mPDF

Other new features in mPDF Version 5.1

- Kerning
- Letter- and word-spacing
- Small-caps improved to work with justified text, and now with kerning, letter- and word-spacing
- Bleed area on @page media
- Colorspace and colour conversion (almost everything except BMP images)
- Spot colours
- PDF/X files
- dir="rtl"
- numeric list-styles for arabic and indic

Kerning

Font kerning is supported. This corrects the inter-character spacing between specific pairs of letters. It is dependent on kerning information being available in the original font file.
You need to set $mpdf->useKerning=true; either in the config.php configuration file, or at runtime. This causes the kerning information to be loaded when fonts are accessed (and will therefore increase memory usage).
You can then set kerning on or off using the draft CSS3 style property "font-kerning". Values of normal or auto will turn kerning on; "none" will turn kerning off.

Off: AWAY To War.
On: AWAY To War.

Letter and word spacing & text justification

Letter- and word-spacing can be set on almost all block and in-line style elements, using the CSS properties letter-spacing and word-spacing. Values of normal or a length can be specified (em or ex recommended). Note that setting the letter-spacing value (including setting it to zero) will prevent any additional letter-spacing to be added when full-justifying text. The word-spacing value, however, is a *minimum* value, and can be increased in order to justify text.

Text-align: justify - no longer uses configurable variable $jSpacing= C | W | ''
The default value is for mixed letter- and word-spacing, set by jSWord and jSmaxChar
If a line contains a cursive script (RTL or Indic [devanagari, punjabi, bengali]) then it prevents letter-spacing for justification on that line - effectively the same as setting letter-spacing:0
Spacing values have been removed from the config_cp.php configuration file, so the "lang" property (in config_cp) no longer determines justification behaviour (this includes the use of Autofont()).
When using RTL or Indic [devanagari, punjabi, bengali] scripts, you should set CSS letter-spacing:0 whenever you use text-align:justify.

Nulla felis erat, imperdiet eu, ullamcorper non, nonummy quis, elit. Suspendisse potenti. Ut a eros at ligula vehicula pretium. Maecenas feugiat pede vel risus. Nulla et lectus. Letter spacing set at 0.2em. Fusce eleifend neque sit amet erat. Integer consectetuer nulla non orci. Morbi feugiat pulvinar dolor. Cras odio. Donec mattis, nisi id euismod auctor, neque metus pellentesque risus, at eleifend lacus sapien et risus. Word spacing set at 1em. Phasellus metus. Phasellus feugiat, lectus ac aliquam molestie, leo lacus tincidunt turpis, vel aliquam quam odio et sapien. Mauris ante pede, auctor ac, suscipit quis, malesuada sed, nulla. Integer sit amet odio sit amet lectus luctus euismod. Donec et nulla. Sed quis orci.

Colours

Whenever a colour can be specified in a style, additional formats are now supported: rgb(), rgba(), hsl(), hsla(), cmyk(), cmyka(), or spot().
Spot colours need to be defined at the start of the script using e.g. $mpdf->AddSpotColor('PANTONE 534 EC',85,65,47,9);
The four values define the CMYK values used when the spot colour is not available.
A tint % can be specified when using the spot colour in the document.

background-color: rgba(150,150,255, 0.5); color: rgb(0,150,150);

background-color: rgba(60%,60%,100%, 0.5); color: rgb(0,60%,60%);

background-color: hsla(180,30%,25%, 0.5); color: hsl(360,100%,50%);

background-color: cmyka(85,65,0,30, 0.3); color: spot(PANTONE 300 EC,80%);

@page media

When using @page to create a print publication with page-size less than sheet-size, the bleed margin is now configurable. Backgrounds/gradients/images now use the bleed box as their "container box", rather than the whole page. (See this document as an example.)
Crop- and cross-marks can now both be used together, and are more configurable. Also, background-image-opacity and background-image-resize have been extended to work with @page CSS.
The following values can be set in the configuration file, config.php:
$this->bleedMargin
$this->crossMarkMargin
$this->cropMarkMargin
$this->cropMarkLength
$this->nonPrintMargin

Colorspace and colour conversion

PDF files can contain objects using different colorSpaces e.g. Grayscale, RGB and CMYK. By default, mPDF creates PDF files using the colours as they are specified: font colour may be set (e.g. #880000) as an RGB colour, and the file may contain JPG images in RGB or CMYK format.
In some circumstances, you may wish to create a PDF file with restricted colorSpaces e.g. printers will often want files which contain only CMYK, spot colours, or grayscale, but *not* RGB.
Additional methods for defining colours can be used (see above), but alternatively you can set mPDF to restrict the colorSpace by setting the value for $mpdf->restrictColorSpace:
1 - allow GRAYSCALE only [converts CMYK/RGB->gray]
2 - allow RGB / SPOT COLORS / Grayscale [converts CMYK->RGB]
3 - allow CMYK / SPOT COLORS / Grayscale [converts RGB->CMYK]
This will attempt to convert every colour value used in the document to the permitted colorSpace(s). Almost everything including images will be converted (except BMP images), and the conversion of images may take significant time.
This example file is set to (3) CMYK; compare the appearance of the Tux penguin in this file and in the previous example file (RGB).

