DRAFT

ArabTEX
a System for Typesetting Arabic

User Manual Version 3.09 [1] [2]

Klaus Lagally

July 22, 1999

[1] Report Nr. 1998/09, Universität Stuttgart, Fakultät Informatik, Breitwiesenstraße 20–22, 70565 Stuttgart, Germany

[2] This Report supersedes Reports Nr. 1992/06 and 1993/11

Overview

ArabTEX is a package extending the capabilities of TEX/LATEX to generate the Perso-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 LATEX2$_\varepsilon$. It is compatible with Babel, CJK, the EDMAC package, and PicTEX (with some restrictions); other additions to TEX have not been tried.

ArabTEX is primarily intended for generating the Arabic writing, but the standard scientific transliteration can also be easily produced. For languages other than Arabic that are customarily written in extensions of the Perso-Arabic script some limited support is available.

ArabTEX defines its own input notation which is both machine, and human, readable, and suited for electronic transmission and E-Mail communication. However, texts in many of the Arabic standard encodings can also be processed.

Starting with Version 3.02, ArabTEX also provides support for fully vowelized Hebrew, both in its private ASCII input notation and in several other popular encodings.

Please send error reports, suggestions and inquiries to the author:

Prof. Klaus Lagally
Institut für Informatik
Universität Stuttgart
Breitwiesenstraße 20-22
70565 Stuttgart
GERMANY

`lagally@informatik.uni-stuttgart.de`

Contents

List of Figures

List of Tables

Chapter 1

Introduction to ArabTEX

Note: This manual describes Version 4 of ArabTEX, which in general is upwards compatible to earlier versions. Incompatible changes are flagged by an asterisk in the margin. *

ArabTEX is a package extending the capabilities of TEX/LATEX to generate an extended Perso-Arabic and/or Hebrew script in addition to the customary left-to-right scripts (called *Roman* in the sequel). Besides Arabic and Hebrew there are provisions for many other languages that use some extensions of the Arabic script; we shall use the term *Arabic* generically to denote any supported right-to-left script, without any cultural or political connotation.

An ArabTEX document is usually multi-lingual and may contain right-to-left insertions within left-to-right paragraphs, and vice versa. There are various possibilities to encode right-to-left insertions: in addition to several standard encodings there are also 7-bit ASCII encodings modelled after various transliteration standards (there is more than one convention, so the intended language must be specified).

ArabTEX, like TEX and LATEX, is not a substitute for a word processor, and does not follow the WYSIWYG paradigm (What You See Is What You Get) where the author has all influence on, and therefore is also completely responsible for, every detail of the visual representation of his text. Instead it is an *off-line system* mirroring the classical publication process where an author, concentrating on the content and the structure of her paper only, will deliver a manuscript to the publisher, who will take care of a reasonable visual rendering of the text.

The publisher involved in our case is TEX, the famous typesetting program by D. E. Knuth; and the amount of technical typesetting knowledge embodied in the algorithms of TEX is such that, in general, the author will tacitly accept the resulting output if she had stated her preferences in a sufficiently precise way; if not, the formatting task has to be repeated under changed specifications, as

also happens in real life.

TEX is easily extensible by providing its own macro programming language, enabling the user to implement packages that contain algorithms embodying additional typesetting knowledge. A prominent example is LATEX by L. Lamport, a package providing standard formatting rules for several common document types. One of the basic concepts of LATEX, but already contained in TEX itself, is "structured markup": the user generally only indicates the logical structure of the document, whereas the formatting rules are laid down within descriptions of standard document classes.

ArabTEX, at the user interface, needs to add only very few features: we provide a way to indicate which parts of the input text are considered to be in Arabic (or some other supported language written from right to left) and thus have to be rendered accordingly, and a way of setting parameters influencing the rendering of Arabic text. Additionally we have to support a representation of the Arabic text that can be handled using the available standard equipment.

Internally, of course, there is much more to it. TEX may be very good at rendering left-to-right texts, but Arabic runs from right to left, so we have to instruct TEX to do it the other way; and if we want to mix Arabic and traditional left-to-right text within the same paragraph, things can get extremely complicated. This shows indeed: ArabTEX is very large and comparatively slow. Fortunately, computer technology is advancing at a high pace, so that the penalty of using very expensive algorithms will decrease just by simple waiting; and we thus never were too much concerned about efficiency (even if we tried to avoid the worst blunders).

In order to see how ArabTEX works, let us consider a very simple example: Figure 1.1 contains the complete TEX input text, Figure 1.2 the corresponding output. In this example (using LATEX) we use a standard "article" format, we activate the ArabTEX package, and we set a few options: our text is supposed to be in Arabic, we want to see all vowel marks (this is not always done in Arabic printing), we additionally want to get the scientific transcription, and, of course, the Arabic writing. The document proper consists of a centered headline and a sequence of short Arabic paragraphs, separated in the input by blank lines[1].

This example is *not* typical insofar as we produce both the scientific transcription of the text, and the Arabic writing, from the same input at the same time, interleaving them. This is rarely required; but in our example it allows us to demonstrate that the input notation is very closely related to the transcription. Details about this correspondence are covered in Chapter 4. What indeed *is* typical is the fact that, apart from the centered headline, we supplied no formatting information whatever; TEX will take care of that.

[1] If you happen to be curious and cannot read Arabic: the text contains a traditional story about a somewhat silly person named Ǧuḥā, trying *not* to lend his donkey to a friend, and failing.

```
\documentclass[12pt]{article}
\usepackage{arabtex}

\begin{document}

\setarab    % choose the language specific conventions
\vocalize   % switch diacritics for short vowels on
\transtrue  % additionally switch on the transliteration
\arabtrue   % print arabic text ... is on by default anyway

\centerline {\RL{^gu.hA wa-.himAruhu}}

\begin{RLtext}
'at_A .sadIquN 'il_A ^gu.hA ya.tlubu minhu .himArahu li-yarkabahu
fI safraTiN qa.sIraTiN wa-qAla lahu:

sawfa 'u`Iduhu 'ilayka fI al-masA'i, wa-'adfa`u laka 'u^graTaN.

fa-qAla ^gu.hA:

'anA 'AsifuN ^giddaN 'annI lA 'asta.tI`u 'an 'u.haqqiqa
laka ra.gbataka, fa-al.himAru laysa hunA al-yawma.

wa-qabla 'an yutimmu ^gu.hA kalAmahu
bada'a al-.himAru yanhaqu fI i.s.tablihi.

fa-qAla lahu .sadIquhu:

'innI 'asma`u .himAraka yA ^gu.hA yanhaqu.

fa-qAla lahu ^gu.hA:

.garIbuN 'amruka yA .sadIqI!
'a-tu.saddiqu al-.himAra wa-tuka_d_dibunI?
\end{RLtext}

\end{document}
```

Figure 1.1: Sample ArabTEX input

جُحَا وَ حِمَارُهُ *ǧuḥā wa-ḥimāruhu*

ʾatā ṣadīqun ʾilā ǧuḥā yaṭlubu minhu ḥimārahu li-yarkabahu fī-safratin qaṣīratin wa-qāla lahu:

أَتَى صَدِيقٌ إِلَى جُحَا يَطلُبُ مِنهُ حِمَارَهُ لِيَركَبَهُ فِي سَفرَةٍ قَصِيرَةٍ وَقَالَ لَهُ:

sawfa ʾuʿīduhu ʾilayka fī ʾl-masāʾi , wa-ʾadfaʿu laka ʾuǧratan.

سَوفَ أُعِيدُهُ إِلَيكَ فِي المَسَاءِ ، وَأَدفَعُ لَكَ أُجرَةً.

fa-qāla ǧuḥā:

فَقَالَ جُحَا:

ʾanā ʾāsifun ǧiddan ʾannī lā ʾastaṭīʿu ʾan ʾuḥaqqiqa laka raġbataka, fa-ʾlḥimāru laysa hunā ʾl-yawma.

أَنَا آسِفٌ جِدًّا أَنِّي لَا أَستَطِيعُ أَن أُحَقِّقَ لَكَ رَغبَتَكَ، فَالحِمَارُ لَيسَ هُنَا اليَومَ.

wa-qabla ʾan yutimmu ǧuḥā kalāmahu badaʾa ʾl-ḥimāru yanhaqu fī ʾṣṭablihi.

وَقبلَ أَن يُتِمَّ جُحَا كَلَامَهُ بَدَأَ الحِمَارُ يَنهَقُ فِي اصطَبلِهِ.

fa-qāla lahu ṣadīquhu:

فَقَالَ لَهُ صَدِيقُهُ:

ʾinnī ʾasmaʿu ḥimāraka yā ǧuḥā yanhaqu.

إِنِّي أَسمَعُ حِمَارَكَ يَا جُحَا يَنهَقُ.

fa-qāla lahu ǧuḥā:

فَقَالَ لَهُ جُحَا:

ġarībun ʾamruka yā ṣadīqī! ʾa-tuṣaddiqu ʾl-ḥimāra wa-tukaḏḏibunī?

غَرِيبٌ أَمرُكَ يَا صَدِيقِي! أَتُصَدِّقُ الحِمَارَ وَتُكَذِّبُنِي؟

Figure 1.2: Sample ArabTEX output

Chapter 2

Input to ArabTEX

ArabTEX is activated by the command `\input arabtex` (with Plain TEX) or `\usepackage{arabtex}` (with LATEX). After activating ArabTEX, select one of the language-specific Arabic writing styles, e.g., `\setarab` (see Chapter 3.2). Your modified TEX/LATEX system will recognize the following items:

- standard TEX/LATEX text and commands,
- *Arabic quotations* [1] as arguments to the command `\RL{ }` (read: "right-to-left") within a *Roman* paragraph. A quotation may also be bracketed by `<` and `>` (or with `\<` and `>` except inside a LATEX `{tabbing}` environment)[2]. An Arabic quotation forms a new group, so any assignments will be local by default. *
- longer Arabic text segments called *Arabic environments*, which are bracketed by the commands `\begin{RLtext}` and `\end{RLtext}` (or also `\begin{arabtext}` and `\end{arabtext}` with the same meaning) (even when using Plain TEX!), An *Arabic environment* consists of one or more paragraphs separated by blank lines or `\par` commands. It forms a group, so assignments will be local by default. *

Arabic quotations and *Arabic environments* will be called *Arabic contexts* in the sequel.

[1] The former restriction that quotations must fit on the current line no more applies.

[2] Quotations closed by `>` may not contain nested insertions; also observe that `\<` must be matched by `>`, *not* by `\>` !

2.1 Arabic text elements

Every *Arabic paragraph* and every *Arabic quotation* is a sequence of the following kinds of *Arabic items,* separated by blank spaces or newlines:

- isolated punctuation marks, interpreted as the corresponding Arabic punctuation mark;
- "numbers", i.e. character sequences starting with a digit, and possibly continued by digits, commas, hyphens, or slashes. A "number" will be processed using the normal writing sequence from left to right; however, a final punctuation mark will be split off and processed separately. *
- "Arabic quotes" coded as two left quotes or two right quotes each, or as `\lq` and `\rq`. They should be written directly adjacent to a word.
- "words", i.e. character sequences starting with a letter or a special (non-digit) character followed by a letter. A final punctuation mark will be split off and processed separately. The (coded) characters of a word will in the output be arranged from right to left.
- a sequence of Arabic text elements (words, numbers, and special characters) enclosed in curly braces `{` and `}` . This introduces a new level of TEX grouping; otherwise the constituents are processed normally. This feature may be nested.

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

Inside an *Arabic Environment,* or in an *Arabic quotation,* you may also have:

- ArabTEX commands with or without parameters. These will be executed immediately.
- Some, but not all, TEX/LATEX commands (see below). These will be executed immediately.
- Short mathematical insertions, bracketed by *single* `$` signs. They must fit on one output line and are processed as usual. TEX display mode (bracketed by `$$`) is not provided within an *Arabic environment*; if it is required, the user has to leave the *Arabic environment* temporarily.
- short left-to right ("Roman") *quotations,* containing text and possibly also TEX/LATEX commands, as argument to `\LR{ }` (read: "left-to-right") or bracketed by `<` and `>`[3]. These introduce a new level of grouping, so if they contain any TEX/LATEX assignments the effects of these will be local by default. When using `"<"` and `">"` this feature is not available *within* an *Arabic quotation.* The alternate notation `\<` is *not* provided.

[3]Quotations closed by `>` may not contain nested insertions.

2.2 Commands within an *Arabic context*

A control sequence inside an *Arabic context* should be separated from the preceding text item by white space or another control sequence, and may be of the following kinds:

- ArabTEX option changing commands. These may also be used outside an *Arabic Context,* and usually follow TEX's grouping rules.
- `\\` for a line break; the current line will be padded out on the left with spaces.
- `\|` or `\break` for a line break; the current line will be spread out. If it comes out very badly spaced, automatic stretching might help (see Section 8).
- `\indent` or `\par` (or a blank line) for a new paragraph, `\noindent` for a new paragraph without indentation (*not inside Arabic quotations*).
- `\emphasize {`*group_of_Arabic_items*`}` will put a bar over the indicated group of *Arabic items.*
- `\setnash, \setnashbf, \setnastaliq` and other font selection commands, see Section 3.3.
- size changing LATEX commands like `\large` etc., only if LATEX is used!
- the following commands: `\footnote` (observe that the syntax for Plain TEX and LATEX is different!), `\marginpar` (also with Plain TEX, analogous to the LATEX usage).
- the TEX/LATEX commands `\smallskip`, `\medskip`, `\bigskip`, `\input`, `\hfill`, `\hfil`, `\vfill`, `\vfil`, `\␣` (for a space), `\space`, `\,` (small space), `\newpage`, `\clearpage`, `\pagebreak` with their usual meaning.
- `\nospace` will place the adjacent items in the output in direct contact, without any intervening space, except in case of a line break.
- `\hspace{`*width*`}` will introduce the indicated amount of spacing in the output. The same is true for `\vspace`, `\hskip` (observe TEX syntax!), and `\vskip`.
- `\mbox{`*text*`}` puts the *text* into a box that will not be split across a line break.
- `\spreadbox{`*width*`}{`*text*`}` spreads out the *text* to the indicated *width.* This may be useful e.g., when typesetting poetry.
  `\spreadbox{`*width*`}{`*text*`\hfill }` will inhibit the spreading,
  `\spreadbox{`*width*`}{\hfill` *text*`\hfill }` will center the *text* inside the box.

`\spreadbox{`*width*`}{\hfill }` or `\spreadbox{`*width*`}{␣}` just introduces the indicated amount of horizontal space, as will `\hspace{`*width*`}`.

If two boxing commands follow each other without any intervening blank space in the input, there will also be no resulting space between the boxes in the output.

- `\centerline{`*text*`}` will start a new line whose contents are centered (*not inside Arabic quotations*).
- `\spreadline{`*text*`}` will start a new line whose contents are spread out over the whole width of the page (*not inside Arabic quotations*). It is approximately equivalent to `\spreadbox{\hsize }{`*text*`}`.
- User defined commands whose expansion produces legal ArabTEX input text may be called by `\docommand{`*command_name and parameters*`}`. The command is expanded exactly once,[4] and the expansion text, after suitable substitution of parameters, will be processed by ArabTEX again.
- User defined commands may also be called directly within an *Arabic context* if they have been previously announced to the ArabTEX processor by `\allowarab{`*command_name*`}`. They are expanded exactly once, and the expansion text (after suitable substitution of parameters) will be processed by ArabTEX again, so it must be legal ArabTEX input text.
- Parameter assignments inside an *Arabic context* may be performed by `\doassign{`*parameter*`}{`*value*`}`. The effect is normally local except if the form `\doassign{\global` *parameter*`}{`*value*`}` is used.
- Any non-recognized command will generate an error message and will be echoed verbatim in the output. Even though ArabTEX tries hard to get into synchronization again, additional spurious errors may occur.
- inside an *Arabic Context* normally no further LATEX or ArabTEX environment may be nested; this restriction does not apply to the yet experimental LATEX document classes `arabrep.cls, arabart.cls, arabbook.cls` which are provided for right-to-left documents.

