



4D SYSTEMS

TURNING TECHNOLOGY INTO ART

Application Note: 4D-AN-P3001

ViSi - Displaying Third-Party Fonts

Document Date: 15th March 2013

Document Revision: 1.0

Description

This Application Note explains how custom fonts can be used on a PICASO module in ViSi Environment:

Here is the list of items required to replicate this application,

- 4D Workshop 4 IDE
- A PICASO based Display Module
- 2GB uSD Card
- 4D Programming Cable

Application Overview

There are 3 built in fonts in the PICASO processor, those are.

- **Font1** (5x7)
- **Font2** (8x8)
- **Font3** (8x12)

User might need more stylish and larger size fonts which is a need addressed in this application.

User can import ANSI or UNICODE fonts.

A customer might need the external fonts to be displayed in two ways,

- You may wish to display some text set in the Design time.
- You may wish to display the test in run time.

Both of these are achievable.

When you set the text in design time, it becomes an image that can be displayed whenever it's needed. Whereas in run time the text could be displayed using `putch()`, `putstr()` or `putnum()` command.

NOTE: The `file_Dir()` command is the only other command that writes the list of directory directly to the screen. This command is also affected by the Fonts change.

Setup Procedure

Foremost, the 4D Workshop 4 IDE has to be downloaded and installed. This is available from the 4D Systems website through the following link:

<http://www.4dsystems.com.au/prod.php?id=172>

Documentation regarding Workshop 4 and its environments, such as ViSi, can also be downloaded from this site.

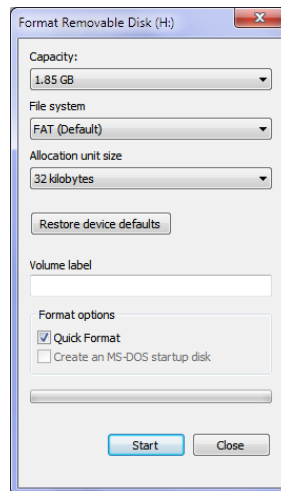
These are the documents that you will need to refer to, to replicate this application.

[Workshop 4 - PICASO 4DGL Internal Functions Reference Manual](#)

[Workshop 4 - 4DGL Programmers Reference Manual](#)

[Workshop 4 - ViSi User Guide](#)

FAT (aka FAT16) format the uSD card using Windows formatting tool,



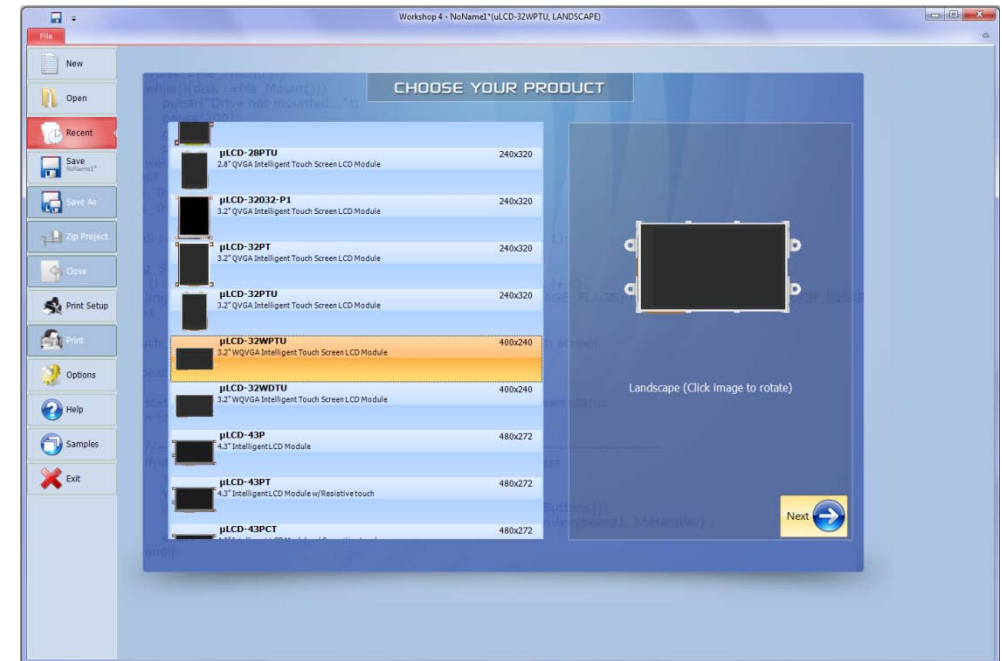
Adding Fonts Procedure

Customizing Fonts using the ViSi environment

Open the Workshop 4 (WS4) IDE and click "Create a new project".



