mPDF Version 4.2
123 Anystreet
Your City
GD12 4LP
☎ 01777 123 567

20th January 2026

mPDF Version 4.2

New Features

- image handling improved
- table layout - additional control over resizing
- vertical-alignment of images - better support for all CSS types
- top and bottom margins collapse between block elements
- improved support for CSS line-height
- display progress bar whilst generating file
- CSS @page selector can be specified when adding a pagebreak
- CSS @page selector allows different margins, backgrounds, headers/footers on :first :left and :right pages
- PNG images with alpha channel fully supported
- ability to generate italic and bold font variants from base font file
- CJK fonts to embed as subsets
- "double" border on block elements
- character substitution for missing characters in UTF-8 fonts
- direct passing of dynamically produced image data
- background-gradient and background-image can now co-exist

Note: automatic top- and bottom-margin to accomodate varying header/footer size was introduced in v4.0 but was not highlighted cf. AutoHeaderMargin in the Manual.

Page backgrounds

Background images, gradients and/or colours can be used together on the same page. On this page, the bars on the left hand side are created using a background-image, whilst a background-gradient sets the background to the whole page.

CSS "double" border

Block elements can now use the CSS property: border(style) = double. See also the tiger logo in the header of this page.

CJK fonts to embed as subsets

When writing documents with Chinese, Japanese or Korean characters, mPDF has previously required the end-user to download Adobe's free CJK font pack. The ability to embed font subsets now makes it feasible to use open license CJK fonts. 2 fonts are now available to download as an additional font-pack:

- zn_hannom_a - contains all characters in the SJIS, BIG-5, and GBK codepages; original file was Han Nom A font (Hi-res version) from http://vietunicode.sourceforge.net/fonts/fonts_hannom.html
- unbatang - contains all the (Korean) characters in the UHC codepage; original file from from http://kldp.net/projects/unfonts/download

The following characters only added an extra 15kB to the size of this PDF file, and approximately 0.15 seconds extra to compile:
Chinese (traditional) 憂鬱 ; chinese (simplified) 来自 ; japanese たゐ ; korean 키스

Artificial Bold and Italic

The text in this block is in ArialUnicodeMS font. Using embedded subsets it covers most characters you want to print - BUT it does not have bold, italic, or bold-italic forms.
From version 4.2, mPDF will create "artificial" font styles if they are not available as separate font files:
The quick brown fox jumps over a lazy dog

The quick brown fox jumps over a lazy dog

The quick brown fox jumps over a lazy dog

Character substitution in UTF-8 files

This paragraph has the font-family set to Trebuchet MS, and the document has the default font set as DejaVuSansCondensed. The following characters are not present in the Trebuchet font, and are substituted from the core Adobe Zapfdingbats font:
✐ ✑ ✒ ✓ ✔ × ✖ ✗ ✘ ✙ ✚ ✛ ✜ ✝ ✞ ✟
The characters are not present in the Trebuchet font, and are substituted from the (default) DejaVuSansCondensed font:
ʀ ʁ ʂ ʃ ʄ ɻ ʃ ɻ ʈ ʉ ʊ ʋ ʌ ʍ ʎ ʏ
Character substitution in UTF-8 files is enabled by setting:

```
$mpdf->useSubstitutionsMB = true;
```

NB In mPDF 5.0 this has changed to

```
$mpdf->useSubstitutions = true;
```

It is not recommended to enable this for regular use, as it will add to the processing time.

Margin-collapse

mPDF has always allowed margins to be collapsed at the top and bottom of pages. This is specified by the custom CSS property "margin-collapse: collapse"

mPDF 4.2 also allows margins to collapse between block elements on the page. This is the default behaviour in browsers, and has been enabled in mPDF 4.2 by default.

