



Smart Widgets: Vertical Slider

DOCUMENT DATE: **22nd November 2019**
DOCUMENT REVISION: **1.2**



Description

This application note shows how to create custom vertical slider for Picaso and Diablo16 touch screen display modules.

Before getting started, the following are required:

- Any Picaso display module such as the following:

Product Name	Description
gen4-uLCD-24PT	2.4 inch resistive touch
gen4-uLCD-28PT	2.8 inch resistive touch
gen4-uLCD-32PT	3.2 inch resistive touch

- The target module can also be a Diablo16 display:

Product Name	Description
gen4-uLCD-24DT	2.4 inch resistive touch
gen4-uLCD-28DT	2.8 inch resistive touch
gen4-uLCD-32DT	3.2 inch resistive touch
gen4-uLCD-35DT	3.5 inch resistive touch
gen4-uLCD-43DT	4.3 inch resistive touch
gen4-uLCD-50DT	5.0 inch resistive touch
gen4-uLCD-70DT	7.0 inch resistive touch
gen4-uLCD-32DCT-CLB	3.2 inch capacitive touch
gen4-uLCD-35DCT-CLB	3.5 inch capacitive touch
gen4-uLCD-43DCT-CLB	4.3 inch capacitive touch
gen4-uLCD-50DCT-CLB	5.0 inch capacitive touch
gen4-uLCD-70DCT-CLB	7.0 inch capacitive touch

Visit www.4dsystems.com.au to see the latest and/or superseded 4D display module products that use the Picaso, Picaso-Lite, and Diablo16 processors. **Non-touch and SB (Super Bright) versions are also available.**

- [4D Programming Cable / uUSB-PA5/uUSB-PA5-II](#)
- [Workshop4 IDE](#) (installed according to the installation document)

This requires the **PRO** version of Workshop4.

Note: Using a non-4D programming interface could damage the processor and void the warranty.

Content

Description	2
Content	3
Application Overview	3
Setup Procedure	4
Create a New Project	4
Design the Project	4
<i>Add Smart Slider to Project</i>	4
<i>Open Smart Widgets Editor</i>	5
<i>Select the Face Image</i>	7
<i>Adding Main Indicator</i>	8
<i>Setting Up Main Indicator Parameters</i>	10
<i>Adding Secondary Indicator</i>	12
<i>Setting Up Smart Slider</i>	16
<i>Adding Gauge for Output</i>	17
Run the Program	18
Proprietary Information	19
Disclaimer of Warranties & Limitation of Liability	19

Application Overview

The Smart Widgets Editor tool enables PRO version users to easily create custom widgets of their own design. It allows the user to create Sliders, Knobs and Gauges.

The purpose of this application note is to introduce the PRO version exclusive tool and to discuss how to create a Smart Slider. To demonstrate, this application note uses the ViSi Genie environment.

Setup Procedure

For instructions on how to launch Workshop4, 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

[First ViSi-Genie Project for Picaso](#)

or

[First ViSi-Genie Project for Diablo16](#)

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

[First ViSi-Genie Project for Picaso](#)

or

[First ViSi-Genie Project for Diablo16](#)


Design the Project

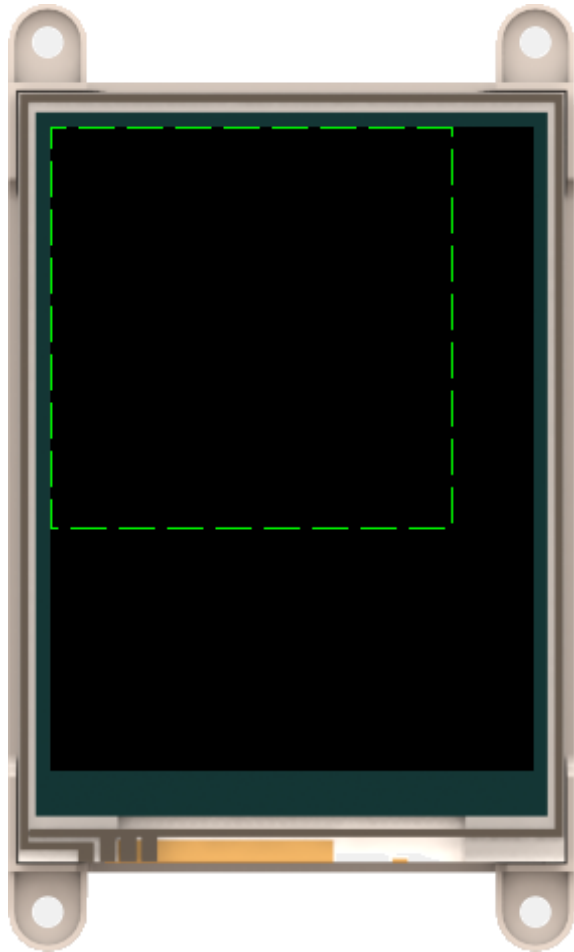
For this application, gen4-uLCD-32DT will be used for the project. Same procedure is applicable for any touch screen Picaso and Diablo16 displays.

