XFree86 Font De−uglification HOWTO

Hal Burgiss

hal@foobox.net

v1.95, 11 February 2002

How to improve X Window fonts. Various tips for improving font handling for XFree86, including sections on font servers, TrueType fonts, Netscape, and related topics.

Table of Contents

1. Introduction

An often heard complaint is the poor default fonts and font settings of X as implemented by many Linux distributions. Some programs use fixed width default fonts when a variable width font would be more appropriate. Other programs use fonts that are so small as to be practically unreadable. Many of the fonts that are bundled with XFree86 are not of the same quality as found on some other platforms. XFree86 does come with a halfway decent courier font, but its Times and Helvetica fonts are simple bitmap fonts that pixelize when scaled. This is changing for the better recently, but a default Linux desktop still often needs some tweaking to get the best fonts possible.

This HOWTO attempts to show how to adjust various font settings, install new fonts, and a few other things that should greatly improve the appearance and readability of fonts on the X Window Desktop. This is done by adjusting the `FontPath` in the `XF86Config` file, by adding switches to X server command line in **startx** or **xdm** (and variants), by adding new fonts, and by making sure a TrueType font server and fonts are installed. TrueType does indeed make a huge difference in many applications.

Comments, corrections, additions and critiques are always welcome. You can reach the author at <`hal@foobox.net`>. Contributions are also welcomed. Especially anyone who really stays current with KDE and/or GNOME issues!

1.1. Conventions Used in this Document

- Where examples of commands are used, a "#" character is used to denote where typically the command would be run as the root user. A "$" is used where typically a non−root user would be executing the command.
- The examples use `/usr/local/share/fonts/ttfonts` as our TrueType font directory. There is no magic to this location, and could conceivably just as well be in any number of other locations. Some distros may have a default location for TrueType fonts, and you may want to use that instead.
- References to "xfs" are to the xfs as packaged by Red Hat (and some other distros) for versions 6.x and later. This differs significantly in some respects from the stock XFree86 xfs.
- References to "Netscape" are to the entire suite of programs from Netscape: Communicator, Navigator, Messenger, etc. And for all intents and purposes, font configuration in Mozilla is very similar (but generally looks better!).
- 'XF86Config' is the X configuration file. This has changed to 'XF86Config−4' for XFree86 v4.x. For the most part, we'll just use 'XF86Config' here.

  Also, while some aspects of XFree86 4.x configuration are the same as 3.3.x, there are some significant differences. We'll only highlight the differences. So unless noted otherwise, any comments or examples will apply to both 3.3.x and 4.x versions.
- File system layout varies somewhat from distribution to distribution. It is impossible to stay on top of every conceivable variation of who keeps which files where. So take the examples here with a grain of salt if the PATHs don't seem to match your system.

1.2. Change Log and What's New

- 1.95: February 11, 2002. A few corrections. Removed the section on Fonts in KDE since this has to have changed, and I don't know anything about KDE (does anyone want to help here?). Added a brief section on gdkxft, which adds anti−aliasing support for GTK+ 1.2 applications.
- 1.9: November 5, 2001. A few new links and some minor catch ups only.
- 1.8: June 25, 2001: Included a new section on Anti−aliasing and Xft from Danny Tholen <obiwan@mailmij.org>. Many thanks on this not so well documented subject. Also, Sebastiano Vigna's neat little package for downloading and installing MS webfonts: http://freshmeat.net/webFonts4Linux. A few other odds and ends.
- 1.70: April 18, 2001: Added links for converting Mac TrueType Fonts (thanks to Karl A. Krueger), links to Unicode TrueType fonts (thanks to Tzafrir Cohen for suggestions and URLs), and added a section on anti−aliasing with X 4.0.2 (or greater). Also, included a reference to cabextract, a utility that is now available for extracting Win32 Fonts (among other things) from a Window's "cab" archive.
- 1.60: March 21, 2001: A few very minor changes. Most notable news is anti−aliasing support now in XFree 4.02 (referenced in the Notes section only). Chinese translation URL added.

1.3. New Versions

The pre−release versions of this document can be found at http://feenix.burgiss.net/ldp/fdu/index.html.

1.4. Copyright

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would very much like to be notified of any plans to redistribute the HOWTOs, this one in particular! Web page authors are free to link to this HOWTO without restriction, though the author would appreciate an email informing him of this, just so he can boost his ego by knowing who else reads and links to this document.

Many of the terms mentioned in this document are trade names. Unless otherwise stated, all trademarks are property of their respective owners.

1.5. Credits

```
  Original author: Doug Holland.
  Email: meldroc@frii.com
  WWW: http://www.frii.com/~meldroc/



 Updated and maintained by Hal Burgiss.
  Email: hal@foobox.net
  WWW: http://feenix.burgiss.net/ldp/fdu/
```

Special thanks go to:

- The developers of the XFree86 Project, for all the hard work and time they have given. Also, Juliusz Chroboczek for his work with xfsft, and XFree86 4.x to help bring TrueType to the hungry masses. And Keith Packard for his anti−aliasing, and other work. This is not to slight the many, many other XFree86 volunteers.
- Font wizard Kristin Aanestad, whose legwork and insight on much of the xfs, TrueType, Netscape, and especially, the fonts.alias sections are much appreciated. More from Kristin at Some Linux for Beginners on a wide range of topics.
- Danny Tholen <obiwan@mailmij.org> is responsible for the nice Xft section, and examples.
- The folks at comp.os.linux.x who gave me a hand in figuring all of this out in the first place.
- The Linux community in general who made all of this possible in the first place. Especially those who have offered suggestions and comments that help to make this HOWTO a better resource. Keep those cards and letters coming ;−)

1.6. Translations

Chinese: http://www.linux.org.tw/CLDP/mini/FDU.html by Yu−Chia Chang.

2. X Server Configuration

There are a few simple configuration tweaks that will help X do its job better.

2.1. Setting The FontPath

The first place to look for curing font problems is the `XF86Config` file. `/usr/X11/lib/X11/XF86Config` or `/etc/X11/XF86Config` are the common locations. (This may be `XF86Config−4` for XFree86 4.x.) If you haven't guessed already, the most important part of this file relating to fonts is the `FontPath.` Before we get into that, this would be a good time to check the other parts of your X configuration. Bad monitor settings can be even more of a headache than bad fonts, so make sure your refresh rate is as high as your monitor can handle (85 Hz is great, 75 Hz is OK, 60 Hz is painful.)

Use your favorite text editor and edit `XF86Config`. Near the top of the file in the "Files" section, you should see something vaguely like this:

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

This much should be the same, or at least similar, for both XFree86 3.x and 4.x. The `FontPath` tells X where to find the fonts it uses to render text on your display. Order is important −− when an X application asks X to render some text, the X server usually has some leeway to choose the font that is used. The X server then goes through the `FontPath` and grabs the first font it sees that matches the X client's criteria, and then displays it. (Note that Red Hat's xfs for versions 6.x and later has a different way of setting the `FontPath`. See the Section 3.2.2 below for more on xfs.)

