**User's Guide » Fonts & Languages**

# Choosing a configuration v6.x

(mPDF >= 6.0)

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_lang2fonts.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 automatic font selection 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' in the initial parameter to override the `config.php` settings and force subsetting of all fonts e.g.

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

**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' => 'chelvetica',
 'helvetica' => 'chelvetica',
 'timesnewroman' => 'ctimes',
 'times' => 'ctimes',
 'couriernew' => 'ccourier',
 'courier' => 'ccourier',
```

## Languages/Scripts which require special fonts

Most browsers / PC applications automatically select / substitute appropriate fonts when required. mPDF does not do this by default (additional processing resources required). there are two situations when you need to consider the method of font selection:

- the text contains characters which are not covered by the initial font selected
- the text contains complex script requiring special treatment i.e. OpenType layout (OTL)

The DejaVu fonts distributed with mPDF contain characters (glyphs) to 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, Indic languages, Vietnamese, Thai, and Arabic languages. With these, you need to tell mPDF to select a suitable font.

There are several 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>
```

2) Write your HTML code using the `lang` attribute to define the language.

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

This needs to be used in conjunction with either:

- `autoLangToFont`
- CSS stylesheet using the `:lang` selector

3) Use `autoScriptToLang` to mark up HTML text by inserting the `lang` attribute, based on the Unicode script block in question, and configurable values in `config_script2lang.php`

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

As for (2) this needs to be used in conjunction with either:

- `autoLangToFont`
- CSS stylesheet using the `:lang` selector

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

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

## Which method should I use?

**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).

**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 3).

If the document is mainly in a common language and may contain only occasional words or characters, the alternative is to use method 4) character substitution.

It is possible to use method 4) together with 1) 2) or 3), to ensure that all special characters are displayed.

### See Also

- RTL & Bidirectional text

Printed on Wed 05 Aug 2015 12:04:44 GMT +0100 (DST)