Add Smart Slider to Project

Add a Smart Slider widget to your ViSi-Genie project. It can be found on the Inputs tab on the Widgets Pane.





Simply click on this icon  to select it. Then place it on the WYSIWYG area.



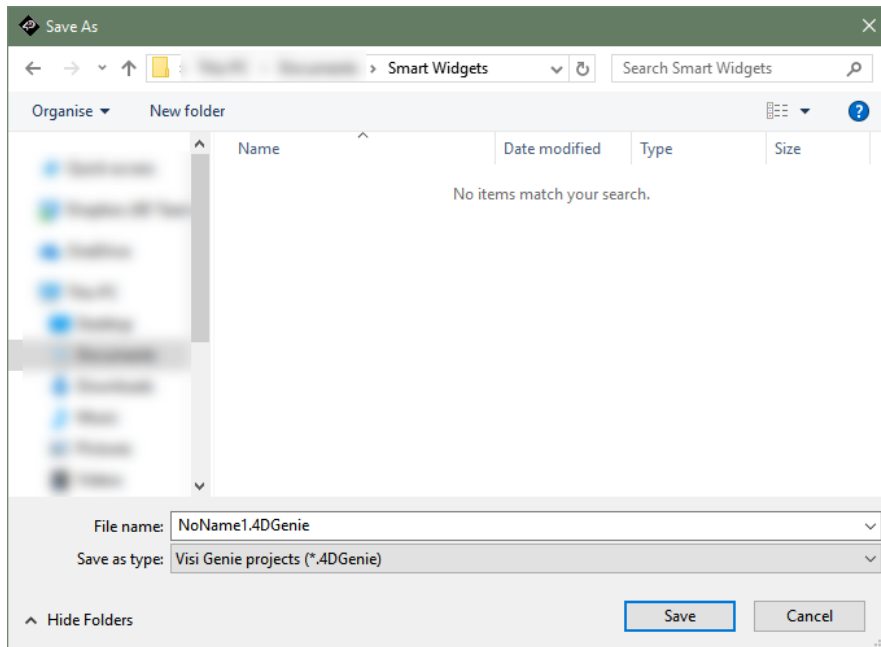
As displayed on the previous image, the widget appears empty when placed in the WYSIWYG.

Open Smart Widgets Editor

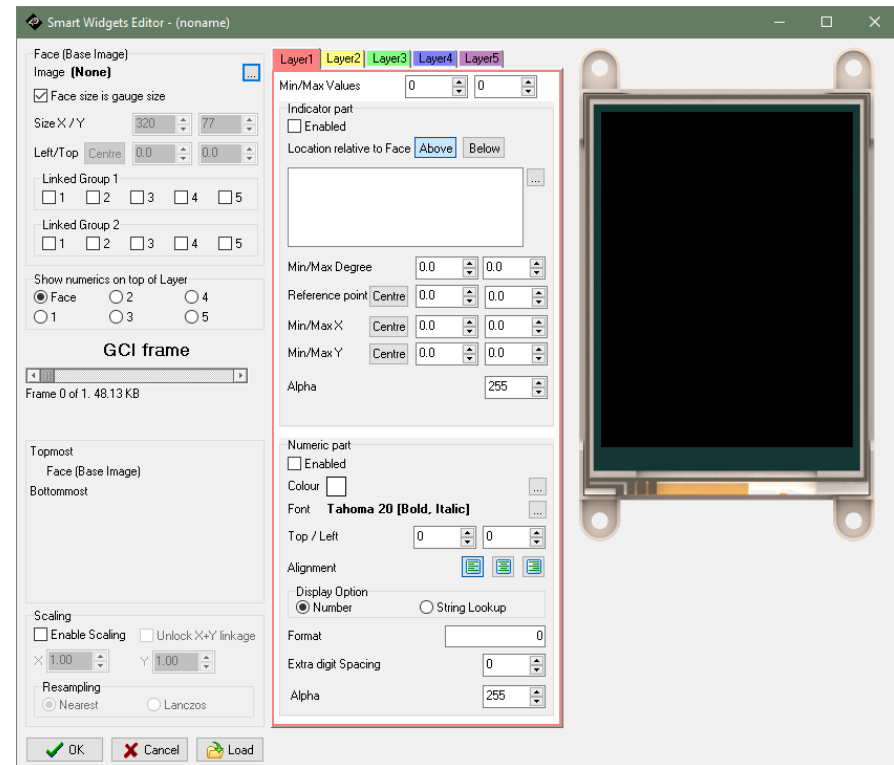
Open the Smart Widgets Editor tool by clicking on  of **Config** in the Object Inspector Properties tab.

