



Smart Widgets: Tank

DOCUMENT DATE: **9th MAY 2020**
DOCUMENT REVISION: **1.0**

WWW.4DSYSTEMS.COM.AU



Description

This application note provides instructions for designing a Custom Tank widget for a ViSi application.

Before getting started, the following are required:

Hardware

- Any [4D Systems display module](#) powered by any of the following processors:
 - Diablo16
 - Picaso
 - Pixxi28/44
- [Programming Adaptor for target display module](#)
- [uSD Card](#)
- [USB Card Reader](#)

Software

- [Workshop4](#)
- This requires the **PRO** version of Workshop4

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

This application note is applicable to all touch screen 4D displays. However, Smart Gauges can also be used on non-touch displays.

Content

Description	2
Content	2
Application Overview.....	3
Setup Procedure	4
Create a New Project	4
Design the Project.....	4
<i>Add a Smart Gauge Object</i>	<i>5</i>
<i>Open the Smart Widgets Editor Tool.....</i>	<i>5</i>
<i>Add a Face or Base Image.....</i>	<i>6</i>
<i>Add a Manipulated Image to Layer 1.....</i>	<i>7</i>
<i>Location of an Image in the Working Area.....</i>	<i>8</i>
<i>Move the Image on Layer 1 Vertically</i>	<i>9</i>
<i>Location of Layer 1 with Respect to the Base Image.....</i>	<i>9</i>
<i>The Need for Multiple Frames.....</i>	<i>11</i>
<i>Set the Minimum and Maximum Values</i>	<i>12</i>
<i>Check the Generated Frames</i>	<i>13</i>
<i>Move the Image on Layer 1 Diagonally</i>	<i>14</i>
<i>Add a Numeric Part to Layer 1</i>	<i>14</i>
<i>Add Static Characters to a Numerical Part</i>	<i>16</i>
<i>Display Negative Values</i>	<i>17</i>
<i>Add a Background Image to Layer 2</i>	<i>19</i>
<i>Add and Configure an Input Object</i>	<i>20</i>

Run the Program 21

Proprietary Information 22

Disclaimer of Warranties & Limitation of Liability 22

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 tank using a Smart Gauge widget. 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

- [ViSi-Genie Getting Started - First Project for Diablo16 Display Modules](#)
- [ViSi-Genie Getting Started - First Project for Picaso Displays](#)
- [ViSi-Genie Getting Started - First Project for Pixxi Display Modules](#)

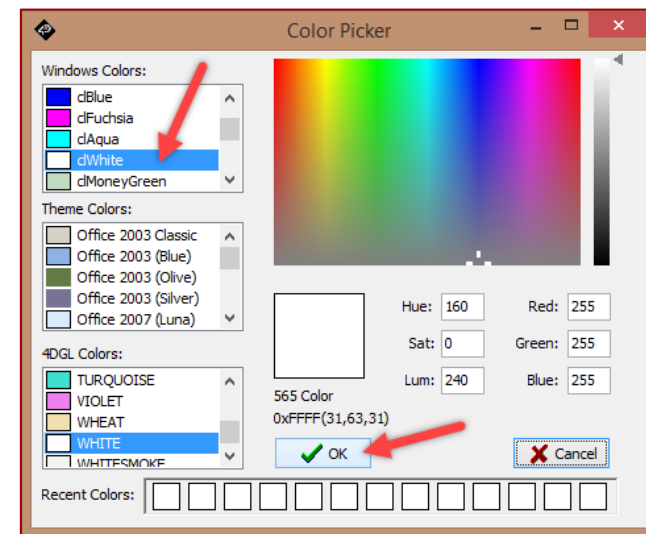
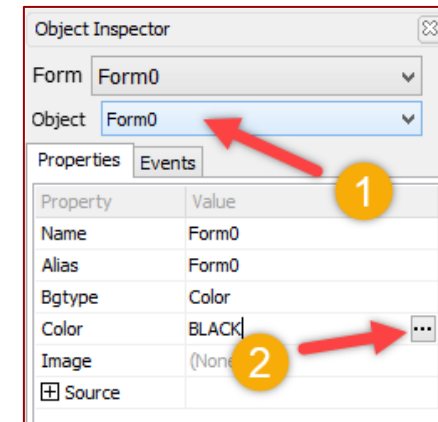
Create a New Project

