



# ViSi-Genie Magic Code Insertion Points

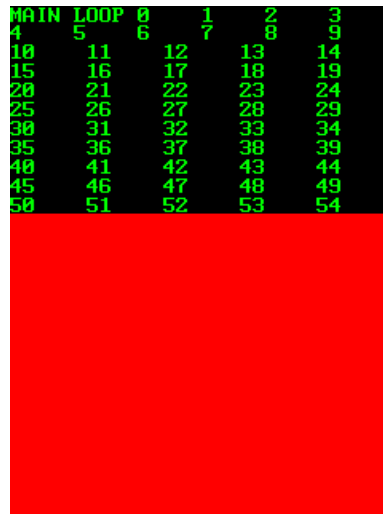
DOCUMENT DATE: **9<sup>th</sup> MAY 2020**  
DOCUMENT REVISION: **1.1**



## 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and other superseded modules which support the ViSi Genie environment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](http://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

<b>Description</b> .....	<b>2</b>
<b>Content</b> .....	<b>3</b>
<b>Application Overview</b> .....	<b>4</b>
<b>Setup Procedure</b> .....	<b>4</b>
<b>Design the Project</b> .....	<b>4</b>
<i>Change Background Color of Form0</i> .....	<b>4</b>
<i>Add Six Magic Codes</i> .....	<b>6</b>
<b>Insertion Points</b> .....	<b>6</b>
<i>Change Insert point</i> .....	<b>6</b>
<i>Code Insertion Points Representation</i> .....	<b>7</b>
<i>MagicCode0(Constants/Global/Data)</i> .....	<b>7</b>
<i>MagicCode1(Main Loop)</i> .....	<b>7</b>
<i>MagicCode2(Pre Activate Form)</i> .....	<b>8</b>
<i>MagicCode3(Pre Genie Initialization)</i> .....	<b>9</b>
<i>MagicCode4(Post Activate Form)</i> .....	<b>9</b>
<b>Build and Upload the Project</b> .....	<b>10</b>
<b>Proprietary Information</b> .....	<b>11</b>
<b>Disclaimer of Warranties &amp; Limitation of Liability</b> .....	<b>11</b>

## 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)

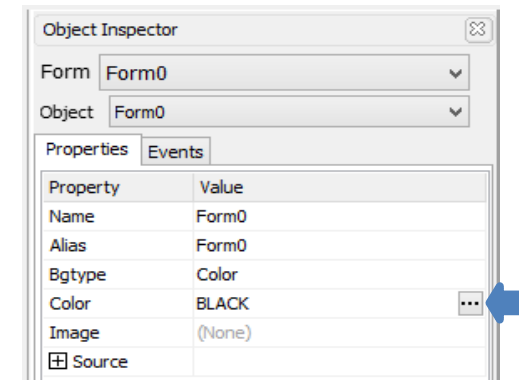
or

[ViSi Genie Getting Started – First Project for Diablo16 Displays](#) (for Diablo16).

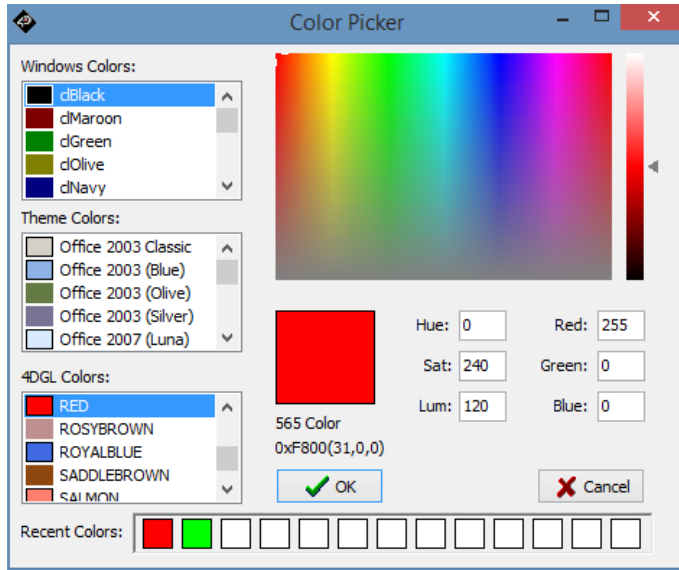
## Design the Project

### Change Background Color of Form0

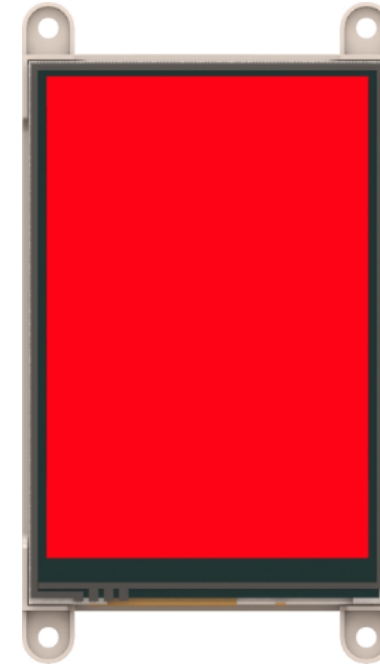
Change in the background color of form0 is shown below



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.