Property	Value
Name	SmartSlider0
Alias	SmartSlider0
Config	(None) 

The tool requires that the project is already saved before the tool opens. Therefore, since on this case, it hasn't been saved yet, Workshop4 will automatically prompt to save



Save the project to desired location. The tool will open after the project has been saved.




As shown in the image, this tool has a lot of parts. The next steps will focus only on the minimum tool functionalities required to make a basic slider.

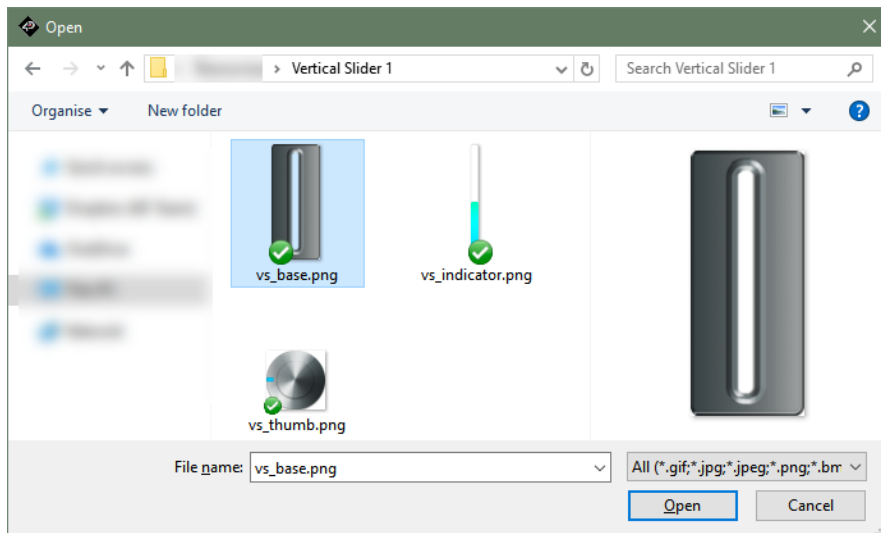
For detailed discussion on how each part works, please refer to the [Smart Widgets Editor User Guide](#).

Select the Face Image

The first step when creating a smart widget is to select the Face (base image).



Click  to select an image.



After selecting a base image, it will be displayed in the preview area.



The size of the output widget depends on the size of this image by default.



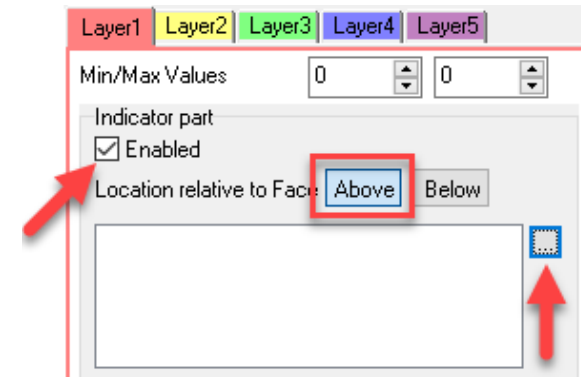
Any part of any image used in a smart widget will not be displayed if outside this area.