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 Diablo16 Display Modules](#)
- [ViSi-Genie Getting Started - First Project for Picaso Displays](#)
- [ViSi-Genie Getting Started - First Project for Pixxi Display Modules](#)

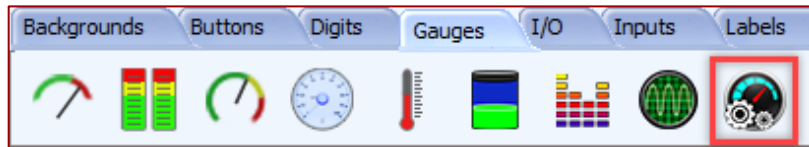
Design the Project


For this application, gen4-uLCD-32DT will be used for the project. Same procedure is applicable for any Picaso and Diablo16 displays. First, set the background color of Form0 to white.

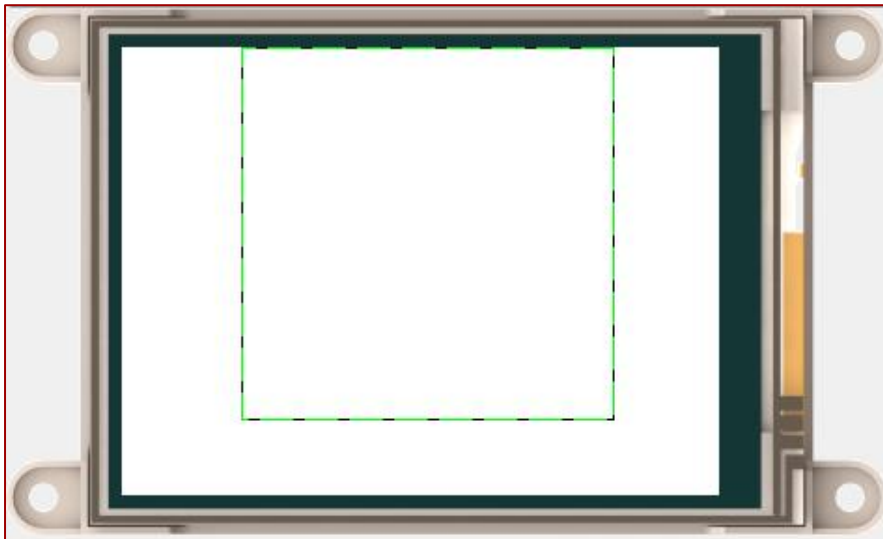


Add a Smart Gauge Object

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




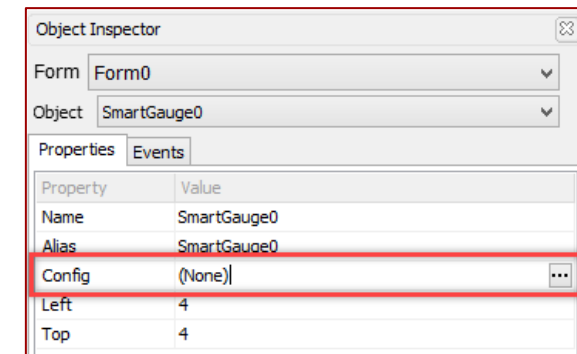
Simply click on this icon  to select it. Then place it on the WYSIWYG area.



