

Впервые столкнувшись с задачей генерации документов PDF на сайте любой программист (надеюсь) начнет свой путь с поиска готовых решений, а именно библиотеки, которая ему эти возможности предоставит. Мне нужна была библиотека на PHP, чтобы не мудрить в дальнейшем, т. к. сайт работал на PHP. Хотелось бы еще чтобы она была бесплатной и позволяла свободно пользоваться ей в коммерческих целях. Нашел информацию по 3-м вариантам:

1.

Расширение PDFLib для PHP ( <http://www.pdflib.com> )

2.

Библиотека FPDF ( <http://www.fpdf.org> )

3.

Библиотека TCPDF ( <http://www.tcpdf.org> )

PDFLib в бесплатной версии не поддерживает utf-8, что плохо. С FPDF почти та же история, поэтому эти варианты я отбросил сразу и не стал с ними заморачиваться. Информации по TCPDF в сети достаточно много. Эта библиотека так же входит в состав многих популярных CMS.

Начнем.

Существуют версии для PHP4 и PHP5. Мне нужна пятерка.

Качаем: <http://sourceforge.net/projects/tcpdf/files/>

Распакуем архив. В каталоге tcpdf увидим следующий список файлов и каталогов:

```
cache config doc — документация (можно выкинуть для облегчения) examples — тоже  
можно выкинуть fonts images 2dbarcodes.php barcodes.php htmlcolors.php tcpdf.php  
unicode_data.php tcpdf.crt tcpdf.fdf CHANGELOG.TXT LICENSE.TXT README.TXT
```

TCPDF при генерации документа включает используемые шрифты в документ. Эти шрифты лежат в каталоге fonts. Каждому шрифту соответствуют три файла:

```
имя_шрифта.php имя_шрифта.z имя_шрифта.ctg.z
```

Кириллических шрифтов там нет, что нас, понятно, не устраивает. Если планируете использовать в документе кириллицу, можете удалить все кроме helvetica.php, т. к. он используется библиотекой по умолчанию, и папки utils, где находиться все необходимое для подготовки кириллических шрифтов.

Допустим нам понабился шрифт Arial нормального начертания. Берем шрифт Arial (начертание - normal) из системной папки Windows (например) – файл arial.ttf – и копируем его в папку fonts/utils.

В этой папке есть утилита ttf2ufm, которой мы должны скормить файл arial.ttf:

```
ttf2ufm -a -F arial.ttf
```

Далее запускаем скрипт makefont.php (предполагается, что путь к php есть в переменной окружения PATH системы):

```
php -q makefont.php arial.ttf arial.ufm
```

Если на вашей локальной машине нет интерпретатора PHP, можете выполнить на сервере файл следующего содержания:

```
<?php require('makefont.php'); MakeFont('times.ttf', 'times.ufm', true, 'iso-8859-5'); ?>
```

Если вы «скормили» несколько файлов и хотите сформировать сразу все шрифты (для которых созданы файлы \*.ufm), имеющиеся в папке, запустите makeallttffonts.php.

После работы данной утилиты у нас в папке utils появятся три файла: arial.php, arial.z, arial.ctg.z, которые мы должны будем переместить в папку fonts. Файлы arial.ttf и arial.ufm нам больше не нужны, поэтому можем смело их удалять.

В итоге мы подготовили шрифт Arial, который будет присоединен к формируемому PDF-файлу. Но при этом нужно учесть, что размер формируемого PDF-документа возрастет примерно на размер файла шрифта arial.z. Также необходимо учесть, что если мы хотим вывести надпись курсивно, нам нужно будет для этого создавать отдельный шрифт ariali, или если полужирным начертанием – arialb.

Пришло время создать наш документ.

```
<?php // подключаем библиотеку require_once('../tcpdf.php');
// create new PDF document $pdf = new TCPDF('P', 'mm', 'A4', true, 'UTF-8', false);
// set document information $pdf->SetAuthor('Ваше имя'); $pdf->SetTitle('Название нашего
документа');
// set default header data $pdf->SetHeaderData(PDF_HEADER_LOGO,
PDF_HEADER_LOGO_WIDTH, PDF_HEADER_TITLE, PDF_HEADER_STRING);
// set header and footer fonts $pdf->setHeaderFont(Array(PDF_FONT_NAME_MAIN, ",
PDF_FONT_SIZE_MAIN)); $pdf->setFooterFont(Array(PDF_FONT_NAME_DATA, ",
PDF_FONT_SIZE_DATA));
// set default monospaced font
$pdf->SetDefaultMonospacedFont(PDF_FONT_MONOSPACED);
//set margins $pdf->SetMargins(PDF_MARGIN_LEFT, PDF_MARGIN_TOP,
PDF_MARGIN_RIGHT); $pdf->SetHeaderMargin(PDF_MARGIN_HEADER);
$pdf->SetFooterMargin(PDF_MARGIN_FOOTER);
//set auto page breaks $pdf->SetAutoPageBreak(TRUE, PDF_MARGIN_BOTTOM);
// set font $pdf->SetFont('times', 'BI', 16);
// add a page $pdf->AddPage();
// print a line using Cell() $pdf->Cell(0, 12, 'Пример 1', 1, 1, 'C');
//Close and output PDF document $pdf->Output('example_1.pdf', 'I'); ?>
```

Это очень простой пример. В папке examples найдется много более сложных примеров, иллюстрирующих почти все возможности библиотеки.

Приведу так же несколько полезных ссылок:

- 1.

Документация по TCPDF [http://www.tecnick.com/public/code/cp\\_dpage.php?aiocp\\_dp=tcpdf\\_docs](http://www.tecnick.com/public/code/cp_dpage.php?aiocp_dp=tcpdf_docs)

2.

Документация по TCPDF в составе Joomla [http://www.tecnick.com/public/code/cp\\_dpage.php?aiocp\\_dp=tcpdf\\_docs](http://www.tecnick.com/public/code/cp_dpage.php?aiocp_dp=tcpdf_docs)

3.

Описание класса TCPDF <http://www.phpmyadmin.net/phpdoc/com-tecnick-tcpdf/TCPDF.html>

4.

Статья, которая мне очень помогла в свое время и материалами которой я пользовался при написании данной статьи <http://i-novice.net/generiruem-pdf-s-pomoshhyu-tcpdf/>

Харчук А. С. 2010 [\[kharchuk.ru\]](http://kharchuk.ru)