If the 100dpi fonts are not listed, they probably did not get installed for whatever reason, so you may want install them now. Default installations may put 75dpi fonts before the 100dpi fonts. If you have a high resolution display (1024x768 or higher), this means very tiny fonts. If this is the case, the first tweak you'll use is to switch the 75dpi and 100dpi FontPath lines:

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

Next, specify that you prefer to use unscaled bitmap fonts. If you've ever used Netscape or any other program that displays titles using big fonts, you'll likely notice that those fonts are pixelized. This is ugly and needs to be corrected. So add `:unscaled` to the ends of the misc, 100dpi and 75dpi fonts. You can even use both unscaled and scaled fonts if you want, just put the unscaled `FontPath` lines first to tell X you prefer unscaled fonts if possible:

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

After making these changes, restart X (and your font server, if installed). Doesn't the desktop look better already?

2.2. X Server Command Line Options

The next thing you need to do is adjust the command line options for the X server. You'll want to use the `−dpi` switch which specifies the display resolution in dots per inch. As a lot of systems use high resolution displays these days, chances are they'll be working at 100 dpi.

If you start X from the console command prompt, type:

```
$ startx −− −dpi 100 −depth 16  # v4.x syntax
```

Or these options can be stored in `~/.xserverrc`. See the **startx** and **xinit** man pages for more on this.

If you use xdm (or friends such as gdm) for graphical logins, you'll want to edit your `/usr/X11/lib/X11/xdm/Xservers` file (or possibly `/etc/X11/xdm/Xservers`) which will have the command line for the Xserver in it. Mine has the line:

```
:0 local /usr/X11R6/bin/X −dpi 100 −gamma 1.6
```

Note that there is no magic to "−dpi 100". Choose something higher if your hardware will support it −− like "−dpi 120".

More information is in the **X**, **Xserver**, **xdm**, **xinit,** and **startx** man pages.

3. TrueType Fonts

Historically, the Unix world relied on Type 1 fonts for high quality scalable fonts. Linux supports Type 1 quite well, both for printing and for screen output. But, Type 1 never was widely adopted by web designers, and on other platforms. TrueType, due to its association with Windows, is the preferred web font.

Because the boys at Redmond are very concerned with the appearance of their software (as opposed to the internal workings ;), they built TrueType font support into Windows. And of course no big surprise, but they got the idea from Apple. In fact, TrueType is a registered trademark of Apple Computer, Inc. Windows 9x, NT, 2K and nearly every other incarnation of Windows comes with Arial, Times New Roman, and Courier New, which are roughly equivalent to Helvetica, Times and Courier. TrueType fonts are scalable, so they look good in large sizes, and they are well hinted, so they are readable at small sizes. Many windows applications come with dozens more TrueType fonts. Don't microwave your Windows CD yet, you'll want to get those fonts first!

Any recent distro will have one or more font servers included. And XFree86 4.x *does have built in support* for TrueType (see Section 4). You won't find many decent TrueType fonts included with any distribution, however. The reason is that there are not many quality TrueType fonts available under a suitable license at this time. Many distributions are including tools for automating the process of adding TrueType fonts. See if that is an option for you. This will be easiest route.

XFree86 3.x does not come with built in TrueType support, so you'll have to add it yourself if you are using a 3.x version. This will mean installing a font server that does support TrueType. And, of course, installing the fonts themselves (see below).

3.1. Making TrueType Fonts Available

Let's start with the fonts first. Any TrueType font included with the various MS Windows incarnations should work. Don't forget word processors and other apps that may include their own fonts. MacOS fonts will only work if converted to a usable format. (See the links section for converter packages.) There are also some 'free' TrueType fonts available for download if you have already nuked that CD (see links section).

Many distributions are now bundling tools for automating the process of including quality TrueType fonts. SuSE, Debian, and Mandrake do (Red Hat 7.x does not at this time). See what packages you might have for this as this will be the most painless way to go. Essentially, these tools help migrate fonts from a Windows installation, or download those available from Microsoft, and then handle the installation and configuration all in one neat utility. If you do have such a utility, the below information may not be necessary!

In order to use TrueType, the fonts will have to be always accessible to X. This means they will have to be on a filesystem that is *always* mounted. This can conceivably be a Windows partition on a dual boot system. Alternately, the fonts can be copied to Linux. First **su** to root:

```
 # su -
 # mkdir -p /usr/local/share/fonts/ttfonts
```

Now, change to the new font directory:

```
 # cd  /usr/local/share/fonts/ttfonts
```

Then, add the fonts to this directory, either by copying them from your Windows system:

```
# cp /mnt/<path_to_fonts>/*ttf  .
```

or by downloading those available directly from Microsoft: http://www.microsoft.com/typography/fontpack/default.htm. These fonts are in self−extracting zip archives. The ones labeled for use with Windows 3.1 can indeed be unpacked under Linux with the Linux **zip** utility:

```
# ls *exe | xargs −n 1 unzip −L
```

The '−L' option will convert to lower case font names (this may be necessary for some versions of xfsft and Red Hat's xfs). Note that the current Linux zip utility does not work with the 32 bit Win9x cab font archives. (It also looks like Microsoft no longer has the 16 bit Arial, Courier and Times−Roman on this site.) But these can be unarchived under Linux with **cabextract**, which can be found http://www.kyz.uklinux.net/cabextract.php3. This would now seem to be the best way to go since there is a better selection of fonts.

A slick solution to this from Sebastiano Vigna is his http://freshmeat.net/webFonts4Linux, which automates the downloading, extracting and installation of the Microsoft fonts all in one neat package.

Or you can get an RPM of WebFonts that contains some of the MS 'Web' TrueTypes from ftp://ftp.rpmfind.net/linux/contrib/noarch/noarch/webfonts−1−3.noarch.rpm. This has enough basic fonts to keep Netscape and other web browsers happy. Something similar for Debian is http://packages.debian.org/unstable/graphics/msttcorefonts.html. This does not include the actual fonts, but facilitates the installation.

If doing it yourself, you will also have to include the new TrueType directory(s) in the X server's fontpath. So with your text editor of choice add the line(s) as appropriate:

```
     FontPath        "/usr/local/share/fonts/ttfonts"
     FontPath        "/usr/X11R6/lib/X11/fonts/misc:unscaled"
     FontPath        "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
     FontPath        "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
     FontPath        "/usr/X11R6/lib/X11/fonts/Type1"
     FontPath        "/usr/X11R6/lib/X11/fonts/Speedo"
     FontPath        "/usr/X11R6/lib/X11/fonts/misc"
     FontPath        "/usr/X11R6/lib/X11/fonts/100dpi"
     FontPath        "/usr/X11R6/lib/X11/fonts/75dpi"
```

3.2. Font Servers

There are several font servers available that will do the job: xfstt, xfsft, and Red Hat's patched version of xfs based on xfsft. While these names are all too similar, these are different packages. One, or more, of these should be included with any recent Linux distribution, and you may have one installed already. If so, use which ever one your distribution is set up to use.

Historically, font servers were used to serve fonts over a network. Font resources could then reside on one host, and clients could access them as needed. But, the developers have enhanced these to include features such as the ability to render TrueType fonts. (XFree86 4.x has this ability included already, and thus an additional font server is not really needed solely for the purpose of having TrueType support.)

3.2.1. xfstt

One such font server is xfstt. xfstt was designed specifically with TrueType fonts in mind.

