



ViSi Genie Magic Slide to Unlock

DOCUMENT DATE: **22nd April 2019**
DOCUMENT REVISION: **1.1**



Description

This application note shows how to use the **Magic Event** object. The Magic Event object is under the Genie Magic pane in Workshop 4 Pro. It contains a 4DGL code and it can be linked to standard Genie objects such as a winbutton, such that any time that the button is touched, the Magic Event object is called (or the 4DGL code is executed). Below are screenshot images of the two forms of the project for this document.



Note 1: The ViSi-Genie project for this application is found in Worskhop. Go to the File menu -> Samples -> ViSi Genie Magic (Picaso/Diablo16) -> SlideToUnlock.4DGenie.

Note 2: Worskhop Pro is needed for this application.

Before getting started, the following are required:

- Any of the following 4D Picaso display modules:

[uLCD-24PTU](#)
[gen4-uLCD-24PT](#)

[uLCD-28PTU](#)
[gen4-uLCD-28PT](#)

[uVGA-III](#)
[gen4-uLCD-32PT](#)

and other superseded modules which support the ViSi Genie environment.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-35D](#)
[Series](#)
[gen4-uLCD-70D](#)
[Series](#)
[uLCD-35DT](#)

[gen4-uLCD-28D](#)
[Series](#)
[gen4-uLCD-43D](#)
[Series](#)
[uLCD-43D Series](#)

[gen4-uLCD-32D](#)
[Series](#)
[gen4-uLCD-50D](#)
[Series](#)
[uLCD-70DT](#)

Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor.

- [4D Programming Cable](#) / [µUSB-PA5/µUSB-PA5-II](#) for non-gen4 displays (uLCD-xxx)
- [4D Programming Cable](#) & [gen4-IB](#) / [gen4-PA](#) / [4D-UPA](#), for gen-4 displays (gen4-uLCD-xxx)

- [micro-SD \(μSD\)](#) memory card
- [Workshop 4 IDE](#) (installed according to the installation document)
- 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

Description	2
Content	3
Application Overview	4
Setup Procedure	4
Create a New Project	5
<i>Create a New Project</i>	5
Design the Project	5
<i>Add a Slider Object to Form0</i>	5
Select the Slider Object Icon	5
Place the Slider Object on the WYSIWYG Screen	5
Change the Orientation of Slider0 to Horizontal	6
<i>Add a Label to Form0</i>	6
Select the Static Text Icon	6
Place the Static Text Object on the WYSIWYG Screen	6
Change the Caption for Statictext0	7
Change the Font for Statictext0	7
<i>Add a New Form – Form1</i>	8
<i>Add a Winbutton Object to Form1</i>	9
Select the Winbutton Icon	9
Place the Winbutton Object on the WYSIWYG Screen	9
Change the Caption for Winbutton0	10

Configure the OnChanged Event Property of Winbutton0	10
<i>Add a Magic Event Object to Form0</i>	11
Navigate Back to Form0	11
Select the Magic Event Icon	11
Link MagicEvent0 to Slider0	12
Write the Code for MagicEvent0	12
Check for the Frame Value of Slider0	13
Activate Form1	14
Write a Value to Slider0	14
Build and Upload the Project	14
Proprietary Information	15
Disclaimer of Warranties & Limitation of Liability	15

Application Overview

When designing GUIs (Graphic User Interfaces), it is often desirable to have a slide-to-unlock feature. This feature is commonly found in smart phones. In the standard version of Workshop 4, a stand-alone application having a slide-to-unlock feature is not possible. A host would be needed to decide when the “unlock” action is triggered, which will occur only when the slider knob is moved to the right. The new objects under the Genie Magic pane in Workshop 4 give the user more control of the program. In this application note for example, the user will learn how to create a stand-alone ViSi-Genie application with a slide-to-unlock feature in Workshop Pro.

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).

