Fonts and FreeBSD

Abstract

This document contains a description of the various font files that may be used with FreeBSD and the syscons driver, X11, Ghostscript and Groff. Cookbook examples are provided for switching the syscons display to 80x60 mode, and for using type 1 fonts with the above application programs.

Table of Contents

1. Introduction . . . 1
2. Basic Terminology . . . 1
3. What Font Formats Can I Use? . . . 2
4. Setting a Virtual Console to 80x60 Line Mode . . . 3
5. Using Type 1 Fonts with X11 . . . 3
6. Using Type 1 Fonts with Ghostscript . . . 6
7. Using Type 1 Fonts with Groff . . . 7
8. Converting TrueType Fonts to a groff/PostScript Format For groff . . . 9
9. Can TrueType Fonts be Used with Other Programs? . . . 11
10. Where Can Additional Fonts be Obtained? . . . 11
11. Additional Questions . . . 12

1. Introduction

There are many sources of fonts available, and one might ask how they might be used with FreeBSD. The answer can be found by carefully searching the documentation for the component that one would like to use. This is very time consuming, so this tutorial is an attempt to provide a shortcut for others who might be interested.

2. Basic Terminology

There are many different font formats and associated font file suffixes. A few that will be addressed here are:

.pfa, .pfb

PostScript® type 1 fonts. The .pfa is the *A*scii form and .pfb the *B*inary form.

.afm

The font metrics associated with a type 1 font.

.pfm

The printer font metrics associated with a type 1 font.

.ttf

A TrueType® font

.fot

An indirect reference to a TrueType font (not an actual font)

.fon, .fnt

Bitmapped screen fonts

The .fot is used by Windows® as sort of a symbolic link to the actual TrueType® font (.ttf) file. The .fon font files are also used by Windows. I know of no way to use this font format with FreeBSD.

3. What Font Formats Can I Use?

Which font file format is useful depends on the application being used. FreeBSD by itself uses no fonts. Application programs and/or drivers may make use of the font files. Here is a small cross reference of application/driver to the font type suffixes:

Driver

vt

.hex

syscons

.fnt

Application

Ghostscript

.pfa, .pfb, .ttf

X11

.pfa, .pfb

Groff

.pfa, .afm

Povray

.ttf

The .fnt suffix is used quite frequently. I suspect that whenever someone wanted to create a specialized font file for their application, more often than not they chose this suffix. Therefore, it is likely that files with this suffix are not all the same format; specifically, the .fnt files used by syscons under FreeBSD may not be the same format as a .fnt one encounters in the MS-DOS®/Windows® environment. I have not made any attempt at using other .fnt files other than those provided with FreeBSD.

4. Setting a Virtual Console to 80x60 Line Mode

First, an 8x8 font must be loaded. To do this, /etc/rc.conf should contain the line (change the font name to an appropriate one for your locale):

```
font8x8="iso-8x8"       # font 8x8 from /usr/share/syscons/fonts/* (or NO).
```

The command to actually switch the mode is vidcontrol(1):

```
% vidcontrol VGA_80x60
```

Various screen-oriented programs, such as vi(1), must be able to determine the current screen dimensions. As this is achieved this through `ioctl` calls to the console driver (such as syscons(4)) they will correctly determine the new screen dimensions.

To make this more seamless, one can embed these commands in the startup scripts so it takes place when the system boots. To do this is add this line to /etc/rc.conf.

```
allscreens_flags="VGA_80x60"    # Set this vidcontrol mode for all virtual screens
```

References: rc.conf(5), vidcontrol(1).

5. Using Type 1 Fonts with X11

X11 can use either the .pfa or the .pfb format fonts. The X11 fonts are located in various subdirectories under /usr/X11R6/lib/X11/fonts. Each font file is cross referenced to its X11 name by the contents of fonts.dir in each directory.

There is already a directory named Type1. The most straight forward way to add a new font is to put it into this directory. A better way is to keep all new fonts in a separate directory and use a symbolic link to the additional font. This allows one to more easily keep track of ones fonts without confusing them with the fonts that were originally provided. For example:

```
Create a directory to contain the font files
% mkdir -p /usr/local/share/fonts/type1
% cd /usr/local/share/fonts/type1

Place the .pfa, .pfb and .afm files here

One might want to keep readme files, and other documentation

for the fonts here also
% cp /cdrom/fonts/atm/showboat/showboat.pfb .
```

```
% cp /cdrom/fonts/atm/showboat/showboat.afm .

Maintain an index to cross reference the fonts
% echo showboat - InfoMagic CICA, Dec 1994, /fonts/atm/showboat >>INDEX
```

