TrueType Fonts in Debian mini−HOWTO

Table of Contents

TrueType Fonts in Debian mini−HOWTO

Bear Giles, `bgiles@coyotesong.com`

v0.3, 10 January 2000

This document describes how to configure a Debian system to use TrueType fonts for display and printing. The most recent version of this document can be obtained in HTML format at `http://www.dimensional.com/~bgiles/debian-tt.html`

---

1. Introduction

A Linux installation typically contains several independent sets of fonts or font metrics. A quick glance at my system shows fonts or font metrics spread across the following directories:

- `XFree86` stores its fonts in `/usr/X11R6/lib/X11/fonts/`
- `Ghostscript` stores its fonts in `/usr/lib/ghostscript/fonts/`
- `TeX` stores its fonts in `/usr/lib/texmf/fonts/`
- The Debian `kbd` package stores its fonts in `/usr/share/consolefonts/`
- `Groff` stores its metrics in `/usr/share/font/devps/`
- `Enscript` (ASCII to PostScript converter) stores its metrics in `/usr/share/enscript/*.afm`

Needless to say, these fonts are not coordinated.

TrueType fonts are designed to eliminate this problem by allowing the same font files to be used for both display and printing. This HOW−TO attempts to show how to use TrueType fonts for display, printing, and document preparation on Debian GNU/Linux systems.

1.1 Disclaimer

The information in this document is, to the best of my knowledge, correct. However this HOWTO is still in preliminary versions and what works for me may not work for you. Even if it does work for you, I am not a professional technical writer and I have been known to gloss over critical details.

So have fun, but play it safe and keep backups.

1.2 Credits

Created by Bear Giles, `<bgiles@coyotesong.com>`

Thanks go to:

- Brion Vibber, `<brion@pobox.com>`, who wrote the `(preliminary) TrueType HOW−TO (http://pobox.com/~brion/linux/TrueType−HOWTO.html)`

- Doug Holland, `<meldroc@frii.com>`, who wrote the `XFree86 Font Deuglification HOW−TO (http://www.frii.com/~meldroc/Font−Deuglification.html)`

1.3 Additional links

These links don't directly discuss Debian systems or packages, but they may still be interesting to readers of this mini−HOWTO.

- Using TrueType Fonts with RedHat Linux `(http://www.kegel.com/linux/tt.html)`
- X Font Tools `(http://www.hex.net/~cbbrowne/xfonts.html)`
- Getting fonts to look pretty under GNU/Linux for applications like StarOffice `(http://king.ccrc.wustl.edu/~zubin/fonts.html)`
- How to use True Type © fonts for StarOffice Under Linux `(http://www.mindspring.net/~john_mcl/adding_fonts.html)`

1.4 Change log

- 0.1. Sept. 16, 1999: First release.
- 0.2. Sept. 16, 1999: added "additional links" section.
- 0.3. Jan. 10, 2000: reader feedback! Added clarification of several details.

1.5 Feedback

Comments, corrections, additions and critiques are always welcome. You can reach me at bgiles@coyotesong.com.

1.6 Related projects

- FreeType http://www.freetype.org/

  The FreeType engine is a free and portable TrueType font rendering engine. The code is a clean−room implementation that is totally independent of the Apple and Microsoft implementations. (A question has recently been raised about a possible Apple *patent*, however.) FreeType is a library, not a font server or a complete text rendering library.

1.7 Distribution

This is the first draft and I expect it to change significantly after publication, I ask that you refer to the latest version at `http://www.dimensional.com/~bgiles/debian−tt.html`. The permanent home for this document will eventually be at `http://www.coyotesong.com/`

---

2. Display (what they never told you)

Before we dive into setting up TrueType fonts under X, we should review the difference between points and pixels... and why we care.

All displayed fonts are measured in *points*. One inch is exactly 72 points. Why 72? Partly because of the limitations of mechanical typesetting machines and partly because it's evenly divisible by 2, 3, 4, 6, 8, 9, 12, 18 and 24. It's also worth noting that the default unit in PostScript is one point.

