Example of How to Convert a TrueType Font to TinyFont for use with the Microsoft .NET Micro Framework

By Sean D. Liming and John R. Malin
SJJ Embedded Micro Solutions / www.sjjmicro.com

January 2008

The TFConvert that is part of the Microsoft .NET Micro Framework SDK converts TrueType Fonts to the TinyFNT format that is used by .NET Micro Framework (.NET MF). TFConvert is a command line tool that has the following syntax:

```
TFConvert <input file> <output file>

        Where
        <input file>  = Font definition file (.fntdef)
        <output file> = Font output file (.tinyfnt)
```

The input file is a font definition file. This is a text file that defines what font is to be converted, the range of characters to convert, and various options to refine the font during conversion. The Visual Studio online help lists 14 options to refine the fonts. As you can guess, creating the fntdef file is where the real work is, but you don't have to be a font expert to use this tool. At a minimum all you need is the following:

- AddFontToProcess – Sets the path to the TTF font to be converted
- SelectFont – Specifies a set of attributes
- ImportRange – Provides a list of Unicode Characters to import

Most of the problems encountered are in the SelectFont options. If there are errors in the file, TFConvert is friendly enough to provide feedback on what the error is and what it found.

For a fun example, we'll convert the Aurabesh font for use in a .NET MF application. The Aurabesh font is from Star Wars™. English characters are replaced 1:1 with a unique set of characters. You can see the Aurabesh font characters used in the movies, video games, and on each of the planets at the center of the www.starwars.com main page:

- Corusant (Coruscant)
- Naboo (Naboo)
- Tatooine (Tatooine)
- Mustafar (Musta far)

You can download the font from various websites. Here are a few sites:

- http://www.fontfiles.com/Fonts/Detailed/56.html
- http://www.geocities.com/TimesSquare/4965/sffont.html
- http://www.fonttrader.com/detailed~name~Aurabesh~font~3638.htm

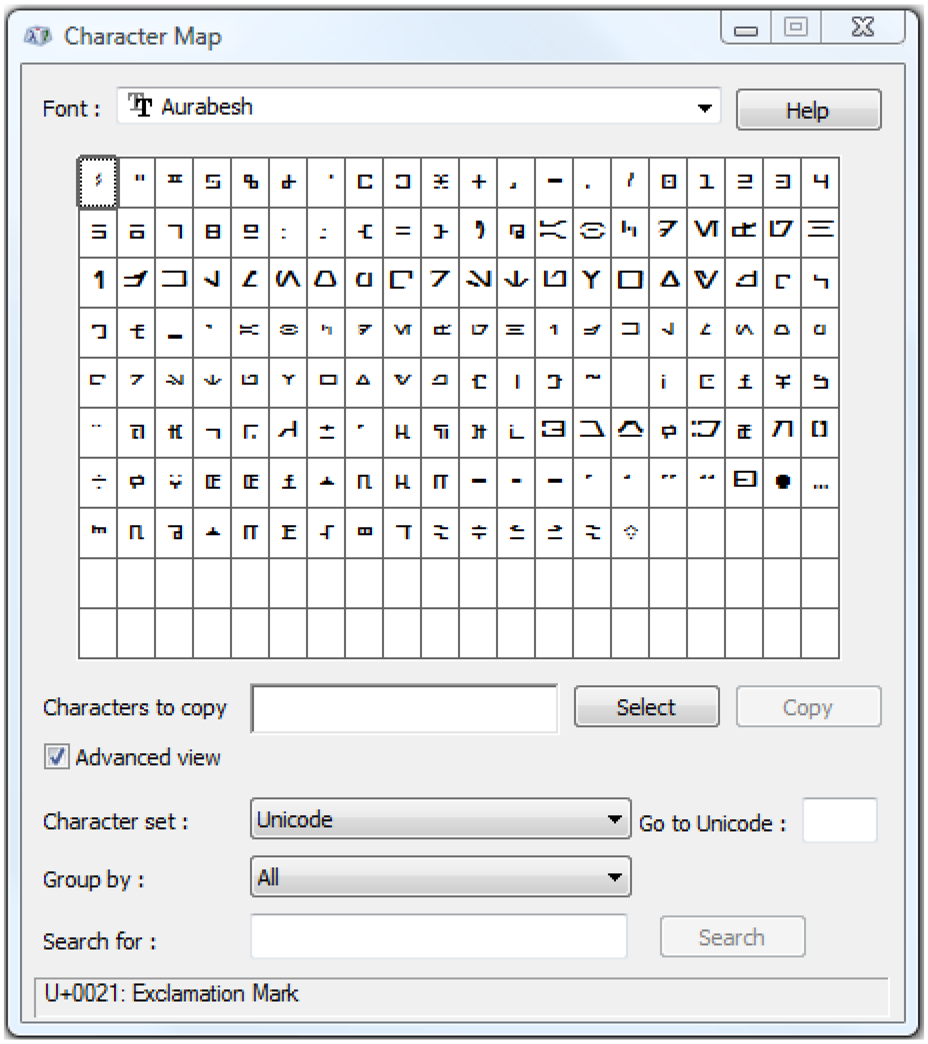
After a little search, a couple examples were found to base the Aurabesh.fntdef. Opening Notepad, the following Aurabesh.fntdef information was entered:

```
# Location of the Original Font
```

```
AddFontToProcess c:\Windows\Fonts\Aurabesh.ttf

# Aurabesh Star Wars
SelectFont "FN:Aurabesh,WE:400,HE:-14,PF:3,FullName:Aurabesh"
```

