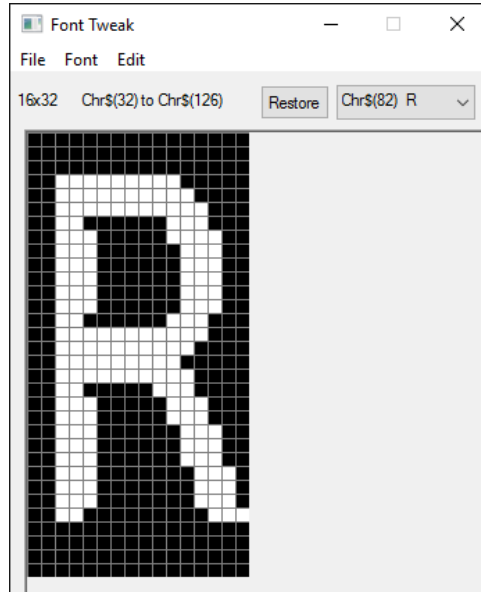# Font Tweak

A program for converting UTFT (and other) fonts to MMBasic BAS format.
By TassyJim on The Back Shed forum.


UTFT fonts are available from http://www.rinkydinkelectronics.com/r_fonts.php

A good source of bitmap fonts is "The Ultimate Oldschool PC Font Pack" from
https://int10h.org/oldschool-pc-fonts/

True Type fonts are readily available including from Google
https://www.google.com/get/noto/

A great font program is "Fony", available from http://hukka.ncn.fi/?fony

FontTweak needs no installation. Just run it from wherever you have permission.
There are a few support files and configuration files, so write permission is required.
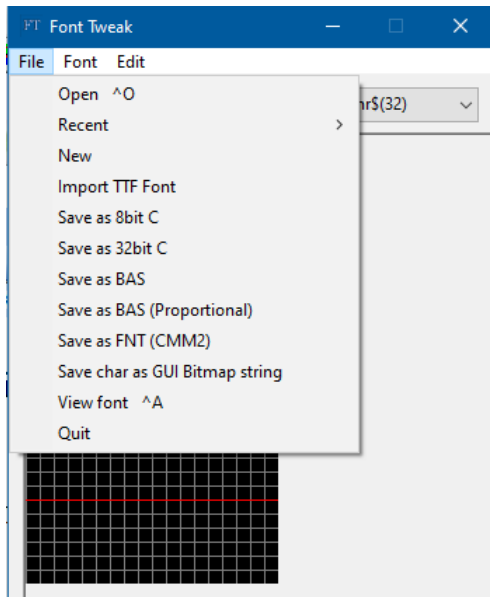
Open opens 'C', 'BAS', 'FNT' or 'BIN' files. Any file extension is OK but the format must comply with the C, BAS or BIN format.
FontTweak will determine the file format and does not rely on the file extension.

C format is the format used in MMBasic C source files.

```
fontdatatype Grotesk16x32[6084] PROGMEM={
0x10,0x20,0x20,0x5F,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,   // <space>
0x00,0x00,0x00,0x00,0x00,0x00,0x01,0xC0,0x01,0xC0,0x01,0xC0,0x01,0xC0
,0x01,0xC0,0x01,0xC0,0x01,0xC0,0x01,0xC0,0x01,0xC0,0x01,0xC0,0x01,0xC
0,0x01,0xC0,0x01,0xC0,0x01,0xC0,0x01,0xC0,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x01,0xC0,0x01,0xC0,0x01,0xC0,0x01,0
xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,   // !
```

It is very useful if you want to manually mix various fonts because it has each character on its own line.

The **BAS** format is used by MMBasic programs to load fonts. The code would normally be inserted into your program or library.

The **BAS (Proportional)** format has all the characters shifted to the left and a DATA section added to the file to allow for proportional font printing.

```
   ' font widths
FONT_WIDTH:
  DATA  32, 95 ' fontstart, fontcount
  DATA  1,2,5
  DATA  10,8,15,10,2,4,4,12
  DATA  8,2,4,2,8,8,8,8
  DATA  8,9,8,8,8,8,8,2
  DATA  2,8,8,8,8,10,11,8
  DATA  9,9,7,7,9,8,2,4
```

```
DATA  8,6,11,8,10,8,10,9
DATA  9,8,7,7,13,9,8,8
DATA  3,8,3,9,16,4,7,7
DATA  7,7,7,5,9,7,2,4
DATA  7,2,12,7,7,7,7,5
DATA  6,5,7,7,12,7,7,5
DATA  5,2,5,8
```

The **FNT** format is either the same as BAS or in the original Maximite FONT format. You can easily update your old Maximite fonts and characters to the newer font formats. If the Maximite font character W x H is not a multiple of 8, the new font will be padded out as required.

