APPLICATION NOTES

4D SYSTEMS
TURNING TECHNOLOGY INTO ART

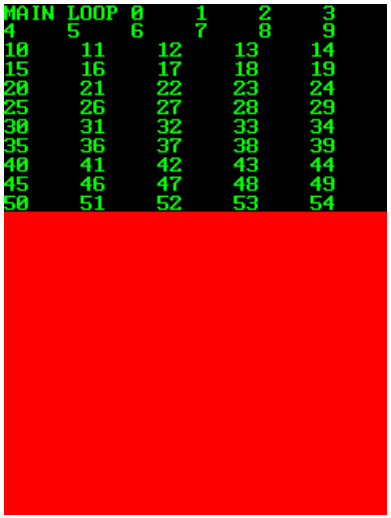
ViSi-Genie Magic Code Insertion Points

DOCUMENT DATE: **9th MAY 2020**
DOCUMENT REVISION: **1.1**

Description

This application note primarily discusses Magic Code insert points. Magic Code can be added at the following points: Constant/Global/Data definition, Main loop, Pre/Post Activate form and Pre/Post Genie Initialization.

The Magic Code is under the Genie Magic pane in Workshop4Pro



This application note requires:

- Any of the following 4D Picaso and gen4 Picaso display modules:

gen4-uLCD-24PT	gen4-uLCD-28PT	gen4-uLCD-32PT
uLCD-24PTU	uLCD-32PTU	uVGA-III

and other superseded modules which support the ViSi Genie environment

- The target module can also be a Diablo16 display

gen4-uLCD-24D series	gen4-uLCD-28D series	gen4-uLCD-32D series
gen4-uLCD-35D series	gen4-uLCD-43D series	gen4-uLCD-50D series
gen4-uLCD-70D series		
uLCD-35DT	uLCD-43D series	uLCD-70DT

Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor. The display module used in this application note is the uLCD-32PTU, which is a Picaso display. This application note is applicable to Diablo16 display modules as well.

- 4D Programming Cable / uUSB-PA5/uUSB-PA5-II
  for non-gen4 displays(uLCD-xxx)
- 4D Programming Cable & gen4-PA, / gen4-IB / 4D-UPA
  for gen4 displays (gen4-uLCD-xxx)
- micro-SD (µSD) memory card
- Workshop 4 IDE (installed according to the installation document)
- Any Arduino board with a UART serial port

- 4D Arduino Adaptor Shield (optional) or connecting wires
- Arduino IDE
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Application Overview

The application presented in this document shows Magic Code object inserted at different points of the program. The ViSi-Genie How to Add Magic Object application note discussed some details on Magic Code but not in full detail.

Setup Procedure

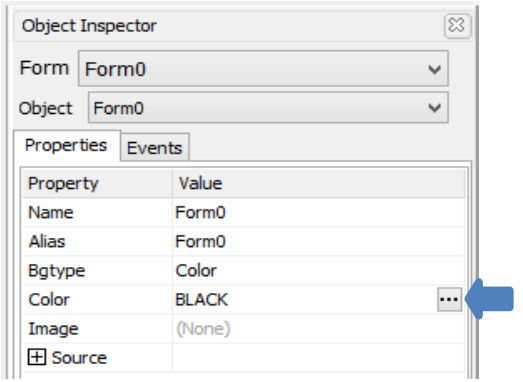
For instructions on how to launch Workshop 4, how to open a ViSi-Genie project, and how to change the target display, kindly refer to the section "**Setup Procedure**" of the application note:

ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)
or
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