For a list of all available commands, consult the Index to this report. As a reminder, the command `\arabstat` will cause a list of all commands that are presently valid inside Arabic text to appear in the TEX log file.

[4] This is no strong restriction as the expansion may contain `\docommand` calls again.

Chapter 3

Running ArabTEX

ArabTEX can be used both with Plain TEX and with LATEX, but is activated differently in either case.

3.1 Activating ArabTEX

With Plain TEX, a small loader program is activated by the command `\input arabtex` at the beginning of an input text. It will define a default font, prepare a minimal environment simulating the (very few) LATEX-like features needed, and load the ArabTEX macro files.

With LATEX2$_\varepsilon$ the command `\usepackage {arabtex}` will do all the loading. Users still running LATEX 2.09 (horror!) should either add the option `arabtex` to the `\documentstyle` command, or upgrade to LATEX2$_\varepsilon$.

ArabTEX loads many internal files automatically, and defines a large numbers of internal commands. These all contain an "at"-sign (`@`) within their names and thus should not interfere with user defined commands. Collisions with other macro packages are possible, however, and may lead to surprises and interesting effects.

ArabTEX tries to diagnose the presence of some other packages with which it could run into conflicts, and sometimes locally modifies itself accordingly. For this to be possible, in case of doubt the ArabTEX package should be loaded last.

3.2 Language selection

The processing of input text in ASCII transliteration encoding is somewhat language dependent. Thus before the first *Arabic quotation* or *Arabic environment*

you have to indicate the desired processing mode by one of the language selection commands `\setarab`, `\setfarsi`, `\seturdu`, `\setpashto`, `\setmaghribi`, etc., [1] or `\setverb` (no special processing; see however Section 4.2). The processing mode may be changed at any time, even inside an *Arabic environment* or an *Arabic quotation.*

Arabic insertions are generally included in `\RL{ }` or bracketed by `\<` and `>`. By selecting a language, the symbols `<` and `>` are also activated as shorthands to bracket short insertions in the chosen language. Whereas this is usually convenient, it also has some drawbacks: the angle brackets can thus no more be used for other purposes, except in mathematical mode where they retain their normal meaning as relational operators. To return them to their normal mode of operation, you can deselect them by `\setnone`.[2]

For further details on supported languages, see Section 6.

3.3 Font selection

For producing the extended Arabic script ArabTEX uses a special strategy to build up character shapes from a collection of fragments, which normally do *not* correspond to individual character glyphs. Therefore none of the available free or commercial Arabic fonts can be used; we provide our own "pseudo-fonts".

Presently the following pseudo-fonts are available:

- `"nash14"` is the default,
- `"nash14bf"` is a bold-face version of `"nash14"`,
- `"xnsh14"` is an improved version of `"nash14"` providing additional shape elements used for some exotic script extensions, *
- `"xnsh14bf"` is a bold-face version of `"xnsh14"`.

`"nash14"` and `"xnsh14"` are the default; use `\setnashbf` to switch to bold-face. `\setnash` will switch back.

With Plain TEX the fonts are available by default at 14 point size only, which cooperates well with the `"cm"` fonts at 10 points. Additional sizes are defined within the file `"arabtex.tex"`; they can be activated whenever needed by the command `\setarabfont{`*font*`}`.

[1] We would prefer to use a single switching command like `language`, `\setlanguage`, or `\selectlanguage`, but these names have already been preempted by TEX3 and the Babel package.

[2] Note for advanced TEX users: All language selecting commands except `\setnone` set the character `<` to be active. If *Arabic insertions* are not needed, or are always started with `\<` or `\RL`, the user may reuse the command `<` for other purposes, or deactivate it by `\catcode ‘\<=12` or `\setnone` to return it to its normal meaning.

With LaTeX, the font size changing commands will also operate on the Arabic fonts.

We strongly recommend migrating to `"xnsh14"` and `"xnsh14bf"` by the command `\newarabfont`; `\oldarabfont` will switch back, if necessary. The old fonts `"nash14"` and `"nash14bf"` are of inferior quality and will be phased out gradually. *

All fonts indicated presently are in the Naskhi style; we had started to write a Nastaliq font for Persian and Urdu, but ran into grave implementation problems, yet unsolved.

Due to a donation by Taco Hoekwater, the fonts `"xnsh14"` and `"xnsh14"` are also available in Postscript T1 format. Their use is highly recommended when using a Postscript interpreter, or converting the output to PDF format; readability is dramatically improved.

In Hebrew mode (see section 7) we can use the standard fonts available on CTAN (after installing them locally). As defaults the fonts `"hclassic"` and `"hcaption"` are provided with ArabTeX; these fonts support vowel points.

Chapter 4

Input encoding conventions

4.1 ASCII Transliteration encoding

The ASCII input notation for Arabic text has been modelled closely after the transliteration standards ISO/R 233 and DIN 31635. These standards do not guarantee unique re-transliteration and are also not 7-bit ASCII compatible, therefore some modifications were necessary. These follow the general rules:

- whenever the transliteration uses a single letter, code that letter;
- whenever the transliteration uses a letter with a diacritical mark, put the punctuation character most closely resembling the diacritical mark *before* the letter (and *not* behind it as in some other coding proposals, as otherwise the readability of the input would suffer, and the encoding could become ambiguous).
- use capital letters for writing variants.

4.1.1 Standard Arabic and Persian characters

The standard encodings for Arabic and Persian consonants are given in Table 4.1 and Table 4.2.

- For long vowels, we use the capital letters `<A>`, `<I>`, `<U>` or also `<aa>`, `<iy>`, `<uw>`, with the same meaning.
- To get the defective writing of long vowels, use `<_a>`, `<_i>`, `<_u>`.
- *'alif maqṣūra* is `<_A>` or `<Y>`.

a	ا	*a*	*'alif*	b	ب	*b*	*bā'*	p	پ	*p*	*pā'*
t	ت	*t*	*tā'*	_t	ث	*t̲*	*t̲ā'*	^g	ج	*ǧ*	*ǧīm*
.h	ح	*ḥ*	*ḥā'*	_h	خ	*ẖ*	*ẖā'*	d	د	*d*	*dāl*
_d	ذ	*d̲*	*d̲āl*	r	ر	*r*	*rā'*	z	ز	*z*	*zāy*
s	س	*s*	*sīn*	^s	ش	*š*	*šīn*	.s	ص	*ṣ*	*ṣād*
.d	ض	*ḍ*	*ḍād*	.t	ط	*ṭ*	*ṭā'*	.z	ظ	*ẓ*	*ẓā'*
‘	ع	‘	*‘ayn*	.g	غ	*ġ*	*ġayn*	f	ف	*f*	*fā'*
q	ق	*q*	*qāf*	v	ڤ	*v*	*vā'*	k	ك	*k*	*kāf*
g	گ	*g*	*gāf*	l	ل	*l*	*lām*	m	م	*m*	*mīm*
n	ن	*n*	*nūn*	h	ه	*h*	*hā'*	w	و	*w*	*wāw*
y	ي	*y*	*yā'*	_A	ى	*ā*	*'alif maqṣūra*	T	ة	*t*	*tā' marbuṭa*

Table 4.1: Standard encodings for Arabic and Persian consonants.

- The short vowels *fatḥa*, *kasra*, *ḍamma* are coded `<a>`, `<i>`, `<u>` and need not normally be written except in the following cases:
  - at the beginning of a word where they generate *'alif*,
  - adjacent to *hamza* where they will influence its carrier,
  - when the transliteration is required,
  - in the `\vocalize` and `\fullvocalize` modes.
- *tanwīn* is coded `<aN>`, `<iN>`, or `<uN>`. A silent *'alif*, if required, is supplied automatically; it may also be explicitly written: `<aNA>`. Likewise, a silent *wāw* may be written `<NU>` as in `<‘amruNU>`.
- *hamza* is denoted by a single *right* quote `<’>`. After selecting the language by `\setarab` the carrier of *hamza* will be determined from the context according to the rules for writing Arabic words; if that is not wanted, “quote” the *hamza* (see Section 4.1.3 below). In the `\setverb` mode, the carrier of *hamza* is determined by the following input character; see Section 4.2.

`c`	ځ	c	*ḥā'* with *hamza*
`^c`	چ	$\check{c}$	*ǧīm* with three dots (below)
`,c`	څ	$\acute{c}$	*ḥ́ā'* with three dots (above)
`^z`	ژ	$\check{z}$	*zāy* with three dots (above)
`^n`	ڭ	$\tilde{n}$	*kāf* with three dots (Ottoman)
`^l`	ڵ	$\tilde{l}$	*lām* with a bow accent (Kurdish)
`.r`	ڕ	$\dot{r}$	*rā'* with a bow below (Kurdish)

Table 4.2: Additional encodings generally available.

- *madda* on *'alif* is generated by a right quote (*hamza*) before `<A>`: `<'A>`. It may also be written `<^A>`; likewise, `<^I>` and `<^U>` will produce *madda* on *yā'* and on *wāw*, as required in some older writing conventions.[1] *

- The coding `<‘>` for *‘ayn* is a single *left* quote, beware of confusing it with *hamza*!

- The "invisible consonant" `<|>` 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 transliteration. At the beginning of a word it will suppress a following short vowel; otherwise it acts like a consonant.

- The sequence `<\,>` will insert a small space, as does `<"|>` (see Section 4.1.3 below). The adjacent characters will not be connected. *

- *šadda* is indicated by doubling the appropriate letter coding. Therefore two equal consonants in sequence have to be separated by a short vowel indicator or `<|>` even in `\novocalize` mode.

- The definite article is separated from the following word by a hyphen. It may be written in the assimilated form (if it exists): `<as-salaamu>`, or always as `<al->`; in that case a subsequent "sun letter" must be doubled: `<al-ssalaamu>`, to receive a *šadda*, and to prevent a *sukūn* on the *lām*. The transliteration in both cases is identical.

- Hyphens `<->` are used for tying words together and for separating prefixes and the article; in these cases they start a new word. Hyphens can also be used to indicate inflectional endings, a connecting vowel in Arabic, or an *

[1] The former use of `<~A>`, `<~I>`, and `<~U>` has been discontinued in ArabTEX version 4.

iẓāfet connection in Persian. Hyphens will show up in the transliteration. Additionally, at the beginning and/or the end of an otherwise isolated word they enforce the use of the connecting 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 (*tatwīl*, *kašīda*) without appearing in the transliteration. It may be used repeatedly. See also Section 8.5: automatic stretching.

  For special applications, it can also be coded `<B>`; and `<|B>` will behave like any ordinary consonant and may carry vowel indicators, *tanwīn*, *sukūn*, and, in the combination `<|BB>`: *šadda*.

4.1.2 Vowelization

There are three modes of rendering short vowels:

- `\fullvocalize`:
  - Every short vowel written will generate the corresponding diacritical mark *fatḥa*, *kasra*, *ḍamma*, except if quoted.
  - If `<N>` follows a short vowel, the corresponding form of *tanwīn* is generated instead.
  - Defective writing: The coding `<_a>` will produce a *Qur'an 'alif* accent (also called *dagger 'alif*) instead of an explicit *'alif* character which would be coded `<A>` or `<aa>`. Likewise, `<_i>` will produce a small *'alif* below the preceding consonant in place of `<I>` (`<iy>`), and `<_u>` will produce an inverted *ḍamma* in place of `<U>` (`<uw>`).
  - 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, *sukūn* will be generated except if a double quote precedes the next consonant. The *lām* of the definite article receives no *sukūn* if a doubled "sun letter" follows.
  - *'alif* at the beginning of a word carries *waṣla* instead of the vowel indicator if the preceding word ended with a vowel.
- `\vocalize`: As above, but *sukūn* and *waṣla* will not be generated except if explicitly indicated by "quoting" (see section 4.1.3).
- `\novocalize`: No diacritics will be generated except if explicitly asked for by "quoting"(see section 4.1.3).

In all modes, a doubled consonant will generate *šadda*, and `<'A>` always generates *madda* on *'alif*.

After `<aN>` the silent *'alif* character is generated automatically if required. The silent *'alif* may also be explicitly indicated by `<aNA>`, or coded literally as `<A>` in `\novocalize` mode. If a silent *'alif maqṣūra* is wanted instead, write `<aN_A>`, `<aNY>`, `<_A>` or `<Y>`.

The *tanwīn fatḥa* is normally positioned on the last consonant of the word, even if a silent *'alif* follows. If it is instead supposed to go onto the *'alif* as required by some modern Arabic writing conventions, or in Persian, this behaviour can be achieved by the option `\newtanwin`. The option `\oldtanwin` will restore the classical behaviour.

A silent *'alif* after *wāw* is indicated by `<UA>` or `<WA>` (with a capital `<W>`!).

4.1.3 Quoting

In `\novocalize` mode (see Section 4.1.2), a double quote `<">` will modify the meaning of the following character as follows:

- if a short vowel follows, the appropriate diacritical mark *fatḥa*, *kasra*, *ḍamma* will be put on the preceding character.
  - If `<N>` follows the short vowel, the appropriate form of *tanwīn* will be generated instead.
  - At the beginning of a word, *'alif* is assumed as the first character.
- 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.

  At the beginning of a word, `<"'>` will generate an isolated *hamza*.
- if the following character is the "invisible consonant" `<|>`, the connection between the adjacent letters will be broken and a small space inserted. This can also be denoted `<\,>` instead of `<"|>`. *

  At the beginning of a word, *'alif* with *waṣla* will be generated.
- otherwise: a *sukūn* will be put on the preceding character. The following character will be processed again.

The double quote will not show up in the transliteration.

In `\vocalize` mode, (see Section 4.1.2), quoting will turn a short vowel *off*; likewise, in `\fullvocalize` mode, quoting will also turn a *sukūn* off. Put in other words: quoting will toggle the generation of short vowel indicators and *sukūn* on and off.

4.1.4 Ligatures

There is no way to explicitly enforce ligatures, as a large number of them are generated automatically. The results will not always look satisfactory, so we recommend inspecting the output after the first run. Any unwanted ligature can be suppressed by interposing the invisible consonant `<|>` between the two letters otherwise combined into a ligature. After `\ligsfalse`, in the middle of a word fewer ligatures will be produced; for some texts this looks better. You can return to the normal strategy by `\ligstrue`.

4.1.5 Coding examples for Arabic [2]

The short vowels *fatḥa, kasra, ḍamma* are denoted, as in the transliteration, by the small letters `a, i, u`:

`mana‘a` مَنَعَ *manaʿa*, `_dahaba` ذَهَبَ *ḏahaba*, `^sariba` شَرِبَ *šariba*,

`qabila` قَبِلَ *qabila*, `‘a.zuma` عَظُمَ *ʿaẓuma*, `‘alu` عَلُ *ʿalu*, `bal` بَلْ *bal*,

`ni‘ma` نِعْمَ *niʿma*, `yaktub` يَكْتُبْ *yaktub*.

The long vowels *ā, ī, ū* are denoted by capitals `A, I, U` or by `aa, iy, uw`:

`qAtala` قَاتَلَ *qātala*, `nUzi‘a` نُوزِعَ *nūziʿa*, `lUmI` لُومِي *lūmī*,

`sIrI` سِيرِي *sīrī*; `lawmI` لَوْمِي *lawmī*, `sayrI` سَيْرِي *sayrī*.

Alif maqṣūra is coded as `_A` (or `Y`.)

`ram_A` رَمَى *ramā*, `_dikr_A` ذِكْرَى *ḏikrā*, `‘al_A` عَلَى *ʿalā*, `bal_A` بَلَى *balā*.

Silent *’alif*: The plural suffixes *-ū, -aw* of the verb are denoted `UA`, `aW` or `aWA`:

`katabUA` كَتَبُوا *katabū*, `yaktubUA` يَكْتُبُوا *yaktubū*,

`ramaWA` رَمَوْا *ramaw*, `yalqaW` يَلْقَوْا *yalqaw*.

