The ArabTEX package

Klaus Lagally

20.11.1993

Contents

1 ArabTEX Version 3 (20.11.1993)

The introduction below is slightly out of date but may be used as a first start.

2 ArabTEX Version 2 (05.11.1992)

2.1 What is ArabTEX?

ArabTEX is a package extending the capabilities of TEX/LATEX to generate the arabic writing from an ASCII transliteration for texts in several languages using the arabic script.

It consists of a TEX macro package and an arabic font in several sizes, presently only available in the Naskhi style. ArabTEX will run with Plain TEX and also with LATEX; other additions to TEX have not been tried.

ArabTEX is primarily intended for generating the arabic writing, but the scientific transcription can be also easily generated. For other languages using the arabic script limited support is available.

2.2 Installing ArabTEX:

The installation procedure is system dependent. You have to install the "nash14" font with its "*.pk" and "*.tfm" files on the font search path of your TEX system, and the "*.sty" files and "arabtex.tex" on the source search path of your system. Possibly you will have to rename the "*.pk" files according to local conventions, and as a last resort you can try to recreate the font from the "*.mf" METAFONT sources. Additional fonts if available are installed analogously.

2.3 Activating ArabTeX:

With Plain TEX, load the ArabTeX macros by "\input arabtex". With LATEX, include the option "arabtex" in the document header. In both cases several additional files will be loaded automatically.

ArabTEX defines several additional commands as indicated below, and also a large number of internal commands which could lead to storage overflow in a small TeX implementation. All internal commands contain an "at" sign <@> in their names and thus should not interfere with any user defined commands (but possibly with TEX extensions we do not know about).

With Plain TEX, the arabic font is only available at the normal 14 point size which ought to cooperate well with the "cm" fonts at 10 points. For other sizes, change the "\magnification" or define additional font identifiers yourself. To change the default, inspect "arabtex.tex" and redefine the "\pnash" command accordingly. With LATEX, the size changing commands will also operate on the arabic font.

2.4 Input to ArabTeX:

After activating ArabTEX, your modified TEX/LATEX system will recognize the following items:

- normal TEX/LATEX text and commands,
- short arabic quotations bracketed by `<` and `>`; these must fit on one line of output, and you have to select one of the Arabic writing styles, e.g `\setarab`, before using this feature. A quotation may also be started with `\<`.
- longer arabic texts bracketed by `\begin{arabtext}` and `\end{arabtext}`, called "Arabic Environments" in the sequel.

An Arabic Environment consists of one or several paragraphs separated by blank lines or `\par` commands. Every paragraph and every arabic quotation is a sequence of the following kinds of items, separated by blank spaces or newlines:

- isolated (legal) special characters, interpreted as the corresponding arabic special character;
- "numbers", character sequences starting with a digit. A "number" will be translated in the normal writing sequence from left to right even if it contains letters and/or special characters;
- "arabic quotes", coded as two left quotes or two right quotes each;
- "words", character sequences starting with a letter or special character followed by a letter. The (coded) characters of a word will in the output be arranged from right to left.
- TEX/LATEX control sequences WITHOUT parameters. These will be executed immediately.
- ArabTEX control sequences with or without parameters. These will be executed immediately.
- a sequence of items enclosed in curly braces `{` and `}`. The output from the constituents will be arranged from right to left and must fit on one output line. As far as TEX is concerned, this is NO group. This feature may not be nested.

Output from all items will be arranged from right to left, lines will be broken as necessary.

Inside an Arabic Environment, but NOT in an arabic quotation, you may also have:

- short mathematical insertions, bracketed by SINGLE `$` signs. They must fit on one output line and are processed as usual;

- short non-arabic text quotations, bracketed by `<` and `>`. These must fit on one output line and introduce a new level of grouping, so if they contain any TEX/LATEX assignments the effects of these will be local.

