

## Programas necesarios

Para generar las nuevas fuentes es necesario disponer del programa **ttf2pt1.exe**. Si accedemos a [este enlace](#) podemos ver, en la zona de descargas, una opción dónde dice *Complete package, except sources* desde el que podemos descargar el fichero **ttf2pt1-3.4.4.exe**.

Al ejecutar este instalador, con la opción por defecto, nos creará un directorio llamado **GnuWin32** (en Archivos de Programa) dentro del cual encontramos un directorio llamado **bin** en el que se encuentra el ejecutable **ttf2pt1**.

El resto de las opciones de esta distribución no son necesarias para nuestros fines.

## Creación del fichero .afm

Este fichero, cuya extensión es el acrónimo de **Adobe Font Metric** contiene las información sobre ancho, alto y kerning de cada carácter de una fuente.

El primer paso ha de ser la creación de ese fichero. El procedimiento para hacerlo es el que tienes descrito al margen.

### ¡Cuidado!

En el directorio **fuentes** tienes los ficheros **cayge.afm** y **caygr.afm** generados según el proceso descrito al margen.

## Definición de las fuentes

El directorio **fontsPDF** de estos materiales (recuerda que es el fruto de renombrar el directorio **font** extraído del [fichero de instalación](#) de la clase FPFD) contiene otro subdirectorio (**makefont**) que incluye un script llamado **makefont.php**. La definición de fuentes requiere la ejecución de una de las funciones que contiene ese script llamada **MakeFont()** y que utiliza los siguientes parámetros:

*MakeFont(f1, f2, c, p, tipo)*

donde **f1** es el nombre y path de la fuente TrueType a utilizar (fichero **ttf**), **f2** es el nombre del fichero **afm**

## Proceso de generación de fuentes

Esta primera imagen es la visualización del directorio fuentes de estos materiales. Podrás observar que contiene fuentes TrueType y el ejecutable **ttf2pt1**.

Hemos incluido dos fuentes TrueType llamadas **bobcayge.ttf** y **bobcaygr.ttf**. Serán las que utilizaremos para añadir nuevas opciones de fuentes a nuestros PDF's.

arial.ttf	267 KB	Archivo de fuentes ...	31/12/2002 11:20
fuentes2.ttf	109 KB	Archivo de fuentes ...	31/12/2002 11:21
ttf2pt1.exe	162 KB	Aplicación	01/01/2004 11:06
bobcayge.ttf	28 KB	Archivo de fuentes ...	12/03/2002 19:22
bobcaygr.ttf	27 KB	Archivo de fuentes ...	12/03/2002 20:32

### ¡Cuidado!

No es necesario que realices la instalación de **ttf2pt1-3.4.4.exe** comentada al margen. En el subdirectorio **fuentes** de estos materiales ya dispones del fichero **ttf2pt1**.

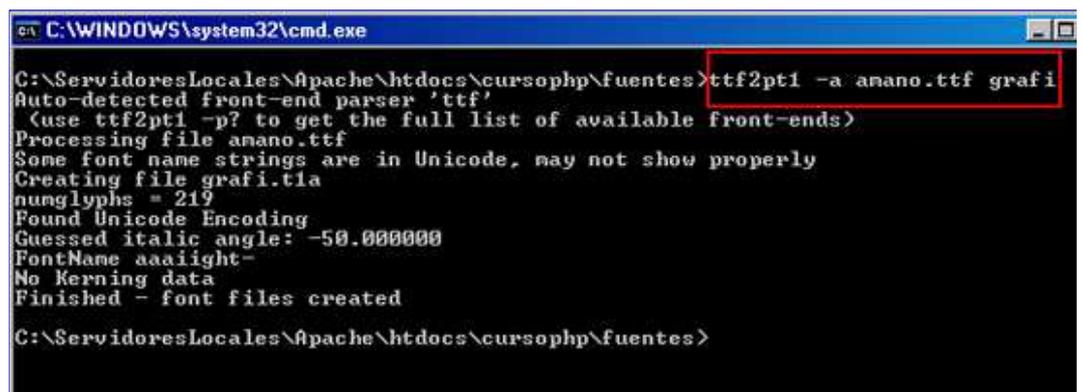
En el caso de Linux Ubuntu puedes instalarlo desde el terminal escribiendo:

```
sudo apt-get install ttf2pt1
```