Create a New Project

Create a New Project

For instructions on how to create a new ViSi-Genie project, please refer to the section “**Create a New Project**” of the application note

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

or

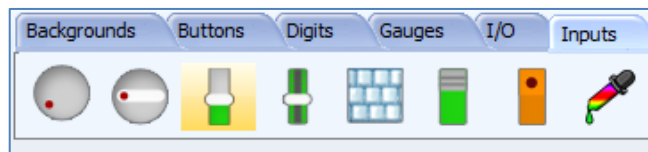
[ViSi Genie Getting Started – First Project for Diablo16 Displays](#) (for Diablo16)

Design the Project

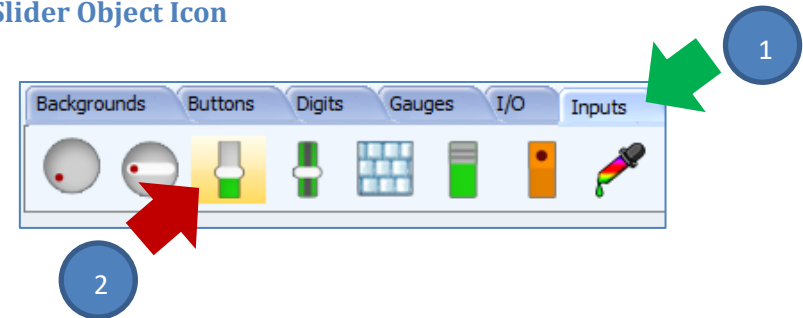
A uLCD-32PTU (portrait orientation) will be used for this application note.

Add a Slider Object to Form0

The slider object is found under the Inputs pane.

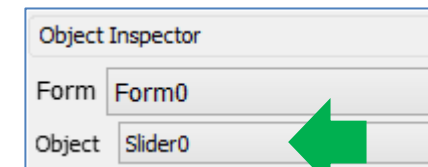
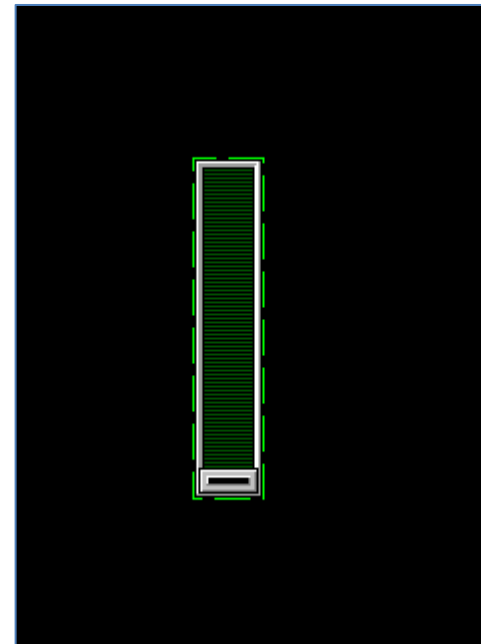


Select the Slider Object Icon



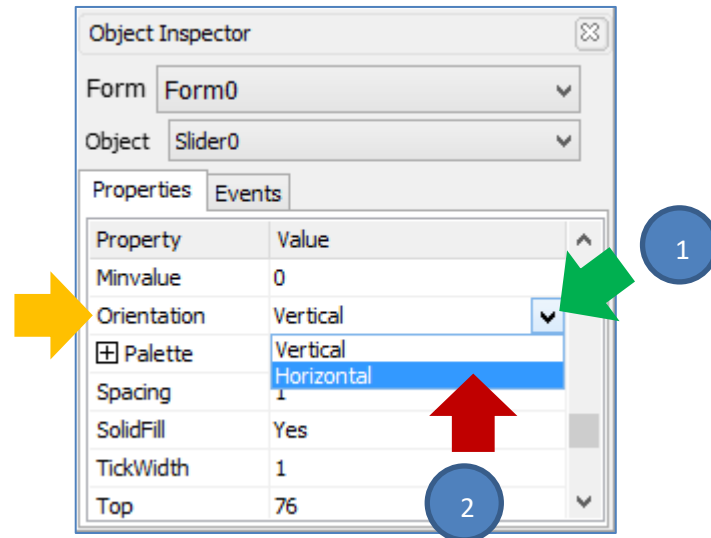
Place the Slider Object on the WYSIWYG Screen

Click on the WYSIWYG screen to place the slider object. This is Slider0.

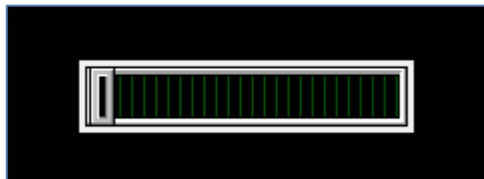


Change the Orientation of Slider0 to Horizontal

In the Object Inspector, look for the property “Orientation”. Change the value from “vertical” (default) to “horizontal”.