(Historical note: I lied. Until the introduction of PostScript an inch was exactly 72.27 points, but that point size was set in the era of mechanical printers with metal stamps in a handful of standard sizes. With computer displays and laser printers, it's easy to get fonts in any size and 72 makes *much* more sense for the reason mentioned above.)

As a general rule, most text should be between 7 and 12 points. Anything smaller than 6 points is literally "the fine print." Line printers used 9 or 12 point type (for 8 or 6 lines/inch, respectively).

In contrast, all video drivers must ultimately measure fonts in *pixels*. To your video driver, your screen is 1024x800 pixels, not 10 by 8 inches (or 720 by 576 points).

To map from points (which we use to specify a font size) to pixels (which we use to blit the video memory) we must know our screen's resolution. This is usually measured in "dots per inch (dpi)," although it's really pixels per inch. These are the units used in the two sets of bitmap fonts included with XFree86: `fonts−75` is intended for use on low−end displays with a resolution of approximately 75 dpi, `fonts−100` is intended for use on mid−range displays with a resolution of approximately 100 dpi. There are no bitmap fonts intended for use on high−end displays with a resolution of over 120 dpi.

As a concrete example, a 13" diagonal screen (11.1" usable) displaying a 640x480 pixel image has a resolution of 72.0 dpi. This Is Not A Coincidence. In fact, most web pages (and Microsoft applications) are designed around a canonical display with a resolution of exactly 72 dpi. XFree86's default configuration assumes a display with a resolution of 75 dpi.

Back in the real world, nobody runs 640x480 video anymore. Nobody uses 13" diagonal screens anymore. Since video cards have improved faster than video monitors it's not uncommon to have a configuration like mine: 19" diagonal screen, (17+" usable), 1600x1200 pixels, 117 dpi resolution.

If I run a stock X configuration, all of my fonts are approximately 2/3 of the intended size. It's not an exaggeration to say that all fonts are cut down by a full size: large fonts (12 pts) appear medium (9). Medium fonts (9 pts) appear small (6).

There are three things we can do to fix this. First, We should tell the X server our actual screen resolution:

/etc/X11/xdm/Xservers

```
#:0 local /usr/X11R6/bin/X -bpp 16
:0 local /usr/X11R6/bin/X -bpp 16 -dpi 120
```

Second, we should ensure that we use the 100 dpi bitmapped fonts in preference to the 75 dpi fonts.

/etc/X11/XF86Config

```
Section "Files"
    RgbPath    "/usr/X11R6/lib/X11/rgb"
```

```
    FontPath   "/usr/X11R6/lib/X11/fonts/100dpi/"
    FontPath   "/usr/X11R6/lib/X11/fonts/75dpi/"
    FontPath   "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath   "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath   "/usr/X11R6/lib/X11/fonts/Speedo/"
EndSection
```

Finally, if one of our bitmapped fonts isn't an exact match the `X` server will attempt to "scale" a similar font via pixel replication. The results are rarely pleasant to use. Assuming we have a reasonably powerful system, we can tell the server to use scaled fonts by default, then bitmaps which are exact matches, and scaled bitmaps as a last resort.

/etc/X11/XF86Config

```
Section "Files"
    RgbPath    "/usr/X11R6/lib/X11/rgb"
    FontPath   "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath   "/usr/X11R6/lib/X11/fonts/Speedo/"
    FontPath   "/usr/X11R6/lib/X11/fonts/100dpi/:unscaled"
    FontPath   "/usr/X11R6/lib/X11/fonts/75dpi/:unscaled"
    FontPath   "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath   "/usr/X11R6/lib/X11/fonts/100dpi/"
    FontPath   "/usr/X11R6/lib/X11/fonts/75dpi/"
EndSection
```

2.1 A look ahead to XFree86 4.0

I've just been informed that XFree86 will support DDC communications, if the video card and monitor both support it. This will allow the X server to query the monitor for its physical dimensions and the server will automatically compute the correct DPI setting.