The defective notation of *ā, ī, ū* can be indicated by `_a, _i, _u` and leads to the appropriate spelling:

`dAru-h_u` دَارُهُ *dāru-hū*, `ri^gli-h_i` رِجْلِهِ *riǧli-hī*,

however: `ramA-hu` رَمَاهُ *ramā-hu*, `yarmI-hi` يَرْمِيهِ *yarmī-hi*;

`_dih_i` ذِهِ *ḏihī*, `h_a_dih_i` هٰذِهِ *hāḏihī*, `tih_i` تِهِ *tihī*, `hAtih_i` هَاتِهِ *hātihī*,

`rabb_i` رَبِّ *rabbī*, `.sAl_i` صَالِ *ṣālī*; `hum_u` هُمُ *humū*;

`qiy_amaTuN` قِيَامَةٌ *qiyāmatun*, `’il_ahuN` إِلٰهٌ *ʾilāhun*,

[2] **Most of the examples are taken from: Wolfdietrich Fischer, Grammatik des Klassischen Arabisch, 2. Auflage, Verlag Otto Harrassowitz, Wiesbaden 1987.**

`sam_awAtuN` سَمٰوَاتٌ *samāwātun*, `_tal_a_tuN` ثَلٰثٌ *ṯalāṯun*,

`l_akin` لٰكِنْ *lākin*, `h_a_dA` هٰذَا *hāḏā*, `'al-ll_ahu` أَللّٰهُ *ʾal-lāhu*,

`'al-rra.hm_anu` أَلرَّحْمٰنُ *ʾar-raḥmānu*, `_d_alika` ذٰلِكَ *ḏālika*.

To reproduce the historical writing correctly, a silent long vowel or *'alif maqṣūra* after `_a` receives no *sukūn* and is ignored in the transliteration:

`.sal_aUTuN` صَلٰوةٌ *ṣalātun*, `.hay_aUTuN` حَيٰوةٌ *ḥayātun*,

`zak_aUTuN` زَكٰوةٌ *zakātun*, `mi^sk_aUTuN` مِشْكٰوةٌ *mišk̄ātun*,

`ar-rib_aU` اَلرِّبٰو *ar-ribā*, `tawr_aITuN` تَوْرٰيةٌ *tawrātun*,

`ram_aYhu` رَمٰىهُ *ramāhu*, `sIm_aYhum` سِيمٰىهُمْ *sīmāhum*.

The short vowel *u* can be written as a long vowel by `_U`:

`'_Ul_A` أُولَى *ʾulā*, `'_UlA'i` أُولَاءِ *ʾulāʾi*, `'_UlU` أُولُو *ʾulū*,

`'_UlAka` أُولَاكَ *ʾulāka*, `'_UlA'ika` أُولَائِكَ *ʾulāʾika*.

***Tanwīn*:** The plural suffixes *-un, -in, -an* are written `-uN, -iN, -aN` or `aNA`. Silent *'alif* in *-an* may be indicated by `A` or omitted; if necessary it is supplied from the context.

`ra^guluN` رَجُلٌ *raǧulun*, `ra^guliN` رَجُلٍ *raǧulin*, `ra^gulaN` رَجُلًا *raǧulan*,

`madInaTaN` مَدِينَةً *madīnatan*, `^gamIlaTaN` جَمِيلَةً *ǧamīlatan*,

`'i_daN` إِذًا *ʾiḏan*, `samA'aN` سَمَاءً *samāʾan*.

There is a special case:

`ribaNU` رِبًو *riban*; `'amruNU` عَمْرٌو *ʿamrun*, `'amriNU` عَمْرٍو *ʿamrin*,

however: `'amraN` عَمْرًا *ʿamran*.

Tanwīn fatḥa is traditionally put on the last consonant even if a silent *'alif* follows. Some modern conventions, and also Persian practice, require to put it on the *'alif* in this case. This behaviour may be switched on by `\newtanwin`, and off by `\oldtanwin`. `\newtanwin` mode is the default for Persian.

`ra^gulaN` رَجُلاً *raǧulan*, `'i_daN` إِذاً *ʾiḏan*.

A silent *'alif maqṣūra* after *tanwīn* is written `aNY` or `aN_A`:

`hudaNY` هُدًى *hudan*, `fataN_A` فَتًى *fatan*;

compare:

`al-hudY` اَلْهُدَى *al-hudā*, `'al-fat_A` أَلْفَتَى *ʾal-fatā*.

Tā' marbuṭa is denoted by `T`:

`kalimaTuN` كَلِمَةٌ *kalimatun*, `kalimaTiN` كَلِمَةٍ *kalimatin*,

`kalimaTaN` كَلِمَةً *kalimatan*; `fatATuN` فَتَاةٌ *fatātun*,

`fatATiN` فَتَاةٍ *fatātin*, `fatATaN` فَتَاةً *fatātan*.

Hamza is indicated by `'`; the appropriate carrier is determined by the context:

`'amruN` أَمْرٌ *ʾamrun*, `'ibiluN` إِبِلٌ *ʾibilun*, `'u_htuN` أُخْتٌ *ʾuḫtun*;

`ra'suN` رَأْسٌ *raʾsun*, `'ar'asu` أَرْأَسُ *ʾarʾasu*, `sa'ala` سَأَلَ *saʾala*,

`qara'a` قَرَأَ *qaraʾa*; `bu'suN` بُؤْسٌ *buʾsun*, `'ab'usuN` أَبْؤُسٌ *ʾabʾusun*,

`ra'ufa` رَؤُفَ *raʾufa*, `ru'asA'u` رُؤَسَاءُ *ruʾasāʾu*; `bi'ruN` بِئْرٌ *biʾrun*,

`'as'ilaTuN` أَسْئِلَةٌ *ʾasʾilatun*, `ka'iba` كَئِبَ *kaʾiba*, `qA'imuN` قَائِمٌ *qāʾimun*,

`ri'AsaTuN` رِئَاسَةٌ *riʾāsatun*, `su'ila` سُئِلَ *suʾila*; `samA'uN` سَمَاءٌ *samāʾun*,

`barI'uN` بَرِيءٌ *barīʾun*, `sU'uN` سُوءٌ *sūʾun*, `bad'uN` بَدْءٌ *badʾun*,

`^say'uN` شَيْءٌ *šayʾun*, `^say'iN` شَيْءٍ *šayʾin*, `^say'aN` شَيْئًا *šayʾan*;

`sA'ala` سَاءَلَ *sāʾala*, `mas'alaTuN` مَسْأَلَةٌ *masʾalatun*,

`saw'aTuN` سَوْأَةٌ *sawʾatun*, `_ha.tI'aTuN` خَطِيئَةٌ *ḫaṭīʾatun*.

Old *Hamza* convention: In an older writing style that is used, e.g., in some Qur'an editions, the *hamza* is sometimes put below its carrier or on the connecting line. This style may be switched on by `\oldhamza` (and off again by `\newhamza`):

`'as'ilaTuN` أَسْئِلَةٌ *ʾasʾilatun*, `ka'iba` كَئِبَ *kaʾiba*, `qA'imuN` قَائِمٌ *qāʾimun*,

`su'ila` سُئِلَ *suʾila*, `^say'aN` شَيْئًا *šayʾan*, `_ha.tI'aTuN` خَطِيئَةٌ *ḫaṭīʾatun*.

Madda in the context *'ā* is generated automatically:

`'AkiluN` آكِلٌ *ʾākilun*, `qur'AnuN` قُرْآنٌ *qurʾānun*, `ra'Ahu` رَآهُ *raʾāhu*.

To reproduce the historic writing correctly, it can also be explicitly indicated by `^A, ^I, ^U` in other contexts:

`'a.sdiq^A'uh_u` أَصْدِقَآؤُهُ *ʾaṣdiqāʾuhū*;

`ya^g^I'u` يَجِيءُ *yağīʾu*, `s^U'ila` سُوئِلَ *sūʾila*.

***Šadda*:** A double consonant must be written twice, even if it is coded by more than one character:

`nazzala` نَزَّلَ *nazzala*, `ba^s^sAruN` بَشَّارٌ *baššārun*, `nawwara` نَوَّرَ *nawwara*,

`sayyiduN` سَيِّدٌ *sayyidun*, `sa''AluN` سَأَّالٌ *saʾʾālun*,

`.sabiyyuN` صَبِيٌّ *ṣabiyyun*, `ʿaduwwuN` عَدُوٌّ *ʿaduwwun*.

Instead of `iyy, uww` one can also write `Iy, Uw`:

`.sabIyuN` صَبِيٌّ *ṣabīyun*, `ʿadUwuN` عَدُوٌّ *ʿadūwun*.

Assimilation: the definite article may be always written `al-`; a following "sun letter" must be written twice like in the Arabic spelling. The transliteration and the use of *sukūn* are adjusted accordingly:

`'al-ddAru` أَلدَّارُ *ʾad-dāru*, `'al-rra^gulu` أَلرَّجُلُ *ʾar-raǧulu*,

`'al-ssanaTu` أَلسَّنَةُ *ʾas-sanatu*, `'al-nnAru` أَلنَّارُ *ʾan-nāru*;

`'al-^gAru` أَلْجَارُ *ʾal-ǧāru*, `'al-bAbu` أَلْبَابُ *ʾal-bābu*;

`'al-llaylaTu` أَلَّيْلَةُ *ʾal-laylatu*, `'al-llisAnu` أَللِّسَانُ *ʾal-lisānu*,

`'al-ll_ahu` أَللّٰهُ *ʾal-lāhu*.

The article may also be written in the assimilated form, with identical result:

`'ad-dAru` أَلدَّارُ *ʾad-dāru*, `'ar-ra^gulu` أَلرَّجُلُ *ʾar-raǧulu*,

`'as-sanaTu` أَلسَّنَةُ *ʾas-sanatu*, `'an-nAru` أَلنَّارُ *ʾan-nāru*.

In some special cases the literal spelling must be used:

`'alla_dI` أَلَّذِي *ʾallaḏī*, `'alla_dIna` أَلَّذِينَ *ʾallaḏīna*, `'allatI` أَلَّتِي *ʾallatī*;

however:

`'al-lla_dAni` أَللَّذَانِ *ʾal-laḏāni*, `'al-llatAni` أَللَّتَانِ *ʾal-latāni*,

`'al-llawAtI` أَللَّوَاتِي *ʾal-lawātī*.

***Waṣla*:** an auxiliary vowel at the beginning of a word is always written, but in the middle of a sentence generally without *hamza*. If a vowel precedes the word, the auxiliary vowel will be omitted in the transcription, and the *waṣla* sign will be used in the spelling:

`wa-ismuhu` وَٱسْمُهُ *wa-ʾsmuhu*, `f--a-in.sarafa` فَٱنْصَرَفَ *fa-ʾnṣarafa*.[3]

This also works across word boundaries:

[3] In vowelized writing, it may sometimes be advisable to introduce a *kašīda* to prevent the vowel marks from bumping into each other.

`yA ibnI` يَا آبْنِي *yā ’bnī*, `h_a_dA ibnuh_u` هٰذَا آبْنُهُ *hāḏā ’bnuhū*,

`qAla u_hru^g` قَالَ آخْرُجْ *qāla ’ḫruǧ*.

An auxiliary vowel at the end of the preceding word may be separated by a hyphen:

`qad-i in.sarafa` قَدِ آنْصَرَفَ *qad-i ’nṣarafa*,

`ra’aW-u al-bAba` رَأَوُا آلْبَابَ *raʾaw-u ’l-bāba*,

`min-i ibnih_i` مِنِ آبْنِهِ *min-i ’bnihī*.

This also works for the article preceding *’alif al-waṣl*:

`’al-i-ismu` أَلِآسْمُ *ʾal-i-’smu*, `’al-i-i^stirA’u` أَلِآشْتِرَاءُ *ʾal-i-’štirāʾu*,

and even if the auxiliary vowel is omitted in the spelling:

`ra^guluN-i ibnatuh_u ^gamIlaTuN` رَجُلٌ آبْنَتُهُ جَمِيلَةٌ *rağulun-i ’bnatuhū ğamīlatun*,
`mu.hammaduN-i al-qura^sIyu` مُحَمَّدٌ آلْقُرَشِيُّ *muḥammadun-i ’l-qurašīyu*.

The particles *li-* and *la-* must be combined with the article except before *lām*:

`lil-rra^guli` لِلرَّجُلِ *lir-rağuli*, `lal-ma^gdu` لَلْمَجْدُ *lal-mağdu*;

however:

`li-llaylaTi` لِلَّيْلَةِ *li-llaylati*, `li-ll_ahi` لِلّٰهِ *li-llāhi*.

The Name of God is written with a special ligature if it is recognized from the input sequence `ll_ah`:

`’al-ll_ahu` أَللّٰهُ *ʾal-lāhu*, `ta-al-ll_ahi` تَآللّٰهِ *ta-’l-lāhi*.

Increased spacing (*Tatwīl*) between adjoining characters may be produced by a double hyphen `--`;

`qabila` قَبِلَ *qabila*, `qa--bi--la` قَـبِـلَ *qabila*, `q--ab--ila` قَـبِـلَ *qabila*.
`q--a--b--i--la` قَـبِـلَ *qabila*, `qa----bi----la` قَـبِـلَ *qabila*

This feature should be used with discretion; automatic spreading usually leads to a better result.

Ties between words are indicated by a single hyphen:

`bi-baladiN` بِبَلَدٍ *bi-baladin*, `ta-al-ll_ahi` تَآللّٰهِ *ta-’l-lāhi*,

`sa-ya’tI` سَيَأْتِي *sa-yaʾtī*, `li-yafra.ha` لِيَفْرَحَ *li-yafraḥa*,

`wa-iswadda` وَٱسْوَدَّ *wa-'swadda*, `ba‘da-mA` بَعْدَمَا *baʿda-mā*,

`.tAla-mA` طَالَمَا *ṭāla-mā*, `fI-ma` فِيمَ *fī-ma*, `‘alA-ma` عَلَامَ *ʿalā-ma*.

A single hyphen at the beginning or end of a word will enforce the use of the joining form of the first resp. the last character, if that form exists (for special uses only):

`s` س *s*, `-s` س *-s*, `-s-` سـ *-s-*, `s-` سـ *s-*

`h` ه *h*, `-h` ه *-h*, `-h-` ـهـ *-h-*, `h-` هـ *h-*

`d` د *d*, `-d` د *-d*, `lA` لا *lā*, `-lA` لا *-lā*

`1400 h-` ١٤٠٠ هـ *1400 h-*

Digit sequences are written in the natural order:

`1234567890` ١٢٣٤٥٦٧٨٩٠ *1234567890*

Hyphen and comma as a decimal separator do not terminate the number:

`123-456,789` ١٢٣-٤٥٦,٧٨٩ *123-456,789*

Ligatures are generated automatically; they can be suppressed by `|`:

`’al-’islAmu` أَلْإِسْلَامُ *ʾal-ʾislāmu*;

`’al-^gAru` أَلْجَارُ *ʾal-ǧāru*, `’al|^gAru` أَلْجَارُ *ʾalǧāru*;

`_tumma` ثُمَّ *ṯumma*, `_tu|mma` ثُمَّ *ṯumma*;

`mu.hammaduN` مُحَمَّدٌ *muḥammadun*, `mu|.ha|mmaduN` مُحَمَّدٌ *muḥammadun*.

Abbreviations and emphasis are indicated by `\emphasize`:

`\emphasize {.sl‘m}` صلعم *ṣlʿm*

`\emphasize {ab\,^g}` أب ج *abǧ*

`\emphasize {‘alayhi as-salAmu}` عليه السّلام *ʿalayhi ’s-salāmu*

4.2 Verbatim input

After disabling language specific processing by `\setverb`, ArabTEX will not use any context information to determine the carrier of *hamza*. Instead the user has to supply this information himself by the next character typed after `<’>`. Generally this character will be used as the carrier; for examples and some exceptions see Table 4.3. A short vowel indicator may follow.

To ease automatic conversion, an initial *’alif* may also be coded `<A>`.

`'a`	أ	*hamza* on *'alif*	`'i`	إ	*hamza* below *'alif*
`'w`	ؤ	*hamza* on *wāw*	`'y`	ئ	*hamza* on a tooth
`'h`	ۂ	*hamza* on *hā'*	`'B`	ء	*hamza* on the line
`'|`	ء	isolated *hamza*	`'A`	آ	*madda* on *'alif*

Table 4.3: Verbatim encodings for the carrier of *hamza*

4.3 Alternate input encodings

The ArabTEX input notation has been very carefully designed for flexibility, readability, and ease of use for linguists confined to standard 7-bit ASCII equipment for processing and transmitting data. However, it does not make much sense re-coding existing machine-readable text files that have been encoded according to other standards. Thus, some alternate reading modules have been written (as there are more than 10 different codings in current use, this is an open-ended activity), and a general code switching procedure has been provided.