PDF/A and PDF/X files

mPDF can produce files which (attempt to) meet the PDF/A and PDF/X specifications. In addition to restricted colorSpace, PDF/A and /X files cannot contain images or colour values with "transparency".
Please note that full compliance with the PDF/A or /X specification is not guaranteed.

RTL (right-to-left) text

Handling of RTL (right-to-left) languages has been significantly rewritten, and is likely to cause changes to the resulting files if you have previously been using mPDF. The changes have made mPDF act more like a browser, respecting the HTML/CSS rules. Changes include:

- the document now has a baseline direction; this determines the
  - behaviour of blocks for which text-align has not been specifically set
  - layout of mirrored page-margins, columns, ToC and Indexes, headers / footers
  - base direction can be set by any of:
    - $mpdf->SetDirectionality('rtl');
    - <html dir="rtl" or style="direction: rtl;">
    - <body dir="rtl" or style="direction: rtl;">
  - base direction is an inherited CSS property, so will affect all content, unless...
- direction can be set for all HTML block elements e.g. <DIV><P><TABLE><UL> etc using
  - CSS property <style="direction: rtl;">
  - 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
- NOTE that block/table margins/paddings are NOT reversed by direction
- language (either CSS "lang", using Autofont, or through initial set-up e.g. $mpdf = new mPDF('ar')) no longer affects direction in any way.
  NB config_cp.php has been changed as a result; any values of "dir" set here are now ineffective
- default text-align is now as per CSS spec: "a nameless value which is dependent on direction"
  NB default text-align removed in default stylesheet in config.php
- once text-align is specified, it is respected and inherited
  NB mPDF <5.1 reversed the text-align property for all blocks when RTL set.
- the configurable value $rtlcss is depracated, as it is no longer required
- improved algorithm for dtermining text direction
  - english word blocks are handled in text reversal as one block i.e. dir="rtl" [arabic text] this will not be reversed [arabic text]
  - arabic numerals 0-9 handled correctly

Although the control of direction for block elements is now more configurable, the control of text direction (RTL arabic characters) remains fully automatic and unconfigurable. <BDO> etc has no effect. Enclosing text in silent tags can sometimes help e.g.: content[arabic text]content

List styles

Additional numerical list-styles are supported. All of these (except Tamil) are consistent with the draft CSS3 specification:
list-style: arabic-indic | bengali | devanagari | gujarati | gurmukhi | kannada | malayalam | oriya | persian | telugu | thai | urdu | tamil

- Arabic
  ١. One
  ٢. Two
  ٣. Three
  ٤. Four
  ٥. Five
  ٦. Six
- Persian
  ١. One
  ٢. Two
  ٣. Three
  ۴. Four
  ۵. Five
  ۶. Six
- Urdu
  ١. One
  ٢. Two
  ٣. Three
  ۴. Four
  ۵. Five
  ۶. Six

- Bengali
  ১. One
  ২. Two
  ৩. Three
  ৪. Four
  ৫. Five
- Devanagari
  १. One
  २. Two
  ३. Three
  ४. Four
  ५. Five

- Gujarati
  - ૧. One
  - ૨. Two
  - ૩. Three
  - ૪. Four
  - ૫. Five
- Gurmukhi
  - ੧. One
  - ੨. Two
  - ੩. Three
  - ੪. Four
  - ੫. Five
- Kannada
  - ೧. One
  - ೨. Two
  - ೩. Three
  - ೪. Four
  - ೫. Five
- Malayalam
  - ൧. One
  - ൨. Two
  - ൩. Three
  - ൪. Four
  - ൫. Five
- Oriya
  - ୧. One
  - ୨. Two
  - ୩. Three
  - ୪. Four
  - ୫. Five
- Tamil
  - க. One
  - உ. Two
  - ங. Three
  - ச. Four
  - ரு. Five
- Telugu
  - ౧. One
  - ౨. Two
  - ౩. Three
  - ౪. Four
  - ౫. Five