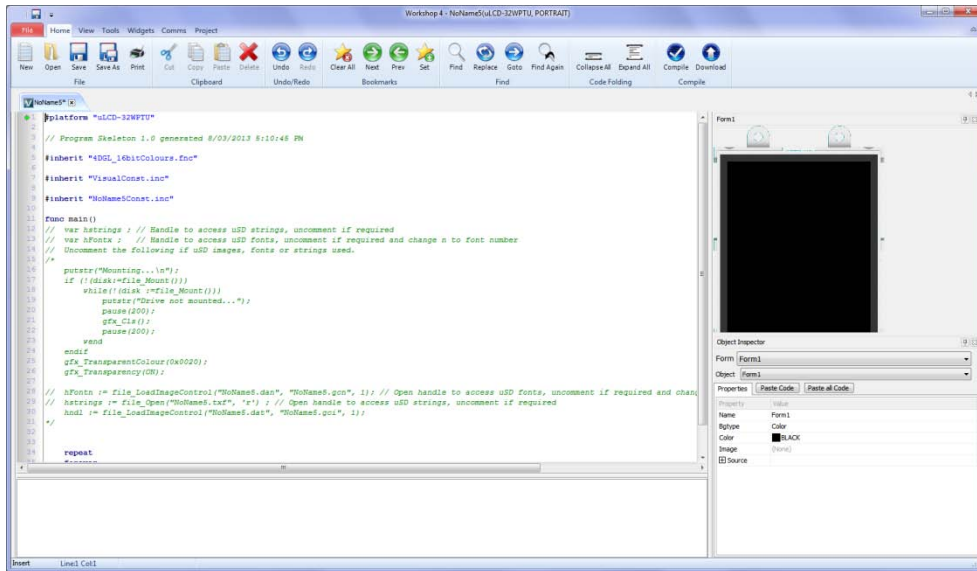
Choose the Display module you want to use. uLCD-32WPTU will be used for this application. Click the image to change the orientation to Portrait and press **Next**.



Select the **ViSi** environment



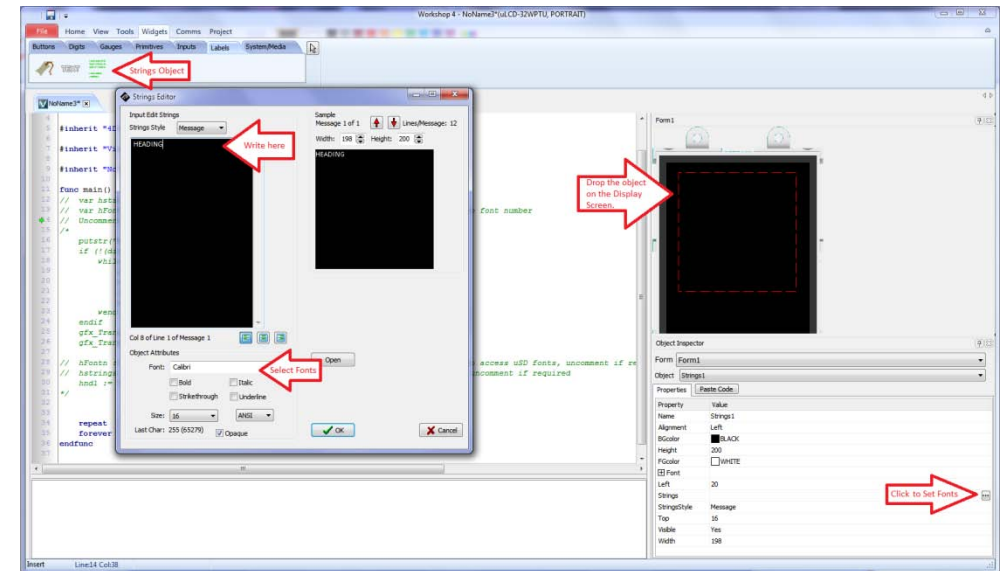
This will open the ViSi development environment window within the WS4 IDE as shown below.