If there are multiple font definitions in any of the above files, only the first one is imported. To import others, put a 'REM' at the start of the first font definition.
" *' DefineFont #8 "* or
" *// fontdatatype Grotesk16x32[6084] PROGMEM={* "
will skip over those definitions.

The **BIN** format is the one used by Fony when 'Exporting' fonts.
You need to take note of the font dimensions when exporting from Fony.

New creates a new file with the character dimensions you enter. The width times the height must be a multiple of 8. The file starts at the ASCII value you entered and initially has one character.

Import TTF lets you use any installed Font or TTF and OTF font files without the need to install them.

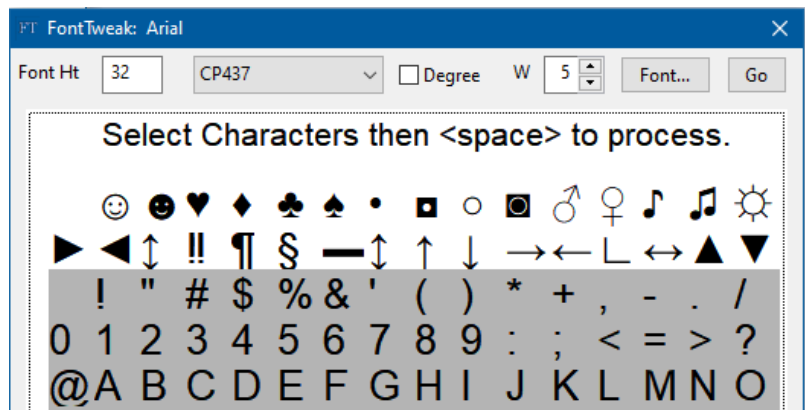**Save as.** You can save in either 'C', 'BAS' or 'FNT' format.
8bit C is the original micromite format while 32bit C is the format used internally in the CMM2.

This allows converting between formats. 'FNT' is the same as 'BAS' without the comments. This is suitable for the Colour Maximite 2. You cannot save in the original Maximite FNT format.

You can also save a character as a GUI BITMAP string.

View font opens a new window with the full font displayed. Clicking on a character will change the edit window to that character.

Import TTF opens a new window.

FT FontTweak: Arial   ✕

Font Ht   32   CP437 ⌄   ☐ Degree   W   5 ⬍   Font...   Go

Select Characters then <space> to process.

☺ ☻ ♥ ♦ ♣ ♠ • ◘ ○ ◙ ♂ ♀ ♪ ♫ ☼
► ◄ ↕ ‼ ¶ § ▬ ↕ ↑ ↓ → ← ∟ ↔ ▲ ▼
! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O

Font HT is the desired height of the resulting font in pixels.

The CP… dropdown list lets you choose from a selection of code pages. You can add more if required to gain access (together with appropriate fonts) to the full range of Unicode characters.

Ticking Degree will replace the back-tick with the degree symbol regardless of the font.

W is the weighting given to characters during the vector to raster conversion. The default of 5 is usually acceptable.

Font… will open up a font selector for installed fonts. You can also choose options such as Italics and Bold. The size is ignored as it is calculated to suit the character height chosen previously.

Holding <Shift> while selecting Font… will open a file selector for you to choose a font file. The font does not get installed into Windows and is only available during the current program.

Next, select the starting and ending characters desired then press Go or <Space>

The font is generated and the font view window opens.