3.2.1.1. Installation

xfstt is *very* easy to install and configure. If it isn't already installed, you'll want to download the tarball, or check your CD. The most current version can be found at http://metalab.unc.edu/pub/Linux/X11/fonts/

Once you have the tarball, unpack it:

```
 $ tar −zxvf xfstt−*tgz
```

Then build and install it. Read the `INSTALL` file for quick instructions, but it's a no brainer.

From the xfstt directory is all you have to do.

```
 # make
 # make install
```

Then start xfstt with:

```
 # xfstt −−sync             # updates xfstt's font database
 # xfstt &                  # runs xfstt in the background.
```

xfstt should be started before the X server starts. Once you have this working correctly, you can add the above lines to `/etc/rc.d/rc.local`, or other suitable start up file. Then type:

```
 $ xset +fp unix/:7101      # tells X about xfstt, and where to look for fonts.
```

or add:

```
      FontPath "unix/:7101"
```

to your `XF86Config` to tell X about the font server. Rerun **xfstt −−sync** any time the FontPath, or contents, change.

3.2.1.2. Adjusting the Default Font Size

If your TrueType fonts appear to be very tiny, the following commands may help.

Add the `−dpi` switch to your X server command line (see section 3 above to do this.)

Use the `−−res` switch to tell xfstt to increase the default resolution. Use the following command line.

```
 # xfstt −−res 120
```

3.2.2. Red Hat's xfs

As of Red Hat Linux 6.0, Red Hat based distributions (Mandrake, etc) have included a specially patched version of xfs, the XFree86 Font Server, and patched X servers as well. Red Hat's xfs includes the xfsft patch set which in turn is built upon the FreeType Font library. Red Hat's xfs provides similar functionality to xfstt. xfs is able to serve both TrueType and Type 1 fonts, as well as legacy X fonts.

If you are using a Red Hat based distro, you should have xfs installed already. If not, it is in the `XFree86−xfs*rpm`. To make sure it runs as one of the default services, either use **ntsysv** or:

```
# chkconfig −−add xfs
```

Now xfs will start every time you boot.

3.2.2.1. Setting the xfs FontPath

The default Red Hat installation of xfs serves fonts via a Unix Domain Socket. We'll need to tell the X server where to look for xfs, and thus fonts. The FontPath in `/etc/X11/XF86Config` must include for Red Hat 6.x:

```
      FontPath   "unix/:−1"
```

This is changed for Red Hat 7.x to:

```
      FontPath   "unix/:7100"
```

At least for a default configurations. This is a reference to the socket where xfs is listening. You may include additional FontPaths, but these will be handled by the X server, and not xfs. A clean install of Red Hat 6/7 should have this already set up, but if you are upgrading from an older version, you may have to change this yourself!

xfs then has its own, separate FontPath stored in `/etc/X11/fs/config`. This is where it will look to find fonts. This is over and above the X server's FontPath in `XF86Config`. You can either add the new path(s) with a text editor, or use the **chkfontpath** command:

```
# chkfontpath −−add /new/font/path
```

The FontPath must exist before running **chkfontpath**. The relevant section of `/etc/X11/fs/config` should now look something like this:

```
     catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,
                 /usr/X11R6/lib/X11/fonts/100dpi:unscaled,
                 /usr/X11R6/lib/X11/fonts/75dpi:unscaled,
                 /usr/X11R6/lib/X11/fonts/Type1,
                 /usr/X11R6/lib/X11/fonts/Speedo,
                 /usr/X11R6/lib/X11/fonts/misc,
                 /usr/X11R6/lib/X11/fonts/100dpi,
                 /usr/X11R6/lib/X11/fonts/75dpi,
                 /new/font/path
```

When adding a new FontPath for TrueType fonts, you will want to do this step after installing and preparing the fonts. See the next section.

3.2.2.2. Getting the Fonts Ready

We still have a bit of work to do before we can actually use any TrueType fonts. xfs requires a few things to be in order. First, all font files must have lower case names for xfs. Secondly, they shouldn't have embedded spaces. And then, we will need to create a couple of files to make things go.

Su to root, and change to the directory where the TrueType fonts are.

```
 # su −
 # cd /usr/local/share/fonts/ttfonts
```

If there are any upper case font names, you can use the following script to convert all names to lower case:

```
 #!/bin/sh
 #
 ## −−−−−−−− convert upper to lower case −−−−−−−−−

 ls * | while read f
  do
```

```
   if [ -f $f ]; then
     if [ "$f" != "`echo \"$f\" | tr A-Z a-z`" ]; then
      #Note that 'This' will overwrite 'this'!
      mv -iv "$f" "`echo \"$f\" | tr A-Z a-z`"
     fi
   fi
 done

## eof
```

Note the punctuation −− the backquotes are important! Remove any spaces from font names too. Once the TrueType fonts are properly installed, you must create both `fonts.dir` and `fonts.scale` files. The following commands do this:

```
# ttmkfdir -o fonts.scale
# mkfontdir
```

As of Red Hat 7.1, the above commands are run from the xfs init script. So restarting xfs (**/etc/rc.d/init.d/xfs restart**) will accomplish the same thing.

You should now have `fonts.dir` and `fonts.scale` files in your TrueType font directory. **ttmkfdir** is in the `Freetype` RPM, and must be run before **mkfontdir**. With Debian based distros, there is a similar utility called **mkttfdir**, and is in the `fttools` Deb package. Though this apparently does not generate as many encodings as **ttmkfdir**. These commands may not always report errors, so verify that they were created and are not empty files:

```
$ ls -l fonts.*
 -rw-r--r--  1 root   root    11657 Aug 17 10:31 fonts.dir
 -rw-r--r--  1 root   root    11657 Aug 17 10:31 fonts.scale
```

If you encounter any problems, try **ttmkfdir** with the **− m** switch. This will discard bad characters from the font file. Specify a number such as 50 or l00 (**ttmkfdir −m 50**). The files themselves are text files. Have a look:

```
$ less  fonts.dir
 114
 webdings.ttf  -microsoft-Webdings-medium-r-normal--0-0-0-0-p-0-microsoft-symbo
 verdanaz.ttf  -microsoft-Verdana-bold-i-normal--0-0-0-0-p-0-ascii-0
 verdanaz.ttf  -microsoft-Verdana-bold-i-normal--0-0-0-0-p-0-fcd8859-15
 verdanaz.ttf  -microsoft-Verdana-bold-i-normal--0-0-0-0-p-0-iso8859-15
 verdanaz.ttf  -microsoft-Verdana-bold-i-normal--0-0-0-0-p-0-iso8859-9
 verdanaz.ttf  -microsoft-Verdana-bold-i-normal--0-0-0-0-p-0-iso8859-1
[...]
```

Next, update the FontPath and xfs:

```
 # chkfontpath --add /usr/local/share/fonts/ttfonts
 # /etc/rc.d/init.d/xfs restart
```

You should now be in business. You can check which fonts are available to X:

```
$ xlsfonts | less
```

or check them out further with **xfontsel**, or **gfontsel**. If they are visible to **xlsfonts**, then they are available to X and vice versa. If they are not there, try restarting X with Ctrl−Alt−BS.