Adding Main Indicator

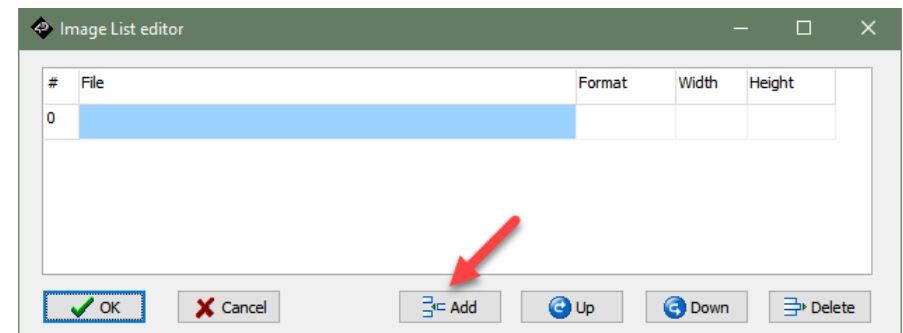
After selecting the image, a slider needs a thumb and an optional indicator.

In this project, Layer 1 will manipulate the thumb while Layer 2 will manipulate the other indicator.

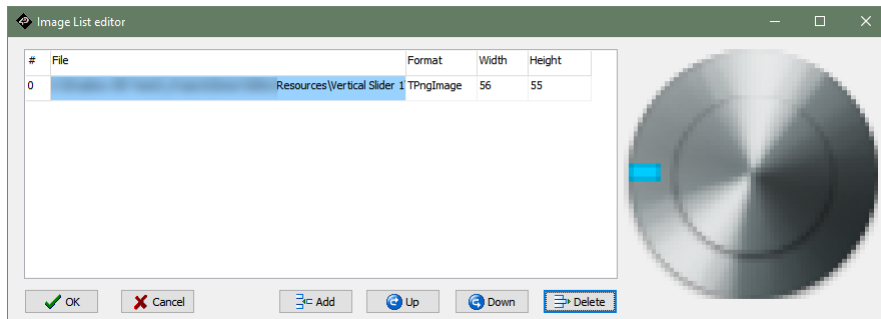
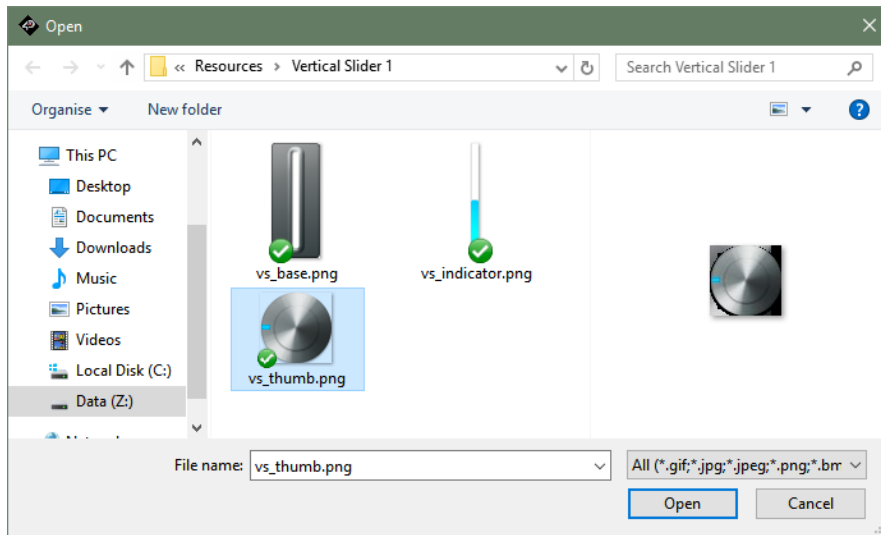
Enable Layer 1 and select the image similar to how the face image was selected.



Obviously, the thumb should be above the base image.



Select the image and click **Open**



When done adding the images for the layer, simply click **OK**

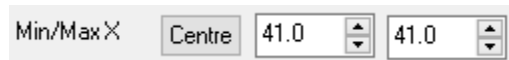
You'll see the knob at the top left of the base image.



The next thing to do is to select its starting and end positions.

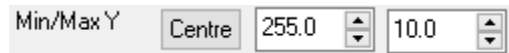
Setting Up Main Indicator Parameters

First, we want the knob to be centered horizontally all the time.