Control sequences in an Arabic Environment may be of the following kinds:

- ArabTEX option changing commands. These may also be used outside an Arabic Environment and generally have a global effect;
- `\\` for a new line;
- `\par` (or a blank line) for a new paragraph, `\noindent` for a new paragraph without indentation (NOT in arabic quotations);
- `\emphasize` ⟨*item*⟩ will put a bar over the next ⟨*item*⟩;
- `\setnash`, `\setnashbf`, `\setnastaliq` font selection commands, see below;
- size changing LATEX commands like `\large` etc., only if you use LATEX;
- most other TEX/LATEX commands make no sense in an Arabic Environment.
- you MUST NOT nest another LATEX environment inside an Arabic Environment (except possibly display math which we did not test, and might work);
- if you really need to use a control sequence with parameters, define a new TEX macro or enclose the whole construct in curly braces `{` and `}`.

2.5 Font selection:

For space economy, only the Naskh font is available by default. With LATEX, additional fonts can be loaded by the document style options "`nashbf`" and/or "`nastaliq`" (when available). Users of Plain TEX can load and define suitable fonts themselves.

The following font selection commands are available:

- `\setnash` (default) selects the Naskh font.
- `\setnashbf` selects a bold-face version of Naskh.
- `\setnastaliq` selects the Nastaliq font.

If a font is not available or has not been loaded, the corresponding command will select the default font.

With LATEX, the size changing commands will also operate on the additional fonts.

2.6 Input coding:

The ASCII input notation for arabic text is modelled closely after the transliteration standards ISO/R 233 and DIN 31 635. As these standards do not guarantee unique re-transliteration and are also not ASCII compatible, some modifications were necessary. These follow the general rules:

- if the transliteration uses a single letter, code that letter;
- if the transliteration uses a letter with a diacritical mark, put a special character similar to the diacritical mark BEFORE the letter.

2.6.1 Additional characters generally available:

`b`	bah	`d`	dal	`.s`	ssad	`f`	fah	`h`	hah	`'`	hamza
`t`	tah	`_d`	dhal	`.d`	ddad	`q`	qaf	`w`	waw	`N`	tanween
`_t`	thah	`r`	rah	`.t`	ttah	`k`	kaf	`y`	yah	`Y`	alif maqsoura
`^g`	geem	`z`	zay	`.z`	tthah	`l`	lam	`g`	gaf	`_A`	alif maqsoura
`.h`	hhah	`s`	seen	`‘`	‘ain	`m`	meem	`p`	pah	`T`	tah marbouta
`_h`	khah	`^s`	sheen	`.g`	ghain	`n`	noon	`v`	vah	`W`	waw (see below)

2.6.2 Standard arabic and persian characters:

`c`	hhah with hamza
`^c`	gim with three dots (below)
`,c`	khah with three dots (above)
`^z`	zay with three dots (above)
`~n`	kaf with three dots (Ottoman)
`~l`	law with a bow accent (Kurdish)
`~r`	rah with two bows (Kurdish)

See also “Urdu” and “Pashto” below.

2.6.3 Additional coding rules:

- For long vowels, use capital letters `A`, `I`, `U`, or `_a`, `_i`, `_u`.
- As the transliteration is ambiguous, use `T` for ⟨tah marbouta⟩, `N` for ⟨tanween⟩, `Y` or `_A` for ⟨alif maqsoura⟩.
- Short vowels `a`, `i`, `u` need not generally be written except in the following cases:
  - at the beginning of a word where they generate ⟨alif⟩,
  - adjacent to ⟨hamza⟩ where they will influence the carrier,
  - when the transcription is wanted,
  - in `\fullvocalize` mode.