The SelectFont options were a guess based on the parameters used in other font definition files. We will come back to this a little later. The next step is to define the characters to convert. The Character Map (charmap.exe) tool can be used to view the characters that comprise the font.



Font : Aurabesh

Characters to copy :

Advanced view [✓]

Character set : Unicode

Go to Unicode :

Group by : All

Search for :

U+0021: Exclamation Mark

Figure 1 - Character map application shows all the characters in the font file, as well as, the Unicode number for the font.

At the bottom of Character Map is the Unicode value for each character. In the Figure 1, the Exclamation mark is U+0021. The number is a Hex value, and the integer equivalent is required for the ImportRange. Also, the character numbers are not contiguous. There are gaps and skips between numbers. Walking through each character in Character Map, all the character ranges and individual characters to be converted were found. Although all the characters did not need to be converted, for completeness of this example, the full character set was converted. Calc.exe was used to convert the hex values to the integer equivalents. The table below shows the final result.

Character Map Range	Import Range Final Values
U+0021 – U+007E	33 – 126
U+00A0 – U+00B6	160 – 182
U+00BB – U+00BB	187 – 187
U+00BF – U+00BF	191 – 191
U+00C6 – U+00C7	198 – 199
U+00C9 – U+00C9	201 – 201
U+00D8 – U+00D8	216 – 216
U+00DF – U+00DF	223 – 223
U+00E6 – U+00E6	230 – 230
U+00F1 – U+00F1	241 – 241
U+00F3 – U+00F3	243 – 243
U+00F7 – U+00F8	247 – 248
U+00FF– U+00FF	255 – 255
U+0152 – U+0153	338 – 339
U+0192 – U+0192	402 – 402
U+0394 – U+0394	916 – 916
U+03A9 – U+03A9	937 – 937
U+03BC – U+03BC	956 – 956
U+03C0 – U+03C0	960 – 960
U+2010 – U+2010	8208 – 8208
U+2013 – U+2014	8211 – 8212
U+2018 – U+2019	8216 – 8217
U+201C – U+201D	8220 – 8221
U+2020 – U+2020	8224 – 8224
U+2022 – U+2022	8226 – 8226
U+2026 – U+2026	8230 – 8230
U+2122 – U+2122	8482 – 8482
U+2126 – U+2126	8486 – 8486
U+2202 – U+2202	8706 – 8706
U+2206 – U+2206	8710 – 8710
U+220F – U+220F	8719 – 8719
U+2211 – U+2211	8721 – 8721
U+221A – U+221A	8730 – 8330
U+221E – U+221E	8734 – 8734
U+222B – U+222B	8747 – 8747
U+2248 – U+2248	8776 – 8776
U+2260 – U+2260	8800 – 8800
U+2264 – U+2265	8805 – 8805
U+22F2 – U+22F2	8946 – 8946
U+25CA – U+25CA	9674 – 9674

Table 1 - Unicode Hex and Integer values found using the CharMap utility

***Note**: The dashes between the ranges should not be used in the font definition file. There were placed in the table for readability.*