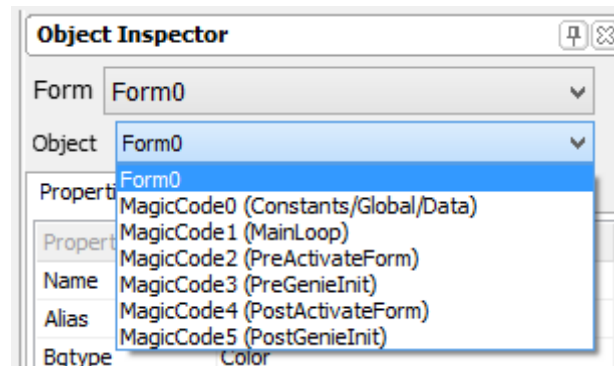
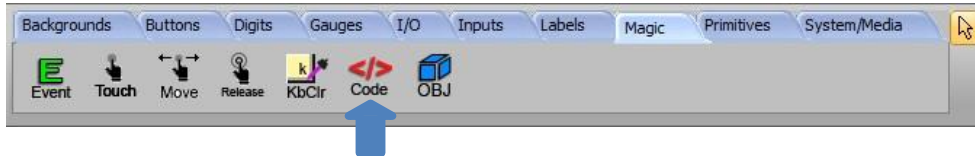
The image above shows the Color Picker window. The application in this document used the color 'RED'. It is necessary to change the form's color because it will indicate if the form is activated or not. This will be explained further.



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**.

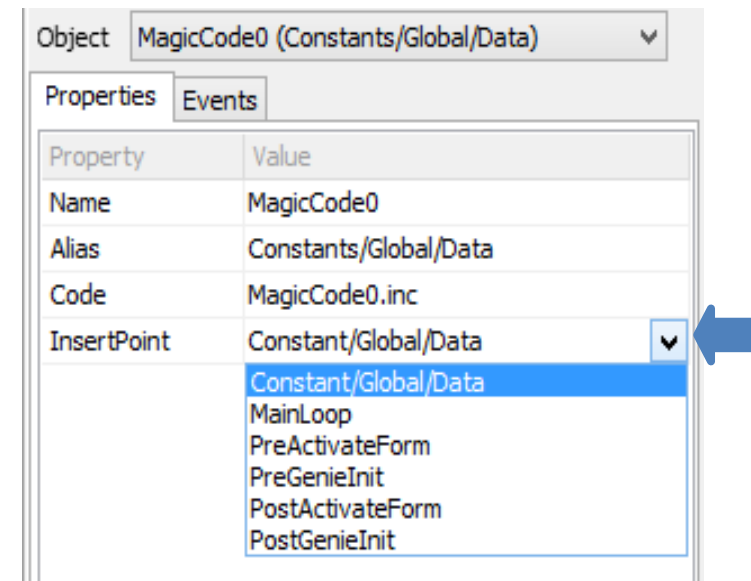


To know more about adding Magic Events and Magic Code refer to the application note, [ViSi-Genie How to Add Magic Objects](#)

## 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:

```
MAIN LOOP 0 1 2 3
4 5 6
10 11 12 13 14
15 16 17 18 19
20 21 22 23 24
25 26 27 28 29
30 31 32 33 34
35 36 37 38 39
40 41 42 43 44
45 46 47 48 49
50 51 52 53 54
```

### 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:

```
PreActivateForm:
Activating Form
```



### MagicCode3(Pre Genie Initialization)

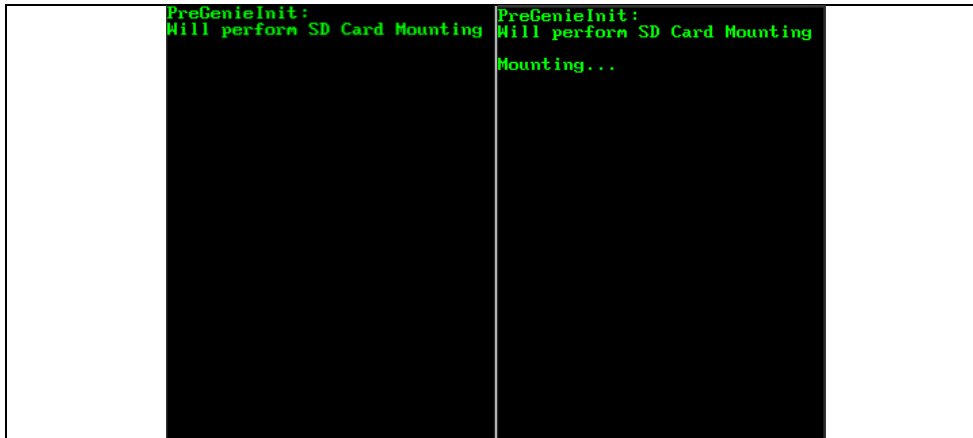
The content of MagicCode3 is shown below:

```
// 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:



```
PreGenieInit:
Will perform SD Card Mounting

PreGenieInit:
Will perform SD Card Mounting
Mounting...
```



```
Drive not mounted...

PreGenieInit:
Will perform SD Card Mounting
Mounting...
```

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)

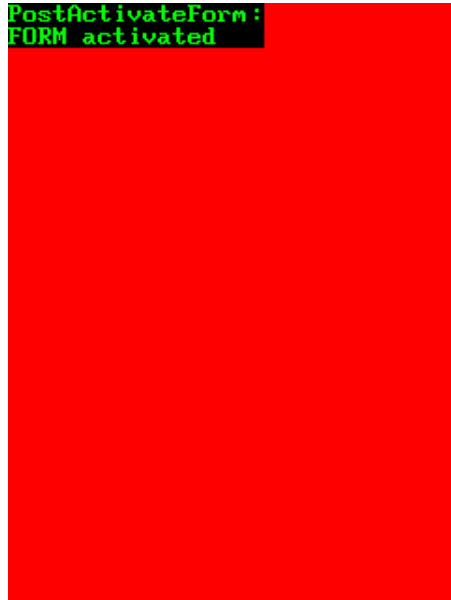
The content of MagicCode4 is shown below:

```
// 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

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.