

What's new in Version 4 - Archived

Major additions

Ability to embed font subsets (creating much smaller files)

Much improved support for Arabic languages

Support for Indic languages including consonant conjuncts

Support for Fixed position block elements

New utility to help create your own fonts

PNG alpha channel transparency supported

New utility to create smaller mPDF script with reduced functionality (less memory)

Multiple Barcode types supported

Note: Some changes may alter behaviour from previous versions - see below for details

Additional CSS support

- parsing for stylesheets has been altered and generally improved
- position:fixed or position:absolute supported (partially) for block elements
(together with overflow = visible|hidden|auto, and top,left,bottom,right)
- border-top-color, border-top-width and border-top-style (+ left, bottom and right) are now supported
- border-color, border-style and border-width are now supported as shorthand forms
- "font" is supported as shorthand form - as for CSS2 i.e. [font-style | font-variant | font-weight]? font-size [/ line-height]? font-family
- line-height now supports actual length units e.g. px or mm or em
- border specified as "em" now works correctly
- shorthand form of margin and padding with 3 parameters supported e.g. "margin: 20px 20px 20px;"
- inline elements e.g. p.classname span { } now (partially) supported in CSS stylesheets
- image properties height or width now support values set as "em"
- image now supports property "padding"
- table cell - CSS height property supported (previously only supported as HTML attribute)

Changed behaviour from previous versions of mPDF

Justified text

Word and character spacing are continued on the final line of justified text. This follows practice in programs such as Word and produces a more pleasing appearance. The word spacing and character spacing are limited to maximum values which can be altered in the `config.php` file (`$this->jSmaxCharLast`, `$this->jSmaxWordLast`). Setting these both to 0 will turn off the effect.

Automatic language detection

Auto-detection of languages using `AutoFont()` has been improved:

- avoids nesting of `s` generated by using `AutoFont()`
- now distinguishes Chinese and Vietnamese
- better algorithm to distinguish Arabic, Persian, Urdu, Sindhi and Pashto
- improved detection of Indic text including zero-width (non-)joiners etc.

`AutoFont()` now marks text with a class as well e.g. `` to aid CSS stylesheet use:

Example

```
<style>
.lang_hi { font-size: 18pt; }
</style>
```

```
// Any text which is detected by AutoFont() as Hindi/devanagari, will be tagged with the
// class, and thus acquire the CSS properties.
```

Character Substitutions

`$useSubstitutions` is now selected automatically when instantiating class. Character substitution is turned on automatically if using mPDF with win-1252, win-1251, iso-8859-x, or a CJK codepage. Otherwise it is turned off (it was always on <= 3.2).

Character substitution no longer works for UTF-8 documents. If you want to override the above, and turn it off, use:

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

Character substitution now uses an appropriate sans, serif, or mono font and keeps the current style (italic/bold etc.)

Support for CSS "lang"

`$useLang` is now set to **TRUE** by default i.e. mPDF will use the CSS property "lang" to restrict fonts - as specified by `GetCodePage()` in `config_cp.php`

Prior to v4.0, if the main codepage is set to e.g. `mPDF('he')` this restricts available fonts (so that only fonts which contain hebrew characters will be selected). If you then use `<p lang="ps">...</p>` to include some Pashto language in the document, mPDF would not select an appropriate font as it was not included in the initial restricted list.

This has been changed so that "lang" will select restricted fonts from the full list.

RTL text

mPDF will now automatically check all HTML (`WriteHTML()`) for any RTL characters. This will set `$biDirectional` to enable processing of RTL text. `$biDirectional` no longer needs to be set explicitly.

The variable `$rtlAsArabicFarsi` is deprecated/ignored/no longer needed, with the new fonts used in v4.0.

The Scheherazade font is no longer included with mPDF. All RTL language is now "joined" and uses Presentation Forms for initial-medial-final forms (which are not included in the Scheherazade font).

Joining of arabic text has been improved for arabic (esp. characters U+0670 and U+0653), and additionally for persian(farsi) pashto, urdu, and sindhi.

Brackets are now reversed appropriately in RTL text: `[]{}()`

New Alias

`$mirrorMargins` has been introduced instead of `$useOddEven`. `$useOddEven` is deprecated but will continue to be supported.

Defining _MPDF_PATH

`_MPDF_PATH` is now defined automatically when including the `mpdf.php` file, so you don't need to explicitly define it.

Changes to default CSS

The default CSS for mPDF (contained in `mpdf.css`) has been changed

- `line-height` for UL, OL removed - to allow inheritance of line-height to lists

NB Although CSS states line-height is inherited by all, browsers do not inherit to tables (and neither does mPDF).