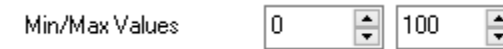
For this slider, the **Min/Max X** needs to be both at 41. This will make the X position of the knob be constant from minimum value to maximum value.

Now, to create frames that moves the thumb vertically, an initial Y position (**Min Y**) and final Y position (**Max Y**) must both be specified.



In this case, the thumb moves from 255 to 10 (moving upward).

Lastly, it is required to assign the minimum and maximum values (**Min/Max Values**).



For this application note project, 0 to 100 was used. That is a total 101 frames moving from 255 to 10 Y positions.

Here's the 26th frame (index 25):



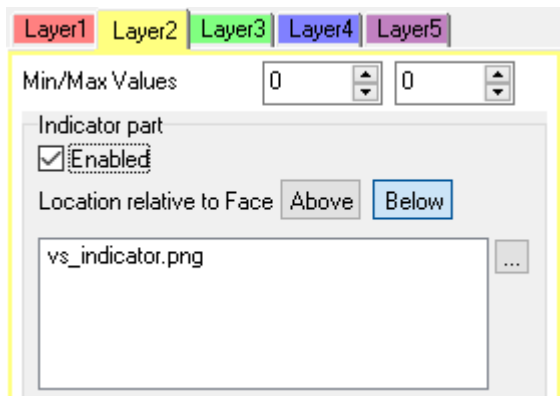
While this is the last frame (101st / index 100)



At this state, this slider is already sufficient for setup for it to be used in the project. However, in this application note project, additional layers will be used to for better design.

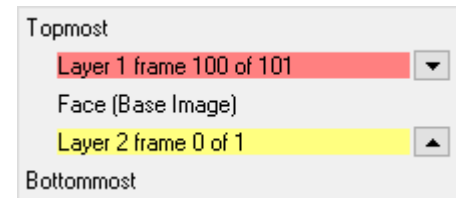
Adding Secondary Indicator

Next, add another indicator for the slider. Enable layer 2 and select the image the same way as Layer 1.

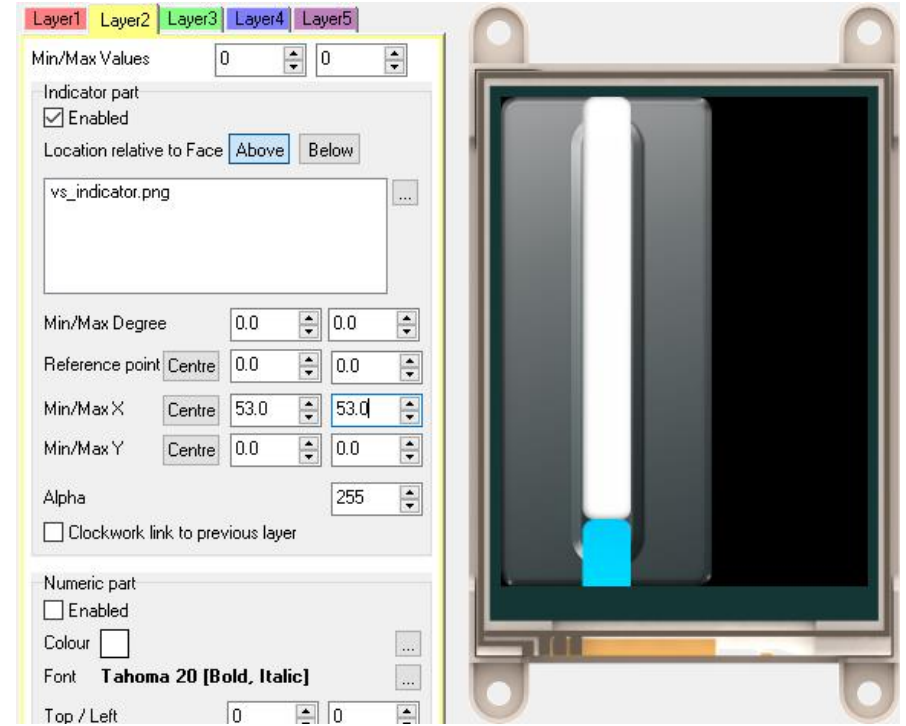


Layer 2 will be below the Face (base image)

Layer 1	vs_thumb.png
Face	vs_base.png
Layer 2	vs_indicator.png



For Layer 2, similar to Layer 1, it also needs to be positioned horizontally centered on the image. When positioning images, it is ideal to place the image above the face image temporarily. If necessary, other layers can be temporarily disabled.