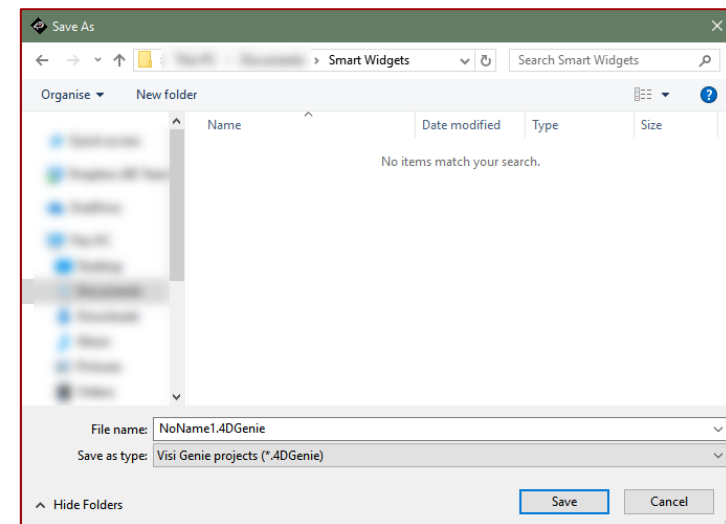
As displayed on the previous image, the widget appears empty when placed in the WYSIWYG area.

Open the Smart Widgets Editor Tool

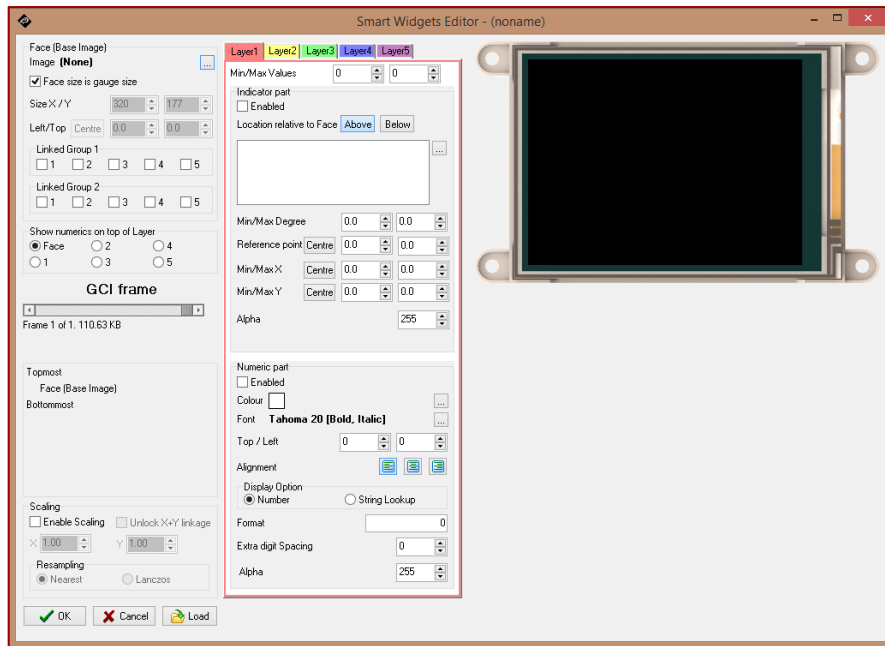
Open the Smart Widgets Editor tool by clicking on  of **Config** in the Object Inspector Properties tab.