3.2.3. xfsft

xfsft is a TrueType solution from Juliusz Chroboczek. xfsft is based on the FreeType font library as developed by Mark Leisher and others. It is essentially is a patch for XFree86's xfs and related libraries −− xfs + ft. Red Hat's xfs is essentially xfsft with a few minor modifications. Also, XFree86 4.x includes the `freetype` font module which is also the result of Juliusz's work, and is one of the TrueType solutions available for XFree86 4.x.

Building xfsft requires having at least some of the XFree86 source available, in addition to xfsft itself, so this is not for the faint of heart. Instructions for building and configuring xfsft are in the tarball, so I won't go into details here. They are pretty straight forward. There are links to binaries available at the xfsft home page (see above).

Note that you *must* also create `fonts.scale` and `fonts.dir` files for xfsft. `fonts.scale` can be created manually (ugh!), or with the **ttmkfdir** utility. This is not included with xfsft but you can get it here: http://www.joerg−pommnitz.de/TrueType/ttmkfdir.tar.gz, or probably on many Linux archives sites too. Red Hat has this as part of the `Freetype` RPM. And for Debian it is called **mkttfdir** and is in the `fttools` package.

You will also need a configuration file. Here is a sample:

```
     −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

     clone−self = off
     use−syslog = off

     client−limit = 20

     catalogue = /usr/local/share/font/ttfonts

     error−file = /home/jec/fonts/xfs.errors

     # in decipoints
     default−point−size = 120

     # x,y
     default−resolutions = 100,100,75,75
```

```
      −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
```

You can then run start xfsft:

```
# xfs −port 7100 −config /path/to/your/config/file &
```

You can then add xfsft to the X server's `FontPath`:

```
$ xset +fp tcp/localhost:7100
```

If all goes well, you could then add this `FontPath` to `XF86Config`.

3.3. The fonts.alias File

`fonts.alias` is yet another font configuration file that can be used to tweak how fonts are handled. Like `fonts.scale` and `fonts.dir`, `fonts.alias` must be in the same directory as the fonts you are aliasing. It is not mandatory however, but does solve certain potential problems. Here is an example from the first line of`/usr/X11R6/lib/X11/fonts/misc/fonts.alias` on a Red Hat system:

```
 fixed    −misc−fixed−medium−r−semicondensed−−13−120−75−75−c−60−iso8859−1
```

`fixed` is the 'alias' here. Any time this is requested, we actually get the font definition from the second column. Font too small? Just change the definition. (Warning: this is a critical file, at least on Red Hat.) The same principle applies to all fonts, including TrueType. In fact, if you don't have TrueType, you could conceivably use this trick to have a comparable Type 1, or other, font aliased as a TrueType.

`fonts.alias` is important for some applications that don't handle the data provided by `fonts.scale` well. Most notably here is Netscape. Without a `fonts.alias` you will find that Netscape will only show point sizes of 0 and 12 available. `fonts.alias` fixes this. You might also find that if you a specify another size with the `scalable font` option under `Preferences`, Netscape will not remember this setting. Annoying! This is also fixed. So we really need this file. Sample excerpt from a `fonts.scale`:

```
 arial.ttf   −monotype−Arial−medium−r−normal−−0−0−0−0−p−0−ascii−0
 arial.ttf   −monotype−Arial−medium−r−normal−−0−0−0−0−p−0−fcd8859−15
 arial.ttf   −monotype−Arial−medium−r−normal−−0−0−0−0−p−0−iso8859−15
 arial.ttf   −monotype−Arial−medium−r−normal−−0−0−0−0−p−0−iso8859−1
```

These are scalable so we don't get any predefined point sizes. We will need to create our `fonts.alias` something like this excerpt for Arial:

```
 −monotype−Arial−medium−r−normal−−6−60−0−0−p−0−iso8859−1 \
            −monotype−Arial−medium−r−normal−−9−90−75−75−p−0−iso8859−1
 -monotype-Arial-medium-r-normal--7-70-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--9-90-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--8-80-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--10-100-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--9-90-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--11-110-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--10-100-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--12-120-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--11-110-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--12-120-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--12-120-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--12-120-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--13-130-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--13-130-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--14-140-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--14-140-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--15-150-0-0-p-0-iso8859-1 \
             -monotype-Arial-medium-r-normal--15-150-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--18-180-0-0-p-0-iso8859-1 \
            -monotype-Arial-medium-r-normal--18-180-75-75-p-0-iso8859-1

 -monotype-Arial-medium-r-normal--24-240-0-0-p-0-iso8859-1 \
            -monotype-Arial-medium-r-normal--24-240-75-75-p-0-iso8859-1
```

(*Please note that I have split each line for readability*. There should be two columns all on one line, without the "\", and separated by at least one space.) This will keep Netscape happy. Also, if font names should have embedded spaces, then you should enclose the filename in quotes. You might also note the `pointsize` discrepancy between the first and second columns of the first few rows. The first column of the first entry has a '6', whereas this is aliased to a '9' in the second column, and thus '9' point. This is by design and is an excellent way to overcome the Netscape 'damn tiny fonts' syndrome. Adjust to suit your tastes, resolution, and eyesight.

This file can be created manually with a text editor, or conceivably with some fancy sed or awk scripting. There is an excellent discussion of this file, and other font related topics at Kristin Aanestad's site at http://home.c2i.net/dark/linux.html. There is also a link to a python script which can reportedly automatically generate a `fonts.alias` file at this same site. Thanks to Kristin whose work and insight was the inspiration for this section!

Another potential use of `fonts.alias` would be to map one font to something quite different. Say you don't have TrueType fonts, and didn't want to install Microsoft's. You could alias nice, scalable Type 1 fonts to a TrueType. That way when the system (or some web page) wants a TrueType, you'd get something of comparable quality instead of bitmap that doesn't scale well.

Note that with XFree86 4.0.2 and greater, there are new font handling mechanisms available via the Xft extensions. Font aliasing is done in Xft's own configuration file: `XftConfig`. This is the preferred method where anti−aliasing, and the other new rendering features are desired. See the Anti−aliasing Section for more on this and de−mystification. This is only true where the application (i.e. the toolkit, e.g QT) itself supports the new extensions! At this time, not all do (yet).

4. XFree86 4.x

XFree86 4.0 introduced native support for TrueType fonts, along with other new features. The enhanced font support is based on xfsft from Juliusz Chroboczek, which in turn is based on the FreeType font library originally from Mark Leisher, so the configuration is similar to xfsft and Red Hat's patched xfs. As of 4.0.2, XFree86 begins to support anti−aliasing which is a technique for smoothing font outlines (see section below).

The `FontPath` is still in `XF86Config`, as always. For Red Hat 6/7 using a stock XFree86 4.x (i.e. NOT the Red Hat 7.x supplied version), this will mean moving the Red Hat xfs `FontPath` from `/etc/X11/fs/config` back to `XF86Config`. A separate font server is no longer needed just for TrueType support. You may disable it, unless it is needed to serve fonts to other clients in a network environment. See the section below for Red Hat 7.x specific configuration issues.

```
       Section "Files"
        FontPath "/usr/X11R6/lib/X11/fonts/misc:unscaled"
        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/Type1"
        FontPath "/usr/X11R6/lib/X11/fonts/Speedo"
        FontPath "/usr/share/fonts/default/Type1"
        FontPath "/usr/local/share/fonts/ttfonts"
        FontPath "/usr/X11R6/lib/X11/fonts/100dpi"
        FontPath "/usr/X11R6/lib/X11/fonts/75dpi"
       EndSection
```

In order to use TrueType, you must also specify which font `module` the X server should be using in the "Module" section:

```
       Section "Module"
        Load  "freetype"
        Load  "speedo"
        Load  "type1"
        <load other modules....>
       EndSection
```

Note that there can be only one 'Module' section, so include any other modules here as well.

