

Choosing a configuration v5.x

(mPDF >= 5.0)

In mPDF there are a number of ways to configure your set-up. There is often a trade-off between file size, processing speed, appearance (support for different fonts), and reliability (i.e. ensuring that text is always displayed, at least in some form).

Some of the things you can change are:

- the initial parameter used to call mPDF e.g. `$mpdf=new mPDF('c')`
- configuration variables set in `config.php` see mPDF Variables - Overview (many of those configuration variable can also be set at runtime on a 'per-script' basis)
- font details in `config_fonts.php`
- language/font details in `config_cp.php`
- initial style settings in `config.php` or `mpdf.css`

Main choices

Some of the major considerations are:

- whether to restrict the document to core non-embedded fonts
- whether, or how, to subset embedded fonts
- handling of languages which require special fonts, including `auto_font` and character substitution

Core non-embedded fonts

PDF files have certain standard fonts: Arial/Helvetica, Times and Courier in the win-1252 character set, and Zapfdingbats and Symbol character sets. These fonts should be available to any PDF reading program, and do not need to be embedded in the PDF document.

Advantages: Small file size, fast processing, small memory usage.

Disadvantages: Limited choice of fonts for appearance. Will not display characters which are not in the win-1252 Symbols, or Dingbats codepages (suitable for most Western European languages).

To use core fonts only, use 'c' for the initial parameter:

```
$mpdf = new mPDF('c');
```

Embedded Unicode fonts

The alternative (the default setup) uses TrueType Unicode fonts, and the only limitation of characters to display is determined by the font files themselves.

Subsetting fonts

Fonts with good coverage of all characters you may require can be very large. If you embed the whole font file in the PDF document, the file can become very large - especially if you use a number of fonts. mPDF can embed subsets of the fonts i.e. just including the characters used in the PDF document.

Advantages of subsetting: Manageable file size (typically between 20-200kB)

Disadvantages of subsetting: Increase in processing time and memory usage (not always), as mPDF has to rebuild font files for each document.

By default, mPDF will embed subsets of fonts if less than 30% of the characters contained in the font are used in the document; otherwise it will embed the whole font file. You can override this by changing the configurable variable `$this->percentSubset`

For backwards compatibility, you can use '-s' or 's' in the initial parameter to override the `config.php` settings and force subsetting of all fonts e.g.

```
$mpdf = new mPDF('s');  
$mpdf = new mPDF('ar-s'); // also defining arabic language code  
$mpdf = new mPDF('utf8-s'); // for backwards compatibility - the utf-8 does nothing
```

Using core non-embedded fonts in a Unicode document

If your document uses Unicode fonts as above, you can force mPDF to use the core (non-embedded) PDF fonts in parts of the document by selecting the fontnames: `chelvetica`, `ccourier` and `ctimes` e.g.

```
<p style="font-family:chelvetica">This paragraph will use core fonts</p>
```

You could force mPDF to always use core fonts when Arial/Helvetica/Courier are specified, by editing the font translation variable `$this->fonttrans` in `config_fonts.php` e.g.:

```
$this->fonttrans = array(
    'arial' => 'chHelvetica',
    'helvetica' => 'chHelvetica',
    'timesnewroman' => 'cTimes',
    'times' => 'cTimes',
    'couriernew' => 'ccourier',
    'courier' => 'ccourier',
```

Languages which require special fonts

Indic languages always require special handling (cf. Indic fonts). Other than this, whether you need to do anything special is determined by the choice of fonts you use in the document, and whether they contain the necessary characters to display your text. The DejaVu fonts distributed with mPDF will usually display most Western and Eastern European languages, Cyrillic text, Baltic languages, Turkish, and Greek. Languages which usually need special consideration are: CJK (chinese - japanese - korean) languages, Vietnamese, Thai, and Arabic languages. With these, you need to tell mPDF to select a suitable font.