You will probably want to Centre the font next.
Small fonts will need a LOT of work to tidy the up. Large fonts of 32 pixels high look good without any extra work.
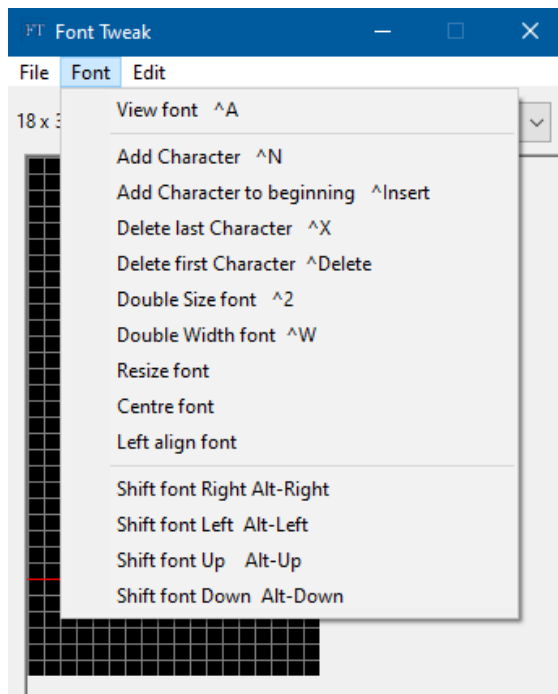
Item under the "Font" menu affect the full font file rather than the individual character.

View font opens a new window with the full font displayed. Clicking on a character will change the edit window to that character.

Add Character (ctrl-N) adds a new character to the end of the font file.

Add Character to beginning adds a new character to the start of the font file.

Delete Last Character (ctrl-X) or 'Delete' removes the last character in the file.

Delete First Character (ctrl-Delete) removes the first character in the file.

Double size (ctrl-2) increases the character width and height by 2 times. There is a limit to the maximum size the program permits.

Double width (ctrl-W) increases the character width by 2 times, keeping the height the same.

Resize font. Fonts imported from Fony often have a size unsuitable for MMBasic. Here you can resize them to something useable.

Centre font. After resizing, some imported fonts have the characters left justified. You can centre all characters horizontally in their box.

Left Align font will shift all characters to the left. Usefull if you want to reduce the width of a font.
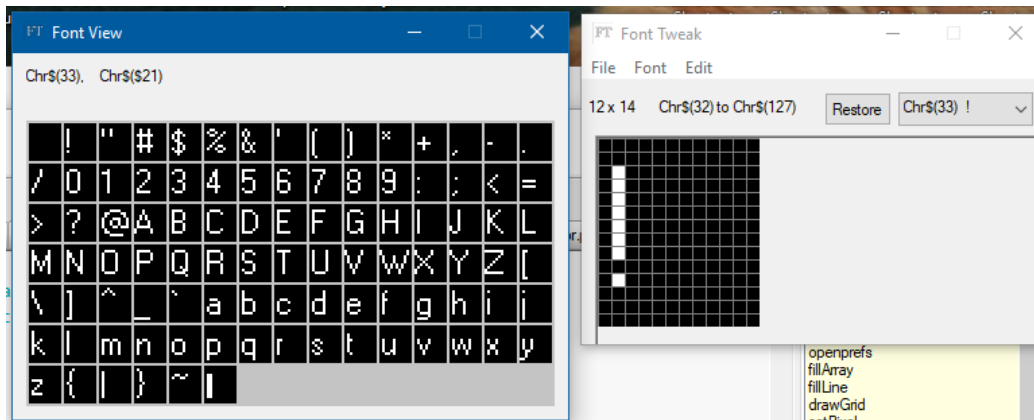
Alt-Right/Left/Up/Down (Right/Left/Up/Down) moves all characters one row/column at a time. Any pixels that go off the grid will be lost.

Example:

With the help of Fony, I exported one of the Ultimate old school fonts and opened it in Font Tweak: If you don't get the dimensions correct, it will be obvious when you view the full font.
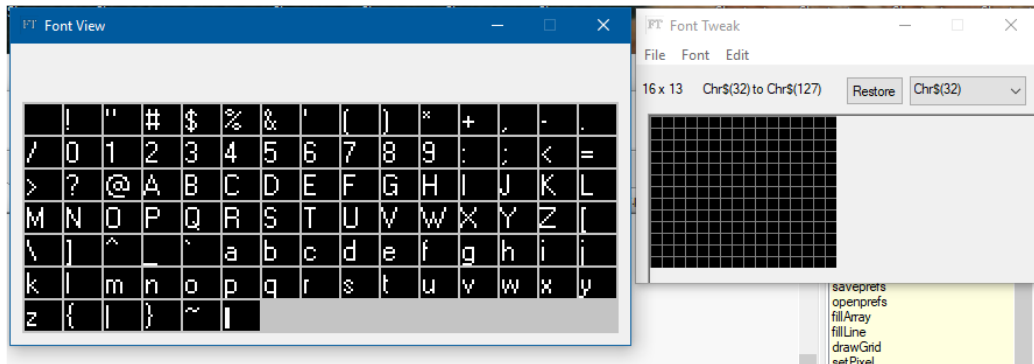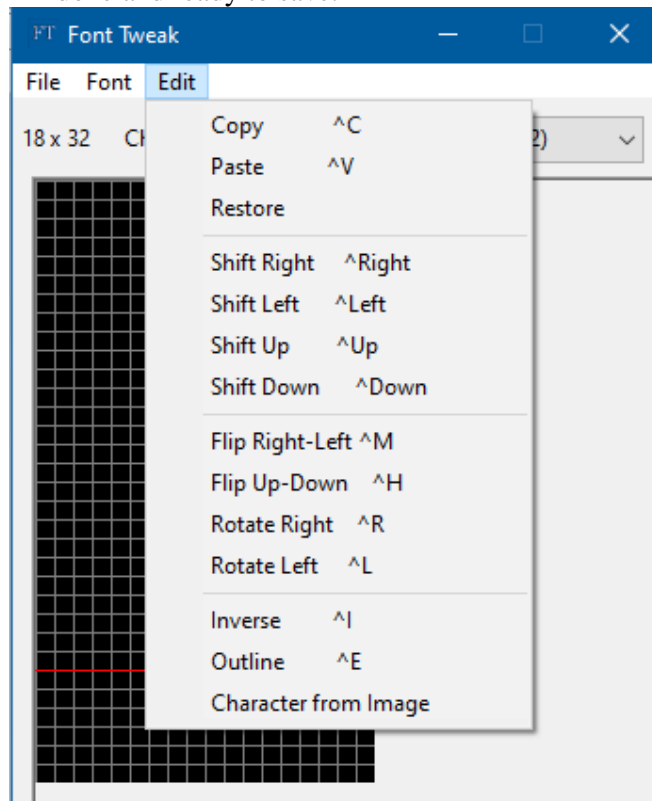

There is a lot of wasted space on the right of the font so I reduced it down to 12 pixels wide. To keep the multiple of 8 rule, I had to change from 13 to 14 pixels high.

Looking better but it needs to be centred:
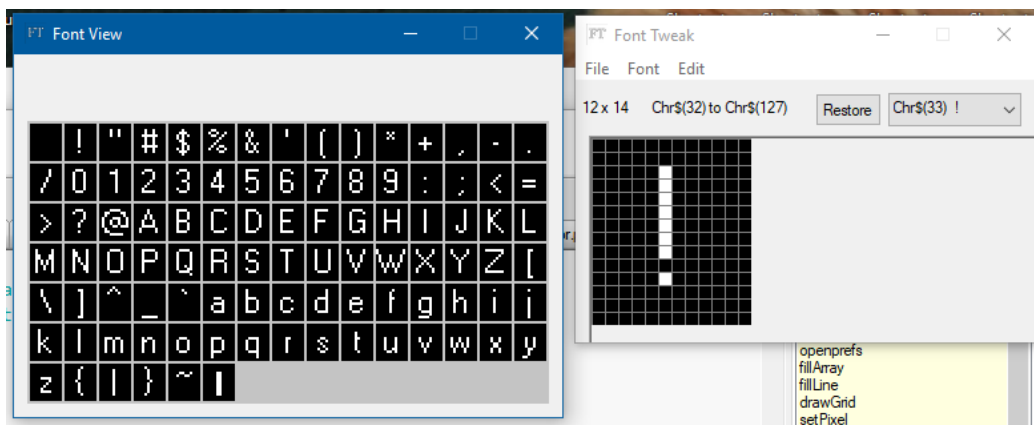
All done and ready to save.

Items under the "Edit" menu operate on individual characters.

Select the character to edit using the dropdown list or use right and left to step through the characters.

Copy and **Paste** allow you to copy a character to a different character position in the font.
Restore recovers the character as it was when selected.

Ctrl-Right/Left/Up/Down (Right/Left/Up/Down) moves the character one row/column at a time. Any pixels that go off the grid will be lost.

Flip Right-Left (ctrl-V) gives a mirror image, flipped about the vertical centreline.

Flip Up-down (ctrl-H) gives an inverted image, flipped about the horizontal centreline.
Rotate Right/Left (ctrl-R, ctrl-L) rotates the character. If the character size is not equal, the top left portion if the character is rotated and the remaining pixels cleared.

Inverse (ctrl-I) changes all pixels. Black to white and white to black.