You also must to create `fonts.scale` and `fonts.dir` file for each TrueType font directory, just like for xfsft and Red Hat's xfs. ttmkfdir will come in handy for `fonts.scale`. See the xfs Section 3.2.2 above for more details and examples.

`xtt` is another available TrueType module that is best known for supporting ideographic (Oriental) type fonts. You can use either, but only one at a time.

X server command line options are still the same as previous versions of X:

```
$ startx −− −dpi 100
```

4.1. Anti−aliasing

Anti−aliasing is a technique for producing even smoother, crisper looking fonts by subtly softening the sharp, jagged edges. This has long been a feature of Apple and Microsoft font rendering, and is now making it's way into X via the X Rendering Extension specification thanks to Keith Packard. The new extensions provide other benefits as well. Distributions that support anti−aliasing with their stable/official versions are now being released.

That is the good news. The bad news is that not all drivers support anti−aliasing yet. This is a moving target, so you will have to dig around to find whether your chipset is supported or not. The recently released 4.2 should have near universal support. More not−so−good news is that few applications are actually taking advantage of this yet. We will have to wait for the various toolkits (TK, GTK, Xaw, etc) to catch up.

If you are reading this long after the publication date (February 2002), hopefully most of these shortcomings will have been overcome. All hardware will eventually be supported, mainstream distros will have shipped releases that include the new extensions, and they will be enabled by default. Many apps will look better since they will be "AA" aware, and we won't have to jump through any configuration hoops just to make it work. In the meantime, read on ...

4.1.1. Requirements

Minimum requirements for Anti−Aliasing:

- XFree86 4.0.2 or later.
- You graphic card's driver has to support anti−aliasing. If 4.0.2 (or greater) is already installed, you can get this information direct from the driver with **xdpyinfo**. Run this and look for "Number of Extensions:". If this lists "RENDER" among them, then you should be good to go. If not, well, it isn't going to work, and you will have to wait for an updated driver.
- The Freetype2 library available from http://www.freetype.org, and also now bundled with XFree86. XFree needs to be linked against this, so install and build first if building from scratch. Your distro should have Freetype packages as well. Just make sure it is `freetype−2`.
- TrueType fonts are best for display purposes. Type1 is also good. See above sections.
- For KDE users, KDE supports anti−aliasing as of 2.x. This will require QT−2.3.0 or later, and built with Xft support. A nice font HOWTO from Troll Tech for KDE and QT can be found: http://trolls.troll.no/~lars/fonts/qt−fonts−HOWTO.html.

  GNOME does not support anti−aliasing at this time in stable releases (as of 1.4). Awaiting GTK implementation (probably for GTK 2.0). Apparently this will be available later this year, so stay tuned. GNOME 2.0 will have native support for anti−aliasing.
- Applications that "know" about anti−aliasing. Not necessarily at the individual application level, but the libraries and toolkits (GTK, TK, etc.) that the application are built against, must be able to use the new features. At this time, there are scant few. KDE/QT is first out of the box. Also, **xterm** supports the new extensions.
- The new rendering extensions configuration file, `XftConfig`, must be configured for the fonts you want to use.

- The new extensions supplant much of what we have been doing with font servers like xfs. `fonts.alias` and similar configuration files, for instance, are not used for fonts that are being controlled by the new extensions.

4.1.2. Installation

Keith Packard has a very brief summary of the steps required for building, installing and configuring from source at http://www.xfree86.org/~keithp/render/aafont.txt. No need to reprint it here.

Newer distro releases are likely to have the foundation support for anti−aliasing available now. Red Hat, for instance, has it available as of Red Hat 7.1.

To verify the necessary components, first make sure the "freetype" module (and any others) are loaded. Check the X server output:

```
(II) LoadModule: "freetype"
(II) Loading /usr/X11R6/lib/modules/fonts/libfreetype.a
(II) Module freetype: vendor="The XFree86 Project"
       compiled for 4.0.3, module version = 1.1.9
       Module class: XFree86 Font Renderer
       ABI class: XFree86 Font Renderer, version 0.2
(II) Loading font FreeType
```

Then verify if the "RENDER" extension is available, either check with **xdpyinfo**, or check the X server log, typically `/var/log/XFree86.0.log`:

```
(II) Initializing built-in extension MIT-SHM
(II) Initializing built-in extension XInputExtension
(II) Initializing built-in extension XTEST
(II) Initializing built-in extension XKEYBOARD
(II) Initializing built-in extension LBX
(II) Initializing built-in extension XC-APPGROUP
(II) Initializing built-in extension SECURITY
(II) Initializing built-in extension XINERAMA
(II) Initializing built-in extension XFree86-Bigfont
(II) Initializing built-in extension RENDER
```

If "RENDER" is there, anti−aliasing and the other advanced rendering extensions should be available.

4.1.3. Xft Configuration

By Danny Tholen <obiwan@mailmij.org>

Xft is an interface to the freetype rasterizer written by Keith Packard, member of the XFree86 Project, Inc. It allows applications to use fonts from the new X render extension using a unified font naming scheme. In `/etc/X11/XftConfig` (or `/usr/X11R6/lib/X11/XftConfig`) you will find a configuration file which can be adapted to suit your personal taste. In this section I will explain the syntax and demonstrate some things you can do with this file.

The following information is based on 4.0.3. 4.1 is just released, and there may be a few new wrinkles not touched on here.

4.1.3.1. XftConfig Structure

The basic structure revolves around a 'pattern'. A pattern is a set of name/value−list pairs, each value−list contains one or more typed values. A certain application requests a font, for example:

```
family: "Arial"
size: 12
encoding: "iso8859−1"
```

A size 12 arial font in latin−1 encoding. The Xft extension will now try to patch this pattern to all of the fonts available in the system. And selecting the one with the best score. Before the matching is done Xft looks in `XftConfig`. The requested pattern can here be extended before use. An example is:

```
match any family == "Arial" edit antialias = true;
```

This will enable anti−aliasing for all fonts of the family Arial.

Also, the X server is queried to list all of its fonts; the XLFD contains just enough information to match fonts roughly.

Here's a list of attributes used in matching fonts (in priority order, this may not be up to date anymore!):

```
foundry        font foundry (string, like "monotype")
encoding       font encoding (string, like "iso8859−1")
spacing        font spacing (integers or proportional (0), mono (100),
               charcell (110))
bold           is the font bold? (boolean)
italic         is the font italic? (boolean)
antialias      is the font anti−aliased? (boolean)
family         font family (string)
size           font size (double)
style          font style (string, like "Bold Italic")
slant          font slant (roman, italic, oblique)
weight         font weight ( integers or light, medium (100), demi−bold,
               bold, black)
rasterizer     not yet used (probably "TrueType", "Type1", ...)
outline        are outlines available? (boolean)
```

4.1.3.2. XftConfig Syntax

- #### dir

  Adds a directory to the list of places Xft will look for fonts. There is no particular order implied by the list; Xft treats all fonts about the same.