- `text-align` and `text-indent` have been removed from P but included for BODY

Previously `<div><h1>Here</h1></div>` lost the font-size set for H1

(This has also been changed in function `MergeCSS` so that the default tag CSS is not overwritten by "inherited" styles)

In-line "background-color"

The in-line property `background-color` now puts a fill of the specified colour around the font - not the full cell size (which includes the line-height). This makes the appearance match that of a browser more closely.

Underline and Strikethrough

The position and line thickness used for underline and strikethrough text is now based on information from the font file (the up and ut values in the font metrics). Previously this was standard for every font/style/weight.

This has required changes to the core font files e.g. times.php to add up/ut values.

Changes to font files

A fresh install is recommended for mPDF 4.0. One of the reasons is numerous changes to the font files including:

- all the dejavu* and free* font files are updated to the latest releases
- the core font files have been changed - renaming the variable returning the character-width data
- up and ut have been added to some of the font files (cf. underline and strikethrough)

Other Font changes/improvements

Text can now be copied & pasted correctly from a completed PDF doc to PC clipboard when using single-byte, non-win-1252 codepage e.g. win-1251.

- NB Requires /maps/ folder with copies of all maps in it (included as part of mPDF 4.0)
- NNB This doesn't work if you have made custom "diffs" - but this only affects the copy/paste text, not how it appears in the PDF

A font that is declared (e.g. default font) but never used, is not now embedded in the PDF file when using TrueTypeUnicode (i.e. utf-8) fonts (not when using embedded font subset). This can save up to 300KB off the size of the PDF file.

All the dejavu* and free* fonts have been updated to the latest versions (affects unifont files only). Note that the freeserif file has much greater coverage of characters now - but it adds 768kb to a PDF file if not using subsets!

Developing

If you testing out different configurations of mPDF, try setting `$mpf->showStats = true` (default = false). This will display a few figures about mPDF instead of outputting the actual PDF file e.g. memory usage, files size, time taken.

HTML Headers and Footers

All HTML headers and footers are now actually "written" at the end of the document. This allows the page numbers (especially the page total) to be accurately written. The variable `$this->headerPageNoMarker = "!"`; is no longer required.

Alternative modes of making the top/bottom margins dependent on the header/footer sizes have been added. See Headers & Top Margins

Image handling

As well as support for PNG alpha channel transparency, a number of other improvements have been made for images.

GIF files take a very long time to process using the included gif.php class. This allows GIF files to be included without requiring built-in PHP support. However since most people probably have the GD library included with their PHP and GIF support, mPDF now attempts first to use this. If successful, it results in much faster processing of GIF images.

mPD now also attempts to use the GD library (if present) to handle interlaced, transparent and filtered PNG files - with better results and greater support for different types of PNG images.

There is improved recognition of image files with no extension and no/incorrect PHP Header sent (again using GD library where possible).

Bug fixes

Bug fixes (parsing CSS)

- `<link href="" ...` was not recognised if `>` 1 space between words
- `#Content p em { font-style:italic; }` was applied to `"#Content p"`
- `@import url()` embedded in a stylesheet file requires path fixed relative to stylesheet file
- `background-image url()` embedded in a stylesheet file requires path fixed relative to stylesheet file

- comment tags inside CSS `<style>` embedded in the HTML were removed
Now fixed so `<style><!-- ... --></style>` works; `<!-- <style>...</style> -->` is removed

Bug fixes (other)

- clear (CSS property for floating elements) caused properties for that element to reset to defaults
- width: auto caused collapse of border and padding on L & R of ordinary block elements
- text-indent not inherited correctly (including em and % values)
- named colour "steelblue" corrected RRGGBB hex code
- table cell widths in %: error if width of table cells set to $\geq 100\%$, and not all columns are set. (This was fixed in 3.2 but led to problem where 2 cols: 1) 80% and 2) not set (see Table sizing test) Now fixed again to work for both - hopefully
- parse PNG error fixed
- background-image not correctly positioned in HTMLFooter and HTMLHeader (Not fixed properly in 3.2!)
- fonts not supported with 0-9 in the name
- font list in GetCodepage() in htmltoolkit.php (now config_cp.php) containing space " " not recognised
- list number positioning
- list font size set in CSS for UL/OL not working for first level list
- table width (real value, not %) not working in nested table
- GIF file failed if PDF file not compressed
- list-style-type incorrectly inherited
- line-height inheritance in lists
- SetColumns added a new line - not required if at start of document/page
- footer_line_spacing did not work
- table cellPadding="" overwrote cell padding set on cell CSS
- could not turn off Default non-HTML footer LINE
- border specified as "em"
- default values set in mpdf.css overridden by inherited properties e.g. `<div><h1>Here</h1></div>` lost font-size for H1

Printed on Wed 05 Aug 2015 12:36:49 GMT +0100 (DST)