You will still need to make the appropriate changes to your font path, since the server should not second guess your explicit configuration.

2.2 Review: Preparation before changing `X` configuration files.

Any time we make changes to the X11 configuration files, it's an excellent idea to disable `XDM` by putting `exit 0` somewhere near the top of the file. If you don't do this and `X` is unable to start for some reason, `XDM` will put your system into a nasty busy loop that is an unspeakable pain to correct. You Have Been Warned.

3. Display (`X` and the font servers)

Strictly speaking, it is not absolutely necessary to set up font servers to use TrueType fonts with `X` servers. If you wish to use static files instead of a font server, please see the instructions for setting up TrueType fonts for `ghostscript`.

3.1 Setting up the `xfs` Font Server

At this point I assume you have a working `/etc/X11/XF86Config` file that loads explicitly specifies each directory in the `FontPath`. We will convert it to use the `xfs` Font Server.

- Install `XFS`

  If you have not already done so, install `main/binary-*/x11/xfs_*.deb`.

- Configure `XFS`

  Edit `/etc/X11/xfs/config` and change the `catalogue` to contain the contents of your `FontList`. You may also wish to change the `default-resolutions` value.

  /etc/X11/xfs/config

  ```
  # paths to search for fonts
  catalogue =
      /usr/X11R6/lib/X11/fonts/100dpi/:unscaled,
      /usr/X11R6/lib/X11/fonts/75dpi/:unscaled,
      /usr/X11R6/lib/X11/fonts/misc/,
      /usr/X11R6/lib/X11/fonts/Speedo/,
      /usr/X11R6/lib/X11/fonts/Type1/,
      /usr/X11R6/lib/X11/fonts/100dpi/,
      /usr/X11R6/lib/X11/fonts/75dpi/
  # x1,y1,x2,y2,...
  default-resolutions = 100,100,75,75
  ```

- Restart `XFS`

  Restart `XFS` in the usual Debian manner:

  root shell

  ```
  # /etc/init.d/xfs restart
  ```

- Verify that `XFS` is working

  Before we change our `XF86Config` file, we should verify that the `xfs` server is working by listing them through the font server:

  user shell

  ```
  $ fslsfonts -server unix/:7100
  -adobe-courier-bold-i-normal--0-0-0-0-m-0-iso8859-1
  -adobe-courier-bold-o-normal--0-0-100-100-m-0-iso8859-1
  -adobe-courier-bold-o-normal--0-0-75-75-m-0-iso8859-1
  ```

```
....
```

- Change `/etc/X11/XF86Config` to use `xfs`

  We're now ready to tell the X server to use the `xfs` font server. We keep one static font path as a fallback position in case of a problem with `xfs`. (We keep "misc" since it contains `fixed`, the default font.)

  /etc/X11/XF86Config

```
Section "Files"
    FontPath   "unix/:7100"
    FontPath   "/usr/X11R6/lib/X11/fonts/misc/"
EndSection
```

- Restart `X`

  Restart `X` with `startx`, unless you like living life dangerously. If that's the case, name me the beneficiary of your unusually large life insurance policy and restart `xdm`.

- Verify that the `XFS` fonts are visible

  Once we have an X session established, we can verify that our server sees all of the `xfs` fonts by listing them through the `X` server.

  user shell

```
$ xlsfonts
-adobe-courier-bold-i-normal--0-0-0-0-m-0-iso8859-1
-adobe-courier-bold-o-normal--0-0-100-100-m-0-iso8859-1
-adobe-courier-bold-o-normal--0-0-75-75-m-0-iso8859-1
....
```

3.2 Setting up the `xfstt` Font Server and installing TrueType fonts

At this point I assume you have a working `xfs` font server and wish to add TrueType support via `xfstt`.

- Install `XFSTT`

  If you have not already done so, install `main/binary-*/x11/xfstt_*.deb`.