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 the user to save




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

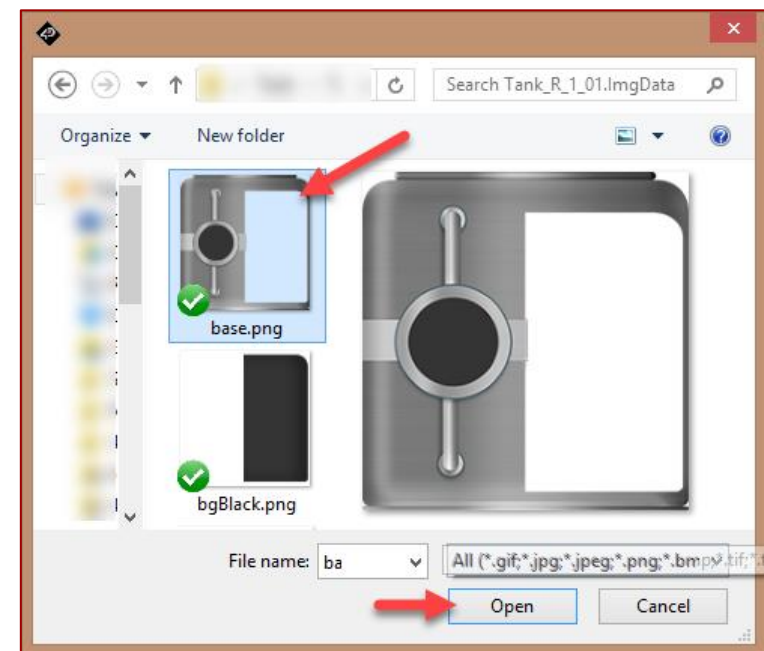


As shown in the image, this tool has a lot of parts. The next steps will focus only on the minimum tool functionalities required to make a tank.

For detailed discussion on how each part works, please refer to the **Smart Widgets Editor User Guide**.

Add a Face or Base Image

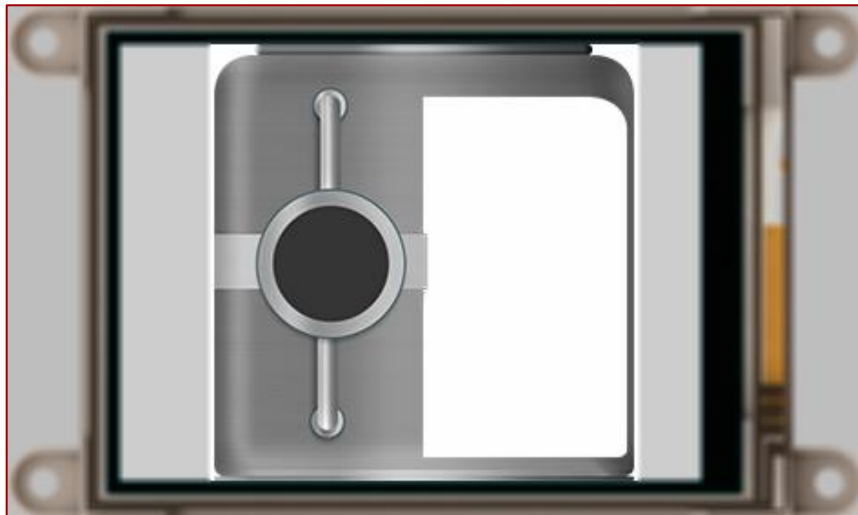
The first step when creating a smart widget is to select the face (base image). The face or base image is a static part of the widget. Click  to select an image.



After selecting a base image, it will be displayed in the preview area. Note that the base image has a transparent part.



Also, note that, by default, the area occupied by the base image is the working area.