An alternate reading module, e.g. `asmo449.sty` for the ASMO 449 code, is installed by `\usepackage{asmo449}` or by `\input asmo449.sty`. Afterwards, a *code_name* (in this case `asmo449`) is defined. Input encoding is switched by the command `\setcode{`*code_name*`}` that changes the coding for *Arabic text* globally. Encoding may be switched several times in the same document, provided the appropriate reading modules are installed; `\setcode{arabtex}` or `\setcode{standard}` returns to the standard ArabTEX notation.

As texts coded in an alternate encoding are always rendered verbatim, the commands `\novocalize`, `\vocalize`, `\fullvocalize` and the language selection commands `\setarab` etc. generally make no sense and are temporarily disabled.

4.3.1 ASMO 449 = ISO 9036

ASMO 449 (see Table 4.4) is a 7-bit code, differing from ASCII (ISO 646) mainly by replacing the Roman letters by the Arabic letter characters and diacritical marks; the Arabic digits share their positions with the ASCII digits. The positions of special and control characters in both codes are identical. ASMO 449 is supported by Arabic MS-DOS.

The file `asmo449.sty` contains a reading module for the ASMO 449 code (identical to ISO 9036). It is installed by the LATEX command `usepackage {asmo449}` or by `\input asmo449.sty`. The module is activated by `\setcode {asmo449}` or `\setcode {iso9036}`; all following Arabic text will be considered to be coded

	0	1	2	3	4	5	6	7
00	NUL	DLE	SP	٠	@	ذ	ـ	ـً
01	SOH	DC1	!	١	ء	ر	ف	ـّ
02	STX	DC2	”	٢	آ	ز	ق	ـْ
03	ETX	DC3	#	٣	أ	س	ك	ـًّ
04	EOT	DC4	$	٤	ؤ	ش	ل	ـٌّ
05	ENQ	NAK	%	٥	إ	ص	م	ـٍّ
06	ACK	SYN	&	٦	ئ	ض	ن	ـَّ
07	BEL	ETB	’	٧	ا	ط	ه	ـُّ
08	BS	CAN	)	٨	ب	ظ	و	ـِّ
09	HT	EM	(	٩	ة	ع	ى	
10	LF	SUB	*	:	ت	غ	ي	
11	VT	ESC	+	؛	ث	[	ـً	}
12	FF	IS4	‘	>	ج	\	ـٌ	ا
13	CR	IS3	−	=	ح	[	ـٍ	{
14	SO	IS2	.	<	خ	^	ـَ	~
15	SI	IS1	/	؟	د	_	ـُ	DEL

Table 4.4: ASMO 449 code table

according to the ASMO 449 standard.

Texts in ASMO 449 are usually not fully vowelized; thus the transliteration cannot be expected to be correct. This is especially true for Egyptian texts which commonly do not differentiate between *yā'* and *'alif maqṣūra*.

A minimal driver file for processing existing ASMO 449 text, e.g. in a file `asmotext.dat`, could look as follows:

```
\documentclass {article}
\usepackage{arabtex}
\usepackage{asmo449}
\begin {document}
\setcode {asmo449}
\begin {RLtext}

\input asmotext.dat

\end {RLtext}
\end {document}
```

4.3.2 ASMO 449E = ISO 8859 - 6

The file `iso88596.sty` contains a reading module for the ISO 8859-6 code (extended ASMO 449 = ASMO 449E). It is installed by the LaTeX command `\usepackage{iso88596}` or by `\input iso88596.sty`. The module is activated by `\setcode{iso8859-6}`; all following Arabic text will be considered to be coded according to the ISO 8859-6 standard. The ArabTeX notation may be reactivated by `\setcode{arabtex}`.

ISO 8859-6 (see Table 4.5) is an 8-bit code closely related both to 7-bit ASCII and to ASMO 449; whereas the lower 128 positions are identical to ASCII (ISO 646), the upper 128 positions contain the Arabic characters of ASMO 449 in the analogous places, plus a few additional graphic and control characters.

We exploit the close relationship of these codes by reusing the ASMO 449 reading routines, after suitable modification of the input. This only works correctly if the input text does not contain genuine ASCII letters, as we project the Arabic characters onto their locations in ASMO 449. Some of the code switching messages in the log file are spurious; do not worry.

The notes on vowelization and transliteration of ASMO 449 apply also.

The driver file indicated for ASMO 449 will be usable after the obvious modifications; however, your TeX installation must be capable of processing 8-bit data input. This is nowadays usually the case; otherwise you can try to locally find some utility program that will strip the highest order bit off the characters in your file, and process the result via ASMO 449.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	٠	@	P	‘	p			NSP		ASP	ذ	ـ	ـِ
01	SOH	DC1	!	1	١	A	Q	a	q					ء	ر	ف	ـّ
02	STX	DC2	”	2	٢	B	R	b	r					آ	ز	ق	ـْ
03	ETX	DC3	#	3	٣	C	S	c	s					أ	س	ك	ـًّ
04	EOT	DC4	$	4	٤	D	T	d	t			¤		ؤ	ش	ل	ـٌّ
05	ENQ	NAK	%	5	٥	E	U	e	u					إ	ص	م	ـٍّ
06	ACK	SYN	&	6	٦	F	V	f	v					ئ	ض	ن	ـَّ
07	BEL	ETB	’	7	٧	G	W	g	w					ا	ط	ه	ـُّ
08	BS	CAN	(	8	٨	H	X	h	x					ب	ظ	و	ـِّ
09	HT	EM	)	9	٩	I	Y	i	y					ة	ع	ى	
10	LF	SUB	∗	:		J	Z	j	z					ت	غ	ي	
11	VT	ESC	+	;		K	[	k	{				؛	ث		ـً	
12	FF	IS4	,	<		L	\	l	|			‘		ج		ـٌ	
13	CR	IS3	−	=		M	]	m	}			SHY		ح		ـٍ	
14	SO	IS2	.	>		N	^	n	~					خ		ـَ	
15	SI	IS1	/	?		O	_	o	DEL				؟	د		ـُ	

Table 4.5: ISO 8859-6 code table

4.3.3 CP 1256 = Arabic Windows Encoding

The file `arabwin.sty` contains a reading module for the Arabic part of the code page 1256 used within *MS Arabic Windows*. It is installed by the LaTeX command `\usepackage{arabwin}` or by `\input arabwin.sty`. The module is activated by `\setcode{arabwin}` or `\setcode{cp1256}`; all following Arabic text will be considered to be coded according to the *MS Arabic Windows* standard. The ArabTeX notation may be reactivated by `\setcode{arabtex}`.

The code page 1256 used in *MS Arabic Windows* (see Table 4.6) is an 8-bit code closely related to 7-bit ASCII; whereas the lower 128 positions are identical to ASCII (ISO 646), some of the upper 128 positions contain the Arabic characters plus additional graphic and control characters.

We reuse the ASMO 449 reading routines, after suitable modification of the input. This only works correctly if the input text does not contain genuine ASCII letters, as we project the Arabic characters onto their locations in ASMO 449. Please note that only the characters that appear in Table 4.6 are processed correctly. Some of the code switching messages in the log file may be spurious; do not worry.

The notes on vowelization and transliteration of ASMO 449 apply also.

The driver file indicated for ASMO 449 will be usable after the obvious modifications; however, your TeX installation must be capable of processing 8-bit data input.

4.3.4 ISIRI 3342

The file `isiri.sty` contains a reading module for the ISIRI 3342 Persian Standard Code. It is installed by the LaTeX command `\usepackage{isiri}` or by `\input isiri.sty`. The module is activated by `\setcode{isiri}`; all following Arabic text will be considered to be coded according to the ISIRI 3342 standard. The ArabTeX notation may be reactivated by `\setcode{arabtex}`.

The ISIRI 3342 code (see Table 4.7) is an 8-bit code closely related to 7-bit ASCII; whereas the lower 128 positions are identical to ASCII (ISO 646), some of the upper 128 positions contain the Arabic/Persian characters plus additional graphic and control characters.

The notes on vowelization and transliteration of ASMO 449 apply also.

The driver file indicated for ASMO 449 will be usable after the obvious modifications; however, your TeX installation must be capable of processing 8-bit data input.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	‘	p	پ	گ	NSP		ه	ذ		ً
01	SOH	DC1	!	1	A	Q	a	q			‘		ء	ر	ل	ٌ
02	STX	DC2	”	2	B	R	b	r					آ	ز		ٍ
03	ETX	DC3	#	3	C	S	c	s					أ	س	م	َ
04	EOT	DC4	$	4	D	T	d	t			¤		ؤ	ش	ن	
05	ENQ	NAK	%	5	E	U	e	u					إ	ص	ه	ُ
06	ACK	SYN	&	6	F	V	f	v					ئ	ض	و	ِ
07	BEL	ETB	’	7	G	W	g	w					ا			
08	BS	CAN	(	8	H	X	h	x		ک			ب	ط		ّ
09	HT	EM	)	9	I	Y	i	y					ة	ظ		
10	LF	SUB	*	:	J	Z	j	z	ٹ	ڑ	ھ	؛	ت	ع		ْ
11	VT	ESC	+	;	K	[	k	{					ث	غ		
12	FF	IS4	,	<	L	\	l	|					ج	ـ	ى	
13	CR	IS3	−	=	M	]	m	}	چ		SHY		ح	ف	ي	LRO
14	SO	IS2	.	>	N	^	n	~	ژ				خ	ق		RLO
15	SI	IS1	/	?	O	_	o	DEL	ڈ	ں		؟	د	ك		ے

Table 4.6: Windows CP 1256 code table

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	‘	p			SP	۰	آ	س	ه	ـَ
01	SOH	DC1	!	1	A	Q	a	q			PSP	۱	ا	ش	ی	ـِ
02	STX	DC2	”	2	B	R	b	r			PCN	۲	ء	ص	]	ـُ
03	ETX	DC3	#	3	C	S	c	s			!	۳	ب	ض	[	ـً
04	EOT	DC4	$	4	D	T	d	t			¤	۴	پ	ط	{	ـٍ
05	ENQ	NAK	%	5	E	U	e	u			%	۵	ت	ظ	}	ـٌ
06	ACK	SYN	&	6	F	V	f	v			.	۶	ث	ع	»	ـّ
07	BEL	ETB	’	7	G	W	g	w			,	۷	ج	غ	«	ـْ
08	BS	CAN	(	8	H	X	h	x			(	۸	چ	ف	*	أ
09	HT	EM	)	9	I	Y	i	y			)	۹	ح	ق	-	ؤ
10	LF	SUB	*	:	J	Z	j	z			×	:	خ	ک	ا	إ
11	VT	ESC	+	;	K	[	k	{			+	؛	د	گ	\	ئ
12	FF	IS4	,	<	L	\	l	|			،	>	ذ	ل		ة
13	CR	IS3	−	=	M	]	m	}			−	=	ر	م		ك
14	SO	IS2	.	>	N	^	n	~			/	<	ز	ن		ي
15	SI	IS1	/	?	O	_	o	DEL			/	؟	ژ	و		

Table 4.7: ISIRI 3342 code table

4.3.5 UNICODE Arabic

The file `utf8.sty` contains a reading module for the Arabic and the Hebrew segment of UNICODE in UTF-8 encoding. It is installed by the LaTeX command `\usepackage{utf8}` or by `\input utf8.sty`.

UTF-8 (UNICODE Transmission Format, see tables 4.8, 4.9, and 7.3) is a multi-byte encoding which, for Arabic and Hebrew, uses two bytes per character whereas ASCII characters use a single byte. Far-eastern languages are encoded in three bytes per character. This is in contrast to UNICODE itself which always uses two bytes per character.

The module is activated by `\setcode{utf8}`; all following Arabic and Hebrew text will be considered to be coded according to the UTF-8 encoding standard. To use the correct font, select the appropriate language. The ArabTeX notation may be reactivated by `\setcode{arabtex}`.

	060	061	062	063	064	065	066	067
0				ذ	ـ	ـِ		ـٰ
1			ء	ر	ف	ـّ	١	آ
2			آ	ز	ق	ـْ	٢	أ
3			أ	س	ك	ـٓ	٣	إ
4			ؤ	ش	ل	ـٔ	٤	ء
5			إ	ص	م	ـٕ	٥	ٱ
6			ئ	ض	ن		٦	ٶ
7			ا	ط	ه		٧	ٷ
8			ب	ظ	و		٨	ٸ
9			ة	ع	ى		٩	ٹ
A			ت	غ	ي		%	ٺ
B		؛	ث		ً		,	پ
C	،		ج		ٌ		,	ټ
D			ح		ٍ		*	ٽ
E			خ		َ			پ
F		؟	د		ُ			ٿ

Table 4.8: UNICODE Arabic, Part 1

	068	069	06A	06B	06C	06D	06E	06F
0	ﭖ	ﭦ		ﮒ	ﮬ	ﭘ		۰
1	ﭠ	ﭬ	ﮢ	ﮔ	ﮭ	ﭙ		۱
2	ﭡ	ﭭ	ﮣ	ﮕ	ﮮ	ﮯ		۲
3	ﭨ	ﭮ	ﮤ	ﮖ	ﮯ	ﮰ		۳
4	ﭩ	ﭯ	ﮥ	ﮗ	ﮰ	ﮱ		۴
5	ﭪ	ﭰ	ﮦ	ﮘ	ﮱ	﮲		۵
6	ﭫ	ﭱ	ﮧ	ﮙ	﮲			۶
7	ﭬ	ﭲ	ﮨ	ﮚ	﮳			۷
8	ﭭ	ﭳ	ﮩ	ﮛ	﮴			۸
9	ﭮ	ﭴ	ﮪ	ﮜ	﮵			۹
A	ﭯ	ﭵ	ﮫ	ﮝ	﮶			؊
B	ﭰ	ﭶ	ﮬ	ﮞ	﮷			؋
C	ﭱ	ﭷ	ﮭ	ﮟ	﮸			،
D	ﭲ	ﭸ	ﮮ	ﮠ	﮹			؍
E	ﭳ	ﭹ	ﮯ	ﮡ	﮺			؎
F	ﭴ	ﭺ	ﮰ	ﮢ	﮻			

Table 4.9: UNICODE Arabic, Part 2

Chapter 5

Transliteration

In addition to the arabic writing, the standard scientific transliteration may also be obtained from a fully vowelized input text. This mode is activated by `\transtrue` and may be switched off again by `\transfalse`.[1] If *only* the transliteration 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 font used for the transliteration is normally *italic*, it can be changed by `\settransfont{`*font*`}`. *

5.1 ZDMG transliteration style

The "ZDMG transliteration" is in fact a family of closely related, but slightly different, transliteration conventions for several languages using the Perso-Arabic script. Therefore for producing it correctly, the appropriate language mode must have been selected.

For Arabic text, the following special cases are handled:

- after the definite article, a double consonant will be assimilated;
- an initial vowel will be replaced by an apostrophe whenever the preceding word ended with a vowel (in this case a *waṣla* appears in the Arabic writing). If that is not wanted, start with *hamza*.
- a silent *'alif* or *'alif maqṣūra* after `<N>` (*tanwīn*) and `<U>` is omitted in the transliteration. The same happens after *wāw* if it is written as a capital `<W>`.
- To correctly reproduce some historical writings, a silent long vowel after `<_a>` is omitted in the transliteration. For examples, see section 4.1.5.

[1] The former option "atrans" is no more necessary.

For Persian texts, the Izafet connection is handled specially, and a final silent *h* will be omitted in the transliteration.