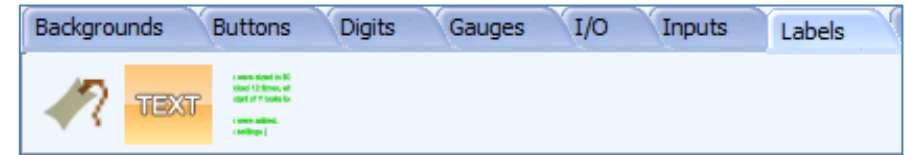


Experiment with the other properties of the slider. The appearance of the slider can be modified by changing several of the property values. Below is the final appearance of Slider0.



Add a Label to Form0

The static text object under the Labels pane can be used to create labels.

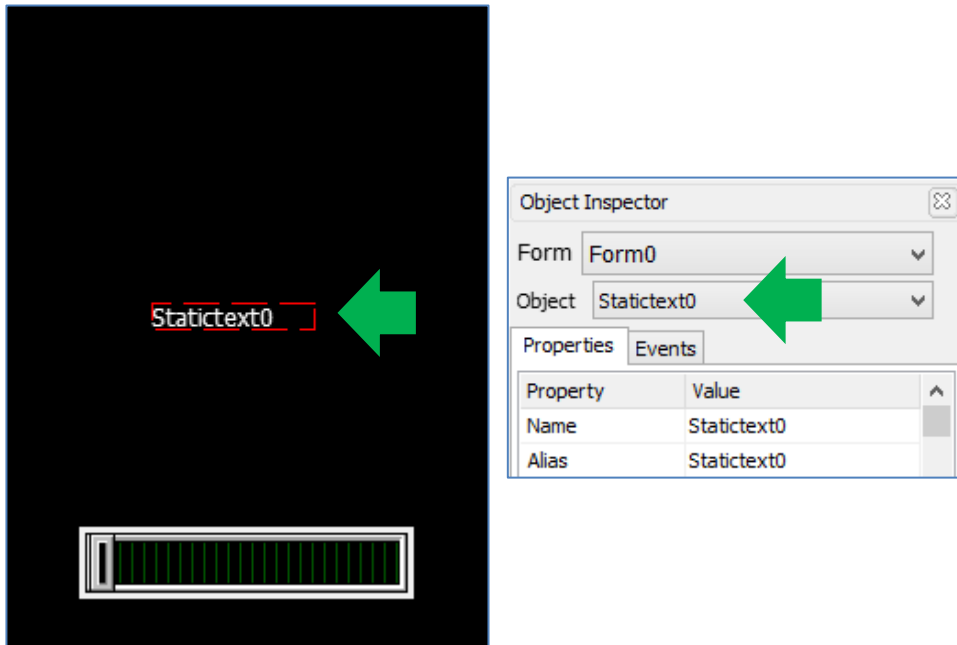


Select the Static Text Icon



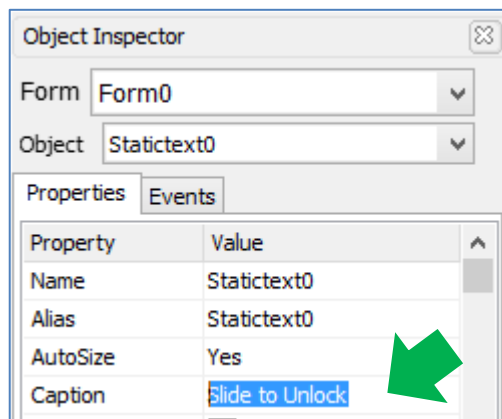
Place the Static Text Object on the WYSIWYG Screen

Click on the WYSIWYG screen to place the static text object. This is Statictext0.