- Installing the TrueType Fonts

  Copy your TrueType fonts into the `/usr/share/fonts/truetype` directory. These files usually have a `.ttf` extension, and they should have `0444` permissions.

* Restart the `XFSTT` Font Server

  Restart the `xfstt` server with the `force-reload` flag

  root shell

  ```
  # /etc/init.d/xfstt force-reload
  ```

* Verify that `XFSTT` is working

  Before we change our `XF86Config` file, we should verify that the `xfstt` server is working.

  Important: the Debian `xfstt` server listens to port 7101, not 7100. Also, the default permissions will require you to run this query as root.

  user and root shells

  ```
  $ fslsfonts -server unix/:7101
  _FSTransSocketUNIXConnect: Can't connect: errno = 111
  fslsfonts:  unable to open server "unix/:7101"

  # fslsfonts -server unix/:7101
  -ttf-arial black-medium-r-normal-regular-0-0-0-0-p-0-iso8859-1
  -ttf-arial mt black-medium-r-normal-regular-0-0-0-0-p-0-iso8859-1
  -ttf-arial narrow-bold-i-normal-bold italic-0-0-0-0-p-0-iso8859-1
  ```

* Change `/etc/X11/XF86Config` to use `xfstt`

  We're now ready to tell the X server to use the `xfstt` font server. We want it to use TrueType fonts in preference to all others.

  /etc/X11/XF86Config

  ```
  Section "Files"
      FontPath   "unix/:7101"
      FontPath   "unix/:7100"
      FontPath   "/usr/X11R6/lib/X11/fonts/misc/"
  EndSection
  ```

* Verify that `XFSTT` fonts are visible

  Once we have an X session established, we can verify that our server sees all of the TrueType fonts by listing them.

  ```
  $ xlsfonts | grep ttf
  -ttf-arial black-medium-r-normal-regular-0-0-0-0-p-0-iso8859-1
  ```

```
-ttf-arial mt black-medium-r-normal-regular-0-0-0-0-p-0-iso8859-1
-ttf-arial narrow-bold-i-normal-bold italic-0-0-0-0-p-0-iso8859-1
....
```

3.3 Using TrueType fonts

At this point it should be possible to use a TrueType font in applications like `GIMP`, `Netscape` or `StarOffice`. Since most TrueType fonts aren't monospaced you probably don't want to use one of them with `xterm` – these programs use monospacing the size of the largest character cell.

3.4 Installing additional TrueType fonts

If you are using the `xfstt` font server, it's trivial to install additional TrueType fonts.

- Copy the new font(s) into `/usr/share/fonts/truetype/`
- Restart `xfs` with `/etc/init.d/xfs restart`

3.5 Internationalization

`xfstt` has the ability to generate multiple font encodings, provided that the TrueType font contains the necessary glyphs. To enable fonts other than iso8859−1/unicode−1, you must manually edit the `/etc/init.d/xfstt` script:

`/etc/init.d/xfstt`

```
- start-stop-daemon --start --quiet --exec $XFSTT -- \
      --port $portno --daemon
+ start-stop-daemon --start --quiet --exec $XFSTT -- \
      --port $portno --encoding iso8859-1,koi8-r,windows-1252,symbol-0 \
      --daemon
```

The recognized encodings in Debian 2.1 are:

- iso8859−1 (Latin 1 – Western Europe)
- iso8859−2 (Latin 2 – Central and Eastern Europe)
- iso8859−3 (Latin 3 – Esperanto and Maltese)
- iso8859−4 (Latin 4 – superceded by Latin 6)
- iso8859−5 (Cyrillic)
- iso8859−6 (Arabic)
- iso8859−7 (Greek)
- iso8859−8 (Hebrew)
- iso8859−9 (Latin 5 – Latin 1 with Turkish, not Icelandic)
- iso8859−10 (Latin 6 – Nordic languages, replaces Latin 4)
- koi8−r (Cyrillic)
- windows−1250 (Central Europe)
- windows−1251 (Cyrillic)
- windows−1252