Multiple ImportRange statements were added to the font definition file from the values listed in Table 1:

```
ImportRange 33 126
ImportRange 160 182
:
:
:
ImportRange 8946 8946
ImportRange 9674 9674
```

Now, we are ready to use TFConvert to convert the definition file…

```
C:\> TFConvert aurabesh.fntdef aurabesh.tinyFNT
```

…and this first attempt failed. The output from TFConvert indicates several times that the FullName in the font definition file was wrong:

Font cannot be found matching SelectFont descriptor FullName. Expected 'Aurabesh', found 'Macromedia Fontographer 4.1 Aurabesh'.

To correct this, we will change the SelectFont option to the following:

```
SelectFont "FN:Aurabesh,WE:400,HE:-14,PF:3,FullName:Macromedia
Fontographer 4.1 Aurabesh,Style:Regular"
```

Running the conversion again yields another error:

Font cannot have negative InternalLeading
Metrics : A:11 D:2 I:-1 E:0
Adjusted Metrics: A:11 D:2 I:-1 E:0

This stumped us for a second. Not being a Font experts, who knows what the problem is. The only negative value in the SelectFont options was for the height (HE:-14). By simply removing this option the conversion was successful. What HE:-14 has to do with InternalLeading, we don't know, but some dumb guesses turn out okay. In the end, we did an Internet search and the Internalleading error was a direct result of the HE parameter. The final aurabesh.fntdef file is as follows:

```
# Location of the Original Font
AddFontToProcess c:\Windows\Fonts\Aurabesh.ttf

# Aurabesh Star Wars
SelectFont "FN:Aurabesh,WE:400,PF:3,FullName:Macromedia Fontographer
4.1 Aurabesh,Style:Regular"

# Import Ranges from Charmap.exe

ImportRange 33 126
ImportRange 160 182
ImportRange 187 187
ImportRange 191 191
ImportRange 198 199
ImportRange 201 201
ImportRange 216 216
ImportRange 223 223
ImportRange 230 230
ImportRange 241 241
ImportRange 243 243
ImportRange 247 248
ImportRange 255 255
ImportRange 338 339
ImportRange 402 402
ImportRange 916 916
ImportRange 937 937
ImportRange 956 956
ImportRange 960 960
ImportRange 8208 8208
ImportRange 8211 8212
ImportRange 8216 8217
ImportRange 8220 8221
ImportRange 8224 8224
ImportRange 8226 8226
ImportRange 8230 8230
ImportRange 8482 8482
ImportRange 8486 8486
ImportRange 8706 8706
ImportRange 8710 8710
ImportRange 8719 8719
ImportRange 8721 8721
ImportRange 8730 8330
ImportRange 8734 8734
ImportRange 8747 8747
ImportRange 8776 8776
ImportRange 8800 8800
ImportRange 8805 8805
ImportRange 8946 8946
ImportRange 9674 9674
```

The next step is to test the font in a .NET MF application. Out of the box, a .NET MF Window Application project is already setup to display hello world. The font used is a small font that gets automatically generated when you first create the project. We will replace this font with the Aurabesh.tinyFNT.