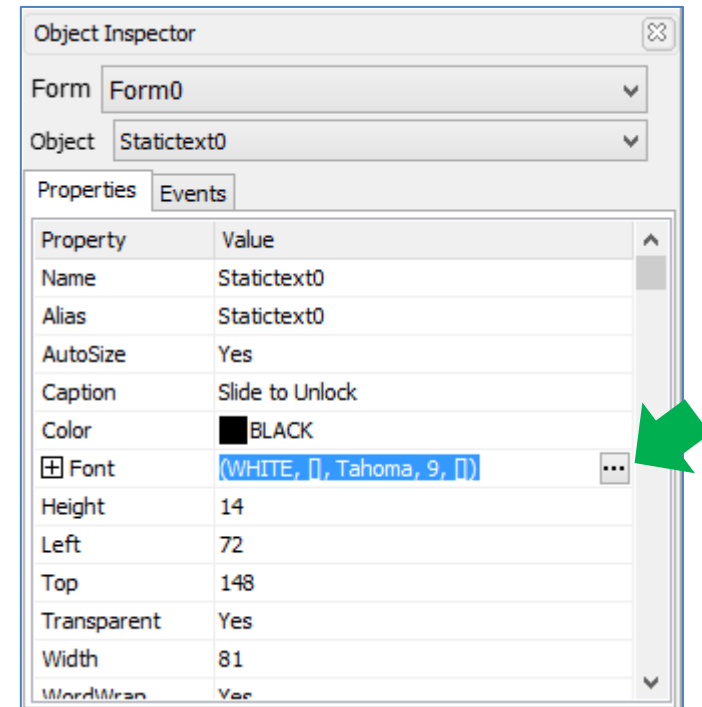
Change the Caption for Statictext0

Type the caption "Slide to Unlock" for Statictext0.

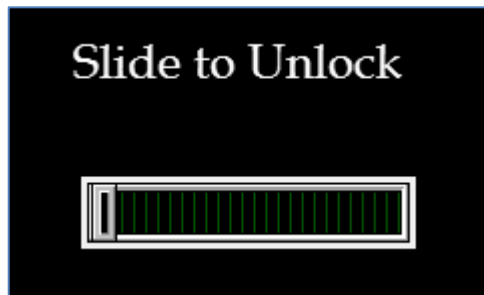
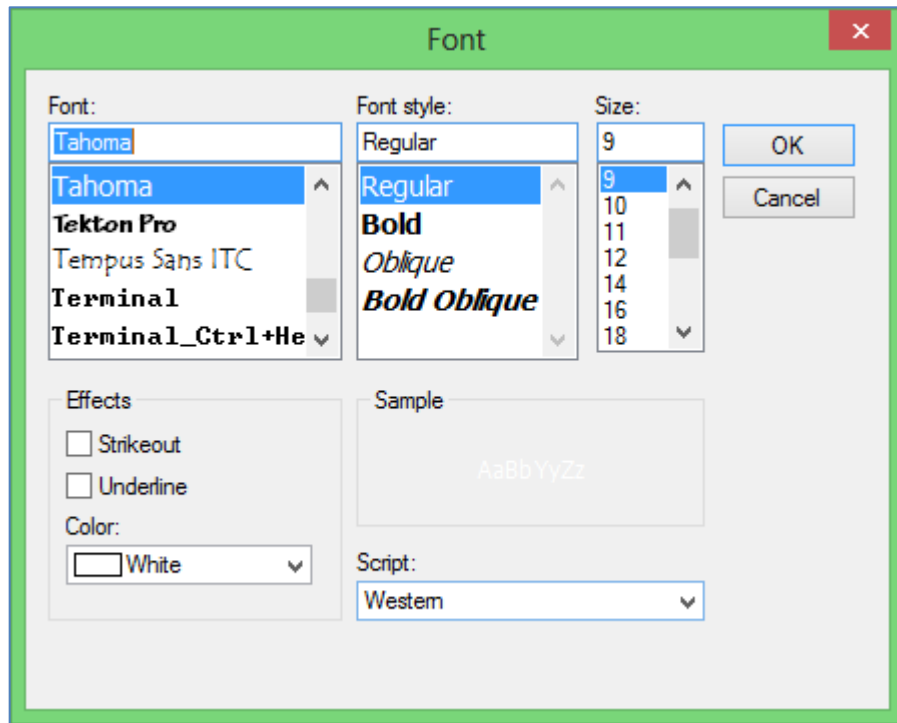


Change the Font for Statictext0

Click on the ellipsis icon of the property "Font".