Now, to use a new font with X11, one must make the font file available and update the font name files. The X11 font names look like:

```
-bitstream-charter-medium-r-normal-xxx-0-0-0-0-p-0-iso8859-1
     |        |      |    |   |     | | | | | | |    \    \
     |        |      |    |   |     \ \ \ \ \ \ \     +----+- character set
     |        |      |    |   \      \  \ \ \ \ \ +- average width
     |        |      |    |    \      \  \ \ \ \ +- spacing
     |        |      |    \ \      \  \ \ \ +- vertical res.
     |        |      |     \     \  \  \ \ +- horizontal res.
     |        |      |      \     \  \  \ +- points
     |        |      |       \     \      \  +- pixels
     |        |      |        \     \      \
  foundry  family  weight   slant  width  additional style
```

A new name needs to be created for each new font. If you have some information from the documentation that accompanied the font, then it could serve as the basis for creating the name. If there is no information, then you can get some idea by using strings(1) on the font file. For example:

```
% strings showboat.pfb | more
%!FontType1-1.0: Showboat 001.001
%%CreationDate: 1/15/91 5:16:03 PM
%%VMusage: 1024 45747
% Generated by Fontographer 3.1
% Showboat
 1991 by David Rakowski.  Alle Rechte Vorbehalten.
FontDirectory/Showboat known{/Showboat findfont dup/UniqueID known{dup
/UniqueID get 4962377 eq exch/FontType get 1 eq and}{pop false}ifelse
{save true}{false}ifelse}{false}ifelse
12 dict begin
/FontInfo 9 dict dup begin
 /version (001.001) readonly def
 /FullName (Showboat) readonly def
 /FamilyName (Showboat) readonly def
 /Weight (Medium) readonly def
 /ItalicAngle 0 def
 /isFixedPitch false def
 /UnderlinePosition -106 def
 /UnderlineThickness 16 def
 /Notice (Showboat
 1991 by David Rakowski.  Alle Rechte Vorbehalten.) readonly def
end readonly def
/FontName /Showboat def
```

```
--stdin--
```

Using this information, a possible name might be:

```
-type1-Showboat-medium-r-normal-decorative-0-0-0-0-p-0-iso8859-1
```

The components of our name are:

Foundry
: Lets just name all the new fonts `type1`.

Family
: The name of the font.

Weight
: Normal, bold, medium, semibold, etc. From the strings(1) output above, it appears that this font has a weight of *medium*.

Slant
: *r*oman, *i*talic, *o*blique, etc. Since the *ItalicAngle* is zero, *roman* will be used.

Width
: Normal, wide, condensed, extended, etc. Until it can be examined, the assumption will be *normal*.

Additional style
: Usually omitted, but this will indicate that the font contains decorative capital letters.

Spacing
: proportional or monospaced. *Proportional* is used since *isFixedPitch* is false.

All of these names are arbitrary, but one should strive to be compatible with the existing conventions. A font is referenced by name with possible wild cards by an X11 program, so the name chosen should make some sense. One might begin by simply using

```
...-normal-r-normal-...-p-...
```

as the name, and then use xfontsel(1) to examine it and adjust the name based on the appearance of the font.

So, to complete our example:

```
Make the font accessible to X11
% cd /usr/X11R6/lib/X11/fonts/Type1
% ln -s /usr/local/share/fonts/type1/showboat.pfb .
```

```
Edit fonts.dir and fonts.scale, adding the line describing the font
and incrementing the number of fonts which is found on the first line.
% ex fonts.dir
:1p
25
:1c
26
.
:$a
showboat.pfb -type1-showboat-medium-r-normal-decorative-0-0-0-0-p-0-iso8859-1
.
:wq

fonts.scale seems to be identical to fonts.dir...
% cp fonts.dir fonts.scale

Tell X11 that things have changed
% xset fp rehash

Examine the new font
% xfontsel -pattern -type1-*
```

References: xfontsel(1), xset(1), The X Windows System in a Nutshell, O'Reilly & Associates.

6. Using Type 1 Fonts with Ghostscript

Ghostscript references a font via its Fontmap. This must be modified in a similar way to the X11 fonts.dir. Ghostscript can use either the .pfa or the .pfb format fonts. Using the font from the previous example, here is how to use it with Ghostscript:

```
Put the font in Ghostscript's font directory
% cd /usr/local/share/ghostscript/fonts
% ln -s /usr/local/share/fonts/type1/showboat.pfb .

Edit Fontmap so Ghostscript knows about the font
% cd /usr/local/share/ghostscript/4.01
% ex Fontmap
:$a
/Showboat        (showboat.pfb) ; % From CICA /fonts/atm/showboat
.
:wq

Use Ghostscript to examine the font
% gs prfont.ps
Aladdin Ghostscript 4.01 (1996-7-10)
Copyright (C) 1996 Aladdin Enterprises, Menlo Park, CA.  All rights
reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
Loading Times-Roman font from /usr/local/share/ghostscript/fonts/tir_____.pfb...
```

```
 /1899520 581354 1300084 13826 0 done.
GS>Showboat DoFont
Loading Showboat font from /usr/local/share/ghostscript/fonts/showboat.pfb...
 1939688 565415 1300084 16901 0 done.
>>showpage, press <return> to continue<<
>>showpage, press <return> to continue<<
>>showpage, press <return> to continue<<
GS>quit
```