5.2 Other transliteration styles

Since there is no general agreement on transcriptions, a number of variants have been provided:

- `\settrans{english}` will switch to the style of the Encyclopedia of Islam which is close to the conventions of the Library of Congress.
- `\settrans{iranica}` will produce the style used in the Encyclopedia Iranica.
- `\settrans{farsi}` produces a variant of the style used in the Encyclopedia Iranica.
- `\settrans{lazard}` switches to the conventions of Gilbert Lazard: "La langue des plus anciens documents de la prose persane".
- `\settrans{urdu}` switches to the conventions for Urdu used in the ALA-LC tables.
- `\settrans{kashmiri}` switches to the conventions for Kashmiri used in the ALA-LC tables; this mode is also chosen automatically by `\setkashmiri`.
- `\settrans{turk}` will produce a style similar to modern Turkish; it only makes sense for Ottoman texts.
- `\settrans{standard}` or `\settrans{zdmg}` will revert to the standard ZDMG mode.
- Transcription conventions for Hebrew are given in section 7.8.

The transliteration mode may be switched at any time. If the input text is not fully vowelized, the transcription cannot be expected to be correct.

5.3 Capitalization

If transcription output is used as part of a Roman text, it may be desirable to have some words start with a capital letter. This can be achieved by prefixing the command `\cap` to the word in question. If the first letter is *hamza* or *'ayn*, the next letter will be capitalized. This feature may also be used after the article or a prefix, and even in other arbitrary positions; `\cap` will only influence the following letter. The Arabic writing is not affected.

Chapter 6

Support for other languages using Perso-Arabic script

ArabTEX is primarily intended for typesetting texts in classical and modern Arabic, but it also provides some support for several other languages that are customarily written using the Arabic alphabet or some extension of it.

In order to switch to the conventions for one of these languages, say `\setfarsi`, `\seturdu`, `\setpashto`, `\setmaghribi`, etc.; `\setverb` will switch off any language specific processing. `\setarab` can be used to switch back to the Arabic conventions.

After selecting the language, `<` and `>` are active as delimiters for quotations; `\setnone` will return `<` and `>` to their normal TEX meaning. Quotations still can be bracketed by `\<` and `>` or by using `\RL{ }`.

This part of ArabTEX relies heavily on contributions from the user community; we want to especially mention Ivan Derzhanski who completely reimplemented the routines for processing Persian. As we extensively modified these contributions again while integrating the system, we are solely responsible for any remaining, or newly introduced, errors.

6.1 Persian (Farsi, Dari), also Ottoman and Kurdish

The Persian mode is activated by `\setfarsi`.

- 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> without a vowel indicator. <H> denotes final silent *hā'*. This *hā'* receives no *sukūn* even in fully vowelized mode.

- For *fatḥa* or *kasra* followed by a final silent *hā'* you can also write <,a> or <,e> in place of <aH> and <eH> (deprecated).
- The *iẓāfet* connection may always be written <-i> or <-e> (with hyphen); then ArabTEX tries to determine the correct spelling from the context. Likewise the *yā'-i-waḥdat* can always be written <-I> or <-E>.
- The present tense forms of the copula are coded <-am>, <-I>, <-ast>, <-Im>, <-Id>, <-and>. In the output they are written as separate words after a little space.
- The final *yā'* carries no dots. Farsi uses the Nasta'liq font if available, otherwise Naskh.

6.1.1 Coding examples for Persian[1]

The short vowels *æ (ă)*, *e (ĕ)*, *o (ŭ)* are denoted by the lowercase letters a, e or i, o or u:

bar بَرْ *bar*, beh بِهْ *beh*, bon بُنْ *bon*.

The long vowels *a (ā)*, *i (ī, ē)*, *u (ū, ō)* are denoted by the capital letters A, I or E, U or O. *Ælef mædde* is automatically generated for word-initial *a*:

Ab آبْ *āb*, bAd بَادْ *bād*, bId بِيدْ *bīd*, bUd بُودْ *būd*.

Note that I yields a *ya-ye mæ'ruf* (with *zīr*), whilst E yields a *ya-ye mæjhul* (without *zīr*). Similarly, U yields a *waw-e mæ'ruf* (with *piš*), whilst O yields a *waw-e mæjhul* (without *piš*):

tIr تِيرْ *tīr*, tE.g تيغْ *tēġ*; dUr دُورْ *dūr*, zOr زورْ *zōr*.

The diphthongs *éı̂* and *óû* are written ay and aw:

pay پَيْ *pay*, naw نَوْ *naw*.

Intervocalic *ḥæmze* is written ':

pA'Iz پَائِيزْ *pā›īz*; miyA'I مِيَائِی *miyā›ī*, mIgU'I مِيگُوئِی *mīgū›ī*;

tawAnA'I تَوَانَائِی *tawānā›ī*, zanA^sU'I زَنَاشُوئِی *zanāšū›ī*.

[1] **We gratefully acknowledge the voluntary help by Ivan Derzhanski who wrote this chapter, *and* implemented the language-specific processing. As we extensively modified his routines during system integration, all responsibility for any remaining, or new, errors rests with us.**

Silent word-final *wāw* is generated by `_U` or `O`:

`t_U` تُو *tu*, `d_U` دُو *du*; `tO` تو *tō*, `dO` دو *dō*.

Waw-e mæʻdul is written `w`; it is omitted in the transliteration and the preceding *xe* receives no *jæzm*:

`_hwAb` خوَابْ *ḫāb*, `_hwI^s` خوِيشْ *ḫīš*, `_hwod` خوُدْ *ḫod*.

Ha-ye hæwwæz-e mæxfi is generated by `H`, or optionally by `,e`, `,a` or `,A`. It does not receive a *jæzm* even in fully vocalised mode and is not joined to a following letter:

`_hAneH` خَانِه *ḫāneh*, `^c,e` چه *čeh*, `naH` نَه *nah*,

`yal_aH` يَلٰه *yalāh*, `yal,A` يَلٰه *yalāh*

`_hAneHhA` خَانِهَا *ḫānehhā*, `_hAneH-hA` خَانِه هَا *ḫāneh-hā*.

Short eḍafe is written `-e` or `-i`:

`ketAb-e U` كِتَابِ اُو *ketāb-e ū*, `rAh-e t_U` رَاهِ تُو *rāh-e tu*,

`nAmeH-i man` نَامِهِ مَنْ *nāmeh-i man*,

`bInI-e An mard` بِينِئِ آنْ مَرْدْ *bīnī-e ān mard*,

`pA-i In zan` پَايِ اِينْ زَنْ *pā-i īn zan*,

`bAzU-i In zan` بَازُوِي اِينْ زَنْ *bāzū-i īn zan*.

Long eḍafe is written `-_i`:

`dAr-_i man` دَارِ مَنْ *dār-ī man*, `_hU-_i t_U` خُوِي تُو *ḫū-ī tu*.

Ḥæmze* as *ya-ye wæḥdæt/nesbæt/xeṭab is likewise written `-_i`:

`nAmeH-_i` نَامِهِ *nāmeh-ī*, `sormeH-_i` سُرْمِهِ *sormeh-ī*,

`gofteH-_i` گُفْتِهِ *gofteh-ī*.

Ye-ye wæḥdæt is written `-I` or `-E`:

`ketAb-I` كِتَابِى *ketāb-ī*, `rAh-I` رَاهِى *rāh-ī*, `nAmeH-I` نَامِه اِى *nāmeh-ī*;

`dAnA-I` دَانَائِى *dānā-ī*, `pArU-I` پَارُوئِى *pārū-ī*;

`dAnA-I-keH` دَانَائِيكِه *dānā-ī-keh*, `pArU-I-keH` پَارُوئِيكِه *pārū-ī-keh*.

The present tense forms of the verb *budæn* and the pronominal clitics are written as they are spoken:

`rafteH-am` رَفْتِه اَمْ *rafteh-am*, `rafteH-Im` رَفْتِه اِيمْ *rafteh-īm*,

`rafteH-I` رَفْتِه اِی *rafteh-ī*, `rafteH-Id` رَفْتِه اِیدْ *rafteh-īd*,

`rafteH-ast` رَفْتِه اَسْتْ *rafteh-ast*, `rafteH-and` رَفْتِه اَنْدْ *rafteh-and*;

`mard-Id` مَرْدِیدْ *mard-īd*, `asb-etAn` اَسْبِتَانْ *asb-etān*;

`An^gA-st` آنْجَاسْتْ *ānǧā-st*, `U-st` اُوسْتْ *ū-st*, `t_U-st` تُوسْتْ *tu-st*;

`ketAb-I-st` کِتَابِیسْتْ *ketāb-ī-st*, `nAmeH-I-st` نَامِه اِیسْتْ *nāmeh-ī-st*.

The preposition *be-* can be written with or without a hyphen:

`be-man` بِمَنْ *be-man*, `be-t_U` بِتُو *be-tu*;

`be-An` بِآنْ *be-ān*, `be-In` بِاِینْ *be-īn*, `beU` بِاُو *beū*.

The components of compounds can be separated by `\,`, or `"|`:

`.sA.heb\,_hAneH` صَاحِب خَانِه *ṣāḥebḫāneh*,

`ta_ht-e-"|_hwAb` تَخْتِ خوَابْ *taḫt-e-ḫāb*;

`pas\,andAz` پَس اَنْدَازْ *pasandāz*, `naw"|AmUz` نَو آمُوزْ *nawāmūz*,
`bI\,_hwod` بِی خوُدْ *bīḫod*.

Digit sequences are written in their natural order:

`1234567890` ۱۲۳۴۵۶۷۸۹۰ *1234567890*

6.2 Maghribi

This works nearly like Arabic, but using a different writing convention. *fā'* is written with one dot below the letter, *qāf* with one dot above the normal letter form of *fā'*. The three dots of *vā'* are put below the letter.

Switch to this mode by `\setmaghribi`.

6.3 Urdu

The Urdu mode is activated by `\seturdu`.

- For Urdu, additional codings are available, see Table 6.1. Some of the given codings also occur in Pashto but with a different meaning, see Section 6.4.
- Urdu uses the Nasta'liq font if available, otherwise Naskh.

6.3.1 Coding examples for Urdū[2]

The short vowels *ă*, *ĭ*, **and** *ŭ* are encoded by the lowercase letters `a`, `i`, and `u`, and are marked respectively by *zabar*, *zēr*, and *pēš*:

`par` پَر *par* , `dam` دَم *dam* / `fir` فِر *fir* , `din` دِن *din* /

`sukh` سُکھ *sukh* , `dukh` دُکھ *dukh*

The long vowels *ā*, *ī*, *ū*, *ē*, **and** *ō* are encoded by the capital letters `A`, `I`, `U`, `E`, and `O`:

∘ Note: *'alif madda* is automatically generated for word-initial *ā*:

`Ap` آپ *āp* / `Am` آم *ām*

`tIn` تِین *tīn* , `la,rkI` لَڑکِی *laṙkī* / `dUr` دُور *dūr* / `dEr` دیر *dēr* ,

`ba,rE` بَڑے *baṙē* / `mOr` مور *mōr*

∘ Note: `I` yields a *ya-ye ma'ruf* (with *zēr*), while `E` yields a *ya-ye mağhul* (without *zēr*).

`tIn` تِین *tīn* , `rItI` رِیتِی *rītī* / `mErE` میرے *mērē* , `la,rkE` لَڑکے *laṙkē*

The diphthongs *ae* **and** *ao* are encoded `ae`, and `ao`:

`kaesA` کَیسَا *kaesā* / `paodA` پَودَا *paodā*

∘ Note: `U` yields a *wāw-e-ma'ruf* (with *pēš*), while `O` yields a *wāw-e-mağhul* (without *pēš*), and `ao` is indicated by a *zabar*. Compare:

`pUr` پُور *pūr* / `pOtA` پوتَا *pōtā* / `paodA` پَودَا *paodā*

Intervocalic *ḥamza* is written `'`:

`^cA'E` چَائے *čā'ē* / `ma'I` مَئِی *ma'ī* / `kO'I` کوئِی *kō'ī*

Aspiration is produced by coding a `h` after the consonant to be aspirated. Aspiration in Urdū is produced by adding *dō čašmī he* after the consonant:

`khEt` کھیت *khēt* / `ghar` گھَر *ghar* / `mujhE` مُجھے *muğhē* / `dharm` دھَرم *dharm*

Nasalization is indicated by *nūn-e-ġunnah* coded as `.n`. Note that the *nuqṭa* for *nūn* is not written when it is used to represent nasalization:

`mae.n` مَیں *maeṇ* / `a,hi.nsA` اَہِنسَا *ahiṇsā*

[2] **Please contact Anshuman Pandey [*apandey@u.washington.edu*] with questions or comments regarding this section.**

URDŪ	NAME	CODE
اَ *a*	*'alif*	a
ب *b*	*be*	b
بھ *bh*	*bhe*	bh
پ *p*	*pe*	p
پھ *ph*	*phe*	ph
ت *t*	*te*	t
تھ *th*	*the*	th
ٹ *ť*	*ťe*	,t
ٹھ *ťh*	*ťhe*	,th
ث *t̲*	*t̲e*	_t
ج *ǧ*	*ǧīm*	j / ^g
جھ *ǧh*	*ǧhīm*	jh / ^gh
چ *č*	*če*	^c
چھ *čh*	*čhe*	^ch
ح *ḥ*	*baŕī he / ḥe*	.h
خ *ẖ*	*ẖe / k̲he*	_h
د *d*	*dāl*	d
دھ *dh*	*dhāl*	dh
ڈ *ď*	*ďāl*	,d
ڈھ *ďh*	*ďhāl*	,dh
ذ *d̲*	*d̲āl*	_d
ر *r*	*re*	r
رھ *rh*	*rhe*	rh
ڑ *ŕ*	*ŕe*	,r
ڑھ *ŕh*	*ŕhe*	,rh
ز *z*	*ze*	z
ژ *ž*	*že*	^z
س *s*	*sīn*	s
ش *š*	*šīn*	^s
ص *ṣ*	*ṣād*	.s
ض *ḍ*	*ḍād*	.d
ط *ṭ*	*ṭoi*	.t
ظ *ẓ*	*ẓoi*	.z

URDŪ	NAME	CODE
ع ‘	*‘ain*	‘
غ *ġ*	*ġain*	.g
ف *f*	*fe*	f
ق *q*	*qāf*	q
ک *k*	*kāf*	k
کھ *kh*	*khāf*	kh
گ *g*	*gāf*	g
گھ *gh*	*ghāf*	gh
ل *l*	*lām*	l
لھ *lh*	*lhām*	lh
م *m*	*mīm*	m
مھ *mh*	*mhīm*	mh
ن *n*	*nūn*	n
نھ *nh*	*nhūn*	nh
و *w*	*wāw*	w / U, O, ao
ہ *h*	*čhōťī he*	,h
ھ *h*	*dō čašmī he*	h
ی *y*	*čhōťī ye*	y / I, E
ے *-ē*	*baŕī ye*	E / ae
ں *ṇ*	*nūn-e-ġunnah*	.n
ء ’	*ḥamza*	’
ۃ *t*	*te marbuṭa*	T
ـَ *a*	*a*	a
ـا *ā*	*ā*	A
ـِ *i*	*i*	i
ـِی *ī*	*ī*	I
ـُ *u*	*u*	u
ـُو *ū*	*ū*	U
ـے *ē*	*ē*	E
ـَے *ae*	*ae*	ae
ـو *ō*	*ō*	O
ـَو *ao*	*ao*	ao

Table 6.1: The Urdu Alphabet

The *"hanging he"* **or** *Ha-ye hawwaz-e-mahfī* is generated by `H`. It does not receive a *jazm* even in fully vocalised mode and is not joined to a following letter:

`rAjaH` رَاجَہ *rāğah* / `kiH` کِہ *kih* / `naH` نَہ *nah* / `waH` وَہ *wah* /

`nAmaH` نَامَہ *nāmah* / `yaH` یَہ *yah*

Tanwīn is coded by `aN`:

`taqrIbaN` تَقرِیباً *taqrīban* / `faoraN` فَوراً *faoran*

Tašdīd is produced by coding the consonant twice:

`mi,t,tI` مِٹّی *miṭṭī* / `unnIs` اُنّیس *unnīs*

○ Note that double consonants in Urdū verbs are written without *tašdīd*. *Tašdīd* is disabled by inserting `|` between the double consonants:

`ban|nA` بَننَا *bannā* / `gin|nA` گِننَا *ginnā* / `jAn|nA` جَاننَا *ğānnā*

Iḍāfat is written `-e` or `-i`:

`^sEr-e panjAb` شیرِ پَنجَاب *šēr-e panğāb*

`Ab-e .hayAt` آبِ حَیَات *āb-e ḥayāt*

Wāw-e ma'dūla, or the "*wāw* which is passed over", is written `w`; it is omitted in the transliteration and the preceding *ḫe* receives no *jazm*:

`_hwAb` خوَاب *ḫāb* / `_hwAja` خوَاجَ *ḫāğa* / `_hwud` خوُد *ḫud*

Wāw-e 'aṭf, or the "*wāw* of conjunction" is coded as `-O`

`gul-O bulbul` گُلو بُلبُل *gul-ō bulbul*

`sarw-O sanObar` سَروو سَنوبَر *sarw-ō sanōbar*

`tar-O tAzaH` تَرو تَازَہ *tar-ō tāzah*

'Alif maqṣūra is encoded as `_A` or `Y`:

`fatw_A` فَتوَی *fatwā* / `da'wY` دَعوَی *daʿwā*

Digit sequences are written in the natural order:

`0123456789` ۰۱۲۳۴۵۶۷۸۹ *0123456789*

,t	ټ	*ṫ*	*tā'* with a small loop
,d	ډ	*ḋ*	*dāl* with a small loop
,r	ړ	*ṙ*	*rā'* with a small loop
.n	ڼ	*ṇ*	*nūn* with a small loop
g	ګ	*g*	*gāf* with a small loop instead of a bar
,z	ږ	*ź*	*rā'* with one dot above and one below
,s	ښ	*ś*	*sīn* with one dot above and one below
ae	ېَ	*ae*	the diphtong *ae*
Ee	ېـ	*ey*	the diphtong *ey*
ee	ېَ	*ey*	the diphtong *ey*
E	ېِ	*ē*	the long vowel *ē*
O	ۆ	*ō*	the long vowel *ō*
U	ۇ	*ū*	the long vowel *ū*

Table 6.2: Additional codings for Pashto.