- #### include and includeif

  Cause Xft to load more configuration parameters from the indicated file. "includeif" doesn't elicit a complaint if the file doesn't exist. If the file name begins with a "~" character, it refers to a path relative to the home directory of the user. This is useful for user−specific configurations.

- #### match edit

  If a pattern from an application matches the pattern after "match", it is edited with the instructions in `edit`. The pattern match is done as follows:

  ```
    match qual FIELD−NAME COMPARE CONSTANT
  ```

  where qual is either `any` (matches one specific font) or `all` (matches all fonts). An example:

  ```
   match all foundry==¨monotype¨
  ```

  which will match (and edit) all fonts belonging to the foundry "monotype".

  ```
   match any family==¨arial¨
  ```

  will match (and edit) one specific font with the family name "arial".

  `FIELD−NAME` is any one of the properties found in the above section Structure, or additionally:

  ```
   pixelsize         font size in pixels (integer)
   charspace         character space (integer)
   minspace          minimal spacing (integer)
   rgba              color hinting (string ¨rgb¨ or ¨bgr¨ and
                     vertical hinting ¨vrgb¨ ¨vbgr¨), aka sub−pixel hinting
   xlfd              x server font (string, type xlsfonts to
                     see a list of your xlfd strings)
   file              the font file (string)
   core              use X core fonts? (boolean)
   render            use render fonts? (boolean)
   index             I have no idea what this does:)
   scalable          is the font scalable (boolean)
  ```

```
scale            scale the font (integer)
charwidth        character width (integer)
charheight       character height (integer)
matrix           no idea (not really at least)
```

`COMPARE` can be `<`, `>`, or `==`.

`CONSTANT` is the value of the field−name in the appropriate type (see above section Structure).

You can use multiple matches before you use the "edit" statement:

```
edit FIELD−NAME ASSIGN EXPR SEMI
```

Where `ASSIGN` can be one of `=`, `+=` or `=+`. With `=`, the matching value in the pattern will be replaced by the given expression. `+=` or `=+` will prepend/append a new value to the list of values for the indicated field.

`EXPR` sets the `FIELD−NAME` to a value.

`SEMI` is a semicolon (`;`). You can use multiple instructions, separated by a semicolon. See below for examples if this is confusing.

4.1.3.3. XftConfig Examples

And now I´ll try to list a few useful configurations and explain them. Note that it is configured for my system, and I may use different fonts than you, so try to adapt the examples to your own needs.

1. How do I make fonts available to Xft?

   List your Type 1 and TrueType font directories with "dir". On my system (Mandrake 7.2) this becomes:

   ```
   dir "/usr/X11R6/lib/X11/fonts/Type1"
   dir "/usr/X11R6/lib/X11/fonts/drakfont"
   ```

2. How do I use a user specific XftConfig file?

   Put an `.xftconfig` file in your user directory and add:

   ```
   includeif   "~/.xftconfig"
   ```

   to your standard `XftConfig`. This will enable a user specific configuration file, but it will not complain if there is no such file.

3. How do I make aliases for my fonts?

I noted that my KDE console asks for "mono" fonts when it is looking for a fixed font. "console" is used when I select "linux" in the font menu of the KDE konsole. Therefore, I used two aliases for fonts which are also named "fixed":

```
match any family == "fixed"     edit family =+ "mono";
match any family == "console"   edit family =+ "mono";
```

4. Anti−aliasing my fonts are blurry and makes me dizzy!

   Although there is a big fuzz around AA in X, good fonts actually look better if they are not anti−aliased. The anti−aliasing blurs the fonts by adding gray pixels to the edges, and this may strain your eyes if you looking at them for a long time. (Your eyes will try to get the fonts sharper, which of course is not working because they are blurred;) However, for very small fonts, anti−aliasing may increase the readability of the fonts, because with sharp edges, there are too little pixels available for your mind to figure out what it means. And for bigger fonts, the edges become very jagged when not anti−aliased, so here you also might want to have aliased fonts. Of course you can also turn off the anti−aliasing for specific fonts. In other operating systems, most truetype fonts are not anti−aliased between 8 and 12 pixels, while only large Type1 fonts are anti−aliased.

   Use the following in your `XftConfig` to anti−alias only fonts of specific sizes:

```
match
      any size > 8
      any size < 15
edit
      antialias = false;
```

5. My fixed fonts do not appear or look _very_ wrong in the KDE konsole or similar programs!

   I noted that somehow a lot of fixed font do not tell Xft that they are fixed, and thus, mono spaced. Therefore only a part of the font is displayed. We can manually set the spacing for these fonts (this assumes you have fixed aliased with mono as in question 3 above):

```
match
      any family == "mono"
edit
      spacing = mono;
```

6. My Symbol, Webdings, etc. fonts do not show up!

   For some reason some (symbol) fonts are not correctly recognized, and Xft will show your default font, or a font which has the closest match (which is generally not what you mean at all). For Adobe Symbol and MS−webdings I did the following to get them working:

```
match
      any family == "webdings"
edit
      antialias = false;
```

```
        encoding += "glyphs−fontspecific";


match
        any family == "symbol"
edit
        antialias = false;
        encoding += "glyphs−fontspecific";
```

A useful way of figuring out these things is to activate debugging with:

```
export XFT_DEBUG=1024
```

This will generate a lot of output, especially if you have many fonts, because it lists the properties and scores of every font available. You can also use other values. For a nice summary of what happens (requested font, XftConfig substitutions, X server additions and the finally matched font), you can use XFT_DEBUG=2.

7. Why do my KDE programs start now soooo slooow?

   The Xft mechanism in XFree prior to 4.1 had to parse the `XftConfig` file each time a program was started. And the info of all these fonts had to be re−read. As of X 4.1.0, a cache is used and starting applications using Xft is much faster. Especially if you have many fonts this can be very useful. So, upgrading XFree86, and related packages, is a good idea.
8. I have a LCD screen on my laptop, can I use sub−pixel hinting instead of normal anti−aliasing?

   Yes you can. Sub−pixel hinting uses colors instead of gray pixels to do the AA. I do not have a LCD screen so I do not have any idea of how it looks but you can play with the "rgba" setting. Try:

   ```
   match edit rgba=bgr;
   ```

   or use rgb if you have a different type of monitor. For vertical AA you can try vbgr and vbgr.
9. My fonts still look bad!

   Good quality fonts are needed to start with. If you do not have some good TrueType fonts, it is worth it to go and look for them on the Internet. Other reasons why your fonts still look bad can be because of your build of freetype2. Snapshots versions before 2.0.2 were compiled with an option that had some patent issues. Therefore, the standard 2.0.2 and 2.0.3 compiles without this option. To fix this, download the freetype2 source rpm and change in include/freetype/config/ftoption.h line 314:

   ```
    #undef TT_CONFIG_OPTION_BYTECODE_INTERPRETER
   ```

   to:

```
#define TT_CONFIG_OPTION_BYTECODE_INTERPRETER
```

and rebuild with this modified source. See the freetype2 `README` file for details. Adobe Courier looks terrible on my system, so I made an alias so that Lucida console is displayed instead. If anyone can get it to display nicely I would appreciate knowing about it.