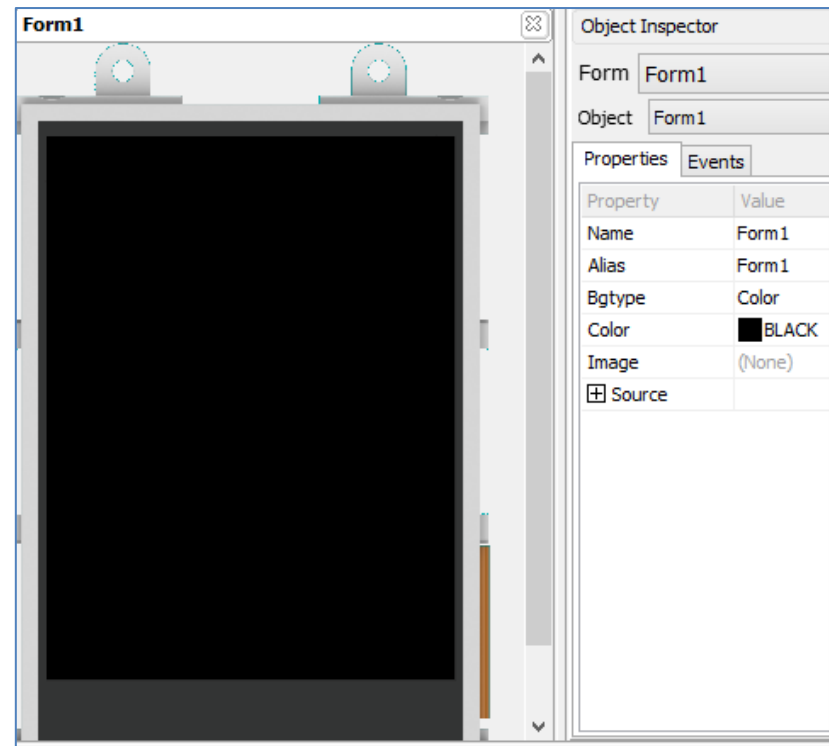
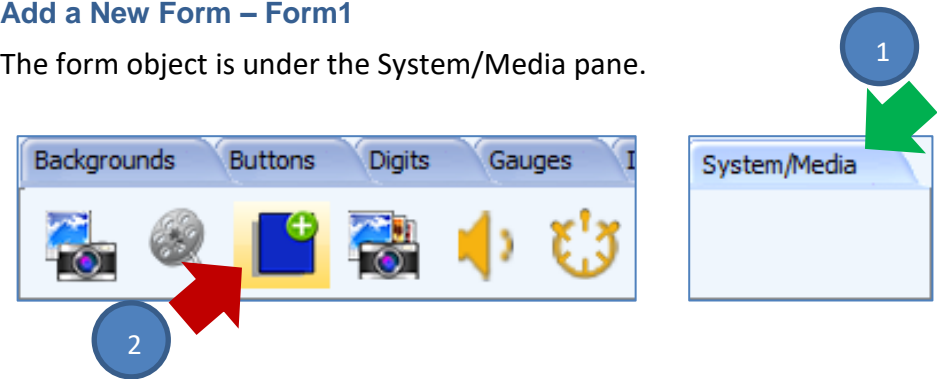


The Font window will appear. Experiment with the different fonts properties. Further below is the final appearance of Statixtext0.



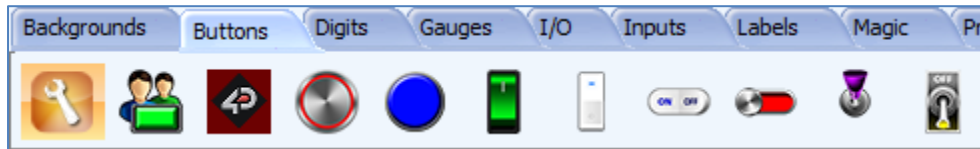
Add a New Form – Form1

The form object is under the System/Media pane.

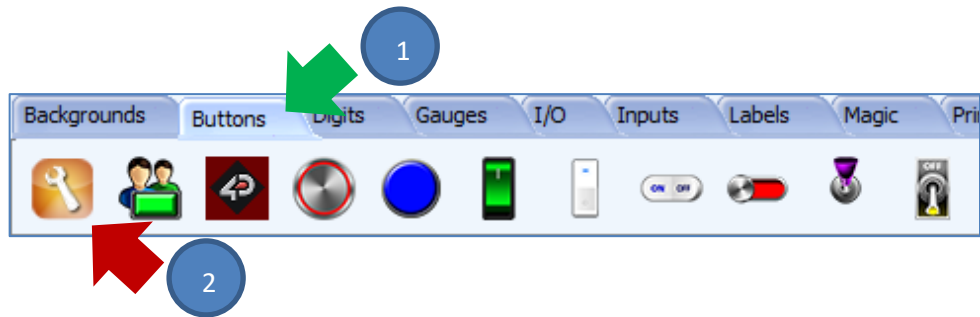


Add a Winbutton Object to Form1

The winbutton object is under the buttons pane.



