<code>\overline</code>	51, 55	<code>\SetArbDflt*</code>	6, 59
P			
<code>\par</code>	90, 92, 126, 151, 166, 254, 257	<code>\SetArbEasy</code>	6, 59
<code>paralist</code> (package) .	45, 46	<code>\SetArbEasy*</code>	6, 59
<code>\pardir</code>	38, 39, 92, 129, 135, 141, 146, 214, 216	<code>\SetArbUp</code>	36, 74
<code>\parindent</code>	242	<code>\setbox</code>	278, 300, 301, 313
<code>\parsep</code>	241	<code>\SetHemistichDelim</code>	27, 171 , 174
<code>\parshape</code>	247	<code>\SetInputScheme</code>	38, 57 , 120
<code>\parskip</code>	241, 265	<code>\setkeys</code>	166, 170
<code>\penalty</code>	283	<code>\setlength</code>	154, 155, 156, 157
<code>polyglossia</code> (package) .	4, 50, 56	<code>\setLR</code>	39, 43, 168
<code>\presetkeys</code>	163	<code>\setRL</code> .	38, 43, 168, 169
<code>\prname</code>	35, 83	<code>\SetTranslitConvention</code>	34, 72
<code>\ProcessOptionsX</code> . . .	18	<code>\SetTranslitFont</code> .	34, 68
<code>\ProvidesPackage</code> .	2, 224	<code>\SetTranslitStyle</code>	34, 70 , 84
Q			
<code>quoting</code> (package)	44	<code>\spewnotes</code>	200
<code>quran</code> (package)	48	<code>\sRLbox</code>	298, 313
R			
<code>\refstepcounter</code> . . .	295	T	
<code>reledmac</code> (package) . . .	48	<code>\text</code>	51, 55
		<code>\textdir</code>	38, 39, 41, 42, 44, 45, 85, 87, 92, 96,
		<code>\RequirePackage</code>	101, 106, 111, 129, 135, 141, 146, 213, 215, 217
		<code>\rightmargin</code>	232, 243 , 245
		<code>\RL</code>	42, 43
		<code>\RLfootnote</code> . . .	42, 214
		<code>\rmfamily</code>	41, 42, 44, 45, 68
		<code>\romannumeral</code>	235
		S	
		<code>\savenotes</code>	177
		<code>\sbox</code>	299
		<code>scrextend</code> (package) .	43, 57
		<code>\SetArbDflt</code>	6, 59
		<code>\SetArbDflt*</code>	6, 59
		<code>\SetArbEasy</code>	6, 59
		<code>\SetArbEasy*</code>	6, 59
		<code>\SetArbUp</code>	36, 74
		<code>\setbox</code>	278, 300, 301, 313
		<code>\SetHemistichDelim</code>	27, 171 , 174
		<code>\SetInputScheme</code>	38, 57 , 120
		<code>\setkeys</code>	166, 170
		<code>\setlength</code>	154, 155, 156, 157
		<code>\setLR</code>	39, 43, 168
		<code>\setRL</code> .	38, 43, 168, 169
		<code>\SetTranslitConvention</code>	34, 72
		<code>\SetTranslitFont</code> .	34, 68
		<code>\SetTranslitStyle</code>	34, 70 , 84
		<code>\spewnotes</code>	200
		<code>\sRLbox</code>	298, 313
		T	
		<code>\text</code>	51, 55
		<code>\textdir</code>	38, 39, 41, 42, 44, 45, 85, 87, 92, 96,
		<code>\RequirePackage</code>	101, 106, 111, 129, 135, 141, 146, 213, 215, 217
		<code>\rightmargin</code>	232, 243 , 245
		<code>\RL</code>	42, 43
		<code>\RLfootnote</code> . . .	42, 214
		<code>\rmfamily</code>	41, 42, 44, 45, 68
		<code>\romannumeral</code>	235
		S	
		<code>\savenotes</code>	177
		<code>\sbox</code>	299
		<code>scrextend</code> (package) .	43, 57
		<code>\SetArbDflt</code>	6, 59
		<code>\SetArbDflt*</code>	6, 59
		<code>\SetArbEasy</code>	6, 59