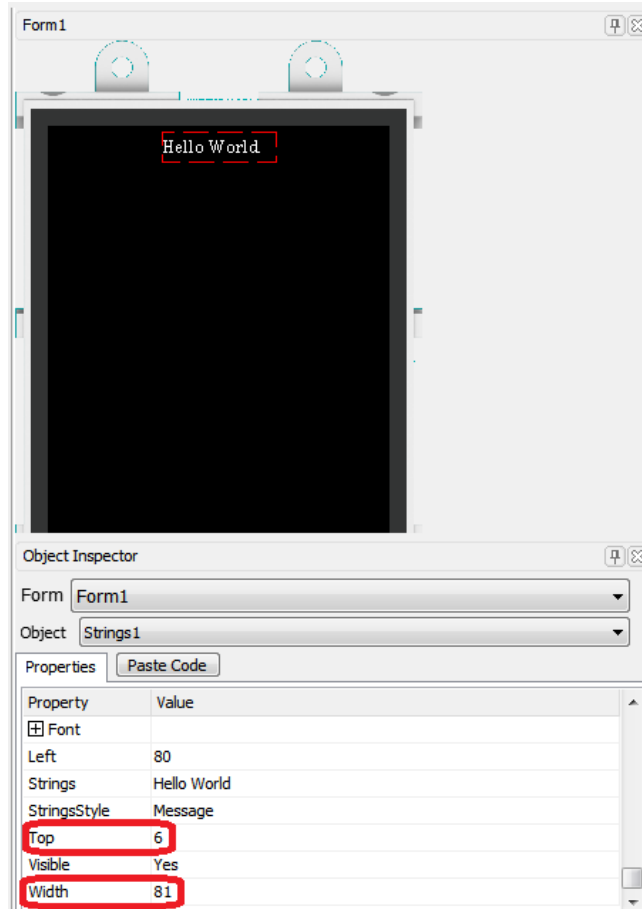
The new project starts as 'noname' project. Save the project as, say AddFonts.

- Go to **Widgets**, select **Strings** object under the **Labels** tab.
- Click on to the screen to drop the Strings object.
- Click **Strings** property in the Properties section at the bottom right to setup the Fonts.
- Write something on the left window.
- Select the Fonts and adjust other properties as required.
- For this application, we have only adjusted the following properties,
 - Written "Hello World"
 - Set Fonts to 'Cambria'
 - Font size is set to 16

Press OK.

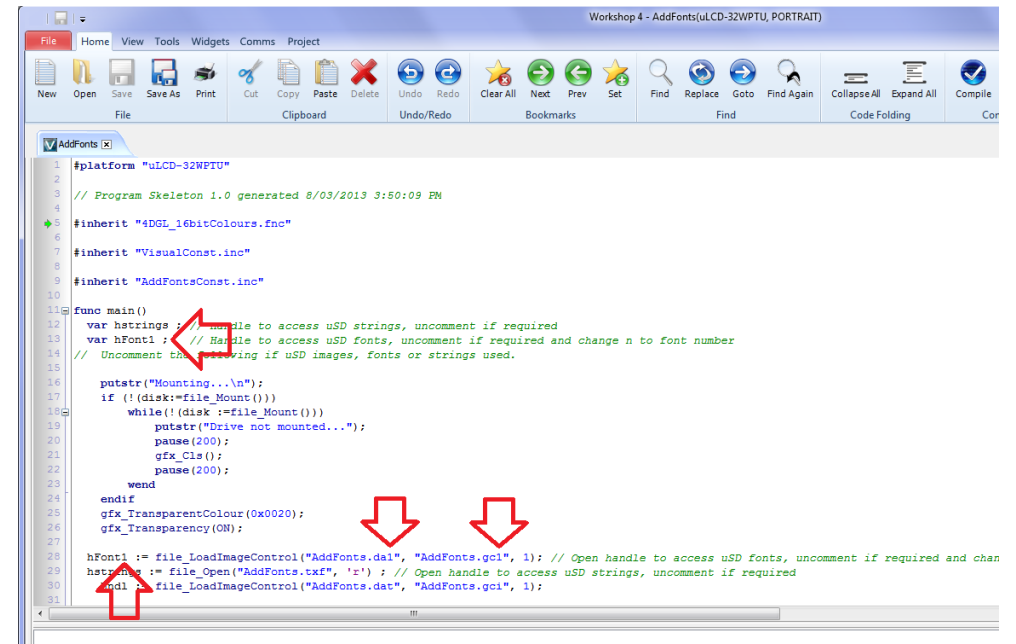


- Adjust the size of the object to fit the text.
- Adjust it in the center.
- The exact location could also be set using the object properties.