6.4 Pashto (Afghanic)

Switch to this mode by `\setpashto`. For writing some Pashto words in the Urdu style, write the command `\seturdu` and afterwards switch back.

- For Pashto, additional codings are available, see Table 6.2. Some of the given codings also occur in Urdu but with a different meaning.
- The codings `<H>`, `<,a>` and `<,e>` are used as in Persian. The rules for *iẓāfet* and *yā'-i-waḥdat* apply.
- The short vowel `<e>` is indicated by a *zwarakay*, `<o>` by an inverted *ḍamma*. Observe also the following codings:

  `<w"'>` ﺅ *hamza* on *wāw*

  `<h"'>` ۂ *hamza* on *hā'*, if not generated by *iẓāfet*

6.5 Sindhi

a	ا	*a*	~n	ڃ	*ñ*	z	ز	*z*	kh	ک	*kh*
b	ب	*b*	^c	چ	*č*	s	س	*s*	g	گ	*g*
:b	ٻ	*ḅ*	^ch	ڇ	*čh*	^s	ش	*š*	:g	ڳ	*g̣*
bh	ڀ	*bh*	.h	ح	*ḥ*	.s	ص	*ṣ*	gh	گھ	*gh*
t	ت	*t*	_h	خ	*ẖ*	.d	ض	*ḍ*	:n	ڱ	*n̤*
th	ٿ	*th*	d	د	*d*	.t	ط	*ṭ*	l	ل	*l*
,t	ٽ	*t́*	dh	ڏ	*dh*	.z	ظ	*ẓ*	m	م	*m*
,th	ٺ	*t́h*	:d	ڊ	*d̤*	‘	ع	‘	n	ن	*n*
_s	ث	*t̲*	,d	ڌ	*d́*	.g	غ	*ġ*	,n	ڻ	*ń*
p	پ	*p*	,dh	ڍ	*d́h*	f	ف	*f*	w	و	*w*
j	ج	*ǧ*	_d	ذ	*d̲*	ph	ق	*ph*	,h	ه	*h*
:j	ڄ	*j̈*	r	ر	*r*	q	ق	*q*	h	ھ	*h*
jh	ڃھ	*ǧh*	,r	ڙ	*ŕ*	k	ک	*k*	y	ي	*y*
a	ـَ	*a*	e	ـِ	*e*	i	ـِ	*i*	o	ـُ	*o*
u	ـُ	*u*	A	ـا	*ā*	E	ـيـ	*ē*	I	ـِي	*ī*
O	ـو	*ō*	U	ـُو	*ū*	ae	ـَيـ	*ae*	ao	ـَو	*ao*
i	ـِ	*i*	_A	ـَى	*ā*	’A	آ	ʾ*ā*	’a	أ	ʾ*a*
’i	إ	ʾ*i*	’y	ـئ	ʾ*y*	’w	ؤ	ʾ*w*	’|	ء	ʾ

Table 6.3: The Sindhi Alphabet

To activate the Sindhi mode, select the language by `\setsindhi`. Sindhi input texts are encoded in a modification of the standard ArabTEX encoding. The alphabet is given in Table 6.3 on page 49.

- Use hyphens to resolve ambiguities with aspired consonants.
- There are two special codings: `\MIN` ـٖ, `\IN` ـٕ .
- The user might want to break some ligatures by inserting a vertical bar to get the correct writing, or just for a better appearance of the script.

6.6 Kashmiri

a	اَ	a	d	د	d	.d	ض	$\dot{z}$	m	م	m
b	ب	b	,d	ڈ	$\dot{d}$	.t	ط	$\underline{t}$	n	ن	n
p	پ	p	_d	ذ	$\underline{z}$	.z	ظ	$\dot{z}$	w	و	w
t	ت	t	r	ر	r	‘	ع	‘	,h	ہ	h
,t	ٹ	$\dot{t}$	,r	ڑ	$\dot{r}$	.g	غ	$\underline{gh}$	y	ی	y
_t	ث	$\underline{s}$	z	ز	z	f	ف	f	h	ھ	h
j	ج	j	^z	ژ	ts	q	ق	q	E	ے	$\bar{e}$
^c	چ	c	s	س	s	k	ک	k	’	ء	’
.h	ح	$\dot{h}$	^s	ش	$\acute{s}$	g	گ	g	T	ۃ	t
_h	خ	$\underline{kh}$	.s	ص	$\dot{s}$	l	ل	l	.y	ۄ	$\dot{y}$
a	ـَ	a	i	ـِ	i	u	ـُ	u	.o	ۅ	$\dot{o}$
A	ـا	$\bar{a}$	I	ـِیـ	$\bar{\imath}$	U	ـُوٗ	$\bar{u}$	.O	ۅاَ	$\bar{\dot{o}}$
.a	ـٔ	$\dot{a}$	.u	ـٕ	$u’$	o	ـوٗ	o	e	ـیٖ	e
.A	ـأ	$\bar{\dot{a}}$	.U	ـٕیـ	$\bar{u}’$	O	ـو	$\bar{o}$	E	ـیـ	$\bar{e}$

Table 6.4: The Kashmiri Alphabet

Select Kashmiri by `\setkashmiri`. The input codes are given in Table 6.4. The transcription follows the ALA-LC romanization conventions.

6.7 Uighuric

Switch to this mode by `\setuighur`.

Uighuric input texts are encoded in a modification of the standard ArabTEX encoding, see column 5 of Table 6.5. Please observe that in Uighuric all characters are coded verbatim.

	1	2	3	4 = 5	(6) 7
01			ـا	ا = a	(01) a
02			ـه	ه = :a	(02) ä
03			ـد	د = d	(09) de
04			ـر	ر = r	(10) re
05			ـز	ز = z	(11) ze
06			ـژ	ژ = ^z	(12) že
07			ـو	و = o	(25) o
08			ـۆ	ۆ = :o	(27) ö
09			ـۇ	ۇ = u	(26) u
10			ـۈ	ۈ = :u	(28) ü
11			ـۋ	ۋ = w	(29) we
12	بـ	ـبـ	ـب	ب = b	(03) be
13	پـ	ـپـ	ـپ	پ = p	(04) pe
14	تـ	ـتـ	ـت	ت = t	(05) te
15	نـ	ـنـ	ـن	ن = n	(23) ne
16	جـ	ـجـ	ـج	ج = j	(06) je
17	چـ	ـچـ	ـچ	چ = ^c	(07) če

	1	2	3	4 = 5	(6) 7
18	خـ	ـخـ	ـخ	خ = x	(08) xe
19	ئـ	ـئـ	ـئ	ئ = i	(31) i
20	ېـ	ـېـ	ـې	ې = e	(30) e
21	يـ	ـيـ	ـي	ي = y	(32) y
22	سـ	ـسـ	ـس	س = s	(13) se
23	شـ	ـشـ	ـش	ش = ^s	(14) še
24	غـ	ـغـ	ـغ	غ = ^g	(15) ğe
25	فـ	ـفـ	ـف	ف = f	(16) fe
26	قـ	ـقـ	ـق	ق = q	(17) qe
27	كـ	ـكـ	ـك	ك = k	(18) ke
28	ڭـ	ـڭـ	ـڭ	ڭ = ^n	(20) ηe
29	گـ	ـگـ	ـگ	گ = g	(19) ge
30	لـ	ـلـ	ـل	ل = l	(21) le
31	مـ	ـمـ	ـم	م = m	(22) me
32	هـ	ـهـ	ـھ	ھ = h	(24) he
33	أـ	ـءـ		= ’	( )
34			ـلا	لا = la	( )

1. initial shape
2. medial shape
3. final shape
4. isolated shape
5. external encoding
6. sorting position
7. name

Table 6.5: ArabTEX encoding of Uighuric

6.8 Old Malay

The preliminary ArabTEX language mode `\setmalay` is provided for processing Old Malay texts in the extended Arabic writing.

Old Malay (Jawi) input texts are encoded in a modification of the standard ArabTEX encoding, see below.

Additional encodings:

p	ڤ	ng	غ
g	ک	ny	ث
v	ۏ	c	چ

This language mode is strictly experimental and expected to contain many errors; it will be adapted to the users' requirements. Please report your experience and suggestions for changes and improvements to the author.

6.9 Other extensions of the Perso-Arabic script

This is up to experimentation by the user. If `\setarab` or `\setfarsi` will not produce the desired result, try `\setverb` for verbatim mode.

The vowelization and the transliteration cannot generally be expected to be correct, but might work by accident.

In case some character variants not yet provided are needed, feel free to ask the author for help. There is no simple way for the user to modify the script directly.

Chapter 7

Hebrew mode

On the request of some users, starting with Version 3.02 ArabTEX has been extended by some modules adding support for Hebrew. Whereas the initial applications only called for short Hebrew quotations within Roman texts, possibly containing Arabic insertions too, adding "Hebrew environments" proved comparatively easy. We also added most commands provided by the HebrewTEX package (an alternative TEX extension developed in Israel, that requires TEX--XET). The Hebrew date quite probably will *not* work correctly.

To process Hebrew input with ArabTEX, proceed as follows:

- for use with Plain TEX, say `\input hebtex`; a small loader module will load both ArabTEX *and* the Hebrew extension.
- with LaTeX2$_\varepsilon$ say: `\usepackage{hebtex}`.

The extension provides a language mode `\sethebrew`, and several common encodings of texts in Hebrew, that may be switched by the `\setcode` command. One (nameless) encoding is compatible with Dov Grobgeld's editor HED, so files prepared for HebrewTEX are supposed to be compatible. In addition, the standard ArabTEX encoding has been extended to cater for Hebrew too.

7.1 Language switching

`\sethebrew` switches to Hebrew mode, `\setarab` back to Arabic.

Remember to switch the encoding and the vowelization mode too!

7.2 Standard Hebrew encoding

`\setcode{standard}` or `\setcode{arabtex}` will switch to the standard ArabTEX Hebrew encoding, defining the consonants as follows:

’	א	aleph	b	ב	beth	g	ג	gimel	d	ד	daleth
h	ה	heh	w	ו	waw	z	ז	zayin	_h	ח	chet
_t	ט	teth	y	י	yod	k	כ	kaph	l	ל	lamed
m	מ	mem	n	נ	nun	s	ס	samekh	‘	ע	ayin
p	פ	peh	.s	צ	sade	q	ק	qof	r	ר	resh
,s	שׂ	sin	^s	שׁ	shin	S	ש	s(h)in	t	ת	taw

Note: without punctuation, the characters *sin*, *shin* and *s(h)in* look identical; otherwise *sin* שׂ has a dot to the left, *shin* שׁ has a dot to the right, *s(h)in* ש is the form without a dot.

There are alternative encodings for soft consonants: `<v>` for `<b>`, `<f>` for `<p>`.

Vowels are encoded as follows:

short vowels			long vowels			defective			half vowels		
a	◌ַ	pathach	A	◌ָ	qames				.a	◌ֲ	chateph patach
e	◌ֶ	segol	E	◌ֵי	sere yod	_e	◌ֵ	sere	.e	◌ֱ	chateph segol
i	◌ִ	chireq	I	◌ִי	chireq yod				.i	◌ְ	shewa
o	◌ָ	qames chatuph	O	◌ֹו	cholem waw	_o	◌ֹ	cholem	.o	◌ֳ	chateph qames
u	◌ֻ	qibbus	U	◌ּו	shureq				.u		no vowel mark

The *matres lectionis* can also be written explicitly, e.g., `<_ey>` for `<E>`, `<iy>` for `<I>`, `<_ow>` for `<O>`.

- `\vocalize` (default) activates vowel points and special punctuation; `\novocalize` switches them off again.
- *patach furtivum* is written `<.a>` *before* its carrier: `<rU.a_h>` רוּחַ.
- *dagesh lene* with `<b g d k p t>` and *mappiq* with `<h>` is expressed by prefixing a dot: `<.b>`, `<.g>`, `<.d>`, `<.k>`, `<.p>`, `<.t>`; `<.h>`

- *dagesh forte* is expressed by doubling the consonant; thus two equal consonants in sequence (even in `\novocalize` mode) must be separated by some short vowel indicator (or `<.u>`), if the standard encoding is used.
- *dagesh orthophonicum* is coded like *dagesh forte*.
- *meteg* is indicated by `<|>` after the vowel.
- *maqqef* is `<-->` (en-dash; a single hyphen will be ignored)
- Prefixes may be separated by a single hyphen, which appears in the transcription without changing the Hebrew writing.
- For those rare cases where a consonant is missing, input `<|">` (bar quote); this may also carry vowels.
- *raphe*, accents, and cantillation marks are not supported.

Abbreviations may not be used in this mode as we know of no obvious way of denotating them. Suggestions are welcome.

7.3 ISO 8859-8 and Hebrew MS-Windows

ISO 8859-8 is an 8-bit encoding extending 7-bit ASCII with Hebrew letters. Within Hebrew MS-Windows, the code page CP 1255 provides a superset of ISO 8859-8, containing a full complement of Hebrew vowels, and a host of miscellaneous special characters.

ArabTeX provides within the package `cp1255.sty` a reading module for a subset of CP1255, containing all Hebrew characters from ISO 8859-8 and all Hebrew vowels of CP1255, but omitting the extra special characters (see Table 7.1). This encoding is activated by the switching command `\setcode{cp1255}` or `\setcode{hwin}`.