		<code>\SetArbEasy*</code>	6, 59
		<code>\SetArbUp</code>	36, 74
		<code>\setbox</code>	278, 300, 301, 313
		<code>\SetHemistichDelim</code>	27, 171 , 174
		<code>\SetInputScheme</code>	38, 57 , 120
		<code>\setkeys</code>	166, 170
		<code>\setlength</code>	154, 155, 156, 157
		<code>\setLR</code>	39, 43, 168
		<code>\setRL</code> .	38, 43, 168, 169
		<code>\SetTranslitConvention</code>	34, 72
		<code>\SetTranslitFont</code> .	34, 68
		<code>\SetTranslitStyle</code>	34, 70 , 84
		<code>\spewnotes</code>	200
		<code>\sRLbox</code>	298, 313
		T	
		<code>\text</code>	51, 55
		<code>\textdir</code>	38, 39, 41, 42, 44, 45, 85, 87, 92, 96,
		<code>\RequirePackage</code>	101, 106, 111, 129, 135, 141, 146, 213, 215, 217
		<code>\rightmargin</code>	232, 243 , 245
		<code>\RL</code>	42, 43
		<code>\RLfootnote</code> . . .	42, 214
		<code>\rmfamily</code>	41, 42, 44, 45, 68
		<code>\romannumeral</code>	235
		S	
		<code>\savenotes</code>	177
		<code>\sbox</code>	299
		<code>scrextend</code> (package) .	43, 57
		<code>\SetArbDflt</code>	6, 59
		<code>\SetArbDflt*</code>	6, 59
		<code>\SetArbEasy</code>	6, 59
		<code>\SetArbEasy*</code>	6, 59
		<code>\SetArbUp</code>	36, 74
		<code>\setbox</code>	278, 300, 301, 313
		<code>\SetHemistichDelim</code>	27, 171 , 174
		<code>\SetInputScheme</code>	38, 57 , 120
		<code>\setkeys</code>	166, 170
		<code>\setlength</code>	154, 155, 156, 157
		<code>\setLR</code>	39, 43, 168
		<code>\setRL</code> .	38, 43, 168, 169
		<code>\SetTranslitConvention</code>	34, 72
		<code>\SetTranslitFont</code> .	34, 68
		<code>\SetTranslitStyle</code>	34, 70 , 84
		<code>\spewnotes</code>	200
		<code>\sRLbox</code>	298, 313
		T	
		<code>\text</code>	51, 55
		<code>\textdir</code>	38, 39, 41, 42, 44, 45, 85, 87, 92, 96,
		<code>\RequirePackage</code>	101, 106, 111, 129, 135, 141, 146, 213, 215, 217
		<code>\rightmargin</code>	232, 243 , 245
		<code>\RL</code>	42, 43
		<code>\RLfootnote</code> . . .	42, 214
		<code>\rmfamily</code>	41, 42, 44, 45, 68
		<code>\romannumeral</code>	235
		S	
		<code>\savenotes</code>	177
		<code>\sbox</code>	299
		<code>scrextend</code> (package) .	43, 57
		<code>\SetArbDflt</code>	6, 59
		<code>\SetArbDflt*</code>	6, 59
		<code>\SetArbEasy</code>	6, 59
		<code>\SetArbEasy*</code>	6, 59
		<code>\SetArbUp</code>	36, 74
		<code>\setbox</code>	278, 300, 301, 313
		<code>\SetHemistichDelim</code>	27, 171 , 174
		<code>\SetInputScheme</code>	38, 57 , 120
		<code>\setkeys</code>	166, 170
		<code>\setlength</code>	154, 155, 156, 157
		<code>\setLR</code>	39, 43, 168
		<code>\setRL</code> .	38, 43, 168, 169
		<code>\SetTranslitConvention</code>	34, 72
		<code>\SetTranslitFont</code> .	34, 68
		<code>\SetTranslitStyle</code>	34, 70 , 84
		<code>\spewnotes</code>	200
		<code>\sRLbox</code>	298, 313
		T	
		<code>\text</code>	51, 55
		<code>\textdir</code>	38, 39, 41, 42, 44, 45, 85, 87, 92, 96,
		<code>\RequirePackage</code>	101, 106, 111, 129, 135, 141, 146, 213, 215, 217
