How to use font-based icons in isCOBOL GUI programs

Author: Veryant Support

When your GUI application is going to need a large variety of different bitmaps for any kind of GUI control that accepts a bitmap for its icon, you can use font-based bitmaps instead of trying to get different images for each control. Bitmap fonts provide a comprehensive set of glyphs, or icons.
In this article we will use one of the most popular bitmap fonts: **FontAwesome** . You can download it from: www.fontawesome.com .
However, any bitmap font can be used (Material Design Font, dafont.com, etc)
The .otf file that you download can be located in any folder that you choose. You create a named font with W$CREATEFONT that points to the path (full or relative) of the font.

```
move "Font Awesome 5 Free Solid" to font-name
call "w$createfont" using "files/Font Awesome 5 Free-Solid-900.otf"
                          font-name
```

Then you load a font with W$FONT based on that named font.

```
initialize wfont-data
set wfdevice-console to true
move font-name       to wfont-name
move 10              to wfont-size
call "w$font" using wfont-get-font
                    h-font
                    wfont-data
```

Now, in order to use some of the icons from that font for your program, you need to create a strip of selected icons.
The FontAwesome font uses a hexadecimal code of 4 characters to refer to their icons.
Here';s a link to all of the available icons and their hex codes: https://fontawesome.com/v5/cheatsheet
Each of those hex codes need to be converted to a decimal value and then combined in a national variable as in this example:

```
77  character-1-hex         pic x(4 ).
77  character-1-n           pic n(1 ).
77  character-1-red         pic x(2) comp-x
                                     redefines character-1-n.
...

77  icon-characters         pic n any length .

...

  move "f1c3" to character-1-hex
  move "f576" to character-2-hex
  move "f008" to character-3-hex

  move function hex2dec(character-1-hex) to character-1-red
```

```
move function hex2dec(character-2-hex) to character-2-red
move function hex2dec(character-3-hex) to character-3-red

initialize icon-characters
string character-1-n delimited by space
       character-2-n delimited by space
       character-3-n delimited by space
        into icon-characters.
```

Once you have the list of decimal values in a national string, you can create the bitmap strip in memory, using the W$BITMAP routine:

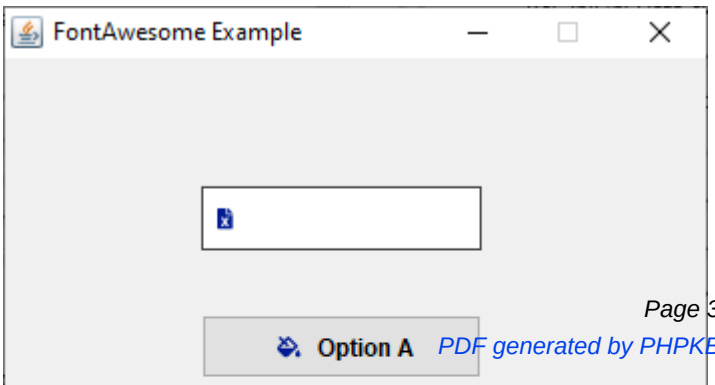
```
call "w$bitmap" using wbitmap-load-symbol-font,
                      h-font
                      icon-characters
                      20
                      icon-color
               giving h-font-icon
```

Now, since the h-font-icon handle of font contains the strip of icons 20 pixels each, you can assign icons from it to a GUI control on the screen section:

```
03 ef-1
   entry-field
   line                    5
   col                     15
   size                    20 cells
   lines                   2 cells
   bitmap-handle           h-font-icon
   bitmap-width            20
   bitmap-number           1
   .

03 pb-1
   push-button
   line                    9
   col                     15
   lines                   2 cells
   size                    20 cells
   bitmap-handle           h-font-icon
   bitmap-width            20
   title-position          2
   title                   "Option A"
   bitmap-number           2
   .
```

This code creates part of a simple example screen that looks like this:


FontAwesome Example
Option A

Attached is a zip file with a full sample of using the FontAwesome bitmap font. After downloading it, you can compile:

```
iscc  fontawesome.cbl
```

and run:

```
iscrun FONTAWESOME
```