- ⟨hamza⟩ is denoted by a single RIGHT quote; its carrier will be determined from the context according to the rules for writing arabic words. If that is not wanted, "quote" it (see below).
- ⟨'ain⟩ is a single LEFT quote, don't confuse it with ⟨hamza⟩!
- ⟨madda⟩ is generated by a right quote (⟨hamza⟩) before `A`: `'A`.
- The "invisible letter" `|` may be inserted in order to break unwanted ligatures and to influence the ⟨hamza⟩ writing. It will not show in the arabic output or in the transcription.
- ⟨tashdid⟩ is indicated by doubling the appropriate letter.
- The article is always written `al-` (with hyphen!).
- Hyphens `-` may be used freely, and generally do not change the writing, but will show up in the transcription. At the beginning and the end of a word they enforce the use of the connection form of the adjacent letter (if it exists), like e.g. in the date `1400 h-`.
- A double hyphen `--` between two otherwise joining letters will break any ligature and will insert a horizontal stroke (⟨tatweel⟩, ⟨kashida⟩) without appearing in the transcription. It may be used repeatedly.

2.7 Quoting:

A double quote `"` will modify the meaning of the following character as follows:

- if a short vowel follows, the appropriate diacritical mark ⟨fatha⟩, ⟨kasra⟩, ⟨damma⟩ will be put on the preceding character even if the vocalization is off otherwise. If `N` follows the short vowel, the appropriate form of ⟨tanween⟩ will be generated instead. At the beginning of a word, ⟨alif⟩ is assumed as the first character. If the previous word ended with a vowel, ⟨wasla⟩ is generated instead of the vowel indicator.
- if the following character is a single right quote, a ⟨hamza⟩ mark will be put on the preceding character even if in conflict with the ⟨hamza⟩ rules.
- if the following character is the "invisible letter" `|`, the connection between the adjacent letters will be broken and a small space inserted.
- otherwise: a ⟨sukun⟩ will be put on the preceding character. The following character will be processed again.

The double quote will not show up in the transcription.

2.8 Ligatures:

There is no way to explicitly indicate ligatures as a large number of them are generated automatically. Any unwanted ligature can be suppressed by interposing the invisible letter | between the two letters otherwise combined into a ligature. After "`\ligsfalse`" ligatures in the middle of a word will not normally be produced; for some texts this looks better. You can return to the normal strategy by "`\ligstrue`".

2.9 Vocalization:

There are three modes of rendering short vowels:

`\fullvocalize`:
- every short vowel will generate the corresponding diacritic mark ⟨fatha⟩, ⟨kasra⟩, ⟨damma⟩.
- If `N` follows a short vowel, the corresponding form of ⟨tanween⟩ is generated instead.
- `_a` will produce a ⟨qur'an alif⟩ accent instead of an explicit ⟨alif⟩ character which is coded `A`.
- if a long vowel follows a consonant, the corresponding short vowel is implied. The long vowel itself carries no diacritical mark.
- if no vowel is given after a consonant, ⟨sukun⟩ will be generated except if a double "sun letter" follows ⟨lam⟩.
- ⟨alif⟩ at the beginning of a word carries ⟨wasla⟩ instead of the vowel indicator if the preceding word ended with a vowel.

`\vocalize`: as above, but ⟨sukun⟩ and ⟨wasla⟩ will not be generated except if explicitly indicated by "quoting".

`\novocalize`: no diacritics will be generated except if explicitly asked for by "quoting".

In all modes, a double consonant will generate ⟨tashdid⟩, and `'A` always generates ⟨madda⟩ on ⟨alif⟩. After `aN` the silent ⟨alif⟩ character is generated if necessary. The silent ⟨alif⟩ may also be explicitly indicated by `aNa` or `aNA`, or coded literally as `A` in `\novocalize` mode. If a silent ⟨alif maqsoura⟩ is wanted instead, write `aNY`, `aN_A`, `Y` or `_A`. A silent ⟨alif⟩ after ⟨waw⟩ is indicated by `Ua`, `UA` or `Wa`, `WA` (with a capital `W`!).