Select the Winbutton Icon



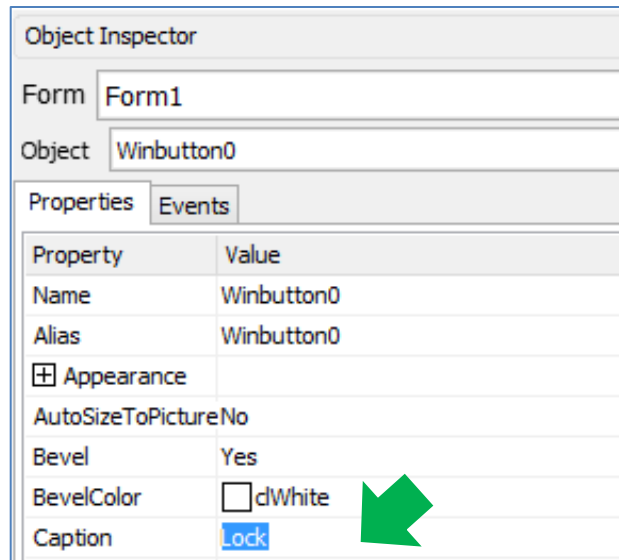
Place the Winbutton Object on the WYSIWYG Screen

Click on the WYSIWYG screen to place the winbutton object. This is Winbutton0.

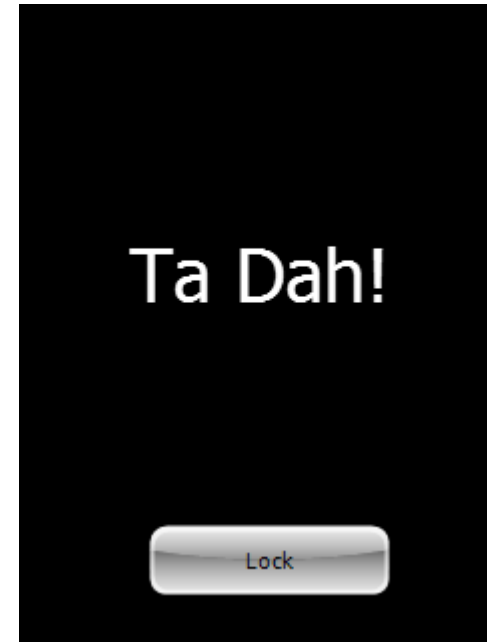


Object Inspector	
Form	Form1
Object	Winbutton0
Properties Events	
Property	Value
Name	Winbutton0
Alias	Winbutton0

Change the Caption for Winbutton0



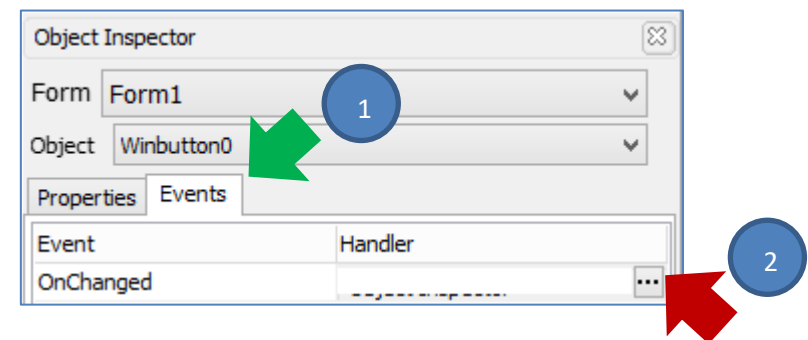
Experiment with the other properties of the winbutton object. The final appearance of Winbutton0 is shown below.



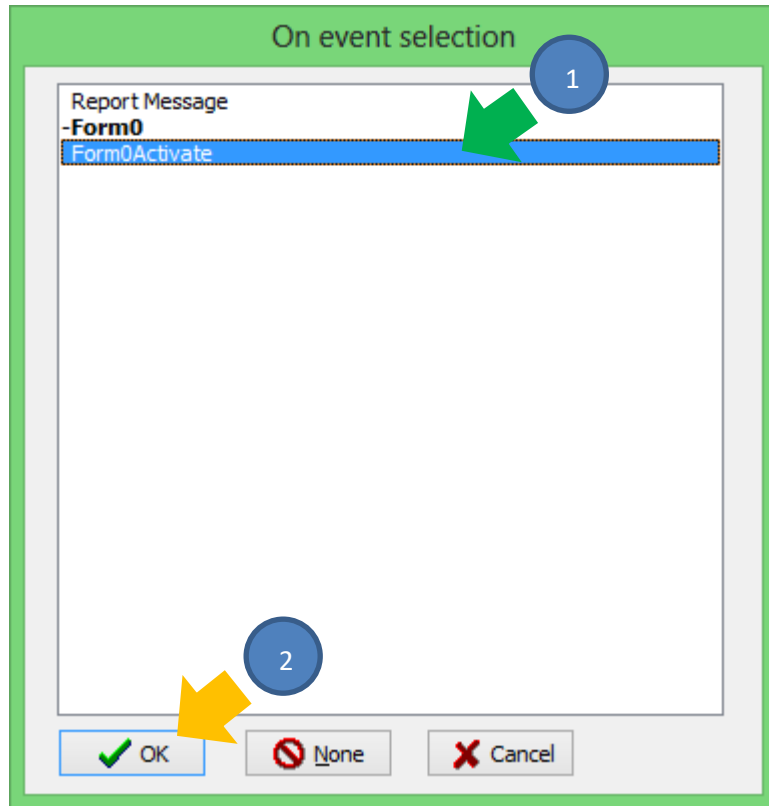
Note that another static text object is added to Form1. This is Statictext1.