This is my `XftConfig`:

```
#
# XftConfig
#
# By: Danny Tholen
#
# Use with Type1 and TrueType fonts
#

dir "/usr/X11R6/lib/X11/fonts/Type1"
dir "/usr/X11R6/lib/X11/fonts/drakfont"
dir "/usr/share/fonts/default/Type1"

#
# alias 'fixed' and 'console' for 'mono'
# (some programs ask for 'mono' if they mean 'fixed';)
#
match any family == "fixed"        edit family =+ "mono";
match any family == "console"      edit family =+ "mono";


#
#Check users config file
#
includeif     "~/.xftconfig"


#
# Use TrueType fonts for defaults
# Danny: disabled
#match any family == "serif"       edit family += "Times New Roman";
#match any family == "sans"        edit family += "Verdana";


#
# Use lucida console as default fixed type font
# and set the spacing of "mono" to 100, this
# fixes broken fonts which are fixed, but do not
# set mono−spacing.
match
      any family == "mono"
edit
      family += "lucida console";
      spacing = 100;


#
# Alias between XLFD families and font file family name, prefer local fonts
#
match any family == "Charter"           edit family += "Bitstream Charter";
match any family == "Bitstream Charter" edit family =+ "Charter";

match any family == "Lucidux Serif"     edit family += "LuciduxSerif";
match any family == "LuciduxSerif"      edit family =+ "Lucidux Serif";

match any family == "Lucidux Sans"      edit family += "LuciduxSans";
match any family == "LuciduxSans"       edit family =+ "Lucidux Sans";

match any family == "Lucidux Mono"      edit family += "LuciduxMono";
match any family == "LuciduxMono"       edit family =+ "Lucidux Mono";


#
# TrueType font aliases
#
match any family == "Comic Sans"        edit family += "Comic Sans MS";
match any family == "Comic Sans MS"     edit family =+ "Comic Sans";
match any family == "Trebuchet"         edit family += "Trebuchet MS";
match any family == "Trebuchet MS"      edit family =+ "Trebuchet";
match any family == "Monotype"          edit family =+ "Monotype.com";
match any family == "Andale Mono"       edit family += "Monotype.com";
match any family == "Monotype.com"      edit family =+ "Andale Mono";


# Danny:
# set the AA for different fonts
#
# most TT fonts do not need to be aliased between
# 8 and 15 points, although this might be a matter of taste.
match
      any size > 8
      any size < 15
edit
      antialias = false;


# Danny: Courier looks terrible, and I
# cannot get most characters to fit nicely
# in their space. So I use courier 10 pitch
match
      any family == "courier"
edit
      family += "courier 10 pitch";
```

```
# these are symbols, and for some reason this needs to be added!:

match
      any family == "webdings"
edit
      antialias = false;
      encoding += "glyphs−fontspecific";

match
      any family == "symbol"
edit
      antialias = false;
      encoding += "glyphs−fontspecific";

match
      any family == "Standard Symbols L"
edit
      antialias = false;
      encoding += "glyphs−fontspecific";

match
      any family == "dingbats"
edit
      antialias = false;
      encoding += "glyphs−fontspecific";

match
      any family == "Cursor"
edit
      antialias = false;
      encoding += "glyphs−fontspecific";


# maybe arial looks better like this?:
match
      any family == "Arial"
      any size > 7
      any size < 15
edit
      antialias = false;


# end
```

4.1.4. GTK and GNOME

As mentioned above, KDE and QT do have solid anti−aliasing support with recent releases. GNOME, however, does not support anti−aliasing natively in stable releases (as of Feb 2002). But it is under developement and will be available when 2.0 is released, which should not be far away. If you are adventurous, you might consider getting the current development snapshot, and play with that.

In the meantime, there is the gdkxft project available at http://sourceforge.net/projects/gdkxft/. This will add anti−aliasing support to GTK+ 1.2 applications. This is a stand−alone library and not a patch. It will only effect GTK+ widgets (many of them but not all). There are some limitations, but mostly it works as advertized. Read the included `README` closely. An initial installation provides a good starting point. I used some of Danny's suggestions above, and it seems to work mostly. Even with the latest Mozilla (widgets only AFAICT)! GNOME users will need to use the "gdkxft" supplied theme.

For RPM users, there is a spec file in the tarball, and an RPM can be built with "rpm −tb <tarball>.tgz". Then install the binary RPM that is produced from that.

4.1.5. Afterword

You've gone through all the steps, and verified that the "RENDER" extension is available, but you don't perceive a difference? Well, maybe the applications themselves just aren't there yet, and are not able to utilize these new features. Netscape, for instance, is not able to take advantage of anti−aliasing.

So how to know what does and does not make use of anti−aliasing? A quick test is to use something like xmag, or gimp, and enlarge the text considerably. Look for diagonal lines, and if they are clearly stair−stepped with no softened edges. If so, then while anti−aliasing is technically available, it is not being used. With anti−aliasing you should see gradients instead of well defined sharp edges. You can compare this with xterm and "AA":

```
$xterm −fa charter −fs 14
```

One final point: anti−aliasing and TrueType are completely separate issues. One does not depend on the other, though both together can can enhance appearance significantly. Especially, with good quaility TrueType fonts!

4.2. Red Hat 7.x Differences

Red Hat 7.0 introduced some changes to X configuration over previous Red Hat versions. It is also different from the stock XFree86 configuration as addressed above. Notable differences:

- Both XFree86 3.3.6 and 4.x are included. If upgrading you may wind up with 3.3.6. The X configuration file is `XF86Config` for 3.3.6 and `XF86Config−4` for 4.x. Of course, you'll need to know which is which for editing and configuration purposes.
- xfs is still handling all font duties. A default Red Hat 7.x installation does not use the 'modules' section of XF86Config−4 for font handling. Instead it relies on xfs, which has this capability built in. This is different from a stock installation of XFree86 4.x where the X server does all the font work −− including TrueType.
- The socket for xfs is "unix/:7100" with RH 7.x, as opposed to "unix/:−1" in previous versions (i.e. Red Hat 6.x).
- As of Red Hat 7.1, the xfs init script actually runs **mkfontdir** and **ttmkfdir** on font directories known to xfs. So this step is not necessary when new fonts are added. Just restart xfs.

5. Adjusting Fonts in Specific Applications

5.1. Netscape

Let's face it, Netscape is an important application in Linux. We all use it, and we all need it, so let's look at it specifically for a minute. An out of the box Netscape installation is prone to the font problems we've discussed −− large fonts that get pixelized, splotchy looking fonts, fonts so small they are unreadable. In short, ugly. Maybe this is why you are here?

Hopefully, at this point you have followed the above suggestions. These steps can help greatly. TrueType font availability is almost a necessity, and you need a TrueType font server for this. Many web pages specify font families −− like Arial −− that are not typically available to Linux users. This is bad design, but having some of the basic TrueType fonts available will help greatly in overcoming the short−sightedness of some designers. Microsoft −− can't live with 'em, can't live without 'em.