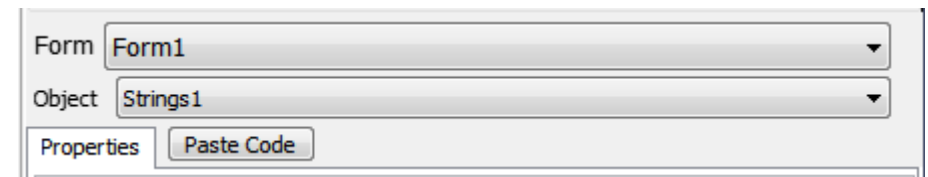


Simulation Procedure

Uncomment the sections as shown in the image below and change the 'n' or 'x' from the variable definition to a distinct number (start with 1) at the code sections pointed in the image below,



Put the cursor on, say line 34 and click Paste Code,



You will see the following code added in the code section.

```

34
35 // Strings1 1.0 generated 8/03/2013 5:54:50 PM
36 txt_FontID(hFont1) ; // Font index correct at time of code generation
37 txt_FGcolour(WHITE) ;
38 txt_BGcolour(BLACK) ;
39 gfx_MoveTo(89 , 4) ;
40 PrintDisk(hstrings, Strings1StartH, Strings1StartL, Strings1Size, i) ; // where i is Message 0 - StringsCount-1
41

```

Change i in the following command line to '0' for the message in the first Row of the String.

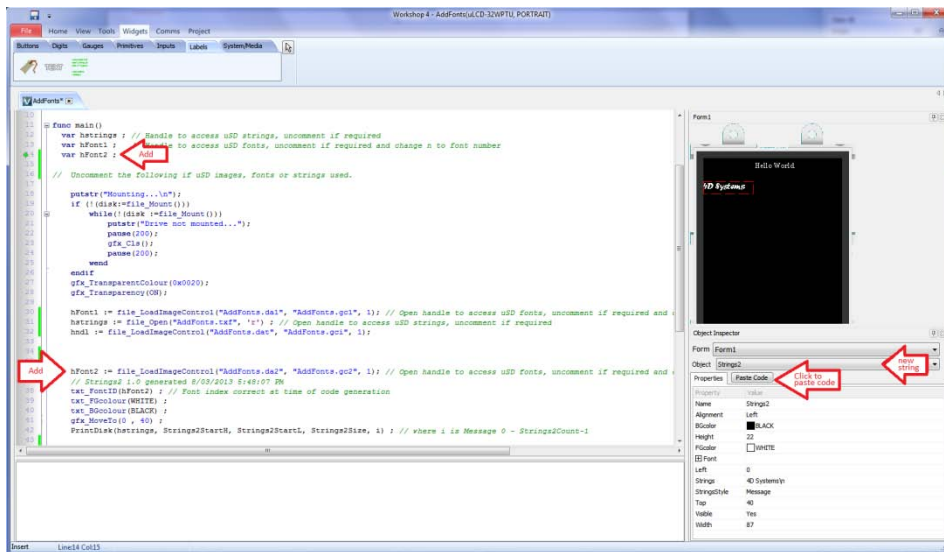
PrintDisk(hstrings, Strings1StartH, Strings1StartL, Strings1Size, 0) ;

Add another Strings object

Drag and drop a new strings object on the screen.

Adjust the properties and select the fonts.

Move the cursor to the blank area, say line 43 and click paste code for the new string object.



Now, add the hFont2 variable definition and add the line as shown in the image above,

hFont2 := file_LoadImageControl("AddFonts.da2", "AddFonts.gc2", 1);

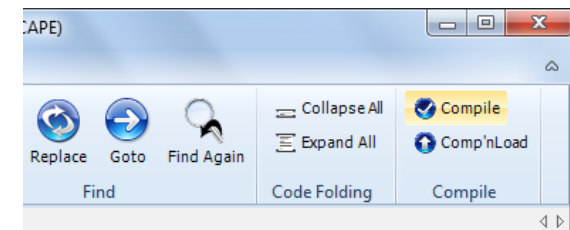
Change i in the following command line to '0' for the message in the first Row of the String.