In the next 2 paragraphs, the first one has the margin-bottom set to 3em, and the second has the margin-top set to 0em. So the vertical-space between paragraphs is 3em:

The quick brown fox jumps over a lazy dog

The quick brown fox jumps over a lazy dog

In the next 2 paragraphs, the first one has the margin-bottom set to 2em, and the second has the margin-top set to 1em. The margins collapse to the larger of the adjoining margins i.e. 2em:

The quick brown fox jumps over a lazy dog

The quick brown fox jumps over a lazy dog

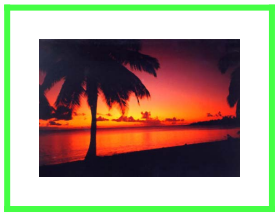
Images

PNG Alpha channel

PNG alpha channel transparency is now fully supported, and works against solid backgrounds, gradients or background images:

Image Border and padding

Image padding is now supported as well as border and margin:



Vertical alignment

From mPDF version 4.2 onwards, most of the values for "vertical-align" are supported: top, bottom, middle, baseline, text-top, and text-bottom.
Note: The default value for vertical alignment has been changed to baseline, and the default padding to 0, consistent with most browsers.

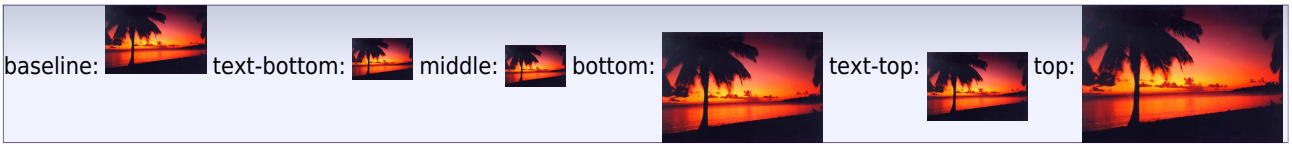
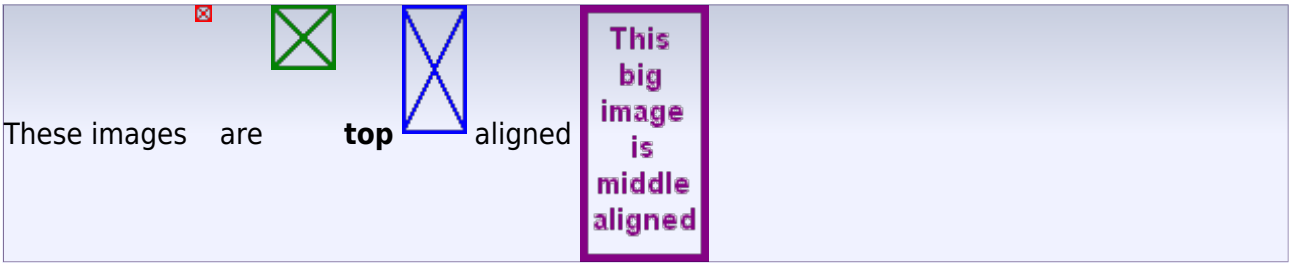
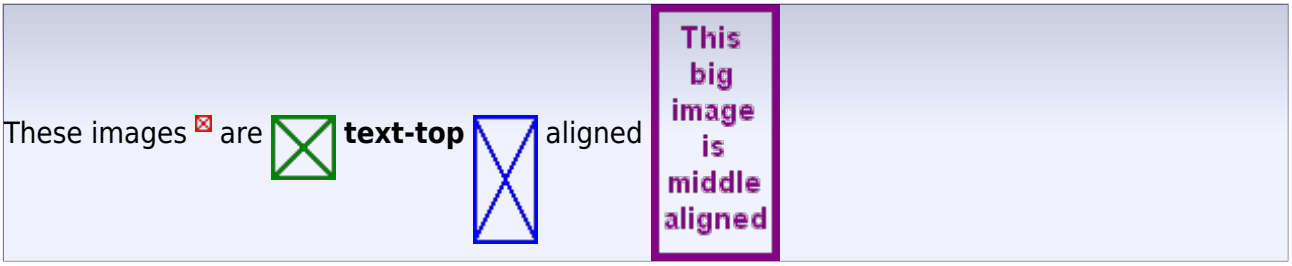
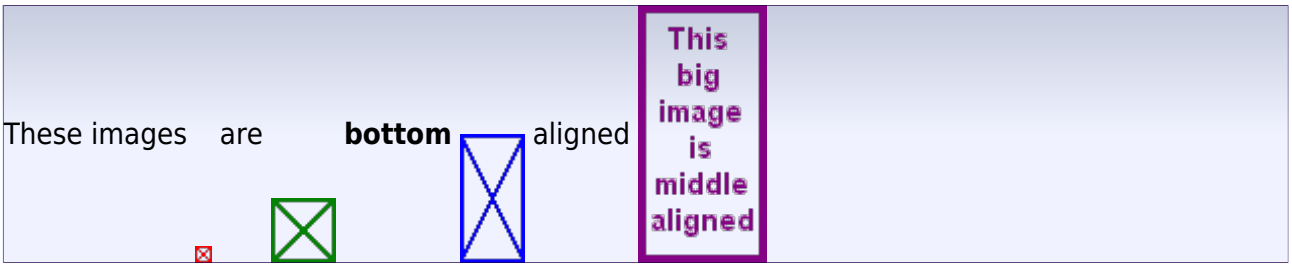