Outline removes the centre of large blocky characters, creating an outline.
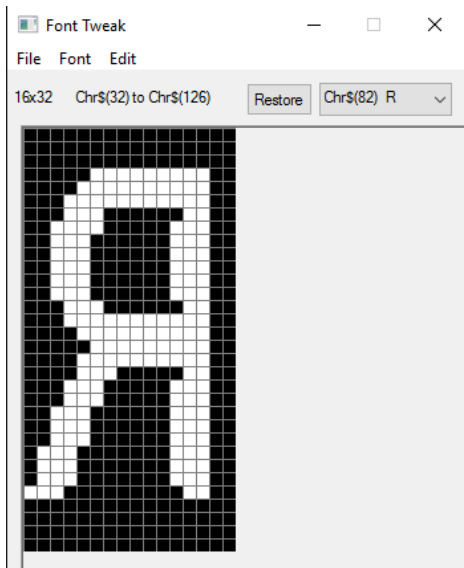
Ctrl_V or Flip Right-Left   Ctrl-H or Flip Up-Down
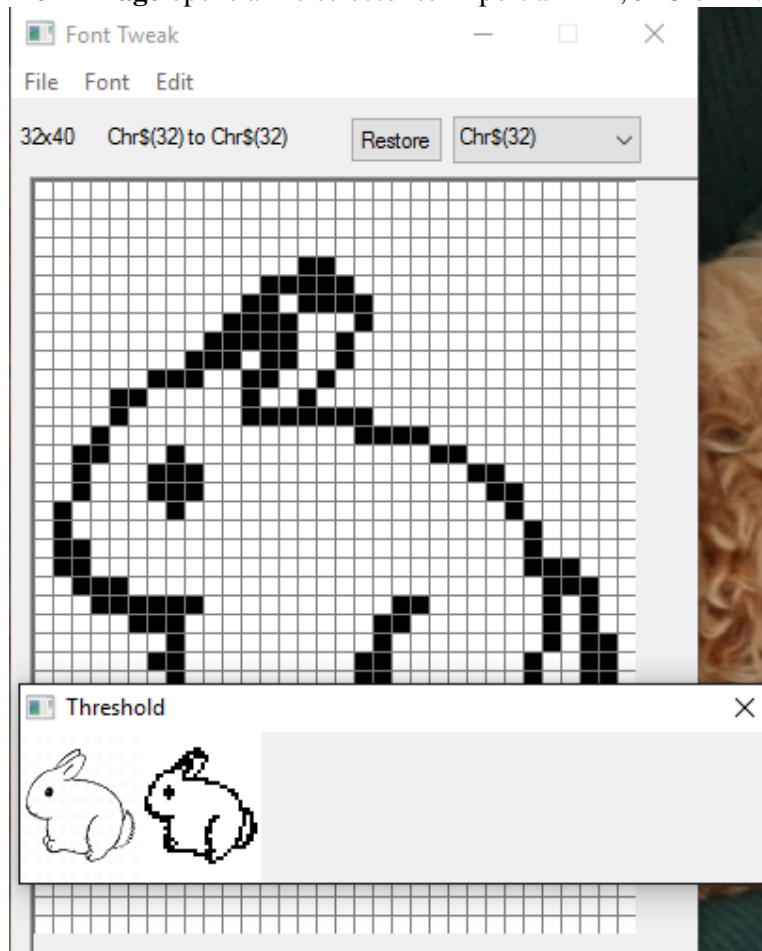
Character from Image opens a file selector to import a BMP, JPG or PNG image.

The size of the image is not important but its proportions are. If your font is 32 x 50, the image width to height should be in the same ratio. The image opens in a new window with the original one the left and a black and white version on the right.

There are two methods used for converting the image.

Threshold changes each pixel to white if it is brighter than the set value, black otherwise. This mode is best for line art or silhouettes.

Region threshold set the pixel white if it is brighter than the surrounding pixels. This mode is usually best for true photos but you may need to experiment.