7.4 HED, PC "oldcode" and "newcode"

There is a default reading module for the Hebrew characters in code positions $96\cdots122$ (HebrewTEX "pccode"), in code positions $128\cdots154$ as generated by the editor HED, and also in code positions $224\cdots250$ (HebrewTEX "newcode", ISO 8859-8). In fact these are three different encodings catered for by a single reading module; see the code assignments in Table 7.2. Observe that the Hebrew characters in ISO 8859-8 are supported both by this reading module and by the Hebrew MS-Windows encoding mentioned in Section 7.3. The code switching commands `\setcode{hed}`, `\setcode{newcode}`, `\setcode{pccode}`, and `\setcode{iso8859-8}` all activate the default reading module.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	‘	p			NSP		◌ִ	|	א	ן
01	SOH	DC1	!	1	A	Q	a	q					◌ֱ	◌ֹ	ב	ס
02	STX	DC2	”	2	B	R	b	r					◌ֲ	◌ֺ	ג	ע
03	ETX	DC3	#	3	C	S	c	s					◌ֳ	:	ד	ף
04	EOT	DC4	$	4	D	T	d	t			א		◌ִ	◌ֻ	ה	פ
05	ENQ	NAK	%	5	E	U	e	u					◌ֵ	וֹ	ו	ץ
06	ACK	SYN	&	6	F	V	f	v					◌ֶ	◌ֽ	ז	צ
07	BEL	ETB	’	7	G	W	g	w					◌ַ	◌ֿ	ח	ק
08	BS	CAN	(	8	H	X	h	x					◌ָ		ט	ר
09	HT	EM	)	9	I	Y	i	y					◌ׁ		י	ש
10	LF	SUB	∗	:	J	Z	j	z					•		ך	ת
11	VT	ESC	+	;	K	[	k	{					◌ׂ		כ	
12	FF	IS4	,	<	L	\	l	|					◌ׄ		ל	
13	CR	IS3	−	=	M	]	m	}			SHY		◌ׅ		ם	LRO
14	SO	IS2	.	>	N	^	n	~					◌ׇ		מ	RLO
15	SI	IS1	/	?	O	_	o	DEL					◌ּ		ן	

Table 7.1: ISO 8859-8 and Windows CP 1255 code table

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	א	נ	א	נ					א	נ
01	SOH	DC1	!	1	A	Q	ב	ס	ב	ס					ב	ס
02	STX	DC2	”	2	B	R	ג	ע	ג	ע					ג	ע
03	ETX	DC3	#	3	C	S	ד	ף	ד	ף					ד	ף
04	EOT	DC4	$	4	D	T	ה	פ	ה	פ					ה	פ
05	ENQ	NAK	%	5	E	U	ו	ץ	ו	ץ					ו	ץ
06	ACK	SYN	&	6	F	V	ז	צ	ז	צ					ז	צ
07	BEL	ETB	’	7	G	W	ח	ק	ח	ק					ח	ק
08	BS	CAN	)	8	H	X	ט	ר	ט	ר					ט	ר
09	HT	EM	(	9	I	Y	י	ש	י	ש					י	ש
10	LF	SUB	*	:	J	Z	ך	ת	ך	ת					ך	ת
11	VT	ESC	+	;	K	]	כ	}	כ						כ	
12	FF	IS4	,	>	L	\	ל	|	ל					ל	
13	CR	IS3	−	=	M	[	ם	{	ם						ם	
14	SO	IS2	.	<	N	^	מ	~	מ						מ	
15	SI	IS1	/	?	O	_	ן		ן						ן	DEL

Table 7.2: HED, CP 1255 and ISO 8859-8 code table

In this encoding vowel points, *dagesh* and *meteg* cannot be used, as they cannot be represented in the input. Abbreviations may be expressed by a single or double apostrophe (right quote). The final and the medial forms of characters are equivalent; ArabTEX will choose the appropriate shape automatically.

7.5 BHS encoding

a"MT"b"001"c"Gen"x1

1 [1] בְּרֵאשִׁית בָּרָא אֱלֹהִים אֵת הַשָּׁמַיִם וְאֵת הָאָרֶץ׃ [2] וְהָאָרֶץ
הָיְתָה תֹהוּ וָבֹהוּ וְחֹשֶׁךְ עַל־פְּנֵי תְהוֹם וְרוּחַ אֱלֹהִים מְרַחֶפֶת עַל־פְּנֵי
הַמָּיִם׃ [3] וַיֹּאמֶר אֱלֹהִים יְהִי אוֹר וַיְהִי־אוֹר׃ [4] וַיַּרְא אֱלֹהִים אֶת־
הָאוֹר כִּי־טוֹב וַיַּבְדֵּל אֱלֹהִים בֵּין הָאוֹר וּבֵין הַחֹשֶׁךְ׃ [5] וַיִּקְרָא
אֱלֹהִים | לָאוֹר יוֹם וְלַחֹשֶׁךְ קָרָא לָיְלָה וַיְהִי־עֶרֶב וַיְהִי־בֹקֶר יוֹם אֶחָד׃
פ [6] וַיֹּאמֶר אֱלֹהִים יְהִי רָקִיעַ בְּתוֹךְ הַמָּיִם וִיהִי מַבְדִּיל
בֵּין מַיִם לָמָיִם׃ [7] וַיַּעַשׂ אֱלֹהִים אֶת־הָרָקִיעַ וַיַּבְדֵּל בֵּין הַמַּיִם אֲשֶׁר
מִתַּחַת לָרָקִיעַ וּבֵין הַמַּיִם אֲשֶׁר מֵעַל לָרָקִיעַ וַיְהִי־כֵן׃ [8] וַיִּקְרָא
אֱלֹהִים לָרָקִיעַ שָׁמָיִם וַיְהִי־עֶרֶב וַיְהִי־בֹקֶר יוֹם שֵׁנִי׃ פ
[9] וַיֹּאמֶר אֱלֹהִים יִקָּווּ הַמַּיִם מִתַּחַת הַשָּׁמַיִם אֶל־מָקוֹם אֶחָד וְתֵרָאֶה
הַיַּבָּשָׁה וַיְהִי־כֵן׃ [10] וַיִּקְרָא אֱלֹהִים | לַיַּבָּשָׁה אֶרֶץ וּלְמִקְוֵה הַמַּיִם
קָרָא יַמִּים וַיַּרְא אֱלֹהִים כִּי־טוֹב׃ [11] וַיֹּאמֶר אֱלֹהִים תַּדְשֵׁא הָאָרֶץ
דֶּשֶׁא עֵשֶׂב מַזְרִיעַ זֶרַע עֵץ פְּרִי עֹשֶׂה פְּרִי לְמִינוֹ אֲשֶׁר זַרְעוֹ־בוֹ
עַל־הָאָרֶץ וַיְהִי־כֵן׃ [12] וַתּוֹצֵא הָאָרֶץ דֶּשֶׁא עֵשֶׂב מַזְרִיעַ זֶרַע לְמִינֵהוּ
וְעֵץ עֹשֶׂה־פְּרִי אֲשֶׁר זַרְעוֹ־בוֹ לְמִינֵהוּ וַיַּרְא אֱלֹהִים כִּי־טוֹב׃ [13] וַיְהִי־
עֶרֶב וַיְהִי־בֹקֶר יוֹם שְׁלִישִׁי׃ פ [14] וַיֹּאמֶר אֱלֹהִים יְהִי מְאֹרֹת
בִּרְקִיעַ הַשָּׁמַיִם לְהַבְדִּיל בֵּין הַיּוֹם וּבֵין הַלָּיְלָה וְהָיוּ לְאֹתֹת וּלְמוֹעֲדִים
וּלְיָמִים וְשָׁנִים׃

Figure 7.1: Hebrew example

The package `bhs.sty` provides support for the encoding that is used in the machine-readable version of BHS (Biblia Hebraica Stuttgartensia). After loading the package, you can switch to this encoding by the command `\setcode{bhs}`.

The line-breaks of the source are (usually) respected. BHS line numbers and comments are only partially supported. For an example, see Figure 7.1.

`\setcode{witbhs}` switches to a variant of the BHS encoding that was developed by the Werkgroep Informatica of the Vrije Universiteit Amsterdam. Activate it by `\usepackage{witbhs}`.

7.6 UNICODE Hebrew

The file `utf8.sty` contains a reading module for the Arabic and the Hebrew segment of UNICODE in UTF-8 encoding. It is installed by the LaTeX command `\usepackage{utf8}` or by `\input utf8.sty`.

UTF-8 (UNICODE Transmission Format, see table 7.3) is a multi-byte encoding which, for Arabic and Hebrew, uses two bytes per character whereas ASCII characters use a single byte. Far-eastern languages are encoded in three bytes per character. This is in contrast to UNICODE itself which always uses two bytes per character.

The module is activated by `\setcode{utf8}`; all following Arabic and Hebrew text will be considered to be coded according to the UTF-8 encoding standard. To use the correct font, select the appropriate language. The ArabTeX notation may be reactivated by `\setcode{arabtex}`.

7.7 Hebrew fonts

- As a default, the fonts `"hclassic"` (default) and `"hcaption"` are distributed with ArabTeX. Switch to `"hcaption"` by `\hp`, and back by `\hc`.

  These fonts have been designed and donated by Joel Hoffman, who also wrote several macro packages from which we took a few ideas for positioning punctuation. There is a variety of other usable fonts on the CTAN archives.

- If no vowel points are required, all the standard fonts `"DeadSea"`, `"OldJaffa"`, `"TelAviv"`, and `"Jerusalem"` can be used if locally available. They are activated by the commands `\ds`, `\oj`, `\ta`, `\jm`; the command `\hc` switches back to the default `"hclassic"` font.

- The `"Shalom"` family of fonts, if available, can be activated by `\shlmold`, `\shlmscr`, and `\shlmstk`. Their vowel points presently do *not* work since they must be handled differently from the ArabTeX strategy.

- In case a font appears to be locally available but is not found, check and, if required, correct the exact spelling of the font name within the file `"uheb.fd"`. We have seen various incompatible variants on CTAN and on the InterNet.

	058	059	05A	05B	05C	05D	05E	05F
0				◌ֹ	׀	א		וו
1				◌ֻ	◌ׁ	ב	ס	וי
2				◌ּ	◌ׂ	ג	ע	יי
3				◌ֽ	׃	ד	ף	׳
4				◌ֿ	◌ׄ	ה	פ	״
5				◌ׁ		ו	ץ	
6				◌ׂ		ז	צ	
7				◌ׄ		ח	ק	
8				◌ׅ		ט	ר	
9				◌ׇ		י	ש	
A				•		ך	ת	
B				◌ׇ		כ		
C				◌ׇ		ל		
D				◌ׇ		ם		
E				◌ׇ		מ		
F				◌ׇ		ן		

Table 7.3: UNICODE Hebrew

- Activating other Hebrew fonts by the command `\sethebfont{`*font*`}` might work.

Note: We recommend to set `\lineskiplimit -20pt` whenever Hebrew and Roman script are used within the same paragraph; this will lead to uniform line spacing. The value of `\baselineskip` may have to be adjusted.

7.8 Hebrew transcription systems

`\transtrue` activates the standard ZDMG transcription, and there are provisions for additional transcription systems:

- `\settrans{zaw}` switches to the conventions of "Zeitschrift für die Alttestamentliche Wissenschaft" (recommended);
- `\settrans{gesenius}` activates the system used in W. Gesenius' Hebrew Grammar, 26th edition (deprecated).
- `\settrans{standard}` restores the standard ZDMG transcription.

Chapter 8

Miscellaneous features

`.k`	ك·	*.k*	*kāf* in the final position without a mark
`^d`	آد	*d*	*dāl* with a dot below
`.f`	ف·	*.f*	*fā'* without a dot
`.b`	ب·	*.b*	*bā'* without a dot
`.n`	ن	*.n*	*nūn* without a dot (not available in Pashto mode)
`Y`	ى	*ā*	*'alif maqṣūra*; *yā'* without dots in all positions

Table 8.1: Additional codings for special purposes.

8.1 Additional codings

To reproduce exotic, erroneous or archaic texts exactly as they are written, some additional codings are available, see Table 8.1.

If further variants are needed, write to the author and indicate:

- the required shape,
- the assumed transliteration,
- a suggestion for the input coding,
- some information on the intended use.

We are willing to consider any suggestion. Adding a new character might be easy, or else it might be outright impossible. ArabTEX is rather flexible, but there are also some technical limitations.

8.2 Dots on *yā'*

Whether *yā'* in the final position carries dots or not is controlled by the chosen language convention. You can override this, after selecting the language, by `\yahdots` and `\yahnodots`.

8.3 Vowel positioning

In vowelized Arabic text, the short vowel marks are by default positioned close to the basic character glyphs. If this is not wanted, they may be raised to approximately uniform height by the command `\accentshigh`. You may revert to the default strategy by `\accentslow`.

8.4 Abjad numerals

The command `\abjad {#1}` will convert its argument, which has to be a legal representation of a number between 1 and 1999, to the Arabic *'abjad* notation used in some mediaeval manuscripts. The result of the conversion will not look perfect, and the legal *'abjad* number 0 can presently not be generated. The command `\abjad{#1}` can be used inside and outside of an *Arabic context.*

This routine profited greatly from suggestions by Dr. Benno van Dalen (Utrecht University).

8.5 Automatic stretching

For special purposes, e.g. for headlines and for Arabic paragraphs containing long mathematical or non-Arabic insertions, the connection between adjacent Arabic letters may be made "elastic", if they form no ligature. Thus a *kašīda* is inserted whose length will be adjusted automatically to uniformly fill the output line.

This feature increases the already high storage demands of ArabTEX, and should therefore be used sparingly. It can be switched on with `\spreadtrue` and switched off again with `\spreadfalse`. Inside an *Arabic Environment,* it will also be switched off automatically at the end of every paragraph.

8.6 Uniform baselines

The Arabic and Hebrew fonts are optically compatible with the standard Roman fonts, but have larger ascenders and descenders; this will lead to unequal distances between the baselines of consecutive lines, especially if Roman and non-Roman text are mixed within the same Roman paragraph.

Typesetting on a grid will improve line spacing. We recommend to set `\lineskiplimit -20pt` whenever Roman script and Arabic and/or Hebrew are used within the same Roman paragraph. The value of `\baselineskip` may have to be adjusted; with LaTeX use `\baselinestretch`.

Also within an *Arabic environment* typesetting on a grid may lead to a better result.

8.7 Verbatim copy of the input

For testing purposes, the Arabic input may be reproduced verbatim after `\showtrue` in addition to the normal output; `\showfalse` switches this feature off again. Commands will not usually be shown. The output will generally not look pleasant, and this feature is only provided in order to trace down errors, or to demonstrate the operation of ArabTeX as in the examples above.

8.8 Progress report

Since ArabTeX is still rather slow (due to evolving technology it is getting faster every year), it will produce some terminal output while running to indicate it is still alive. If that is not wanted, e.g., on a very fast computer system, or while running a batch job, say `\quiet` or `\tracingarab = 0` (outside an *Arabic Environment*; otherwise say `\doassign {\tracingarab }{0}`). The setting `\tracingarab = 1` will only report Arabic paragraphs, a value of 2: Arabic lines and insertions, a value of 3 or more: individual *Arabic items*.

8.9 Module Reporting

A complete list of the modules loaded in a particular run will be put into the TeX log file (before the run statistics), if LaTeX is used. This is believed to be useful when tracing down errors. This list is also available to the user, even with Plain TeX, as the contents of the control sequence `\arabtexconfig`.

Chapter 9

Compatibility issues

ArabTEX relies only on part of the powerful features of the TEX typesetting engine (neither mathematical mode nor the alignment mechanism are used), and few of the features provided by the Plain TEX package and none of LATEX are required, but may be necessary in other parts of a multi-lingual document. Of course, TEX's macro processor is very heavily used.

It turned out that ArabTEX could be made to cooperate with a number of other macro packages, sometimes after some adjustments to ArabTEX when detecting the presence of another system, and sometimes by compatible adjustments to the other system. However there are some problem areas:

- The resource requirements of ArabTEX and usually also of the other packages are very high, and might reach the limits of a small TEX system. Fortunately, nowadays very large TEX implementations are available.
- The running time is not negligible (however, computers are still becoming faster, and typesetting this very document takes only about 20 seconds on a Pentium 233 PC running emTEX).
- Tracking down errors in a combination of several large macro packages might be difficult and time consuming.
- There might be conflicts between the names of internal commands of several packages. The resulting effects can be very obscure; there seems to be no easy solution.
- ArabTEX assumes that the special and punctuation characters have their original category codes both when it is loaded,and when Arabic processing begins. If some macro package changes these codes, Arabic processing will usually be broken. This does not apply to Babel nor to `"german.sty"`; these packages are specially handled.

- Conversely ArabTEX changes the category code of `<` which might break other packages. Loading ArabTEX as the last module usually helps, and enables ArabTEX to detect the presence of other packages.