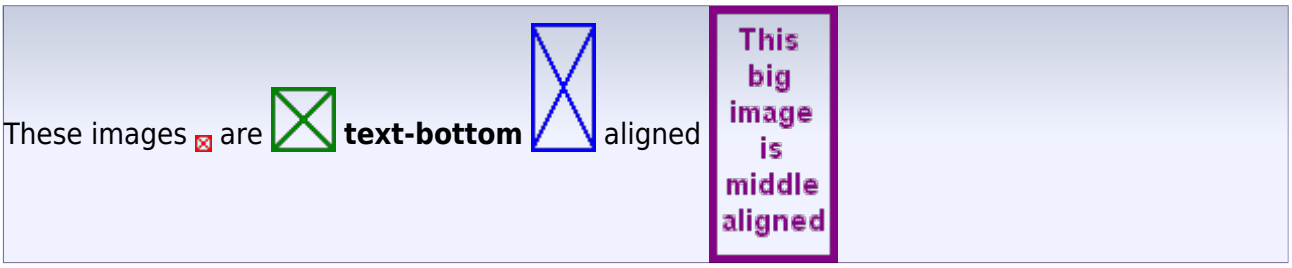
Image Alignment

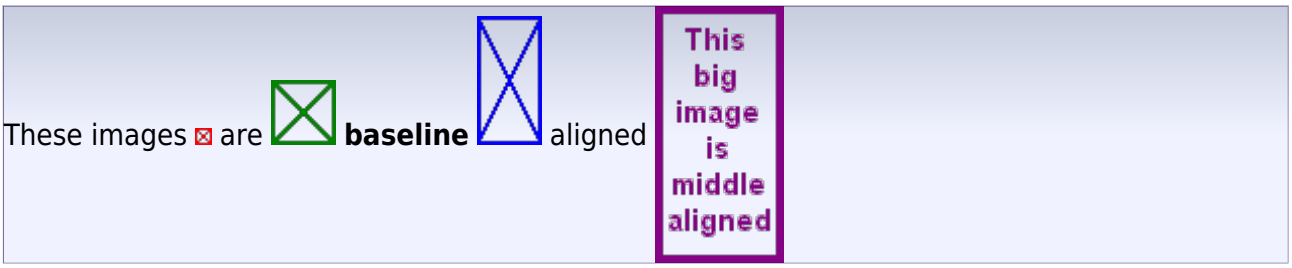
From mPDF version 4.2 onwards, in-line images can be individually aligned (vertically).

