Fonts in Dzongkha

Pema Geyleg
Department of Information Technology
pema.geyleg@gmail.com

Abstract

This paper documents the research conducted while studying Dzongkha open type font support in Linux operating system.

1. Introduction

Fonts play special role for the development of rendering software like Pango and ICU layout engine. Background information on true type fonts and how it is being supported in Linux operating system is being provided here. The Open type fonts for Dzongkha had already been developed by Dzongkha Development Authority (DDA) while enabling Dzongkha computing in Microsoft Windows operating system. We just went about reusing their fonts on Linux operating system. It is by no means a complete reference neither a universal guide for all fonts.

2. Methods

2.1. True Type Fonts

Apple Computer, Inc originally designed the True Type digital font format. It was a means of avoiding per-font royalty payments to the owners of other technologies, and a solution to some of the technical limitations of Adobe's Type 1 format. Originally code named "Bass" (as they were scalable fonts) and later "Royal", the TrueType format was designed to be efficient in storage and processing, and extensible. It was also built to allow the use of hinting approaches already in use in the font industry as well as the development of new hinting techniques, enabling the easy conversion of already existing fonts to the TrueType format. This degree of flexibility in TrueType's implementation of hinting makes it extremely powerful when designing characters for display on the screen. Microsoft had also been looking for an outline format to solve similar problems, and Apple agreed to license True Type to Microsoft.

A digital font is not limited to the characters associatated with a given alphabet or script. Many different kinds of information are included in a True type font file which are utilised by the TrueType rasterizer and the operating system software so that the character s displayed on the computer screen or print out are exactly the same as what the font designer intended them to. A TrueType font's information is arranged in a series of tables. Other than the shapes of each character, information like how the characters should be spaced vertically and horizontally within a block of text, details of character mapping(which governs the variety of characters included in the font and the keystrokes needed to access them), and much more are included. Manufacturer's details, viz, copyrights, names and licensing permissions are also included in the fonts.

One thing that true type include is obviously the shape of each character. So each and every letter contained in a TrueType font is stored as an outline, or more accurately as a mathematical description of the character constructed from a series of points. Hence TrueType is also known as an outline font format. The advantage being that only one character is needed to produce all the sizes of that character. By which a single outline can be scaled to an enormous range of different sizes, enabling the same character to be displayed on monitors of different resolutions, and to be printed out at a large number of different sizes.

The user can never see the outlines stored in an outline font, because a bitmap is produced by the TrueType rasterizer before a character can be displayed on the screen or printer. The reason being both screen displays and printers use dot patterns to represent images. So character outlines contained in the TrueType font are scaled to the requested size, and are converted into bitmaps by turning on the pixels encompassed by the outline by the process known as scan conversion or rasterization.

2.2. True Type Open Fonts

True Type open fonts is an extension to the TrueType font standard. It contains additional information which helps in extending the capabilities of the fonts to support high quality international typography.

- It allows an abundant mapping between characters and glyphs, supporting ligatures, positional forms, alternates and other substitutions.
- It includes supportive information for features for two dimensional positioning and glyph attachment.
- It contains precise script and language information, for text processing application to adjust its behavior accordingly.
- It has an open format to allow developers of the font to define their own typographical features.

A TrueType Open font identifies complicated typographical issues that largely affects people familiar with text-processing applications in multi-lingual and non-Latin environments.

2.3. Glyph

It is not users viewing or printing characters, but its users viewing or printing glyphs. A character is represented by a glyph. A collection of glyphs makes up a TrueType font. While retrieving glyphs, the client uses information in the "cmap" table of the font, which maps the client's character codes to glyph indices in the table.

Combinations of characters and alternative forms of characters(glyphs and characters do not strictly correspond one-to-one) can also be represented by glyphs. For example, a user might type two characters, which might be better represented with single ligature glyph. In a converse way, the same character might take different forms at the beginning, middle or end of a word, thus a font would require a number of different glyphs to represent a single character. TrueType Open fonts contain a table that provides a client with information about possible glyph substitutions.

2.4. XFree86

The organization, XFree86 Project, Inc produces XFree86, which is a freely redistributable open-source administration of the X window system. XFree86 runs largely on UNIX and UNIX-like operating system such as Linux, Sun Solaris x86, Mac OS X, including all of the BSD variants, and other platforms like OS/2 and Cygwin as well.

XFree86 provides a client/server interface between display hardware (the mouse, keyboard and video display) and the desktop environment. It also provides the windowing infrastructure and a standardized application interface (API). Most importantly, it is platform independent, network transparent and extensible.

2.5. XFS

XFS stands for the X font server. It provides a standard mechanism for an X server to communicate with a font renderer, frequently running on a remote machine. It commonly runs on TCP port 7100 or thereabouts.

2.6. Traditional X Font System

The Traditional X font system in XFree86 4.X supports bitmap fonts (BDF, PCF), Type1 and TrueType fonts with the aid of Free Type library. All are managed at the X server side, either by the X server itself via loadable modules or by a dedicated X font server (XFS).