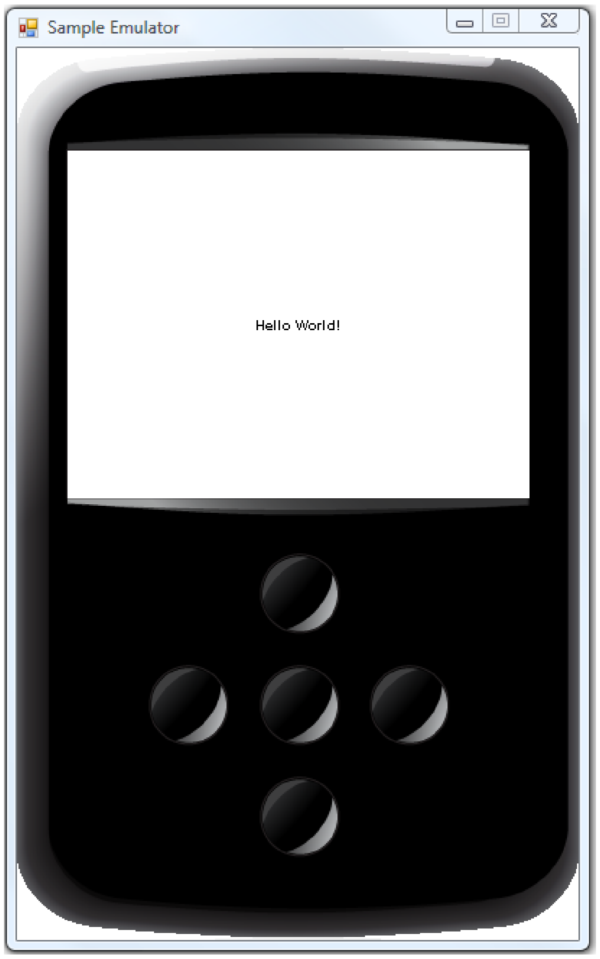
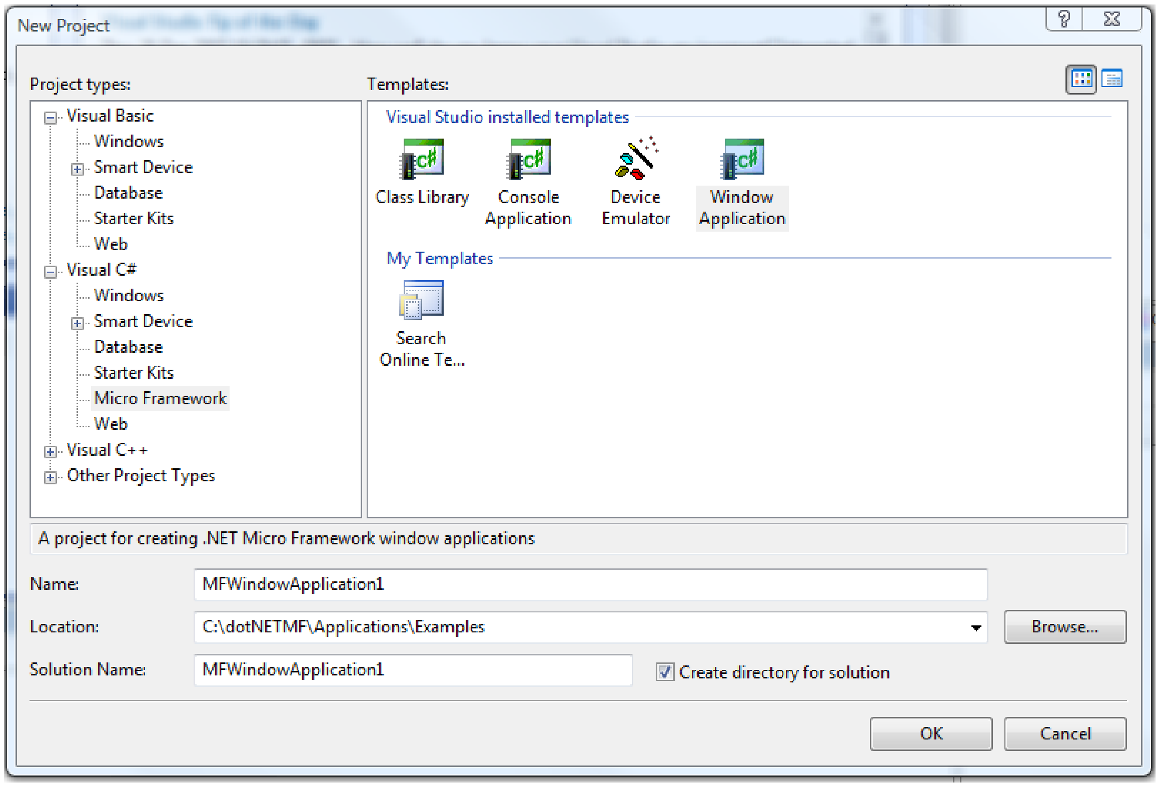


Figure 2 - A basic .NET MF Window application outputs “Hello World!” as you can see from the Microsoft Emulator.

Visual Studio 2005 and the Microsoft .NET Micro Framework SDK are required to perform this basic test.