- windows−1253 (Greek)
- windows−1254
- windows−1255
- windows−1256
- windows−1257
- symbol−0
- wingding−0
- wingreek−0
- cp−437 (various IBM code pages)
- cp−737
- cp−850
- cp−851
- cp−852
- cp−853
- cp−855
- cp−857
- cp−860
- cp−861
- cp−862
- cp−863
- cp−864
- cp−865
- cp−866
- cp−869
- cp−895
- atari−st
- unicode−2

The first 128 characters in the `iso8859-x` encodings is always ASCII. The `windows-` fonts embrace and extend `iso8859-1` with additional characters such as "smart quotes." (Since these extensions, such as "smart quotes" are undefined in `iso8859-1`, they are usually rendered as question marks.)

Excellent source for additional information on character set encodings are at

- `http://ppewww.ph.gla.ac.uk/%7Eflavell/iso8859/iso8859-pointers.html`,
- `http://anubis.dkuug.dk/i18n/` and
- `http://czyborra.com/charsets/iso8859.html`.

3.6 Security Issues

I used Unix sockets above, but the standard Debian packages also configure `xfs` and `xfstt` to listen to TCP/IP ports 7100 and 7101, respectively. Access to these ports should be controlled by the `trusted-clients` field in `/etc/X11/xfs/config`, but this option is not implemented in XFree86 3.3.2.3a.

This means that anyone, anywhere, can connect to your font server. Since `xfs` (and presumably `xfsts`) "clone" to support more users there's a trivial denial−of−service attack against these systems. It's probably safe to use the font servers on dialup lines (since you're present to handle problems), but DSL and cable modem users should use a firewall.

4. Printing (`ghostscript`)

Starting with version 4, `ghostscript` has supported TrueType fonts as a compile−time option. Two Debian packages provide `ghostscript`:

- `main/binary-*/text/gs_*.deb` is DFSG−compliant version 5.10,
- `non-free/binary-*/gs-aladdin_*.deb` is non−DFSG−compliant version 5.50.

Both versions support TrueType fonts.

4.1 Configuring `Ghostscript` to use TrueType fonts

If you have a working `xfstt` server, it is easy to configure `ghostscript` to use TrueType fonts. We simply execute the following command:

```
# xfstt --gslist --sync >> /etc/gs.Fontmap
```

In practice, I've found it beneficial to make several small changes to the font definitions generated by `xfstt`. First, if a font name does not contain any spaces, I change the name to the usual notation. If a font name does contain spaces, I replace all spaces with dashes and the original name is added as an alias to the new name.

Finally, I prepend `TTF-` (or `MS-`) to all font names to minimize problems caused by a TrueType font having an identical name to an preexisting font.

Thus

```
(Arial)               (/usr/share/fonts/truetype/arial.ttf)   ;
(Arial Bold Italic)   (/usr/share/fonts/truetype/arialbi.ttf) ;
```

becomes

```
/MS-Arial             (/usr/share/fonts/truetype/arial.ttf)   ;
/MS-Arial-Bold-Italic (/usr/share/fonts/truetype/arialbi.ttf) ;
(Arial Bold Italic)   /MS-Arial-Bold-Italic                   ;
/Arial                /MS-Arial                               ;
```

The aliases ensure that `ghostscript` and `xfstt` can still specify the same font by a common name.

Much more significantly, with the change in the font names it's possible to instruct `ghostscript` to use TrueType fonts instead of the standard fonts. The documentation claims that this is also possible with parenthetical notation, but I could not get it to work.

For instance, we can instruct `ghostscript` to replace Helvetica fonts with Microsoft's free Arial fonts by appending the following lines to the `/etc/gs.Fontmap` file:

```
/Helvetica               /MS-Arial              ;
/Helvetica-Oblique       /MS-Arial-Italic       ;
/Helvetica-Bold          /MS-Arial-Bold         ;
/Helvetica-BoldOblique   /MS-Arial-Bold-Italic  ;
```

Similar aliases can be defined for the other standard fonts. These aliases would be most useful on `samba` printers serving Windows clients.