9.1 Arabic document classes

The experimental LaTeX2$_\varepsilon$ classes `"arabart"`, `"arabbook"`, `"arabrep"` extend the standard classes `"article"`, `"book"`, and `"report"` in several respects: The overall document layout has been "arabized": page numbers are in Indic numerals, and columns run from right to left. The format of running heads depends on the context of the corresponding sectioning commands.

Within Arabic environments which are bracketed by `\begin{RLtext}` and `\end{RLtext}` most LaTeX commands and environments are allowed, including all sectioning commands, `\tabular`, `\tabbing`, even `\tableofcontents`, and use an "Arabic looking" format. All arguments that denote text to be typeset are interpreted according to the currently activated Arabic encoding. Other arguments keep their LaTeX standard meaning, including the preamble of `\tabular`, whose columns are processed from left to right (visual formatting). Generally only the basic functionality is available; optional arguments in brackets are not yet supported.

The commands `\pagenumbering{abj}` and `\abj{`*ctr*`}` generate "*'abjad*" numerals for page numbers and/or arbitrary LaTeX counters.

The document will start out in Roman mode, but may even be made into a single *Arabic environment*. Outside of Arabic environments the LaTeX commands revert to their standard meaning. The `picture` environment and mathematical displays presently only work in Roman mode, but may contain *Arabic insertions*.

9.2 Using ArabTEX with EDMAC

ArabTEX will cooperate with EDMAC, a Plain TEX macro package for critical editions, written by John Lavagnino and Dominik Wujastyk. If EDMAC is already present when ArabTEX is loaded, the EDMAC commands will, after suitable local modifications, be available inside an *Arabic environment*. Their arguments are considered Roman text but may contain *Arabic quotations*.

For further details, see the EDMAC documentation.

EDMAC has been extended to work with LaTeX too, and ArabTEX still cooperates most of the time. However the three macro packages involved are very complicated and interact in very subtle ways, so the user may sometimes get a surprise. In this case, please contact the author.

9.3 Using ArabTEX with Babel

The Babel package by Johannes Braams provides support for multi-lingual texts in a large number of, mostly European, languages. ArabTEX does not use the language-switching facilities provided, but is otherwise compatible.

If ArabTEX is used in a Babel document, "Roman insertions" within an *Arabic context* are interpreted according to the presently active Babel language mode. Conversely, a "Roman paragraph" in a Babel document may contain *Arabic insertions*.

9.4 Using ArabTEX with PicTEX

With some caution, ArabTEX can be used together with PicTEX. However, PicTEX uses the angle brackets < and > for labeling diagrams, and this requires switching off their special meaning within ArabTEX by the command `\setnone`. Therefore short *Arabic insertions* must be included as arguments of `\RL{ }` or bracketed with `\<` and `>`.

9.5 Using ArabTEX with CJK

The CJK package by Werner Lemberg, supporting typesetting of texts in Chinese, Japanese, and Korean, to our surprise proved to be compatible with ArabTEX (after a very small adjustment). Due to the high resource requirements of both packages, a *Very Big TEX* may be required for processing texts of substantial size.

Chapter 10

Acknowledgments

The development of ArabTEX would not have been possible without the assistance of many people, and it is impossible to acknowledge every individual contribution. Besides our local team, i.e. Udo Merkel and Heribert Schlebbe, helpful advice came, among others, from Chahriar Assad, Benno van Dalen, Ivan Derzhanski, Wolfdietrich Fischer, Ahmed El-Hadi, Yannis Haralambous, Abdelsalam Heddaya, Nicholas Heer, Taco Hoekwater, Yussuf Jabri, Iqbal Khan, Tom Koornwinder, Eberhard Krüger, Asif Lakehsar, Jan Lodder, Richard Lorch, Pierre MacKay, Eberhard Mattes, Fathy Neamat-Allah, Anshuman Pandey, Bernd Raichle, Ulrich Rebstock, Adrian Rezus, Paul Roochnik, Mohamed Saba, Waheed Samy, Annemarie Schimmel, Nariman Shehab, Arian Verheij, Dominik Wujastyk, and Michio Yano. We also have to thank all users who sent error reports, comments, and suggestions.

References

B. Alavi, M. Lorenz: *Lehrbuch der persischen Sprache.*
5. Auflage 1988. VEB Verlag Enzyklopädie, Leipzig.

A. A. Ambros: *Einführung in die moderne arabische Schriftsprache.*
1. Auflage 1969. Max Hueber Verlag, München.

ASMO 449: *7-bit coded Arabic character set for information interchange.*
Arabic Standards and Measurements Organization, 1982.

J. D. Becker: *Arabic Word Processing.*
Comm. ACM **30/7,** 600-610 (1987).

T. Borg: *Arabisch für Ausländer. Ein Lehrbuch für modernes Hocharabisch.*
2. Auflage 1979. Verlag Borg GmbH, Hamburg.

J. A. Boyle: *Grammar of Modern Persian.*
Wiesbaden: Otto Harrassowitz, 1966.

B. Comrie (ed.): *The World's Major Languages.*
Croom Helm, London 1987.

DIN 31 635: *Umschrift des Arabischen Alphabets.*
Deutsches Institut für Normung e.V., 1982.

J. Lavagnino and D. Wujastyk: *An Overview of EDMAC: A* `plain` *TEX format for critical editions.*
TUGboat **11/4,** 623-643 (1990).

L. P. Elwell-Sutton: *Elementary Persian Grammar.*
Cambridge University Press, 1963.

C. Faulmann: *Das Buch der Schrift, enthaltend die Schriften und Alphabete aller Zeiten und aller Völker des gesammten* (sic!) *Erdkreises.*
K. K. Hof- und Staatsdruckerei, Wien 1878.

W.D. Fischer: *Grammatik des Klassischen Arabisch.*
2. Auflage 1987. Verlag Otto Harrassowitz, Wiesbaden.

A. Grohmann: *Arabische Paläographie (Teil I und II).*
Österreichische Akademie der Wissenschaften, Philosophisch-historische Klasse, Denkschriften 94, 1. Wien 1967.

E. Harder, A. Schimmel: *Arabische Sprachlehre.*
15. Auflage 1983. Julius Groos Verlag, Heidelberg.

هاشم محمّد الخطّاط، قواعد الخطّ العربيّ.
Hāšim Muḥammad al-Ḫaṭṭāṭ: Qawā'id al-Ḫaṭṭi al-'Arabī.
Maktaba an-Nahḍa, Baghdad; Dār al-Qalam, Beirut, 1400/1980.

ISO/R 233 - 1961: *International System for the Transliteration of Arabic Characters.*
International Standards Institution, 1961.

ISO 8859 - 6: *Information processing — 8-bit single-byte coded graphic character sets — Part 6: Latin/Arabic alphabet.*
International Organization for Standardization, 1987.

ISO 9036: *Information processing — Arabic 7-bit coded character set for information interchange.*
International Organization for Standardization, 1987.

D. E. Knuth: *The* METAFONT*book.*
Addison Wesley Publishing Comp., Reading, Mass., 1986.

D. E. Knuth: *The TEXbook.*
Sixth printing. Addison Wesley Publishing Comp., Reading, Mass., 1986.

D. E. Knuth and P. MacKay: *Mixing right-to-left texts with left-to-right texts.*

TUGboat **8/1,** 14-25 (1987).

Ann K. S. Lambton: *Persian Grammar.*
Cambridge University Press, 1953.

L. Lamport: *LaTeX, A Document Preparation System.*
Addison Wesley Publishing Comp., Reading, Mass., 1986.

M. Lorenz: *Lehrbuch des Pashto (Afghanisch).*
2. Auflage 1982. VEB Verlag Enzyklopädie, Leipzig.

P. A. MacKay: *Typesetting Problem Scripts.*
BYTE **11/2,** 201-216 (1986).

H. Ritter: *Über einige Regeln, die beim Drucken mit arabischen Typen zu beachten sind.*
ZDMG **100/2,** 577-580 (1951).

Friedrich Rückert: *Grammatik, Poetik und Rhetorik der Perser.*
Wiesbaden: Otto Harrassowitz, 1966.

C. Salemann, V. Shukovski: *Persische Grammatik.*
4. Auflage 1947. Verlag Otto Harrassowitz, Leipzig.

A. Schimmel: *Islamic Calligraphy.*
E.J.Brill, Leiden, Netherlands 1970.

H.J. Vermeer, W. Akhtar, A. Akhtar: *Urdu-Lautlehre und Urdu-Schrift.*
3. Auflage 1985. Julius Groos Verlag, Heidelberg.

Appendix A

Obtaining and installing ArabTEX

A.1 Obtaining ArabTEX

The ArabTEX system is available from the author's institution (by anonymous FTP from `ftp.informatik.uni-stuttgart.de (129.69.211.2)`, in the directory `pub/arabtex`) and from many other common servers, e.g. the CTAN network

- `ftp.dante.de/tex-archive/language/arabtex`
- `ftp.tex.ac.uk/tex-archive/language/arabtex`
- `ctan.tug.org/tex-archive/language/arabtex`

.

The files may be transferred individually or as a package: `arabtex.zip` for PC systems, `arabtex.tar.Z` for U*IX systems; we recommend to get and inspect the file `arabtex.htm` or `readme.txt` first. Successfull operation on the Apple Macintosh in conjunction with OzTEX has also been reported.

At the time of this writing, version 4.00 is current. The Nasta'liq font is still under development; Naskh will be substituted automatically.

ArabTEX is being maintained and extended, if required, on a regular schedule. The current status can be found in the file `arabtex.htm` or by sending an EMail message (with arbitrary content) to

`arabtex@informatik.uni-stuttgart.de`

A.2 Installing ArabTEX

The installation procedure is strongly system dependent, and we recommend securing the assistance of a local TEXpert. You have to install the fonts provided (`"nash14", "nash14bf", "xnsh14", "xnsh14bf", "hclassic", "hcaption"`) with their `"*.pk"` and `"*.tfm"` files on the font search path of your TEX system, and the `"*.sty"` files, `"arabtex.tex"`, and `"hebtex.tex"` on the source search path (usually TEXINPUT) of your system. Possibly you will also have to rename the `"*.pk"` files according to local conventions, and as a last resort you can try to recreate the fonts from the `"*.mf"` METAFONT sources. Additional fonts, whenever available, are installed analogously.

ArabTEX has been found to cooperate well with TEX versions 3.xxx, LATEX2$_\varepsilon$, LATEX versions 2.09 of 1991 or later, MlTêX, NFSS and NFSS2 (not required), and previewers that can handle fonts of more than 128 characters. TEX-XET or TEX--XET are not required, and their additional features are presently not exploited.

The TEX "hash size" should be at least 3000 to 3500, especially when using ArabTEX in conjunction with LATEX, and if the transliteration module is used. Use of a BIG TEX may be necessary when using the NFSS2 due to the latter's high demand on string storage. Space and time requirements are not negligible, and have increased during development; however, ArabTEX currently still runs, albeit slowly, even on a PC XT standard configuration.

Appendix B

Release history

The development of the ArabTEX system began around 1991 as a private project of the author, for his personal use. However it turned out soon that the package, if at all feasible, could be of use for others also who see the need of printing Arabic text without involving a special publishing agency. As prospective users we mainly considered Orientalists which we believed very short on funding (this proved to be a drastic understatement). There was no Arabic word processing available at that time, and using TEX as a platform looked like the only remaining possibility except perhaps implementing some complete system from scratch, which probably would necessitate building multiple versions for various computer platforms and operating systems, which we did neither dare nor could afford.

Basing the design on TEX required the minimal user interface to become extremely lean, to facilitate the use by non-programmers. In fact, only three commands and the input notation conventions have to be learned, once the user is familiar with TEX or LATEX, to be able to use ArabTEX for a standard Arabic document. Additional features can be looked up as required.

Using the TEX typesetting engine as a machine independent platform suggested implementing the internal algorithms in TEX's powerful internal macro language. Unfortunately, it is not easy to use, and errors can be very hard to find and eliminate.

B.1 ArabTEX version 1.00

ArabTEX version 1.00 was a prototype, to check the basic feasibility of our approach, and to get some operating experience. Many of its features were only available in a very primitive form. It is no more supported.

B.2 ArabTEX version 2.00

ArabTEX Version 2 was the first stable version of ArabTEX. It was not fully compatible with Version 1; however, moving to the new version usually caused little problems. 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.

The main differences between versions 1 and 2 were:

- The font size was increased, so the document layout changed. The old font `"nash10"` was abolished and replaced by `"nash14"`; the character locations have been assigned differently.
- Some Arabic characters were now coded differently: *'ayn* is denoted by a left quote, and `<c>`, `<^z>`, `<^t>`, and `<.n>` have been assigned new meanings in order to better conform to the standard transliteration.
- Many more ligatures than before were supplied. This normally did not concern the user.
- `\vocalize` no more generated *sukūn* and *waṣla* except if explicitly indicated by quoting. See `\fullvocalize`.
- Arabic Environments are now always bracketed by the new control sequences `\begin{arabtext}` and `\end{arabtext}` even if only the transliteration is wanted.

B.3 ArabTEX version 3.00

The changes introduced in Version 3.00 fall into one of two categories: error corrections, and upward compatible extensions. Details are not given here, but are documented in the text file `"changes.txt"` that is part of the distribution package of ArabTEX. The earlier change history up to Version 3.00 is described in the text file `"changes2.txt"`.

Version 3 is upwards compatible with version 2. However, many new features were introduced gradually, among them support for additional input encodings and a multitude of languages that use the Perso-Arabic script. We gratefully acknowledge the cooperation of several users who contributed information, documentation, and even helped with the coding.

On some users' request, a Hebrew mode was added, as well as support for nearly all the Hebrew TEX fonts that are available on the CTAN server network.

B.4 ArabTEX version 4.00

Version 4 is an upwards compatible extension of version 3, and many modules have been rewritten. All presently supported features are documented in this manual. It proved impractical to indicate every extension explicitly; the basic user interface is still the same.

In a few instances we had to abolish certain old features that, to our knowledge, were rarely used or not at all, because of ambiguities or conflicts with the extensions. These places are indicated in the manual, and are flagged by an asterisk in the margin. *

The most important incompatible changes are:

- Tilde (<~>) is no more used as a prefix in the transliteration encoding, because of conflicts with TEX's use of tilde for a stable space. Caret (<^>) is now used instead in all cases. *
- The double bar (<||>) for indicating a small unbreakable space has been replaced by <\,>. *

Users who still need the old features should contact the author; there might be a workaround.

Appendix C

Miscellaneous utilities

The following packages are not part of ArabTEX proper, and are not supported in any way, but are distributed along with ArabTEX as possibly a convenience to the users. There is no warranty whatsoever.

C.1 verses.sty

This is a small utility for typesetting classical Arabic poetry in two parallel blocks, such that every line contains two half-verses. For its use, see the file itself.

C.2 twoblks.sty

This LATEX option will define a command `\twoblocks {#1}{#2}` which will place the two parameters `#1` and `#2`, usually two paragraphs, into two boxes side by side, separated by space of length `\colsep`. If necessary, the resulting boxes will be split across a page boundary.

This feature is useful if two versions of a text are to be contrasted. They may be in different languages, and one of them might be in Arabic (if enclosed in `\begin {arabtext}` ··· `\end {arabtext}`).

This sentence has been written twice: in the English language and in the Arabic language.

كُتِبَتْ هٰذِهِ ٱلْجُمْلَةُ مَرَّتَيْنِ: بِٱللُّغَةِ ٱلْإِنْجِلِيزِيَّةِ وَبِٱللُّغَةِ ٱلْعَرَبِيَّةِ.

Otherwise this command does not depend on ArabTEX in any way, and indeed originated in a completely different context.

Beware that the two "blocks" should each not contain much more than one, not too long, paragraph of text, otherwise TEX's main storage might overflow. There must be no `\verbatim` text inside the parameters of `\twoblocks`, nor any `\catcode` changes; and all TEX groups and `\if` ··· `\fi` sequences must be properly nested.

C.3 raw.sty

This is a small utility to ease the processing of input files that have been produced by some OCR reading program. It will deactivate most of TEX's special characters.

This package depends strongly on the special application; if you need it or a variant of it, enquire with the author.

Index