There are a number of different ways to do this:

1) Write your HTML code to specify the exact fonts needed:

```
<p style="font-family: Garuda">????????????????????????????????</p>
<p style="font-family: BIG5">????????</p>
<p style="font-family: sun-exta">????????</p>
<p style="font-family: 'XB Riyaz'">???????? ????</p>
<p style="font-family: ind_hi_1_001">???? ????</p>
```

2) Use the `lang` attribute to define the language. This causes mPDF to use a font as specified in the `config_cp.php` file. The current font-family will be used if it is available in the list defined as `$unifonts`, otherwise the first font specified in the `$unifonts` list will be selected.

```
<p lang="th">????????????????????????????????</p>
<p lang="zh-CN">????????</p>
<p lang="ar">???????? ????</p>
<p lang="hi">???? ????</p>
```

3) If you set the language in the initial call to mPDF i.e.

```
$mpdf = new mPDF('th');
$mpdf = new mPDF('zh-CN'); // You can append +aCJK or -aCJK
$mpdf = new mPDF('ar');
$mpdf = new mPDF('hi');
```

then mPDF will restrict the fonts you can use for the whole document. Any of the fonts defined in `$unifonts` can be used:

```
// If config_cp.php defines:
CASE "th": $spacing = "C";
$unifonts = "garuda,garudaB,garudaI,garudaBI,norasi,norasiB,norasiI,norasiBI";
break;
```

These paragraphs will appear in Garuda (a sans-serif font) and Norasi (serif) respectively:

```
<p style="font-family:sans">????????????????????????????????</p>
<p style="font-family:serif">????????????????????????????????</p>
```

4) You can also set the language for the whole document by setting:

```
<body lang="th">
<body lang="zh-CN">
<body lang="ar">
<body lang="hi">
```

5) Use `SetAutoFont()` to automatically detect these languages. AutoFont inspects the HTML code and inserts a `span` element to mark text which is auto-detected e.g.

```
$mpdf->SetAutoFont(AUTOFONT_ALL);
```

```
<p>This is Thai text: ?????????????????????????????????</p>
```

becomes:

```
<p>This is Thai text: <span lang="th">
```

```
class="lang_th">????????????????????????????????????????</span></p>
```

The `lang` attribute means that mPDF will select a font as described above in 2). You can also use CSS stylesheets to apply additional styles e.g. changing the font-size.

Do not use `AutoFont` as well as methods (1) - (4).

6) Use `$useSubstitutions` to use character substitution. mPDF will inspect every character in the HTML code, and if the character is not represented in the current font, it will try to substitute it from one of the fonts defined in `$this->backupSubsFont` in `config_fonts.php`.

```
$this->backupSubsFont = array('dejavusanscondensed','norasi');
```

Which method should I use?

Let us call Western and Eastern European languages, Cyrillic text, Baltic languages, Turkish, and Greek - **common languages**, and CJK and Indic languages, Vietnamese, Thai, and Arabic - **special languages**.

Individually authored pages

If you are authoring a document in a common language which contains sections of text using special languages, the ideal method to use is 1) or 2).

Document in a Special language

If you are writing documents entirely in a special language, you should use method 3) or 4), but you will be restricted to selecting from the fonts defined in `config_cp.php`

Multilingual documents 'on-the-fly'

If you are creating a PDF document from a page such as this web-page or a forum board which is likely to contain some special language text, and it is not possible to mark-up the special language text, you should use method 5) `AutoFont`.

If the document is mainly in a common language and may contain only occasional words or characters, the alternative is to use method 6) character substitution. This will be slower than 5) if there are substantial sections in special languages.

It is possible to use methods 5) and 6) together, to ensure that all special characters are displayed.

Multilingual website

If you are using a multilingual website (Wiki?) which has document in different languages, but where each page will be in only one language, and you have a common stylesheet etc. you can use method 3). Use the language code (en-GB, zh-CN) from the website page selected to set up the mPDF class.

If you want to use core non-embedded fonts when possible (for Western European languages), you can add '-x' to the language string. This will select core fonts only when the language string is appropriate (as defined in `config_cp.php`) e.g.

```
$mpdf = new mPDF('en-GB-x'); // will only use core non-embedded fonts
$mpdf = new mPDF('de-x');   // will only use core non-embedded fonts (German)
$mpdf = new mPDF('ar-x');   // behaves as though ('ar') called (Arabic)
$mpdf = new mPDF('ru-x');   // behaves as though ('ru') called (Russian)
```

See Also

- RTL & Bidirectional text
- `SetAutoFont()` - Automatically detect language in the input HTML text and use appropriate fonts

There is a useful list of language/country codes at: <http://www.i18nguy.com/unicode/language-identifiers.html>.

Printed on Wed 05 Aug 2015 12:04:37 GMT +0100 (DST)