1. Open Visual Studio 2005
2. Create a .Net Micro Framework Windows Application.



A project for creating .NET Micro Framework window applications

Name: MFWindowApplication1

Location: C:\dotNETMF\Applications\Examples

Solution Name: MFWindowApplication1

[✓] Create directory for solution

Figure 3 - Creating a .NET MF Window Application

3. Once you create the project, the next step is to add Aurabesh to the Resources. In solution Explorer, click to open the Resources.resx.
4. The top-left most drop down, you will see Strings. Change this to **Files**.
5. Now in the Add Resource drop down, select **Add Existing File…**
6. Locate and open the Aurabesh.TinyFNT file or the font that you have converted. The font will be added to Resources and made available to the program

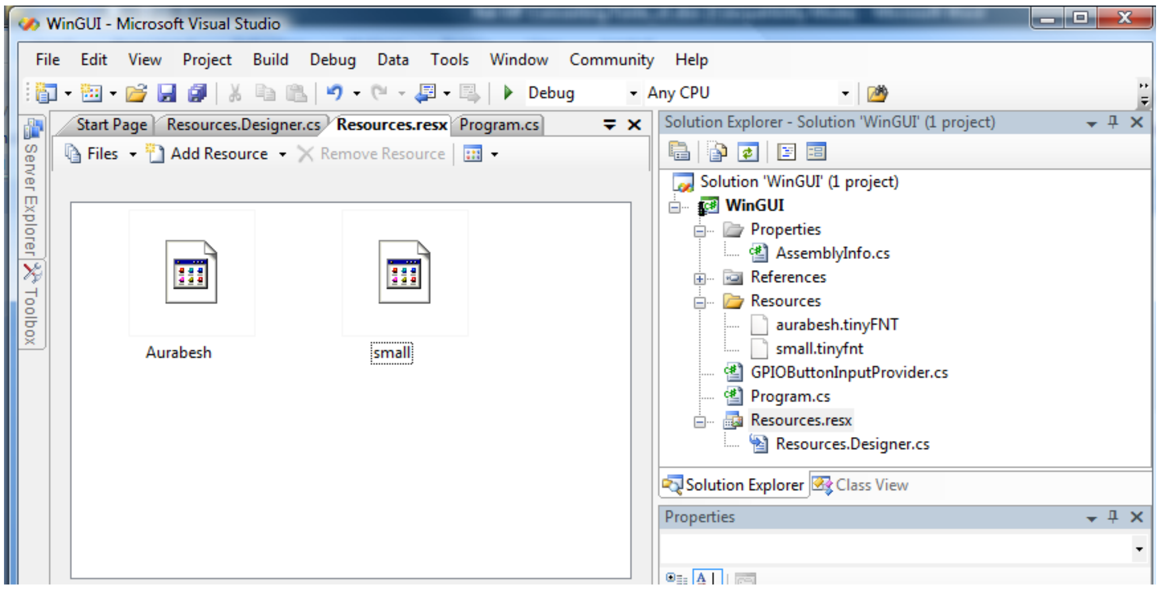


Figure 4 - Adding Aurabesh.tinyFNT as a resource for the project.

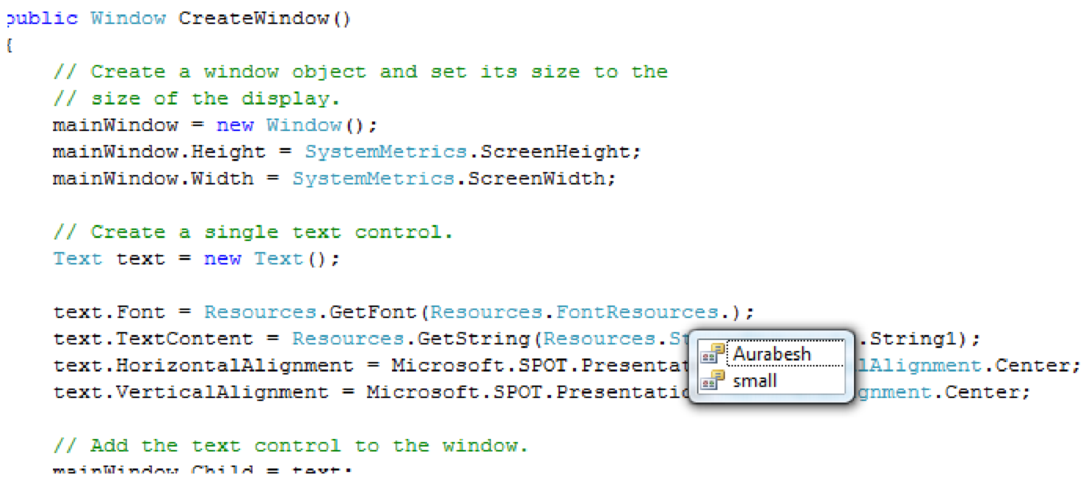
7. Open the Program.cs file.
8. Change the following line from

```
text.Font = Resources.GetFont(Resources.FontResources.small);
```