To switch between the two modes, press the space bar.
To adjust the threshold level, use the right and left arrow keys
To adjust the size of the region (only in Region threshold mode), use the up and down arrow keys.
To transfer the image to the current font character, press return or enter.
Once transferred, some touch-up is usually needed.

```
' proportional fonts demo
' tassyjim April 2020
OPTION EXPLICIT

DIM INTEGER fontwidths(256) ' width of each character
DIM INTEGER txtW(80) ' with of each char in the string to be printed
DIM INTEGER fontstart, fontcount, n

' fill the array with character widths
RESTORE FONT_WIDTH
READ fontstart , fontcount
FOR n = fontstart TO fontstart + fontcount-1
  READ fontwidths(n)
NEXT n

CLS
kern_txt 10,100,"This is IT!!", "L",,RGB(GREEN)
kern_txt 10,120,"WWW is wide", "L",,RGB(GREEN)
kern_txt 10,140,"III is not", "L",,RGB(GREEN)

kern_txt 200,100,"This is IT!!", "C",,RGB(CYAN)
kern_txt 200,120,"WWW is wide", "C",,RGB(CYAN)
kern_txt 200,140,"III is not", "C",,RGB(CYAN)

kern_txt 390,100,"This is IT!!", "R",,RGB(GREEN)
kern_txt 390,120,"WWW is wide", "R",,RGB(GREEN)
kern_txt 390,140,"III is not", "R",,RGB(GREEN)

kern_txt 110,200,"Stay home", "j", 180,RGB(RED)
kern_txt 110,220,"Stay safe", "j", 180,RGB(RED)
kern_txt 110,240,"Save lots of lives!!", "j", 180,RGB(RED)

PRINT
PRINT

SUB kern_txt(x AS INTEGER,y AS INTEGER,txt$,just$,maxW AS INTEGER,c AS INTEGER,bc AS
INTEGER)
  ' maxW = maximum width of text in pixels to justify to.
  ' just$: C = centre, L = left, R = right, J = justify to width maxW
  ' background colour may extend to the right of the last character.
  LOCAL AS INTEGER txtwidth, n, k, px
  FOR n = 1 TO LEN(txt$) ' fill char width array and find total width
    txtW(n) = fontwidths(ASC(MID$(txt$,n,1)))+1
    txtwidth = txtwidth + txtW(n)
  NEXT n
  SELECT CASE just$
    CASE "C","c"
      px = x – txtwidth/2
    CASE "R", "r"
      px = x – txtwidth
    CASE "J", "j" ' justify
      px = x
      IF txtwidth < maxW THEN ' if text width is less than maxW add extra pixel spaces
        k = 0
        FOR n = 1 TO LEN(txt$)
          IF MID$(txt$,n,1)=" " THEN txtW(n)= txtW(n) + 1 ' increase space characters
first
          k = k + 1
          IF (k + txtwidth) >= maxW THEN EXIT FOR
        NEXT n
        DO
          FOR n = 1 TO LEN(txt$)
            txtW(n)= txtW(n) + 1 ' increase all characters
            k = k + 1
            IF (k + txtwidth) >= maxW THEN EXIT FOR
          NEXT n
        LOOP UNTIL (k + txtwidth) >= maxW
      ENDIF
    CASE ELSE ' "L", "l"
      px = x
  END SELECT
  FOR n = 1 TO LEN(txt$)
    TEXT px,y,MID$(txt$,n,1), L,8,1,c,bc
    px = px + txtW(n)
  NEXT n
END SUB
```

