

# Charset2FBFont v2.00

Dieses Programm wandelt, so gut es geht, einen ausgewählten Vektorfont in eine Bitmap um. Eine zusätzliche Datei liefert alle weiteren Informationen um später Zugriff auf die entsprechenden Teilgrafiken innerhalb der Bitmap zu bekommen.

Dieses Programm ist ein kleines Remake, nun in FreeBASIC geschrieben.

Zum Rendern wird die FreeType Library als „Fremdsoftware“ benutzt. Daraus kann noch eine weitere Abhängigkeit an eine weitere Bibliothek entstehen.

Für Windows Rechner werden die entsprechenden Bibliotheken mitgeliefert.

Für Linux Rechner kann ich leider keine Aussage über Lauffähigkeit und Abhängigkeiten treffen.

## Bedienung

Zuerst muß mit „**Open Font**“ ein beliebiger Font geöffnet werden. Es öffnet sich ein Dateidialog.

Bei meinen Tests habe ich aber festgestellt, daß dieser Dialog unter Windows nicht den Ordner „Fonts“ (im Ordner „Windows“) anzeigt. Eine andere Lösung als den Pfad per Hand einzugeben habe ich derzeit nicht parat, danach wird der Inhalt normal angezeigt.

Ist ein Font erfolgreich geöffnet worden, so wird dieser links oben angezeigt.

Mit „**Font Height (pt):**“ und „**Resolution (dpi):**“ läßt sich die Größe (Höhe) des zu rendernden BitmapFonts beeinflussen. Die Fonthöhe wird hier in Pt angegeben. Resolution bezieht sich auf die Auflösung des Ausgabemediums, hier ist es der Monitor.

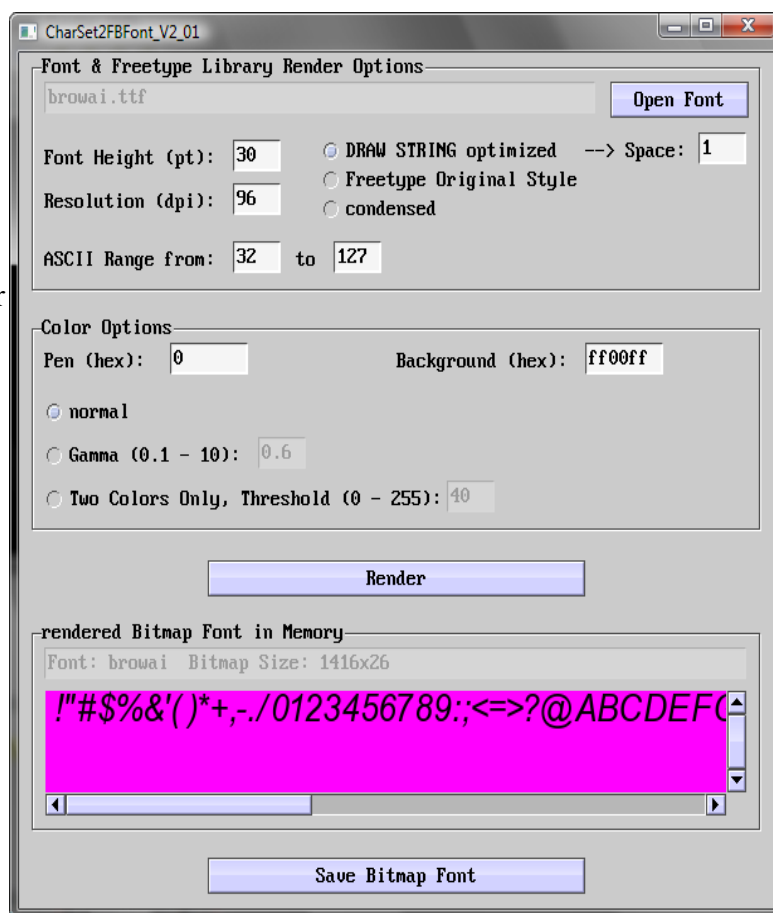
Der entstandene BitmapFont entspricht leider in den seltensten Fällen exakt der gewünschten Pixelhöhe.

Somit hilft derzeit nur ein „Spielen“ mit diesen beiden Werten und nochmaliges Rendern und auch ein wenig Kompromissbereitschaft ;).