to the following:

```
text.Font = Resources.GetFont(Resources.FontResources.Aurabesh);
```

Note: Intelitype shows that Aurabesh is an available font resource.



```
public Window CreateWindow()
{
    // Create a window object and set its size to the
    // size of the display.
    mainWindow = new Window();
    mainWindow.Height = SystemMetrics.ScreenHeight;
    mainWindow.Width = SystemMetrics.ScreenWidth;

    // Create a single text control.
    Text text = new Text();

    text.Font = Resources.GetFont(Resources.FontResources.);
    text.TextContent = Resources.GetString(Resources.St          .String1);
    text.HorizontalAlignment = Microsoft.SPOT.Presentat          lAlignment.Center;
    text.VerticalAlignment = Microsoft.SPOT.Presentatio          gnment.Center;

    // Add the text control to the window.
```

Figure 5 - Intelitype shows Aruabesh as an available font resource.

9. Build the application and run it in the Microsoft Emulator. You can see “Hello World!” as it would appear in the Star Wars universe. If you already knew how this would appear in the Star Wars universe, you’ve been attending too many Star Wars conventions.

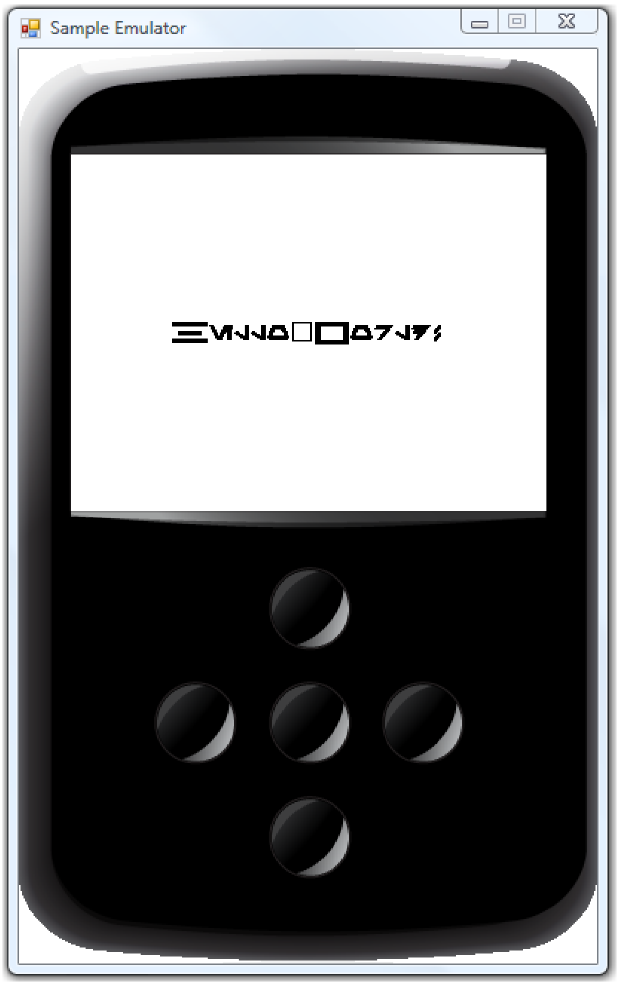


Figure 6 – "Hello World!" now displays using the Aurabesh font

If there are issues with the font, you would have to use the other conversion options.

You can see the size difference between the two font files:

- Aurabesh.TTF – 30KB
- Aurabesh.TinyFNT – 5.3KB

The smaller font is a better solution for .NET MF where memory is a bit constrained.

Windows is a registered trademark of Microsoft Corporation.