DEFINEFONT #8
```
5F201010 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00C000C0 00C000C0 00C000C0 00C000C0 000000C0 00C000C0
00000000 00000000 00D800D8 00D800D8 000000D8 00000000 00000000 00000000
00000000 00000000 80198019 C07F8019 0033C07F 80FF0033 006680FF 00660066
00000000 00000000 003E0008 0068006B 00780068 000F003E 00CB000B 003E006B
00000008 00000000 3066183C 60666066 803DC066 6603BC01 66066606 3C18660C
00000000 00000000 007E003C 00660066 C03C003C C0C6C06C 80E380C7 C03CC07F
00000000 00000000 00C000C0 00C000C0 000000C0 00000000 00000000 00000000
00000000 00000000 00600030 00C00060 00C000C0 00C000C0 00600060 00000030
00000000 00000000 006000C0 00300060 00300030 00300030 00600060 000000C0
00000000 00000000 00060006 70E60006 801FF0FF 8019000F 8010C039 00000000
00000000 00000000 00000000 00180018 00180018 00FF00FF 00180018 00180018
00000000 00000000 00000000 00000000 00000000 00000000 00C000C0 00800040
00000000 00000000 00000000 00000000 00F00000 000000F0 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00C00000 000000C0
00000000 00010000 00020002 00040004 00080008 00100010 00200020 00400040
00800080 00000000 007E003C 00C30066 00C300C3 00C300C3 006600C3 003C007E
00000000 00000000 00380018 00D80078 00180098 00180018 00180018 00FF00FF
00000000 00000000 007E003C 00C300E7 00070003 001C000E 00700038 00FF00FF
00000000 00000000 007E003C 00C300E3 001E0003 0003001E 00E300C3 003C007E
00000000 00000000 000E000E 0036001E 00660036 80FF00C6 000680FF 00060006
00000000 00000000 00FE00FE 00C000C0 00FE00DC 000300E7 00C70003 003C00FE
00000000 00000000 007F003E 00C00063 00FE00DC 00C300E7 006700C3 003C007E
00000000 00000000 00FF00FF 00060006 0018000C 00180018 00300030 00300030
00000000 00000000 007E003C 00C300C3 007E00C3 00C3007E 00C300C3 003C007E
00000000 00000000 007E003C 00C300E6 00E700C3 003B007F 00C60003 0038007E
00000000 00000000 00000000 00C000C0 00000000 00000000 00C000C0 00000000
00000000 00000000 00000000 00C000C0 00000000 00000000 00C000C0 00800040
00000000 00000000 00000000 00070001 0078001E 007800E0 0007001E 00000001
00000000 00000000 00000000 00FF0000 000000FF 00FF00FF 00000000 00000000
00000000 00000000 00000000 00E00080 001E0078 001E0007 00E00078 00000080
00000000 00000000 007E003C 00C300C7 00060003 0018000C 00000018 00180018
00000000 00000000 001F0000 C0408021 40A2409E 80A640A2 00400099 001F8020
00000000 00000000 000E000E 001B001F 803B001B 803F8031 C071C071 E0E0C060
00000000 00000000 00FE00FC 00C300C3 00FE00C3 00C300FE 00C300C3 00FC00FE
00000000 00000000 007F001E 80E18061 00C000C0 00C000C0 806180E1 001E007F
00000000 00000000 00FE00FC 80C300C7 80C180C1 80C180C1 00C780C3 00FC00FE
00000000 00000000 00FE00FE 00C000C0 00FC00C0 00C000FC 00C000C0 00FE00FE
00000000 00000000 00FE00FE 00C000C0 00FC00C0 00C000FC 00C000C0 00C000C0
00000000 00000000 007F001E 80C18063 00C000C0 80C780C7 806380C1 801C807F
00000000 00000000 00C300C3 00C300C3 00FF00C3 00C300FF 00C300C3 00C300C3
00000000 00000000 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0
00000000 00000000 00300030 00300030 00300030 00300030 00300030 00E000F0
00000000 00000000 00C600C3 00CC00CC 00D800D8 00EC00FC 00C600C6 00C300C3
00000000 00000000 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0 00FC00FC
00000000 00000000 E0E0E0E0 E0F1E0F1 60DB60D1 60CA60DB 60CE60CE 60C460C4
00000000 00000000 00E300C3 00F300E3 00DB00F3 00CF00DB 00C700CF 00C300C7
00000000 00000000 003F001E C0C08061 C0C0C0C0 C0C0C0C0 8061C0C0 001E003F
00000000 00000000 00FE00FC 00C300C3 00FE00C3 00C000FC 00C000C0 00C000C0
00000000 00000000 807F001E C0C08061 C0C0C0C0 C0C0C0C0 8061C0C0 001E007F
C001C003 00000000 00FF00FE 80C180C1 00FF80C1 00C600FE 00C300C6 80C100C3
00000000 00000000 003F001E 00608071 003E0078 8003000F 8E380C1 003E007F
00000000 00000000 00FF00FF 00180018 00180018 00180018 00180018 00180018
00000000 00000000 00C600C6 00C600C6 00C600C6 00C600C6 00C600C6 0038007C
00000000 00000000 00C600C6 006C00C6 006C006C 006C006C 00380038 00380038
00000000 00000000 18C718C7 18C718C7 B06DB06D B06DB06D E038E038 E038E038
00000000 00000000 006380C1 00360063 001C001C 0036001C 00630036 80C10063