		<code>\rightmargin</code>	232, 243 , 245
		<code>\RL</code>	42, 43
		<code>\RLfootnote</code> . . .	42, 214
		<code>\rmfamily</code>	41, 42, 44, 45, 68
		<code>\romannumeral</code>	235
		S	
		<code>\savenotes</code>	177
		<code>\sbox</code>	299
		<code>scrextend</code> (package) .	43, 57
		<code>\SetArbDflt</code>	6, 59
		<code>\SetArbDflt*</code>	6, 59
		<code>\SetArbEasy</code>	6, 59
		<code>\SetArbEasy*</code>	6, 59
		<code>\SetArbUp</code>	36, 74
		<code>\setbox</code>	278, 300, 301, 313
		<code>\SetHemistichDelim</code>	27, 171 , 174
		<code>\SetInputScheme</code>	38, 57 , 120
		<code>\setkeys</code>	166, 170
		<code>\setlength</code>	154, 155, 156, 157
		<code>\setLR</code>	39, 43, 168
		<code>\setRL</code> .	38, 43, 168, 169
		<code>\SetTranslitConvention</code>	34, 72
		<code>\SetTranslitFont</code> .	34, 68
		<code>\SetTranslitStyle</code>	34, 70 , 84
		<code>\spewnotes</code>	200
		<code>\sRLbox</code>	298, 313
		T	
		<code>\text</code>	51, 55
		<code>\textdir</code>	38, 39, 41, 42, 44, 45, 85, 87, 92, 96,
		<code>\RequirePackage</code>	101, 106, 111, 129, 135, 141, 146, 213, 215, 217
		<code>\rightmargin</code>	232, 243 , 245
		<code>\RL</code>	42, 43
		<code>\RLfootnote</code> . . .	42, 214
		<code>\rmfamily</code>	41, 42, 44, 45, 68
		<code>\romannumeral</code>	235
		S	
		<code>\savenotes</code>	177
		<code>\sbox</code>	299
		<code>scrextend</code> (package) .	43, 57
		<code>\SetArbDflt</code>	6, 59
		<code>\SetArbDflt*</code>	6, 59
		<code>\SetArbEasy</code>	6, 59
		<code>\SetArbEasy*</code>	6, 59
		<code>\SetArbUp</code>	36, 74
		<code>\setbox</code>	278, 300, 301, 313
		<code>\SetHemistichDelim</code>	27, 171 , 174
		<code>\SetInputScheme</code>	38, 57 , 120
		<code>\setkeys</code>	166, 170
		<code>\setlength</code>	154, 155, 156, 157
		<code>\setLR</code>	39, 43, 168
		<code>\setRL</code> .	38, 43, 168, 169
		<code>\SetTranslitConvention</code>	34, 72
		<code>\SetTranslitFont</code> .	34, 68
		<code>\SetTranslitStyle</code>	34, 70 , 84
		<code>\spewnotes</code>	200
		<code>\sRLbox</code>	298, 313
		T	
		<code>\text</code>	51, 55
		<code>\textdir</code>	38, 39, 41, 42, 44, 45, 85, 87, 92, 96,
		<code>\RequirePackage</code>	101, 106, 111, 129, 135, 141, 146, 213, 215, 217
		<code>\rightmargin</code>	232, 243 , 245
		<code>\RL</code>	42, 43
		<code>\RLfootnote</code> . . .	42, 214
		<code>\rmfamily</code>	41, 42, 44, 45, 68
		<code>\romannumeral</code>	235
		S	
		<code>\savenotes</code>	177
		<code>\sbox</code>	299
		<code>scrextend</code> (package) .	43, 57
		<code>\SetArbDflt</code>	6, 59
		<code>\SetArbDflt*</code>	6, 59
		<code>\SetArbEasy</code>	6, 59
		<code>\SetArbEasy*</code>	6, 59
		<code>\SetArbUp</code>	36, 74
		<code>\setbox</code>	278, 300, 301, 313
		<code>\SetHemistichDelim</code>	27, 171 , 174
		<code>\SetInputScheme</code>	38, 57 , 120
		<code>\setkeys</code>	166, 170
		<code>\setlength</code>	154, 155, 156, 157
		<code>\setLR</code>	39