Configure the OnChanged Event Property of Winbutton0

In the Object Inspector, go to the Events tab and click on the ellipsis icon for the OnChanged property.



The on event selection window appears. Select Form0Activate and click OK.

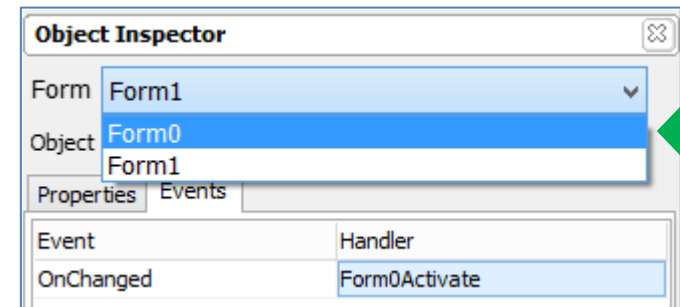


Add a Magic Event Object to Form0

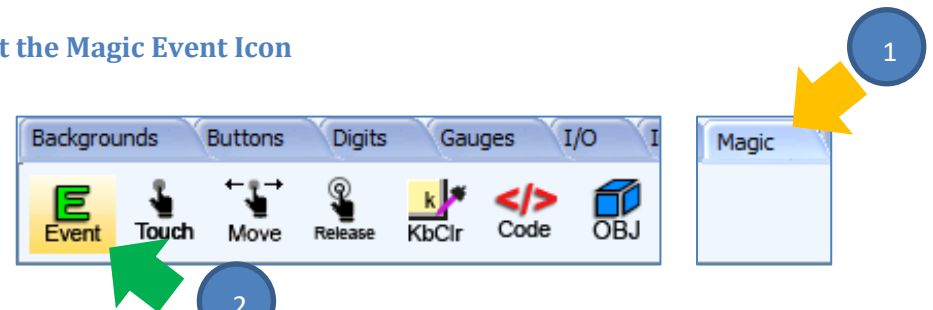
The Magic Event object is under the Magic Pane.



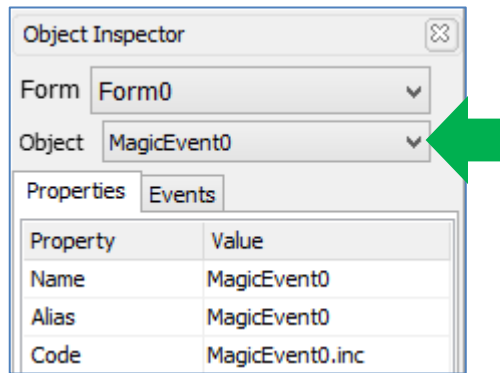
Navigate Back to Form0



Select the Magic Event Icon

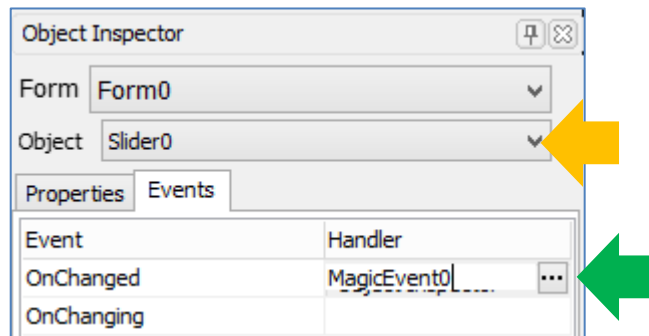


Clicking on the Magic Event icon will automatically cause Workshop to create a magic event object on the current form. Shown below is MagicEvent0.