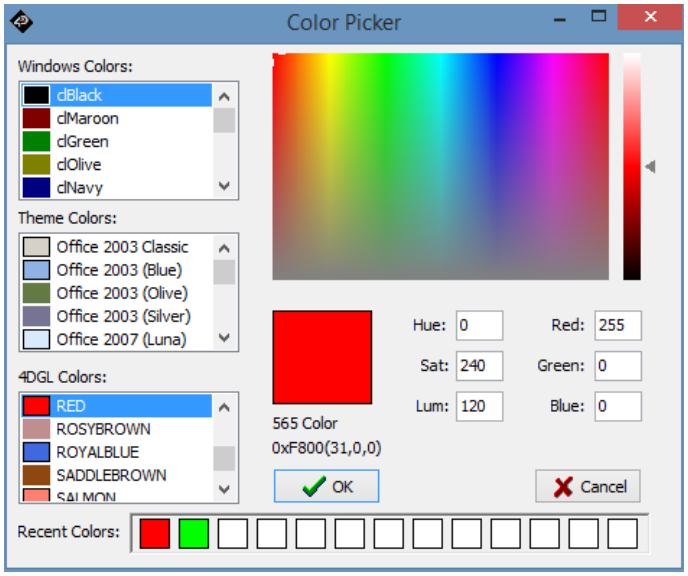
Design the Project

Change Background Color of Form0

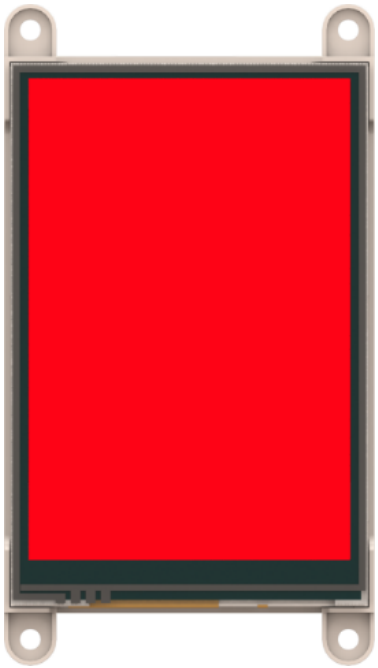
Change in the background color of form0 is shown below



The 'Bgtype' property of form0 should be set to 'Color'. In the 'Color' property click the [...] icon then the 'Color Picker' window will pop up.



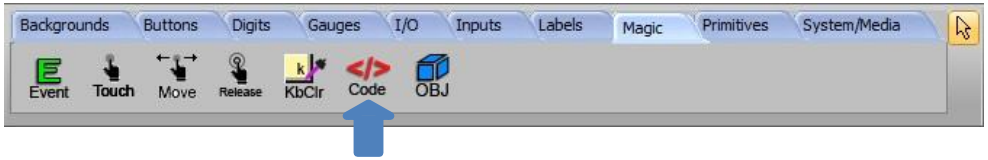
The image above shows the Color Picker window. The application in this document used the color 'RED'. It is necessary to change the form's color because it will indicate if the form is activated or not. This will be explained further.

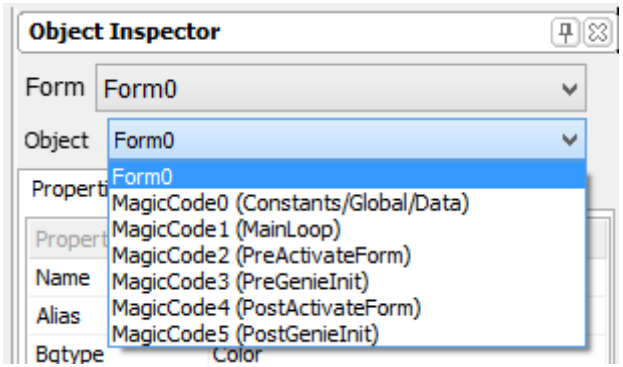


The image above shows the background color of form0 changed to red.

Add Six Magic Codes

Six Magic Codes are added to Form0. This is **MagicCode0, MagicCode1, MagicCode2, MagicCode3, MagicCode4, MagicCode5**.