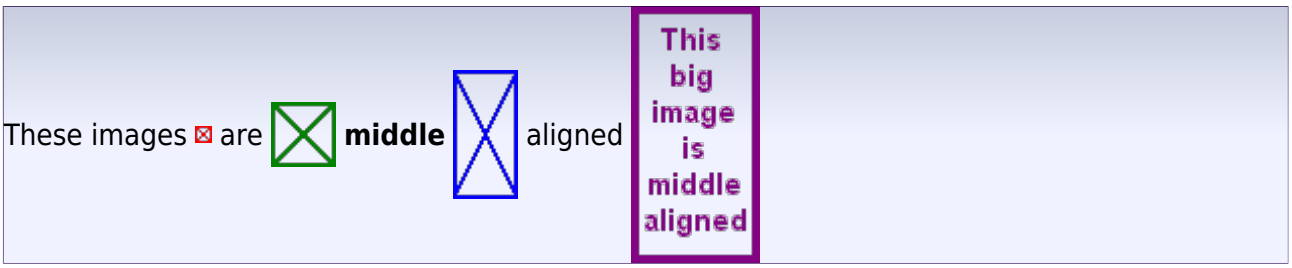


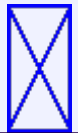






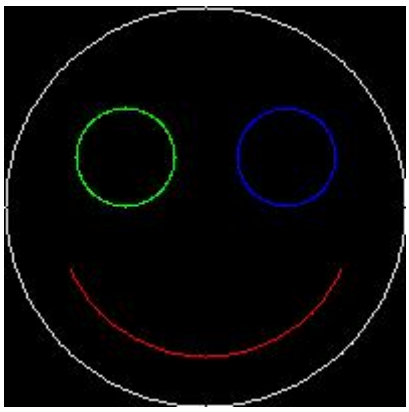




These images ⊠ are  **middle**  aligned 

Images from PHP

Note: This behaviour changed in mPDF 7.0: from direct mPDF properties to internal public `Mpdf::$imageVars` property.



This image was created with the following code:

```
$img = imagecreatetruecolor(200, 200);
$white = imagecolorallocate($img, 255, 255, 255);
$red = imagecolorallocate($img, 255, 0, 0);
$green = imagecolorallocate($img, 0, 255, 0);
$blue = imagecolorallocate($img, 0, 0, 255);
imagearc($img, 100, 100, 200, 200, 0, 360, $white);
imagearc($img, 100, 100, 150, 150, 25, 155, $red);
imagearc($img, 60, 75, 50, 50, 0, 360, $green);
imagearc($img, 140, 75, 50, 50, 0, 360, $blue);
ob_start();
imagejpeg($img);
$mpdf->imageVars['smileyface'] = ob_get_clean();
imagedestroy($img);
```

and written to the document using:

```
<img src="var:smileyface" />
```

Line-height inheritance

Line-height inheritance has been altered to follow the CSS2 recommendation:

- normal is inherited as "normal"
- 1.2 is inherited as a factor
- 120% is converted to an actual value and then inherited as the computed value
- em is converted to an actual value and then inherited as the computed value
- px pt mm are inherited as fixed values

Relative values (e.g. 1.3, normal)

This DIV has the line-height set as "2.0" and font-size as 12pt. The line-height is therefore 24pt, but the factor of 2 is inherited...

Nulla felis erat, imperdiet eu, ullamcorper non, nonummy quis, elit. Suspendisse...

This DIV has the font-size set as 8pt. The line-height of 2 is inherited...

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. Fusce eleifend neque sit amet erat. Integer consectetuer nulla non orci.

This DIV has the font-size set as 18pt. The line-height of 2 is inherited...

Nulla felis erat, imperdiet eu, ullamcorper non...

Absolute values (e.g. 130%, 1.3em, 18pt)

This DIV has the line-height set as "200%" and font-size as 12pt. The computed line-height of 24pt is inherited...