Link MagicEvent0 to Slider0

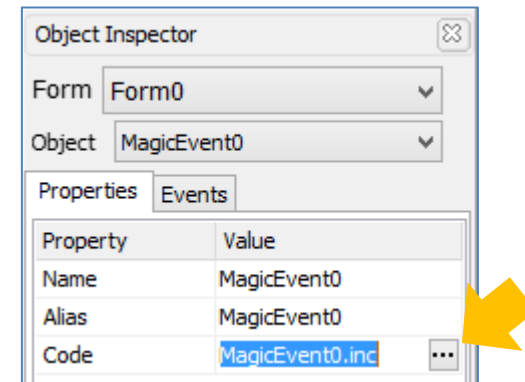
In the Object Inspector, go to the Events tab and set the OnChanged property of Slider0 to “MagicEvent0”.



This means that every time a **touch-and-release** action on Slider0 is detected, the routine inside MagicEvent0 is executed.

Write the Code for MagicEvent0

In the Object Inspector, click on the ellipsis symbol.



The code for MagicEvent0 appears.

```

MagicEvent0.inc
1  //
2  // Added 4/9/2015 10:05:59 AM
3  //
4  func MagicEvent0 (var newval)
5
6  endfunc
  
```

The code for MagicEvent0 is shown below.

```

Event0.inc
1 //
2 // Added 30/10/2014 11:26:21 AM
3 //
4 func MagicEvent0()
5     if (img_GetWord(hndl, iSlider0, IMAGE_INDEX)
6         ActivateForm(1) ;
7     endif
8     WriteObject(tGSlider, iSlider0, 0) ;
9 endfunc

```

The screenshot image was cropped to fit this column. See the SlideToUnlock demo in Worskhop for the full code.

Check for the Frame Value of Slider0

During runtime, the program navigates to Form1 if the slider knob is dragged to and released on the right-most part. Otherwise, the slider knob is moved back to the left-most part and the program stays on Form0. To ensure that the slider knob is on the right-most part, we write:

```

if (img_GetWord(hndl, iSlider0, IMAGE_INDEX) ==
    (img_GetWord(hndl, iSlider0, IMAGE_FRAMES)-1))

```

The function

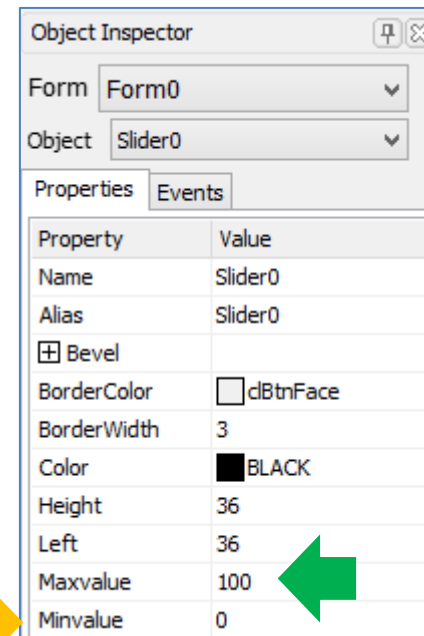
```
img_GetWord(hndl, iSlider0, IMAGE_INDEX)
```

returns the current frame of Slider0. If you happen to have released the slider knob halfway, then this function will return the value “50”.

The function

```
img_GetWord(hndl, iSlider0, IMAGE_FRAMES)-1
```

returns the number of frames (minus 1) of Slider0. Slider0 has 101 frames, since the minimum and values are 0 and 100, respectively.



Property	Value
Name	Slider0
Alias	Slider0
Bevel	
BorderColor	clBtnFace
BorderWidth	3
Color	BLACK
Height	36
Left	36
Maxvalue	100
Minvalue	0

For more information on the 4DGL function “[img_GetWord\(...\)](#)”, see the section “[2.18 Image Control Functions](#)” of the [Picaso or Diablo16 Internal Functions Reference Manual](#).

Activate Form1

To activate Form1, use its index as a parameter in the function

```
6 | ActivateForm(1) ;
```

For more information, see the section “5.1 Genie Magic callable Functions” of the [ViSi-Genie Reference Manual](#).

Write a Value to Slider0

To make the knob of Slider0 return to the left, we write:

```
WriteObject(tGSlider, iSlider0, 0) ;
```

For more information, see the section “5.1 Genie Magic callable Functions” of the [ViSi-Genie Reference Manual](#).

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).

The uLCD-32PTU and/or the uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

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.