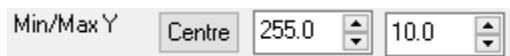


As shown in the image, **Min/Max X** should be constant at 53.

Enable Layer 1 and bring Layer 2 below the face image.

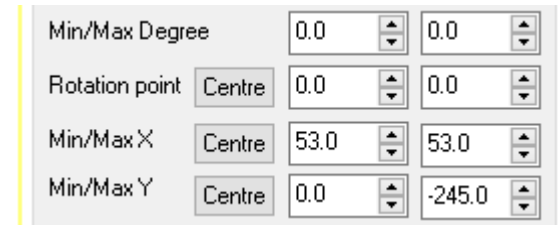


Remember that the thumb moves from this position.

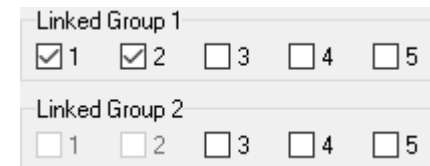


That is a 245-pixel displacement.

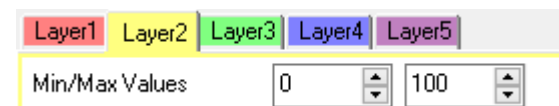
For Layer 2, it should have the same displacement as Layer 1 and **Min/Max Values**.



Additionally, it also needs to be linked to Layer 1 so both of them will go through their minimum and maximum values at the same time.



This layer's Min/Max Values ideally should be same as Layer 1.



This will be the output:

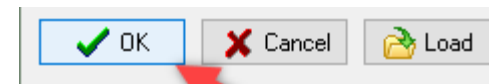
- At 26th frame (index 25)



- At 101st frame (last frame / index 100)



After successfully, configuring the slider, press **OK**.



The slider can be moved to a desired position.



Setting Up Smart Slider

Some properties need to be set before properly using the slider.

Property	Value
Name	SmartSlider0
Alias	SmartSlider0
Config	SmartSlider0
Left	102
Orientation	Vertical
MaxOffset	0
MinOffset	0
Top	0

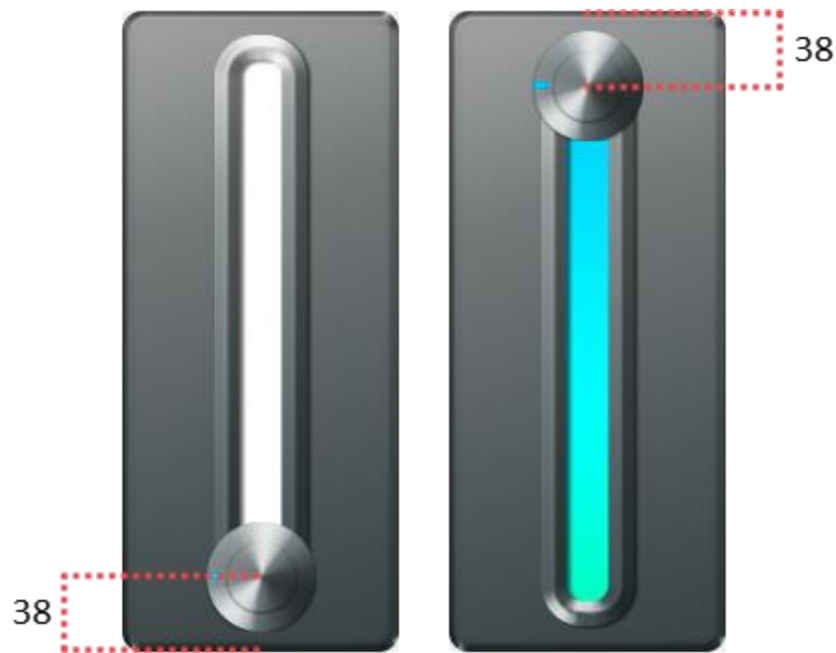
On the Object Inspector's Properties tab, **Orientation**, **MaxOffset** and **MinOffset** need to be provided by the user.