PrintDisk(hstrings, Strings2StartH, Strings2StartL, Strings2Size, 0) ;

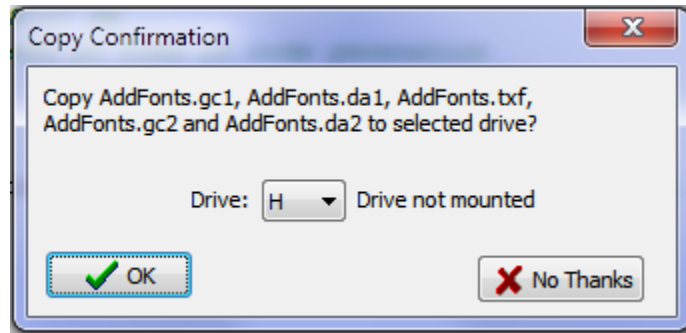
// where i is Message 0 - Strings2Count-1

Completing the ViSi project

- Now, insert the uSD card in the uSD card reader on the PC/Notebook.
- Make sure,
 - The module is connected to the PC/Notebook through 4D Programming Cable
 - uSD card is FAT (aka FAT16) formatted.
 - Correct Com port is selected.
- On the **ViSi** window **Home** ribbon in WS4 click Comp'nLoad.



After clicking on **Comp'nLoad**, you will be prompted to select and confirm the drive to where the font files will be saved. Click **OK** to confirm and start copying.



Press OK to start writing the Font files to the uSD card. Once that is finished, the IDE will continue and load the program to the module. After program is loading is finished, you will see a message on the Display Module's screen,

Mounting...

Or

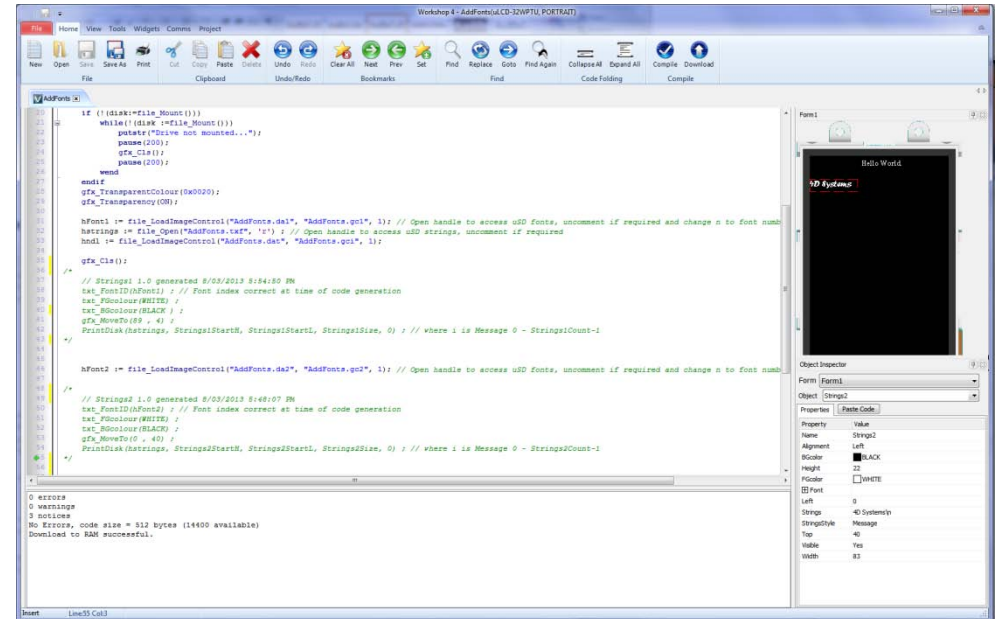
Drive not mounted...

Insert the uSD card in to the Display Module and your program would execute.

Note: Up till now, we have only displayed the text preset in the design time.

Printing text in custom fonts

Now, the font data has been added to the uSD card, you may not wish to use the preset text. Comment out the section that prints the preset text on the screen, as shown in the image below.



You may wish to display text in custom fonts in run time.

You can use simple commands such as,

// Print text in Custom Fonts

txt_FontID(hFont1);

print("\nI am Font1\n");

txt_FontID(hFont2);

print("\nI am Font2\n");

```
putstr("Using Putstr to print Font2\n");  
putnum(HEX, 34);  
print("\n");
```

```
txt_FontID(hFont1);  
file_Dir("*."); //Print the files on screen in Font1
```

Here is what you should see on the screen, complete set of code is available with the application note.



Tips

- The changes you make on the object properties after pasting the code do not reflect on the code. I.e. if you wish to edit the object properties you need to Paste Code after doing so.
- There is a complete STRINGSDEMO.4DViSi example in the 4D Workshop4 IDE. Click 'Samples', select Picaso ViSi – Click for filtered browse, look for,
..\4D Labs\Picaso Visi

Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems.

All trademarks belong to their respective owners and are recognised and acknowledged.

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.