X fonts are referred to by X Logical Font Description (XLFD),in which font properties are described, separated by hyphens:

-fndry-fmly-wght-slant-sWidth-adstyl-pxlsz-ptSz-resx-resy-spc-avgWdth-rgstry- encdng

For example:
-Adobe-times-medium-r-normal--*-100-75-75-p-54-iso8859-1

It means Times family by Adobe, medium weight, Roman upright shape(no slant), normal width, no additional style, any pixel size,10.0 points, 75x75 DPI design resolution, proportional spacing, 5.4 pixels average glyph width, using ISO8859-1 character set.

Some fields of the XLFD can be wildcard ('*') to mean the first matched font in the system. To add new fonts to X, you need to prepare a font list file (fonts.dir) in the directory using the mkfontdir command for bitmap fonts, or mkfontscale followed by mkfontdir for scalable fonts. After that, you just add the font path to the X server or XFS, depending on your X server setup.

To add font path directly to X server (assuming that appropriate font support modules are loaded at X server startup):

```
#xset fp+ /your/font/path
```

```
#xst fp rehash
```

2.7. Disadvantages of using font server:

As all the fonts are held twice, once in the font server and once in the X server, more memory is used up. Which can pose a possible security risk. So its not wise to use it in a system aiming for maximum security. Stability problems might also be caused for the X window system. But with respect to the recent versions of the XFree86 it appears to be adequately stable though.

Editing the XFS configuration file at /etc/X11/fs/config is presumably not a necessary step; however it would be nice to have a quick look at it.

The display mechanism under the X window system works as follows: In contrast with text display terminals, X displays can display text in different sizes and shapes. The X server retrieves the symbols from the font by using character code indexing. Predominantly, there are two types of fonts – fonts based on Bitmap and outline based or curve based fonts. Because its scalable, outline based fonts are becoming very popular. TrueType Font (TTF) is broadly considered as the best scalable fonts for low resolution devices like displays. A freely available font server, Xfstt is for the TrueType fonts and it supplies fonts to the X window system display servers. So apparently, a high level flexibility of customizing the display mechanism and keyboard input is catered for the X Window system also.

3. Results

The open type font supporting Dzongkha script namely

1) Joyig
2) Uchen
3) Tashi

has already been developed at DDA (Dzongkha development authority) while incorporating Dzongkha computing in Microsoft window operating system. It is to be noted that only substitution rules has been specified for these fonts. No positioning rules were specified for these fonts. It was just the case of testing whether it could be reused in Linux operating system. These fonts were successfully installed on Linux operating system as follows.

3.1. Installing Fonts in X Windows

Basic Font installation concept/steps:

- Create a directory & copy the fonts in it
- Generate fonts.dir database
- Set the font path

Restart either the Font server or the X server.

3.1.1.In Fedora core

The most current desktops are using Xfree86 4.2/4.3 support Xft or fontconfig. The Font configuration file is /etc/fonts/fonts.conf or ~/.fonts.conf. With fontconfig, font installation is as simple as copying the fonts in one of the configured directories in /etc/fonts/fonts.conf.

Installation instruction:

- Make a directory called dzongkha in /usr/share/fonts
  # mkdir /usr/share/fonts/dzongkha
- Copy the fonts into that dzongkha directory
  # cp font.ttf /usr/share/fonts/dzongkha/
- Update the font configuration file
- # fc-cache /usr/share/fonts/dzongkha

OR

- # fc-cache -f -v
  to get the font configuration updated globally (-f is to force to update in all configured paths, -v for verbose output).

4. Discussion

The fonts were getting displayed correctly except that it was being displayed at a very small size. Our font sizes of 32 point were approximately equivalent to 12 point size of other fonts. Detailed studies on how to rescale the fonts were carried out and we came to the conclusion that we need to rescale each and every individual glyphs in the font manually. However, there is no people at DDA who can rescale the font according to our need.

5. Conclusion

As mentioned above under discussion section, the fonts need to be rescaled. After discussing with lots of fonts experts we have concluded that the Dzongkha fonts need to be redeveloped with all the features like Positioning rules and Character classification.

Dzongkha localization team at DIT had not been able to fix the rendering glitch in Open Office Application. This has also been attributed to the insufficient rules defined in the existing font which does not allow much flexibility while developing ICU layout engine.

The user guide for font installation and using Dzongkha fonts in Linux operating system is not provided here as it has been already provided with the research on Keyboard driver in Linux operating system in the previous quarterly report.

6. References

[1] J Flowers. And S Gildea. X Logical Font Description Conventions Version 1.5. X Consortium Standard, X Version 11, Release 6.4.

[2] K Packard. Design and Implementation of the X Rendering Extension. In Proceedings of *FREENIX Track, Usenix Annual Technical Conference*, Goston, MA, June 2001. USENIX.

[3] K Packard. The Xft Font Library: Architecture and Users Guide. In Proceedings of *XFree86 Technical Conference*, The XFree86 Project, Inc., October 2001. Usenix Associatation.

[4] K Packard. Font Configuration and Customization for Open Source Systems. In Proceedings of *GNOME Users and Developer European Conference*, 2002.

[5] "Theppitak notes on Thai language Localization for Linux operating system" http://linux.thai.net/thep/presents/PAN/theppitak-linux-l10n-paper.pdf

[6] Get the most out of Open Type Fonts: Tutorial" http://www.adobe.com/tips/ill11openty/main.html

[7] "International components of Unicode in terms of localization" http://oss.software.ibm.com/icu/