2.10 Transcription:

In addition to the arabic writing, the standard scientific transcription may also be obtained from a fully vocalized input text. This is indicated by "`\transtrue`" and may be switched off again by "`\transfalse`". If ONLY the transcription is wanted, you can deactivate the arabic writing by "`\arabfalse`"; it can be reactivated by "`\arabtrue`". If both modes are active their output will be interleaved line by line.

The transcription mode assumes that the input text is in the Arabic language and has been coded according to the rules given above. For words from other languages the transcription might be in error. For Arabic text, the following special cases are handled:

- after the article, a double consonant will be assimilated;
- an initial vowel will be omitted if the preceding word ended with a vowel. If that is not wanted start with ⟨hamza⟩.
- a silent ⟨alif⟩ or ⟨alif maqsoura⟩ after `N` (⟨tanwin⟩) and `U` is omitted in the transcription. The same happens after ⟨waw⟩ if it is written `W`.

For space economy, the transcription module is NOT loaded by default. If you want to use it, add the style option "`atrans`" with LaTeX; and with Plain TeX, say "`\input atrans.sty`".

2.11 Support for other languages:

ArabTeX is primarily intended for writing texts in classical and modern Arabic, but it also provides limited support for several other languages that are customarily written in the arabic alphabet. The vocalization and the transcription cannot generally be expected to be correct, but might work by accident.

In order to switch to the conventions for one of these languages, say "`\setfarsi`", "`\seturdu`", "`\setpashto`", "`\setmaghribi`"; "`\setarab`" is the default and can also be used to switch back to the arabic conventions.

2.11.1 Farsi, Dari:

All characters needed for writing Farsi are available by default. The short vowels `e` and `o` are mapped to ⟨i⟩ and ⟨u⟩, the long vowels `E` and `O` to ⟨I⟩ and ⟨U⟩.

The ⟨izafet⟩ connection may be written literally, which may look awkward in the case of `h"'`, or always as `-i` (with hyphen); then the correct spelling will be determined from the context. Likewise the ⟨yah-i-wahdat⟩ can always be written `-I`.

The final ⟨yah⟩ carries no dots.

Farsi uses the Nasta'liq font if available.

2.11.2 Ottoman:

see Farsi.

2.11.3 Kurdish:

see Farsi.

2.11.4 Urdu:

The additional characters in Urdu are coded as follows:

`h` always denotes the "two-eyed ⟨hah⟩"
`,h` the "wavy" ⟨hah⟩ letter
`,t` ⟨tah⟩ with a small ⟨ttah⟩ accent
`,d` ⟨dal⟩ with a small ⟨ttah⟩ accent
`,r` ⟨rah⟩ with a small ⟨ttah⟩ accent
`.n` ⟨noon⟩ without a dot (modifies a preceding vowel)
`E` ⟨yah bari⟩ in the final position, otherwise mapped to ⟨yah⟩
`O` mapped to ⟨U⟩

The short vowels `e` and `o` are mapped to ⟨a⟩ and ⟨u⟩.

Note: Some of the given codings also occur in Pashto but with a different meaning, see below.

Urdu uses the Nasta'liq font if available.

2.11.5 Pashto:

The additional characters for Pashto are coded as follows:

`,t` ⟨tah⟩ with a small loop
`,d` ⟨dal⟩ with a small loop
`,r` ⟨rah⟩ with a small loop
`.n` ⟨noon⟩ with a small loop
`g` ⟨gaf⟩ is written with a small loop instead of a bar
`,z` ⟨rah⟩ with one dot above and one below
`,s` ⟨sin⟩ with one dot above and one below
`e` like ⟨a⟩, with a ⟨zwarakay⟩ mark if vocalized
`e'i` ⟨yah⟩ with ⟨hamza⟩
`E` ⟨yah⟩ with two dots below aligned vertically
`Ey` ⟨yah⟩ written with a final stroke
`o` mapped to ⟨u⟩
`O` mapped to ⟨U⟩
`w"'` ⟨hamza⟩ on ⟨waw⟩
`h"'` ⟨hamza⟩ on ⟨hah⟩

The ⟨qur'an alif⟩ accent is not available for Pashto.