4.2 Printing TrueType font specimens

The best way to verify that `ghostscript` is properly configured to use TrueType fonts is to print font specimen pages. Assuming that you're running `ghostscript` 5.50 and that it is your default print queue, you can print all TrueType fonts with the following command:

```
# xfstt --gslist --sync | printfont
```

where `printfont` is the following shell script

```
#!/bin/sh

set -e
IFS= ')'

while read fontname rest
do
    cat << EOM | lpr
%!PS
(/usr/lib/ghostscript/5.50/prfont.ps) run
$fontname) DoFont
EOM
done
```

If you wish to print only a few fonts, the following script will be easier to use:

```
#!/bin/sh

set -e
while read -p "Font name, or ^D to exit: " fontname
do
    cat << EOM | lpr
%!PS
(/usr/lib/ghostscript/5.50/prfont.ps) run
$fontname DoFont
EOM
done
```

5. Content Generation

5.1 Generating AFM font metrics

AFM font metrics files are not required for display existing files with TrueType fonts, but they are necessary to create new files. The ghostscript program /usr/lib/ghostscript/5.50/printafm.ps could be used to generate these metric files, but I have found the ttf2afm program from the tetex−bin package to be easier to use.

The following script will generate an afm file for all TrueType fonts in a directory:

```
#!/bin/sh

set -e

for i in *.TTF
do
    /usr/bin/ttf2afm $i > ${i%TTF}afm
done

for i in *.ttf
do
    /usr/bin/ttf2afm $i > ${i%ttf}afm
done
```

One minor problem with ttf2afm is that some applications expect afm files to start with the StartFontMetrics tag, but files created by ttf2afm start with a comment. This "problem" is easily fixed by hitting each file with a text editor.

5.2 Generating font.map files

Once we have our afm files, we need to tell the system how to find them. This is often done via the font.map file.

I have been unable to find documentation on this file format, unlike fonts.dir, fonts.scale, and fonts.alias, all created by the mkfontdir program. However the minimum format appears to be quite simple:

- Font name, without whitespace
- AFM filename, without extension

Aliases appear to be implemented via multiple entries, and the filename extension must be in lowercase.

5.3 Image manipulation: GIMP

GIMP is the Gnu image manipulation and paint program. I did not have to make any additional changes to use TrueType fonts in gimp.

5.4 ASCII to PostScript conversion: `enscript`

`Enscript` is a program that converts ASCII to PostScript. Other programs which serve a similiar purpose are `a2ps` and `mpage`. `Enscript` allows two−up rotation, watermarks, headers, and keyword−based syntax coloring. It does not reformat text and is commonly used to print source listing.

To use TrueType fonts with `enscript`, you must do two things:

- Add `/usr/share/fonts/truetype` to your `AFMPath`.
- Specify a TrueType font, either explicitly or by aliasing a default font.

For details, see the `enscript` documentation.

Once I had made these changes, I had no problem using TrueType fonts.

5.5 Text Formatting and Typesetting: `groff`

`Groff` is the Gnu front end of the groff/troff document formatting system. The power of `Groff` is best seen with `man` pages.

user shell

```
$ zcat /usr/man/man1/groff.1.gz | groff −man | lpr
```

Besides man pages, an incredible amount of Unix documentation uses `troff` formatting with `ms` (and occasionally `me`) macros. The Debian `xbooks` package, for example, has 43 files using `troff` with `ms` macros. With `groff`, this material can be attractively printed.