References: fonts.txt in the Ghostscript 4.01 distribution

7. Using Type 1 Fonts with Groff

Now that the new font can be used by both X11 and Ghostscript, how can one use the new font with groff? First of all, since we are dealing with type 1 PostScript® fonts, the groff device that is applicable is the *ps* device. A font file must be created for each font that groff can use. A groff font name is just a file in /usr/share/groff_font/devps. With our example, the font file could be /usr/share/groff_font/devps/SHOWBOAT. The file must be created using tools provided by groff.

The first tool is `afmtodit`. This is not normally installed, so it must be retrieved from the source distribution. I found I had to change the first line of the file, so I did:

```
% cp /usr/src/gnu/usr.bin/groff/afmtodit/afmtodit.pl /tmp
% ex /tmp/afmtodit.pl
:1c
#!/usr/bin/perl -P-
.
:wq
```

This tool will create the groff font file from the metrics file (.afm suffix.) Continuing with our example:

```
Many .afm files are in Mac format... ^M delimited lines
We need to convert them to UNIX(R) style ^J delimited lines
% cd /tmp
% cat /usr/local/share/fonts/type1/showboat.afm |
    tr '\015' '\012' >showboat.afm

Now create the groff font file
% cd /usr/share/groff_font/devps
% /tmp/afmtodit.pl -d DESC -e text.enc /tmp/showboat.afm generate/textmap SHOWBOAT
```

The font can now be referenced with the name SHOWBOAT.

If Ghostscript is used to drive the printers on the system, then nothing more needs to be done. However, if true PostScript® printers are used, then the font must be downloaded to the printer in order for the font to be used (unless the printer happens to have the showboat font built in or on an

accessible font disk.) The final step is to create a downloadable font. The pfbtops tool is used to create the .pfa format of the font, and download is modified to reference the new font. The download must reference the internal name of the font. This can easily be determined from the groff font file as illustrated:

```
Create the .pfa font file
% pfbtops /usr/local/share/fonts/type1/showboat.pfb >showboat.pfa
```

Of course, if .pfa is already available, just use a symbolic link to reference it.

```
Get the internal font name
% fgrep internalname SHOWBOAT
internalname Showboat
Tell groff that the font must be downloaded
% ex download
:$a
Showboat      showboat.pfa
.
:wq
```

To test the font:

```
% cd /tmp
% cat >example.t <<EOF
.sp 5
.ps 16
This is an example of the Showboat font:
.br
.ps 48
.vs (\n(.s+2)p
.sp
.ft SHOWBOAT
ABCDEFGHI
.br
JKLMNOPQR
.br
STUVWXYZ
.sp
.ps 16
.vs (\n(.s+2)p
.fp 5 SHOWBOAT
.ft R
To use it for the first letter of a paragraph, it will look like:
.sp 50p
\s(48\f5H\s0\fRere is the first sentence of a paragraph that uses the
showboat font as its first letter.
Additional vertical space must be used to allow room for the larger
letter.
```

```
EOF
% groff -Tps example.t >example.ps

To use ghostscript/ghostview
% ghostview example.ps

To print it
% lpr -Ppostscript example.ps
```

References: /usr/src/gnu/usr.bin/groff/afmtodit/afmtodit.man, groff_font(5), groff_char(7), pfbtops(1).

8. Converting TrueType Fonts to a groff/PostScript Format For groff

This potentially requires a bit of work, simply because it depends on some utilities that are not installed as part of the base system. They are:

`ttf2pf`

TrueType to PostScript conversion utilities. This allows conversion of a TrueType font to an ascii font metric (.afm) file.

Currently available at http://sunsite.icm.edu.pl/pub/GUST/contrib/BachoTeX98/ttf2pf/. Note: These files are PostScript programs and must be downloaded to disk by holding down `Shift` when clicking on the link. Otherwise, your browser may try to launch ghostview to view them.

The files of interest are:

- GS_TTF.PS
- PF2AFM.PS
- ttf2pf.ps

The funny upper/lower case is due to their being intended also for DOS shells. ttf2pf.ps makes use of the others as upper case, so any renaming must be consistent with this. (Actually, GS_TTF.PS and PFS2AFM.PS are supposedly part of the Ghostscript distribution, but it is just as easy to use these as an isolated utility. FreeBSD does not seem to include the latter.) You also may want to have these installed to /usr/local/share/groff_font/devps(?).

`afmtodit`

Creates font files for use with groff from ascii font metrics file. This usually resides in the directory, /usr/src/contrib/groff/afmtodit, and requires some work to get going.