Nulla felis erat, imperdiet eu, ullamcorper non, nonummy quis, elit. Suspendisse...

This DIV has the font-size set as 8pt. The computed line-height of 24pt is inherited...

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. Fusce eleifend neque sit amet erat. Integer consectetuer nulla non orci.

This DIV has the font-size set as 18pt. The computed line-height of 24pt is inherited...
Nulla felis erat, imperdiet eu, ullamcorper non...

Line-height & vertical alignment

In these examples, top and bottom padding are set to 0, so the block height = line height.
The inline text (set to a larger font-size) inherits the line-height as a factor of the largest font-size i.e. the line height will expand to reflect the largest font on the line.
Line-height: "normal" (set in mPDF by default as 1.33).

Normal text 16pt font-size Â and normal again

Line-height: 2.0 When using relative line-heights, the text is aligned vertically so that the centre-line of the line goes throught the middle of the largest font.

Normal text 16pt font-size Â and normal again

Line-heights set as a percentages are computed on the base font-size, and are then inherited and treated the same as absolute lengths. This is also true for "em" values. The line-height of this line is set as 200% of the paragraph font-size (10pt).
When using absolute line-heights, the text is aligned vertically so that the centre-line of the line goes throught the middle of the base font.
This means that as far as possible, multiple lines will remain equally spaced
Line-height: 200%

Normal text 16pt font-size Â and normal again

If the line includes a font-size greater than 1.6 times the computed line-height, then the text baseline is dropped so that the text will approximately fit within the line-height.
Line-height: 2em

Normal text 18pt font-size Â and normal again

If the line includes a font-size greater than 2 times the computed line-height, then the line-height is increased to accommodate the larger fontsize.
Line-height: 2em

Normal text 24pt font-size Â and normal again

This broadly reflects the behaviour of IE and Firefox. Note that tall characters such as Â may fall outside the computed line-heights. See the same in an HTML page.

mPDF Version 4.2
123 Anystreet
Your City
GD12 4LP
☎ 01777 123 567

20th January 2026

Extended use of CSS @page selectors

The CSS @page selector, together with the pseudo-selectors :first :left :right have increased support in mPDF 4.2
A named @page can be selected when forcing a new page, e.g. this page was started with:
`<pagebreak page-selector="letterhead" />`
The header and background on this page (and page 1 of the document) are set by the CSS selector: @page letterhead :first {} whilst subsequent pages have no header, a footer, and no background.
CSS @page selectors allow different margins, backgrounds, headers/footers to be set on :first :left and :right pages. Only fixed or mirrored left- and right-margins are supported (i.e. cannot specify different margins for :left and :right).
This layout can be used to produce company letters with only the first page on letterheaded paper.

Table Layout control

mPDF attempts to layout tables according to HTML and CSS specifications. However, because of the difference between screen and paged media, mPDF resizes tables when necessary to make them fit the page. This will happen if the minimum table-width is greater than the page-width. Minimum table-width is defined as the minimum width to accomodate the longest word in each column i.e. words will never be split.

This resizing (minimum-width) can be disabled using a custom CSS property "overflow" on the TABLE tag. There are 4 options:

<table style="overflow: auto"> (this is the default, using resizing)

Verylongwordwithnospacesinitatall	Verylongwordwithnospacesinitatall	Verylongwordwithnospacesinitatall

<table style="overflow: visible"> (disables resizing, but allows overflow to show)

Verylongwordwithnospacesinitatall	Verylongwordwithnospacesinitatall

Verylongwordwithnospacesinitatall

<table style="overflow: hidden"> (disables resizing, and hides/clips any overflow)

Verylongwordwithnospacesinitatall	Verylongwordwithnospacesinitatall	Verylongwordwithnospacesinitatall

<table style="overflow: wrap"> (forces words to break as necessary)

Verylongwordwithnospacesinitatall	Verylongwordwithnospacesinitatall	Verylongwordwithnospacesinitatall