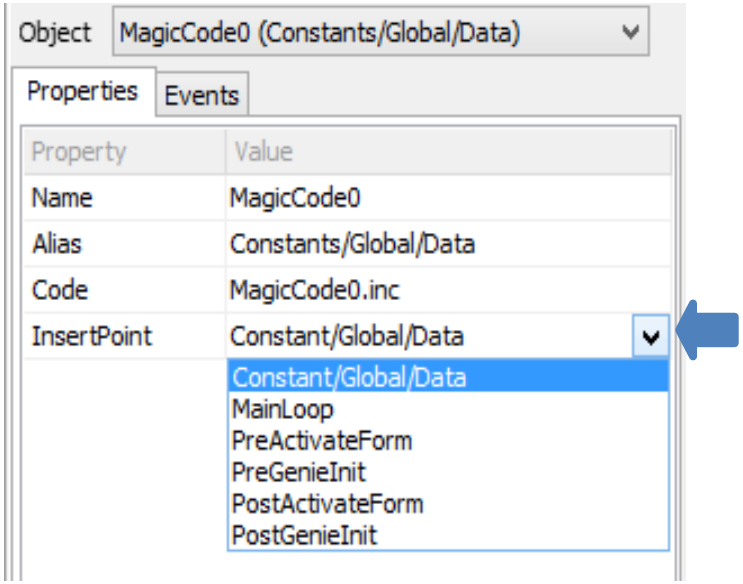


To know more about adding Magic Events and Magic Code refer to the application note, [ViSi-Genie How to Add Magic Objects]

Insertion Points

Change Insert point

Each Magic Code in this application has a different insertion point. Insertion points of Magic Code can be change by clicking in the 'InsertPoint' dropdown button as shown below



Code Insertion Points Representation

```
(CONSTANTS)
Constants/Global/Data
(Functions)

func ActivateForm()
        Pre Activate Form
        (Deactivate Old Form)
        (Display New Form)
        (Enable Inputs)
        Post Activate Form
endfunc

func main()
        Pre Genie Init
        (Mount SD)
        (Open Image Control)
        (Init 'constants')
        (Init Comms)
        (tag real objects)
        (Display Initial Form)
        Post Genie Init
        repeat
                Main Loop
        forever
endfunc
```

In page 65 of the ViSi-Genie Reference Manual the code insertion points is represented in a skeletal program. This should give idea where the magic code is placed inside the program.

MagicCode0(Constants/Global/Data)

The content of MagicCode0 is shown below:

```
// MagicCode0
//
var x;
x := 0;
```

A variable 'x' is declared and initialized with a value of 0. This variable will be used again in the main loop. This is to show that any functions can use this variable because the insertion point of MagicCode 0 is at 'Constants/Global/Data'.

MagicCode1(Main Loop)

The content of MagicCode1 is shown below:

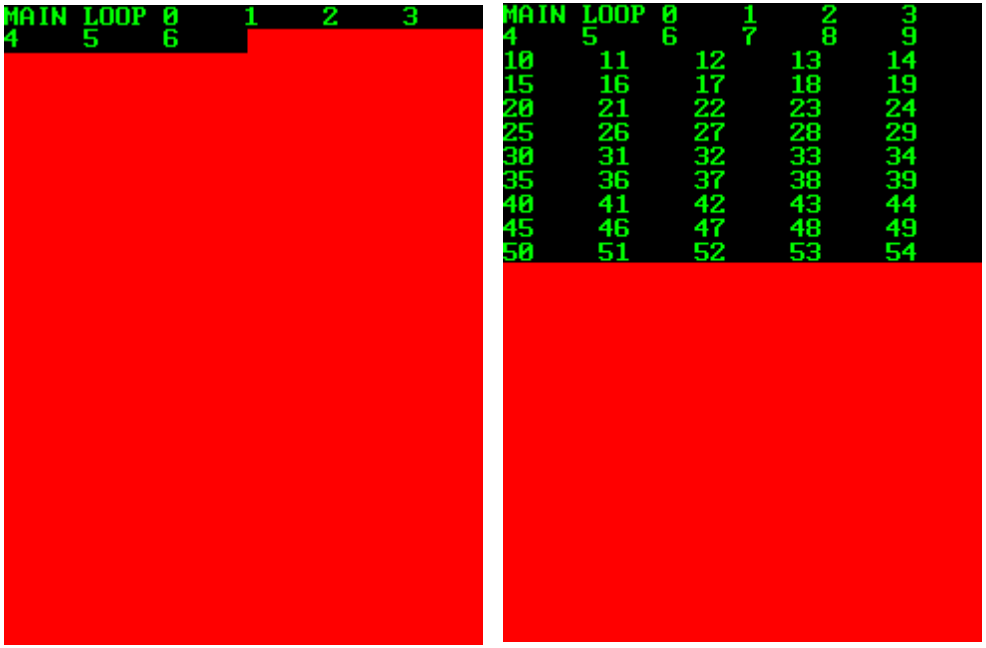
```
// MagicCode1
//

print("MAIN LOOP ");
for(x:=0;x<100;x++)
    print(x,"    ");
    pause(200);
next

gfx_Cls();
gfx_MoveTo(0,0);
```

The insertion point of MagicCode 1 is at 'Main Loop'. The code prints "MAIN LOOP" to tell that MagicCode1 is currently running. I then prints the variable 'x' declared in MagicCode0. It also increments x from 0 to 100 and then clears the screen after to start over.

Sample Output:



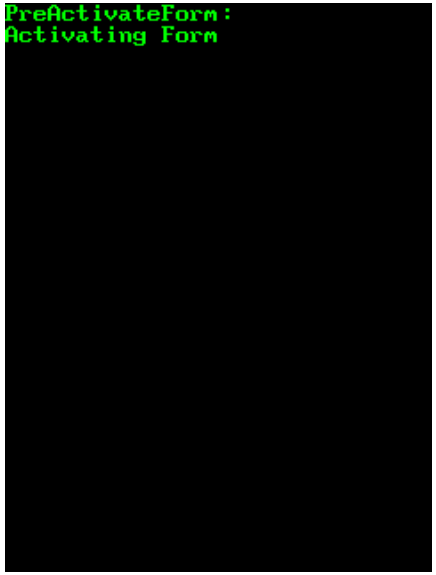
MagicCode2(Pre Activate Form)

The content of MagicCode2 is shown below:

```
// MagicCode2
//
gfx_Cls();
print("PreActivateForm: \nActivating Form\n");
pause(3000);
```

The code simply prints that it is currently in Pre Activate form. As seen in the diagram this code runs before the form is activated

Sample Output:



MagicCode3(Pre Genie Initialization)

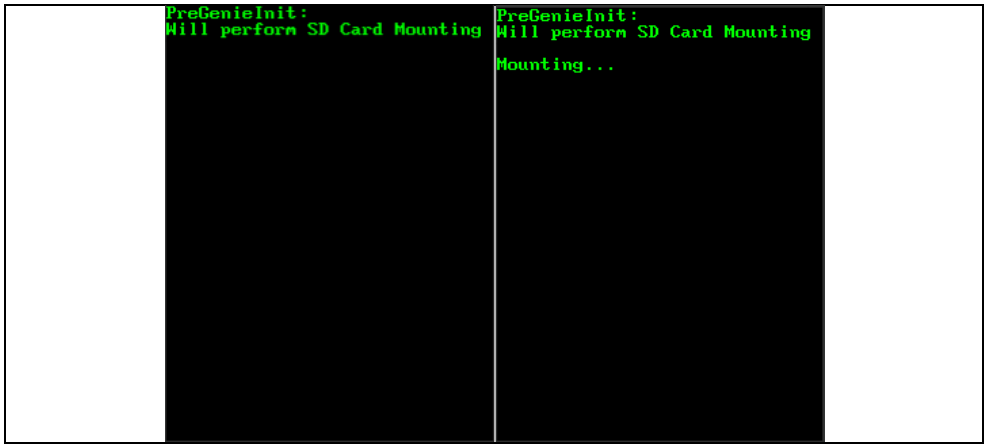
The content of MagicCode3 is shown below:

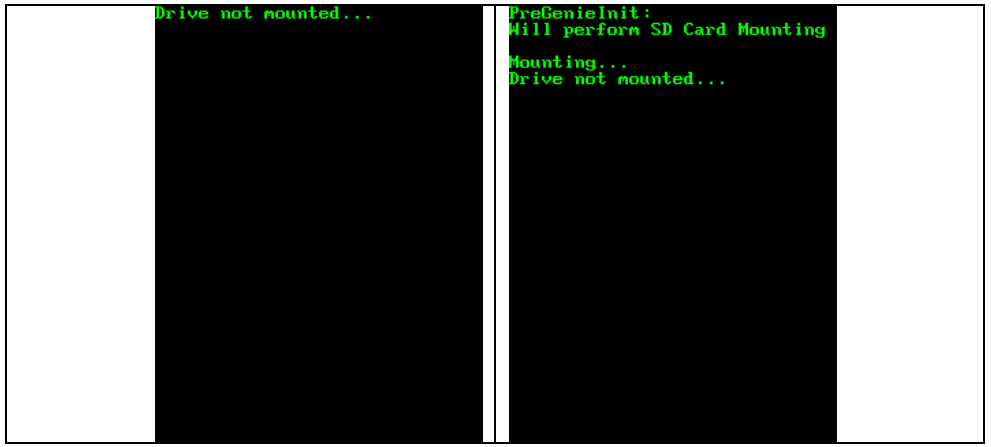
```
// MagicCode3
//

print("PreGenieInit: \nWill perform SD Card Mounting \n");
pause(3000);
```

The code simply prints that it is currently in Pre Genie Init. As seen in the diagram this code runs before the mounting of uSD card.

Output 1:





As seen in the image above, the display prints first the "PreGenie:Init: Will perform SD Card Mounting" and then after 3000 ms, the display will print "Mounting... Drive not mounted...". If an SD card is still not inserted in the display then it will continue to print "Drive not mounted..."

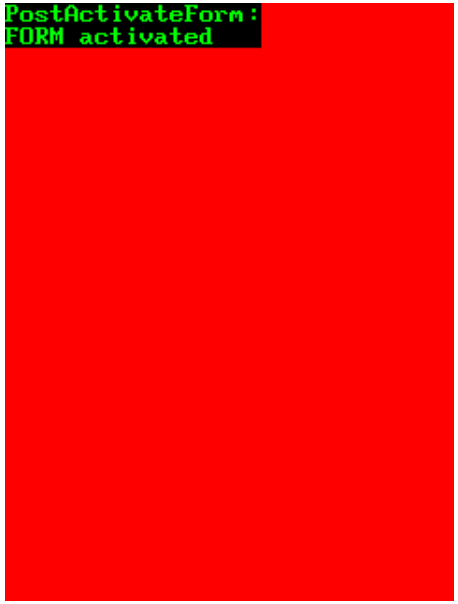
MagicCode4(Post Activate Form)

The content of MagicCode4 is shown below:

```
// MagicCode4
//
gfx_Cls();
print("PostActivateForm: \nFORM activated \n");
pause(3000);
```

The code simply prints that it is currently in Post Activate Form. As seen in the diagram this code runs after the form is activated.

The output below shows that the form is activated. Form0's background color was changed to verify Post Activate Form insert point.



Build and Upload the Project

For instructions on how to build and upload a ViSi-Genie project to the target display, please refer to the section "**Build and Upload the Project**" of the application note

ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)
Or
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

Proprietary Information

Disclaimer of Warranties & Limitation of Liability