If you are paranoid about working in the /usr/src tree, simply copy the contents of the above directory to a work location.

In the work area, you will need to make the utility. Just type:

```
# make -f Makefile.sub afmtodit
```

You may also need to copy /usr/contrib/groff/devps/generate/textmap to /usr/share/groff_font/devps/generate if it does not already exist.

Once all these utilities are in place, you are ready to commence:

1. Create .afm by typing:

   ```
   % gs -dNODISPLAY -q -- ttf2pf.ps TTF_name PS_font_name AFM_name
   ```

   Where, *TTF_name* is your TrueType font file, *PS_font_name* is the file name for .pfa, *AFM_name* is the name you wish for .afm. If you do not specify output file names for the .pfa or .afm files, then default names will be generated from the TrueType font file name.

   This also produces a .pfa, the ascii PostScript font metrics file (.pfb is for the binary form). This will not be needed, but could (I think) be useful for a fontserver.

   For example, to convert the 30f9 Barcode font using the default file names, use the following command:

   ```
   % gs -dNODISPLAY -- ttf2pf.ps 3of9.ttf
   Aladdin Ghostscript 5.10 (1997-11-23)
   Copyright (C) 1997 Aladdin Enterprises, Menlo Park, CA.  All rights reserved.
   This software comes with NO WARRANTY: see the file PUBLIC for details.
   Converting 3of9.ttf to 3of9.pfa and 3of9.afm.
   ```

   If you want the converted fonts to be stored in A.pfa and B.afm, then use this command:

   ```
   % gs -dNODISPLAY -- ttf2pf.ps 3of9.ttf A B
   Aladdin Ghostscript 5.10 (1997-11-23)
   Copyright (C) 1997 Aladdin Enterprises, Menlo Park, CA.  All rights reserved.
   This software comes with NO WARRANTY: see the file PUBLIC for details.
   Converting 3of9.ttf to A.pfa and B.afm.
   ```

2. Create the groff PostScript file:

   Change directories to /usr/share/groff_font/devps so as to make the following command easier to execute. You will probably need root privileges for this. (Or, if you are paranoid about working there, make sure you reference the files DESC, text.enc and generate/textmap as being in this directory.)

   ```
   % afmtodit -d DESC -e text.enc file.afm generate/textmap PS_font_name
   ```

   Where, file.afm is the *AFM_name* created by `ttf2pf.ps` above, and *PS_font_name* is the font

name used from that command, as well as the name that groff(1) will use for references to this font. For example, assuming you used the first `tiff2pf.ps` above, then the 3of9 Barcode font can be created using the command:

```
% afmtodit -d DESC -e text.enc 3of9.afm generate/textmap 3of9
```

Ensure that the resulting *PS_font_name* file (e.g., 3of9 in the example above) is located in the directory /usr/share/groff_font/devps by copying or moving it there.

Note that if ttf2pf.ps assigns a font name using the one it finds in the TrueType font file and you want to use a different name, you must edit the .afm prior to running `afmtodit`. This name must also match the one used in the Fontmap file if you wish to pipe groff(1) into gs(1).

9. Can TrueType Fonts be Used with Other Programs?

The TrueType font format is used by Windows, Windows 95, and Mac's. It is quite popular and there are a great number of fonts available in this format.

Unfortunately, there are few applications that I am aware of that can use this format: Ghostscript and Povray come to mind. Ghostscript's support, according to the documentation, is rudimentary and the results are likely to be inferior to type 1 fonts. Povray version 3 also has the ability to use TrueType fonts, but I rather doubt many people will be creating documents as a series of raytraced pages :-).

This rather dismal situation may soon change. The FreeType Project is currently developing a useful set of FreeType tools:

- The `xfsft` font server for X11 can serve TrueType fonts in addition to regular fonts. Though currently in beta, it is said to be quite usable. See Juliusz Chroboczek's page for further information. Porting instructions for FreeBSD can be found at Stephen Montgomery's software page.
- xfstt is another font server for X11, available under ftp://sunsite.unc.edu/pub/Linux/X11/fonts/.
- A program called `ttf2bdf` can produce BDF files suitable for use in an X environment from TrueType files. Linux binaries are said to be available from ftp://crl.nmsu.edu/CLR/multiling/General/.
- and others ...

10. Where Can Additional Fonts be Obtained?

Many fonts are available on the Internet. They are either entirely free, or are share-ware. In addition many fonts are available in the x11-fonts/ category in the ports collection

11. Additional Questions

- What use are the .pfm files?
- Can one generate the .afm from a .pfa or .pfb?
- How to generate the groff character mapping files for PostScript fonts with non-standard character names?
- Can xditview and devX?? devices be set up to access all the new fonts?
- It would be good to have examples of using TrueType fonts with Povray and Ghostscript.