The rules for ⟨izafet⟩ and ⟨yah-i-wahdat⟩ apply.

Note: Some of the given codings also occur in Urdu but with a different meaning, see above. For writing some words in the Urdu style, write the command `\seturdu` and afterwards switch back to the Pashto conventions by `\setpashto`.

2.11.6 Maghribi:

This is just a different writing convention. ⟨fah⟩ is written with one dot below the letter, ⟨qaf⟩ with one dot above the normal letter form of ⟨fah⟩. The three dots of ⟨vah⟩ are put

below the letter. Otherwise like Arabic.

2.12 Miscellaneous features:

As ArabTEX is slow, it will produce some terminal output while running to indicate it is still alive. If that is not wanted, say "`\quiet`" or "`\tracingarab␣=␣0␣`" (outside an Arabic Environment). "`\tracingarab␣=␣1␣`" will report arabic paragraphs, a value of 2: arabic lines and insertions, a value of 3 or more: individual items.

Whether ⟨yah⟩ in the final position carries dots or not depends on the chosen language convention. You can override this by "`\yahdots`" and "`\yahnodots`".

To reproduce erroneous or archaic texts exactly as they are, the following additional codings are available:

`.k` ⟨kaf⟩ in final position without a diacritical mark
`.f` ⟨fah⟩ without a dot
`.b` ⟨bah⟩ without a dot
`.n` ⟨noon⟩ without a dot (not available for Pashto)
`Y` ⟨alif maqsoura⟩, ⟨yah⟩ without dots in all positions.

2.13 How to move from Version 1 to Version 2

Version 2 is not fully compatible with Version 1; however, moving to the new version should cause little problems, and is recommended as version 1 is no longer supported. Apart from some extensions, most changes were introduced in order to better conform to the transliteration standards, and to have less compatibility problems with TEX and LATEX. Further versions are expected to be upward compatible if no grave bugs turn up.

The main differences between versions 1 and 2 are:

- The font size has increased, so the document layout may change. The old font can no more be used.
- Some arabic characters are now coded differently: ⟨'ain⟩ is denoted by a left quote, and `c`, `^z`, `^t`, and `.n` denote different characters from what they did before. This was changed in order to better conform to the standard transliteration.
- There are a lot more ligatures than before. This normally need not concern the user.
- `\vocalize` will no more generate ⟨sukun⟩ and ⟨wasla⟩ except if explicitly indicated by quoting. See `\fullvocalize`.
- Arabic Environments are always bracketed by the new control sequences `\begin{arabtext}` and `\end{arabtext}` even if only the transcription is wanted.
- Short arabic quotations are now bracketed by `\<` and `>` so `<` has its standard TEX meaning.

We recommend converting existent input files to the new notation. If that is impractical in special cases, the LaTeX option "`oldarabtex`" and/or the command "`\oldarabtex`" will switch back to most of the old conventions (and problems). This shortcut will probably go away in some future version.

2.14 Acknowledgments:

The development of ArabTeX would not have been possible without the assistance of many people. Apart from my local team, helpful advice came among others from Wolfdietrich Fischer, Ahmed El-Hadi, Abdelsalam Heddaya, Iqbal Khan, Tom Koornwinder, Eberhard Krueger, Asif Lakehsar, Jan Lodder, Richard Lorch, Eberhard Mattes, and Bernd Raichle. I also have to thank the many people who sent bug reports and comments.

2.15 Please send bug reports, suggestions and inquiries to the author:

Prof. Klaus Lagally
Institut fuer Informatik
Universitaet Stuttgart
Breitwiesenstrasse 20–22
D-70565 Stuttgart
GERMANY

`lagally@informatik.uni-stuttgart.de`