`Groff` is a very powerful system, but it's the grandchild (or great−grandchild) of a program used to typeset 1960's era printing presses. Font support in `groff` reflects that heritage. `Groff`, in contrast to its predecessors, uses PostScript as the default output format so our earlier work with `ghostscript` takes care of half of the problem −− `groff` does not have to deal with reading TrueType font files. It does need to have accurate font metrics, and this section describes how to regenerate the necessary `groff` files:

Groff PostScript description files

```
/usr/share/groff/font/devps/DESC
    Device description file
/usr/share/groff/font/devps/text.enc
    Encoding used for text fonts
/usr/share/groff/font/devps/generate/textmap
    Standard mapping.
/usr/share/groff/font/devps/generate/Makefile
    Standard makefile
```

We must edit the Makefile,

/usr/share/groff/font/devps/generate/Makefile

```
- afmdir=/usr/local/afm
+ afmdir=/usr/share/fonts/truetype
```

change the name of the fonts to their TrueType equivalent (e.g., if we're using Microsoft's free TrueType fonts we would replace `Helvetica` with `Arial`), and change `TEXTFONTS` and the like to only include those fonts we are redefining.

We must also edit `/usr/share/groff/font/devps/generate/afmname` to use the TrueType font names and `afm` files, and to remove an `"-e"` flag from `awk`.

After all of this, we can rebuild the `groff` tables with

user shell

```
$ cd /usr/share/groff/font/devps
$ make -f generate/Makefile
```

As usual, the best way to verify the changes is to use a visually distinctive font. E.g., if you are using the Microsoft free TrueType fonts you can use `Mistral` for `TR`.

(I expect royalties from everyone who reconfigures their system to print manual pages in Old English fonts next April First!)

5.6 Text Formatting and Typesetting: `TeX`

`TeX` is the other common set of text formatting and typesetting programs on most GNU/Linux systems.

`TeX` fonts can created with `mktexmf`, but I have little information on the exact process. More details will be provided shortly.

6. Unanswered questions

- Applications

  The biggest unanswered question is why `netscape communicator` can use TrueType fonts, `ghostscript` can use the same TrueType fonts, yet pages printed by `netscape` look nothing like the screen.

  The short answer is that `netscape` generates PostScript output with standard fonts (Helvetica and Times−Roman) instead of the user−specified or HTML−specified fonts. The long answer is that I have absolutely no idea why it forces this behavior, or if there is a way to override it.

7. Obtaining TrueType Fonts

If you're looking for TrueType fonts, here are several places where you can start looking:

- c:\windows\fonts on the boat anchor in the corner
- Microsoft's Free TrueType Fonts `http://www.microsoft.com/typography/fontpack/`
- Acid Fonts `http://www.acidcool.com/`
- The Font Ring `http://www.webring.org/cgi-bin/webring?ring=fontring&list`

7.1 A comment about using Microsoft's free TrueType fonts

No article on TrueType support under Linux would be complete without a comment about Microsoft's free TrueType fonts. First the legality of using Microsoft's free core fonts:

`http://www.microsoft.com/typography/faq/faq8.htm`

```
Q. What can I do with these fonts?
   · Anyone can download and install these fonts for
     their own use.
   · [Web page] designers can specify the fonts within
     their own Web pages.
   ...
```

Clearly, it is legal and reasonable for Linux users to download and install these fonts, and I would like to thank Microsoft for making them available.

Since another clause restricts their redistribution "in any form that adds value to commercial products" I don't expect to see these fonts packaged in `main` anytime soon. (Could they be packaged in `non-free`, perhaps as an installer package...?)

Now that that's out of the way, I redirect your attention to the second statement. Microsoft actively encourages web page developers to specify their fonts on web pages, and many HTML editors explicitly name fonts.

Many web sites, when viewed by standard Netscape/Linux, can best be described as... legible. Some aren't even that. Not coincidentally the sites which are, um, legible are also usually the sites that explicitly state all font information in their web pages.

After I installed these fonts I noticed that most of these problematic sites became significantly more attractive. Many were still highly Windows−centric, but at least I didn't flinch every time I loaded their pages.

My recommendation is to install Microsoft's free TrueType fonts for your browser. You are not required to create web pages that use these fonts, nor do you have to configure `ghostscript` to use them.

8. Legalities