Arabic (RTL) text v5.x

> **Note:** Handling of RTL (right-to-left) languages was significantly rewritten for mPDF v5.1

Document Directionality - RTL versus LTR
The document has a baseline direction which is **LTR** or **RTL**; this determines:

- text alignment in blocks for which text-align has not been specifically set
- layout of mirrored page-margins, columns, ToC and Indexes, headers / footers

This base/document directionality is **LTR** by default, and can be set by any of the following:

```
$mpdf->SetDirectionality('rtl');
<html dir="rtl"> or <html style="direction: rtl;">
<body dir="rtl"> or <body style="direction: rtl;">
```

Base direction is an inherited CSS property, so will affect all content, unless direction is specified elswhere.

Block-level Directionality
Direction can be set for any HTML block elements e.g. <p><table> etc using:

```
[HTML]
<div style="direction: rtl;">
or
[CSS stylesheet]
div.right { direction: rtl; }
```

Block-level direction *may* affect text alignment, and will also influence text reversal in **RTL** text.

Note that margin/padding are NOT reversed by direction i.e. left-margin will still be left-margin in **RTL** state.

Text alignment
The default value for text-align is "a nameless value which is dependent on direction". However, once text-align is specified, it is respected and inherited by all descendants.

Directionality in Tables
- direction can only be set on the top-level element of nested lists
- direction can only be set on <table>, NOT on <thead><tbody><td> etc.
- nested tables CAN have different directions

Text Bidirectionality
mPDF analyses any mixed text which contains **RTL** text. The text between HTML tags is divided into "chunks" of **LTR** and **RTL** text.

RTL text chunks are reversed (both letter- and word-order).

If (and only if) the direction of the block is **LTR** then the order of the chunks is reversed as well so that the sentence order is **RTL**.

This process when **RTL** arabic characters are present is fully automatic and unconfigurable. <bdo> etc has no effect.

However enclosing text in silent tags can sometimes help by altering the way the text is broken up into chunks to process e.g.:

```
english text <span>[arabic text]</span> english text
```

Fonts

Arabic is a complex script requiring processing before output. However any appropriate font can be used - as long as it contains the characters in Unicode blocks 'Arabic Presentation Forms' A and B (U+FB50 - U+FDFF, U+FE70 - U+FEFE). Note that quite a large number of fonts contain the isolated characters but not the presentation forms.

2 fonts are bundled with mPDF: XB Zar and XB Riyaz. These are 2 of a number of fonts available from http://wiki.irmug.com/index.php/X_Series_2.

> **Note:** The script handling Arabic text (RTL) was rewritten in mPDF 5.5 with improved support for Pashto/Sindhi/Urdu/Kurdish, especially for joining characters and added new presentation forms.

Non-unicode characters
Some characters in Pashto/Sindhi/Urdu/Kurdish do not have Unicode values for the final/initial/medial forms

of the characters. However, some fonts include glyphs for these characters "un-mapped" to Unicode (including XB Zar and XB Riyaz, which are bundled with mPDF).

By editing `config_fonts.php` and adding to appropriate fonts:

```
'unAGlyphs' => true,
```

this will force mPDF to use unmapped glyphs. It requires the font file to include a Format 2.0 POST table which references the glyphs by name as e.g. uni067C.med or uni067C.medi

XB Riyaz, XB Zar, Arabic Typesetting (MS), Arial (MS) all contain this table. NB If you want to know if a font file is suitable, you can open a .ttf file in a text editor and search for "uni067C.med" - if it exists, it may work!

Using "unAGlyphs" forces subsetting of fonts, and will not work with SIP/SMP fonts (using characters beyond the Unicode BMP Plane).

mPDF maps these characters to part of the Private Use Area allocated by Unicode U+F500-F7FF. This could interfere with correct use if the font already utilises these codes (unlikely).

Alef Maksura

Detailed note on the Alef Maksura for advanced users:
U+0649 Alef Maksura only normally appears at the end of a word (in Arabic)
Initial and Medial forms exist in Unicode as FBE8 and FBE9 but are not in most fonts
So the final form is set in mPDF to show as FEF0; Initial and medial forms are shown as isolated/final, so that it does at least display.
It seems that Initial and Medial forms are used in Koranic text.
I have left options encoded in *function InitArabic()* if you want to alter - to make it double-joining, it also needs to be added to *$arabPrevLink* as "\xd9\x89"

> **Note:** mPDF deletes Unicode characters: U+200C,U+200D,U+200E,U+200F zero-width joiner/non-joiner, LTR and RTL marks so they will not appear - even though some fonts contain glyphs for these characters.

See Also

- useLang - Specify whether to recognise and support the HTML attribute lang
- SetAutoFont() - Use AutoFont to auto-detect text language in HTML input
- autoFontGroupSize - Specify the text chunk size to group when autodetecting text language
- disableMultilingualJustify - Specify whether to disable text justification in multilingual documents
- lang - Information on mPDF support for the HTML attribute lang

Printed on Wed 05 Aug 2015 12:05:22 GMT +0100 (DST)