Assuming you have TrueType working, from the Netscape menu select `Edit -> Preferences -> Fonts`. Open the `Variable Width Font` droplist on the right side of the window. Your TrueTypes should be there along with other fonts. Choose which ever one suits your fancy as the default. Check the `Allow Scaling` checkbox too. If the available point sizes are 0 and 12, you can go down and, and enter your desired point size in the box to the right and click on the `OK` button. The down−side to this is that Netscape will not remember these settings, and you will have to do this each time you start Netscape. *Unless* −− you have `fonts.alias` set up already. Then this will solve these problems. See Section 3.3 for more on `fonts.alias`.

You might consider experimenting with some `~/.Xdefaults` (or perhaps it's`~/.Xresources` on your system) settings too:

```
      Netscape*DocumentFonts.sizeIncrement: 10
      Netscape*documentFonts.maximumPoints: 240
      Netscape*documentFonts.xResolution*iso−8859−1: 120
      Netscape*documentFonts.yResolution*iso−8859−1: 120
```

The 'sizeIncrement' controls how much of a jump Netscape makes when different 'basefont' sizes are specified ala:

```
<basefont size=7>
```

for instance. The default is '20', which is a pretty good jump. Changing this can help Netscape from scaling to too large and too small of a font. The x and y resolutions are roughly equivalent to 'dpi' settings. Any random number within reason can be used here. Experiment.

Then run:

```
$ xrdb −merge ~/.Xdefaults
```

(or `.Xresources` as the case may be) and restart Netscape. There are many settings that can be tweaked or altered this way. Look at the `Netscape.ad` (app defaults) file that should be included with Netscape packages.

If this approach does not get the job done as far as the 'tiny fonts' problem in Netscape, then see the `fonts.alias` section above. You can really fine tune many things with this approach.

5.2. Mozilla

Mozilla configuration should be roughly the same in many respects. You might find, however, that Mozilla does a much better job of handling fonts, and pages will look better overall. Highly recommended! The only caveat is, it seems to need a fairly fast system. It may be pretty sluggish on older systems.

Also, user preferences can be stored in "`user.js`". Not to be confused with "`prefs.js`". Put `user.js` in whatever directory you find `prefs.js` in (this is not a consistent location). Attempt to set a minimum font size:

```
// Don't ever show me a font smaller than this: some samples.
user_pref("font.min−size.variable.", 12);
user_pref("font.min−size.variable.x−western", 12);
user_pref("font.min−size.fixed.x−western", 12);
```

6. Odds and Ends

6.1. Notes

- Unfortunately there is no unified font handling system for Linux. You will have to configure each individual program so you can use TrueType, Type 1 or fonts that pique your fancy. And each program may well have its own way of doing this so you will have to RTFM. Desktop Environments like GNOME and KDE may provide much of this functionality however for apps that are under their control.
- Most GUI apps should be able to use TrueType, and Type 1 fonts too. Wordperfect for Linux, however, cannot use TrueType. (See the links section below for more on Wordperfect.) Text editors, terminal programs and the like need fixed width fonts, and do not play well with TrueType or other scalable fonts.
- Though not discussed here, Type 1 fonts provide many of the same benefits as TrueType and are historically better supported in the Unix world. You likely have many of these installed already. Unfortunately however, Type 1 are not a web standard like TrueType. But they are suitable for many other purposes. They are where it's at for printing. See ghostscript for more on this.
- While it is possible to specify a default point size for the xfs font server, very few applications will actually use this value.
- Abiword comes with a suite of fonts, called 'Abisuite'. Apparently, some of these fonts have the same names as some of the well known MS TrueType fonts: Arial, etc. And apparently, these are of much less quality. And because of the way X searches for fonts, it may find these first and use these, even if the 'real' ones are installed and may be the preferred choice. The solution is to uninstall 'Abisuite'.
- The new Xft rendering extensions of XFree86 4.x will mostly supplant similar features as provided by xfs, and older XFree86 extensions. For instance, font aliasing should be done in `XftConfig` if the new extensions are being used. This would only be true where the application is built against a toolkit (like QT or GTK) that supports the new extensions. This is still not universally supported. In fact, only KDE is wide spread support.

6.2. Links

- The Video Timings HOWTO, the ins and outs of getting the most from your monitor. (Applicable only to XFree86 v.3.x.)
- Font HOWTO Many good tips for installing fonts and for applications such as StarOffice, Applixware, Wordperfect, Ghostscript, TeX/LaTeX.
- A TrueType HOWTO, good tips for printing, and a few application specific tips.
- xfsft Homepage, TrueType font support for X. This is the origin of the "freetype" font module for XFree86 4.x, and Red Hat's xfs. Good site, and good links to other information related to fonts and TrueType.
- Some Linux for Beginners. Great font site, and other Linux topics. Covers many of the topics discussed here in more detail. Some font and other tips for Mozilla: http://home.c2i.net/dark/My_Mozilla_FAQ.html.
- Two guides specifically for Debian from Paul D. Smith: http://www.paulandlesley.org/linux/xfree4_tt.html and for 3.3.x: http://www.paulandlesley.org/linux/debian_tt.html.
- X−TrueType Homepage, and yet another TrueType Font server, especially good for Japanese, Chinese and Korean character sets.
- Tips on font size problems from Netscape.

- Wordperfect for Linux −− Fonts and Printers by Rod Smith, the author of *Using Corel Wordperfect 8 for Linux* from Que. Excellent information on Wordperfect and where TrueType fits in.
- XFree86 Project, the guys and gals who do an incredible amount of work to give us a killer GUI environment. Some info on fonts in XFree86 4.x.
- Microsoft Web Fonts direct from the Lion's den −− and they are free! These can be unpacked in Linux (see above).
- Web Fonts RPM package, contains a few of the MS web browser fonts. http://packages.debian.org/unstable/graphics/msttcorefonts.html is a similar package for Debian.
- Sebastiano Vigna's http://freshmeat.net//webFonts4Linux automates the downloading, extracting and installation of the Microsoft fonts all in one neat package.
- Freeware Connection −− Free Fonts Sites lots of links to lots of sites.
- Bitstream's Geometric Slabserif TrueType Font.
- Two converters for converting a Mac Font "suitcase" to a *nix compatible font: http://www.macinsearch.com/infomac2/font/util/tt−converter−15.html and http://www.netmagic.net/~evan/shareware/#TTFontConvert
- The Unicode HOWTO: http://linuxdoc.org/HOWTO/Unicode−HOWTO.html
- Two sources of 'free' TrueType fonts with large Unicode support are Bitstream Cyberbit, which covers Roman, Cyrillic, Greek, Hebrew, Arabic, combining diacritical marks, Chinese, Korean, Japanese, and more, and is available from ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/Cyberbit.ZIP. And Lucida Sans Unicode, which is included in IBM's JDK 1.3.0beta for Linux, and covers Roman, Cyrillic, Greek, Hebrew, combining diacritical marks. This can be downloaded from ftp://ftp.maths.tcd.ie/Linux/opt/IBMJava2−13/jre/lib/fonts/ as LucidaSansRegular.ttf and LucidaSansOblique.ttf. Thanks to Tzafrir Cohen for these references. He also has a nice page on Hebrew fonts and related topics at http://www.iglu.org.il/faq/?file=133.