Mit „**ASCII Range from / to:**“ läßt sich der Zeichenumfang einstellen, der im BitmapFont enthalten sein soll.

Mit „**DRAW STRING optimized**“ werden die Zeichen innerhalb der Bitmap so angeordnet, daß eine Nutzung des BitmapFonts mit dem DRAW STRING Kommando der FBGFX Bibliothek möglich ist. Nur in diesem Modus läßt sich mit „**Space:**“ der Abstand der Zeichen zueinander einstellen.

Im „**FreeType Original Style**“ wird der Font so in die Bitmap geplottet, wie es die Bibliothek selbst tun würde. Dies ist aber nicht bei allen Fonts zu empfehlen, da „Überschneidungen“ auftreten können. Im Schriftbild mag dies gewollt sein, als Teilbitmap, in der noch Fragmente vom Nachbarzeichen enthalten sind, eher ungünstig.



Die „**condensed**“ Variante versucht alle Zeichen auf engstem Raum unterzubringen, um die Bitmap so klein wie möglich zu halten.

Mit „**Pen (hex):**“ und „**Background (hex):**“ kann der Font und der Hintergrund entsprechend coloriert werden. Die Farbwerte sind hexadezimal anzugeben.

Als Grundlage für Farbberechnungen dient eine 256Grayscale Bitmap, geliefert von der FreeType Bibliothek.

Mit „**normal**“ werden die „Intensitätswerte“ der originalen Bitmap entsprechend farbig umgerechnet.

Mit „**Gamma**“-wert läßt sich der Antialiasbereich der Zeichen etwas wichten. Die Zeichen erscheinen dadurch etwas „dicker“ oder „dünner“.

„**Two Colors Only**“ reduziert tatsächlich die Anzahl der benutzten Farben auf 2. Ein Schwellwert bestimmt hierbei welches Pixel dem Hintergrund bzw. Stift zugeordnet wird.

Wichtig!!! Die gespeicherte Bitmap hat immer noch eine Tiefe von 24bit

Der „**Render**“ Button versucht dann aus allen oben genannten Einstellmöglichkeiten eine Bitmap zu generieren. Eine zusätzliche Datei enthält alle wichtigen Informationen den erzeugten BitmapFont betreffend. Im Bereich unterhalb des Renderbuttons (rendered Bitmap Font in Memory) werden nach erfolgreichem Rendern die „Eckdaten“ des erstmal nur im Speicher(!) befindlichen Bitmapfonts angezeigt.

„**Save Bitmap Font**“ erzeugt einen Ordner mit Namen des Fonts und speichert darin die Bitmap als auch eine sogenannte Descriptor Datei, die alle relevanten Daten zum Font enthält.

## Descriptor Datei

Diese Datei wird zusätzlich zum Font erzeugt und enthält alle Daten zum Font. Hier ein kleiner Auszug aus einer solchen Datei:

```
7309,124,1,97,32,127
32,0,49,0,0,10,97
33,49,39,19,94,10,6
34,88,63,43,41,10,3
.
.
.
.
125,7046,56,36,106,10,6
126,7102,58,38,16,10,8
127,7160,149,129,65,10,31
```

Die erste Zeile enthält folgende Daten in selbiger Reihenfolge durch Komma getrennt:

**Bitmapbreite (px)**

**Bitmaphöhe (px)**

**PlotModus** 1=DRAW STRING optimized, 2=FreeType Style, 3=condensed

**Baseline (px)** Abstand der Grundlinie des Fonts von oben

**ASCIICode erstes Zeichen**

**ASCIICode letztes Zeichen**

Ab der zweiten Zeile erfolgt die Beschreibung der einzelnen Zeichen, Daten ebenfalls durch Komma getrennt:

**ASCIICode des Zeichens** das in dieser Zeile beschrieben wird

**OffsetX (px)** Abstand des Zeichens vom rechten Rand der gesamten Bitmap

**AdvanceX (px)** Laufweite des Zeichens innerhalb des Schriftbildes

**GlyphWidth (px)** Breite des Glyphs

**GlyphHeight(px)** Höhe des Glyphs

**GlyphLeft (px)** Abstand des Glyphs vom linken Rand

**GlyphTop (px)** Abstand des Glyphs vom oberen Rand