Obviously, **Orientation** needs to be set to Vertical.

As for the other two, this project uses:

Property	Value
Name	SmartSlider0
Alias	SmartSlider0
Config	SmartSlider0
Left	102
Orientation	Vertical
MaxOffset	38
MinOffset	38
Top	0

The values are from the desired touch area for the slider. Refer to the images below



After setting everything up, the slider can be used similar to a normal slider.

Note: MaxOffset refers to the offset below the minimum slider position (MaxY position value) while MinOffset refers to the offset above the maximum slider position (MinY position value).

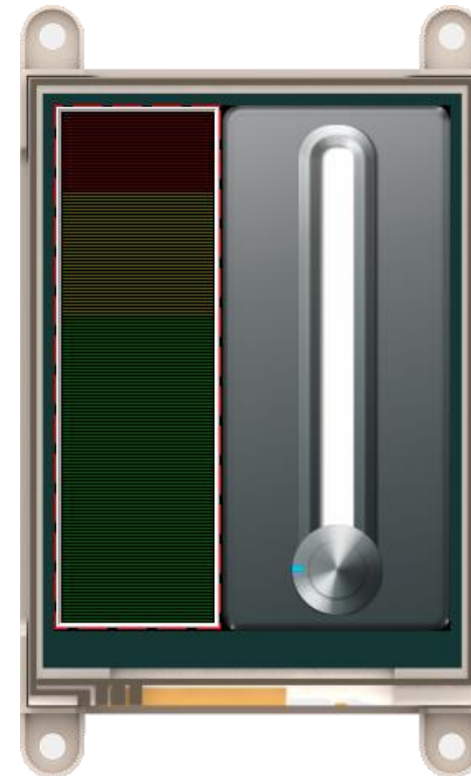
Adding Gauge for Output



Gauges are great options to present information.

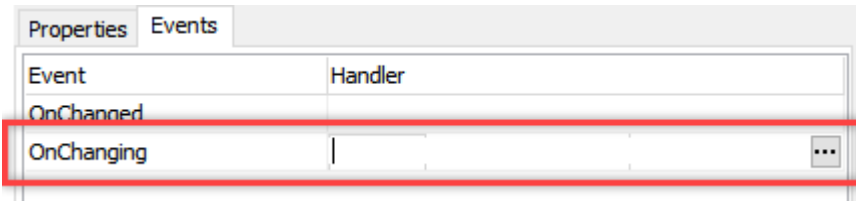


Add a Gauge to the project.

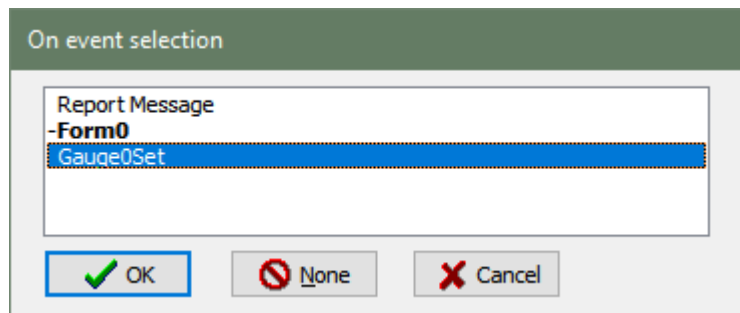


It is important that the Gauge has the same number of frames as the slider. With values 0 to 100 for the Gauge, they have same number of frames.

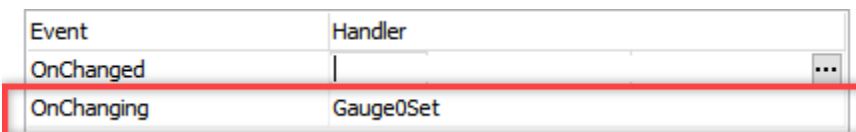
Then for the Slider, under Events tab of the Object Inspector:



Set the event for **OnChanging**:



Select **Gauge0Set** and click on **OK**.



The project is now ready for upload.

Run the Program

For instructions on how to save a **ViSi-Genie** project, how to connect the target display to the PC, how to select the program destination, and how to compile and download a program, please refer to the section “**Run the Program**” of the application note

[First ViSi-Genie Project for Picaso](#)

or

[First ViSi-Genie Project 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.