00000000 00000000 00C300C3 00660066 003C003C 00180018 00180018 00180018
00000000 00000000 00FF00FF 00060006 0018000C 00300018 00600060 00FF00FF
00000000 00E00000 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0
000000E0 00800000 00400080 00200040 00100020 00080010 00040008 00020004
00010002 00E00000 00600060 00600060 00600060 00600060 00600060 00600060
000000E0 001C0000 0036001C 00630036 000080C1 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 FFFF0000
0000FFFF 00000000 00900060 00900090 00000060 00000000 00000000 00000000
00000000 00000000 00000000 00FE007C 001E00C6 00C6007E 00FE00C6 00000076
00000000 00C00000 00C000C0 00FC00D8 00C600C6 00C600C6 00FC00C6 000000B8
00000000 00000000 00000000 007E003C 00C000E6 00C000C0 007E00E6 0000003C
00000000 00060000 00060006 007E0036 00C600C6 00C600C6 007E00C6 00000036
00000000 00000000 00000000 007C0038 00FE00C6 00C000FE 007C00C6 00000038
00000000 00380000 00600078 00F800F8 00600060 00600060 00600060 00000060
00000000 00000000 003E8003 0063007F 007F0063 0060003E 807F007F 00FF80C1
0000007E 00C00000 00C000C0 00FE00DC 00C600E6 00C600C6 00C600C6 000000C6
00000000 00000000 00C000C0 00C00000 00C000C0 00C000C0 00C000C0 000000C0
00000000 00000000 00300030 00300000 00300030 00300030 00300030 00300030
00E00030 00C00000 00C000C0 00CC00C6 00D800CC 00F800F8 00CC00CC 000000C6
```

```
      00000000 00C00000 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0 000000C0
      00000000 00000000 00000000 F0FFE0DC 30C630E7 30C630C6 30C630C6 000030C6
      00000000 00000000 00000000 00FE00DC 00C600E6 00C600C6 00C600C6 000000C6
      00000000 00000000 00000000 007C0038 00C600C6 00C600C6 007C00C6 00000038
      00000000 00000000 00000000 00FC00D8 00C600C6 00C600C6 00FC00C6 00C000D8
      00C000C0 00000000 00000000 007E003A 00C600C6 00C600C6 007E00C6 00060036
      00060006 00000000 00000000 00F800D8 00C000E0 00C000C0 00C000C0 000000C0
      00000000 00000000 00000000 00FC0078 00E000CC 001C0078 00FC00CC 00000078
      00000000 00000000 00600060 00F800F8 00600060 00600060 00780060 00000038
      00000000 00000000 00000000 00C600C6 00C600C6 00C600C6 00FE00CE 00000076
      00000000 00000000 00000000 00C600C6 006C006C 0038006C 00380038 00000010
      00000000 00000000 00000000 30C630C6 606F6066 6069606F C039C039 0000C030
      00000000 00000000 00000000 006C00C6 0038006C 00380038 006C006C 000000C6
      00000000 00000000 00000000 00C600C6 006C006C 006C006C 00380038 00300038
      00600070 00000000 00000000 00F800F8 00300018 00600020 00F800C0 000000F8
      00000000 00380000 00600060 00600060 00800060 00600060 00600060 00600060
      00000038 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0 00C000C0
      000000C0 00E00000 00300030 00300030 00080030 00300030 00300030 00300030
      000000E0 00000000 00FF0073 000000DE 00000000 00000000 00000000 00000000
      00000000 00000000
END DEFINEFONT
  '
  ' font widths
FONT_WIDTH:
  DATA  32, 95 ' fontstart, fontcount
  DATA  1,2,5
  DATA  10,8,15,10,2,4,4,12
  DATA  8,2,4,2,8,8,8,8
  DATA  8,9,8,8,8,8,8,2
  DATA  2,8,8,8,8,10,11,8
  DATA  9,9,7,7,9,8,2,4
  DATA  8,6,11,8,10,8,10,9
  DATA  9,8,7,7,13,9,8,8
  DATA  3,8,3,9,16,4,7,7
  DATA  7,7,7,5,9,7,2,4
  DATA  7,2,12,7,7,7,7,5
  DATA  6,5,7,7,12,7,7,5
  DATA  5,2,5,8
```

Creating new Code Pages.

CodePages define the characters available in the 256 character table.
Originally, they referred to bitmaps and recent files are used to map chr(x) to it's
Unicode equivalent.
Not all font families will have the Unicode character defined so that character will
usually print as an empty square.
FontTweak uses codepage 437 as a default.
Any replacement codepage only needs to specify the Unicode character for characters
that are different to CP437

The file name must start with "CP" and have a ".inf" extension. It must reside in the
same folder as the FontTweak.exe

The file 'CP737.txt' is an example format with the full 256 characters for Codepage
427.
The other CP*.ing files are examples of valid format for some other codepages. These
alternate pages only have the changed characters.