## Creación del fichero .afm

Desde la ventana de MS-DOS y una vez elegido el directorio que contiene el ejecutable **ttf2pt1** podemos crear el fichero **.afm** usando la siguiente sintaxis:

**ttf2pt1 -nombre\_de\_la\_fuente\_TrueType.ttf nombre\_del\_nuevo\_fichero**



```
C:\WINDOWS\system32\cmd.exe
C:\ServidoresLocales\Apache\htdocs\cursophp\fuentes>ttf2pt1 -a amano.ttf grafi
Auto-detected front-end parser 'ttf'
(Use ttf2pt1 -p? to get the full list of available front-ends)
Processing file amano.ttf
Some font name strings are in Unicode, may not show properly
Creating file grafi.t1a
numglyphs = 219
Found Unicode Encoding
Guessed italic angle: -50.000000
FontName aaaiight-
No Kerning data
Finished - font files created
C:\ServidoresLocales\Apache\htdocs\cursophp\fuentes>
```

Nos generará dos ficheros con el nombre asignado al nuevo fichero y extensiones **.afm** y **.t1a** tal como puedes ver en la imagen.

amano.ttf	94 KB	Archivo de fuentes T...	31/12/2006 17:50
grafi.afm	13 KB	Archivo AFM	07/06/2009 8:27
grafi.t1a	305 KB	Archivo T1A	07/06/2009 8:27

## Creación del fichero de definición de fuente

Este es un ejemplo de script que permitiría definir nuevas fuentes mediante la utilidad **makefont**.

creado para esta misma fuente mediante **ttf2pt1**. Estos dos parámetros son obligatorios.

Mediante el parámetro **c** (opcional) se especifica el tipo de codificación (el mapa de fuentes a utilizar). Si no se indica el valor por defecto sería **cp1252**.

El parámetro opcional **c** permite cambiar la codificación de algunos caracteres, y, mediante **tipo** puede indicarse el tipo de fuente original (por defecto se asigna TrueType).

La definición de nuevas fuentes requiere la ejecución de la función MakeFont. Para ello, una de las opciones, es utilizar un script como el que ves a la derecha. En una primera instrucción se incluye el fichero makefont.php y en la segunda se hace una llamada a la función únicamente con los parámetros obligatorios.

La ejecución de ese script daría como resultado la creación de dos ficheros con el mismo nombre. Uno de ellos con extensión **.z** y el otro con extensión **.php**. Ambos, para que puedan ser usados por la clase FPDF, deberán ser incluidos en el directorio **fontsPDF**.

## Diferentes tipografías

Lo habitual es que entre las fuentes TrueType de un tipo determinado existan varios ficheros. Uno para la fuente normal, otro para la cursiva, etcétera. Cuando se trabaja con este tipo de fuentes y se pretende usar estilos distintos lo aconsejable es definir una fuente para cada tipo y tratarlos, a la hora de escribir el código de los script como fuentes independientes.

De esta forma podríamos tener parejas de ficheros del estilo: **fxn\_normal.z** y **fxn\_normal.php**; **fxn\_cursiva.z** y **fxn\_cursiva.php**, etcétera. A la hora de utilizarlas se haría como si se tratara de fuentes de distintas familias y habría de evitarse incluir el estilo en SetFont.

La sintaxis de la asignación de fuentes habría de ser:

```
$obj->SetFont("nombre", "", 9);
```

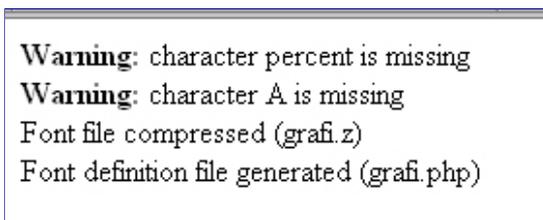
dejando el valor del segundo parámetro (estilo) como una cadena vacía. La única excepción a esta norma sería el caso del subrayado que no requiere otra cosa que asignar el valor U al parámetro estilo.

## Precauciones con las fuentes

El uso de fuentes externas puede producir efectos indeseados si, por alguna circunstancia, tal como ocurre en los ejemplos la fuente

```
<?
# incluimos el fichero makefont.php que está
# en la ruta que se especifica (si lo tuvieramos en otra
# ubicación bastaría cambiar el path del include
include($_SERVER['DOCUMENT_ROOT']."cursophp/fonts/pdf/makefont/makefont.php");
# ejecutamos la función MakeFont indicando como parametros
# la ruta y el fichero ttf a convertir (la fuente trueType)
# y como segundo parámetro la ruta y el nombre del fichero afm
# creado en el proceso anterior
MakeFont($_SERVER['DOCUMENT_ROOT']."/cursophp/fuentes/amano.ttf",
        $_SERVER['DOCUMENT_ROOT']."/cursophp/fuentes/grafi.afm");
?>
```

Al ejecutar el script anterior puede ocurrir que, dependiendo de la fuente ttf que estemos utilizando, aparezcan mensajes de advertencia como los que ves en esta imagen.



Significaría que algunos tipos (generalmente los asociados a los caracteres ASCII ampliados) no estarían disponibles y que, en consecuencia, o no se visualizarían o se visualizarían de forma incorrecta.

Una vez creados los ficheros como las nuevas fuentes (se crean en el mismo directorio en que hayamos ejecutado el script) es necesario incluirlos en el directorio **fontspdf** de modo que estén disponibles a la hora de crear los documentos PDF.



## Ejemplo de utilización

La utilización de nuevas fuentes requiere que

```
objeto->AddFont(nombre_a_utilizar, estilo, fichero_fuente.php)
```

donde el nombre\_a\_utilizar puede ser cualquiera, el parámetro estilo puede ser la cadena vacía y el tercer parámetro es el nombre y extensión del fichero .php que contiene la definición de la fuente y que fué creado por medio de makefont.

Si se omite el tercer parámetro será asignado, de forma automática, la cadena formada por la unión las cadenas **nombre\_a\_utilizar** y **estilo**, agregándole la extensión **.php**.

```
<?
#incluimos el fichero con la clase y definimos la variable FPDF_FONTPATH
# con el mismo criterio comentado en el ejemplo anterior
include("fpdf.php");
define('FPDF_FONTPATH', $_SERVER['DOCUMENT_ROOT'].'/cursophp/fonts/pdf/');
/* establecemos las dimensiones del documento
   creamos un nuevo objeto y añadimos una página*/
$dimensiones=array (140,200);
$miPDF=new FPDF('P','mm',$dimensiones);
# ajustamos al 100% la visualización
$miPDF->SetDisplayMode('real');
#insertamos una página en blanco
$miPDF->Addpage();
# establecemos el color de la letra
$miPDF->SetTextColor(255,0,0);
# añadimos la nueva fuente. Podemos ponerle un nombre cualquiera
# Al no incluir estilo considerará el estilo normal
$miPDF->AddFont('fuentenueva1', '', 'cayge.php');
```

caracteres (eñes, letras con tilde, símbolos). Conviene tenerlo en cuenta para evitar que la elección de una tipografía pueda afectar a los contenidos del documento.

### ¡Cuidado!

En el caso de Ubuntu el proceso es idéntico.

Bastará con abrir el terminal, situarse en el directorio **fuentes**.

Desde allí se ejecuta el comando **ttf2pt1** de forma idéntica a la descrita para MS-DOS.

```
# a indicar SetFont habremos de utilizar el mismo nombre de fuente
# establecido mediante AddFont
$MiPDF->SetFont('fuentenueva1','',24);
$MiPDF->Multicell(0,10,"Prueba de la fuente que hemos
                        llamado fuente nueva1",1,'L',0);

$MiPDF->Ln(10);
# incluimos la fuente caygr.php con nombre fuentenueva2
$MiPDF->AddFont('fuentenueva2','', 'grafi.php');
$MiPDF->SetFont('fuentenueva2','',24);
$MiPDF->Multicell(0,10,"Prueba de la fuente que hemos
                        llamado fuente nueva2",1,'R',0);

$MiPDF->Output();
?>
```

ejemplo144.php

En estos dos ejemplos pueden verse los resultados -con algunos problemas gráficos- de la sustitución de las fuentes en algunos de los ejemplos de páginas anteriores.

ejemplo145.php

ejemplo146.php

Anterior



Índice



Siguiente