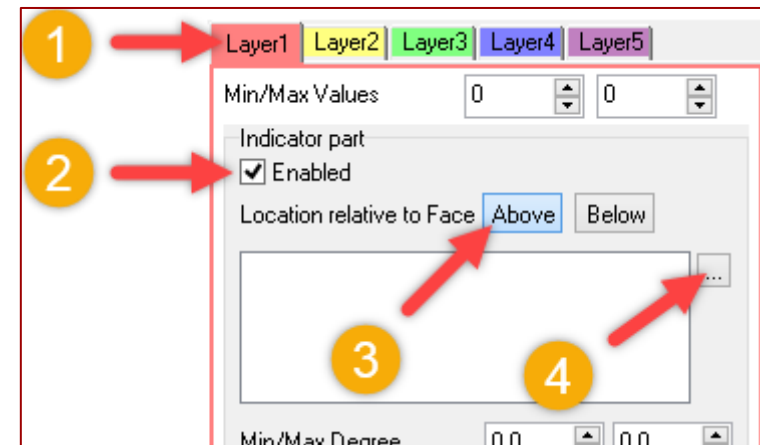


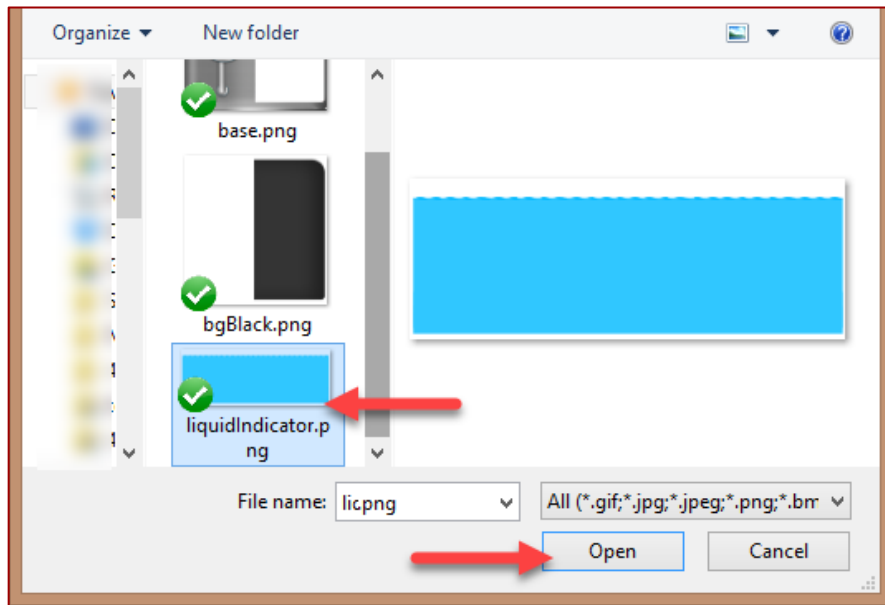
Any part of any image used in a smart widget will not be displayed if outside the working area.

Add a Manipulated Image to Layer 1

The transparent part of the base image will be filled up with liquid. This liquid part can be another image which moves vertically or even diagonally to fill the tank. To implement this, we need a layer containing a manipulated image. We will use Layer 1 for this purpose.

Enable Layer 1 and add a manipulated image to it by following the procedure shown below.



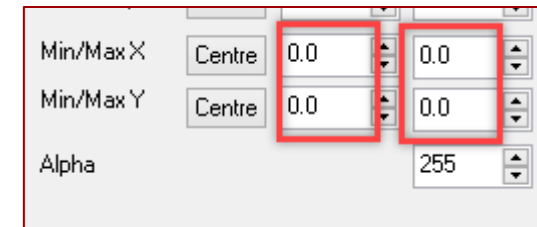


The image on Layer 1 now appears in the working area.



Location of an Image in the Working Area

The location of an image in the working area is set using the **Min/Max X and Y** parameters.

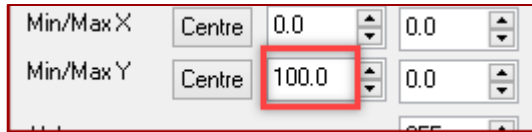


These coordinates are relative to the top-left corner of the working area. By default, Smart Widgets Editor uses the top-left corner of an image as the reference point when it places the image in the working area. As can be seen in the image above, the values of the **Min/Max X and Y** parameters are set to 0 by default. This means that, by default, the top-left corner of the image on Layer 1 coincides with the top-left corner of the working area.

For a detailed discussion on reference points of images and their location in the working area, refer to the document **Smart Widgets Editor User Guide**.

Move the Image on Layer 1 Vertically

For demonstration purposes, set the value of the **Min Y** parameter as indicated below.

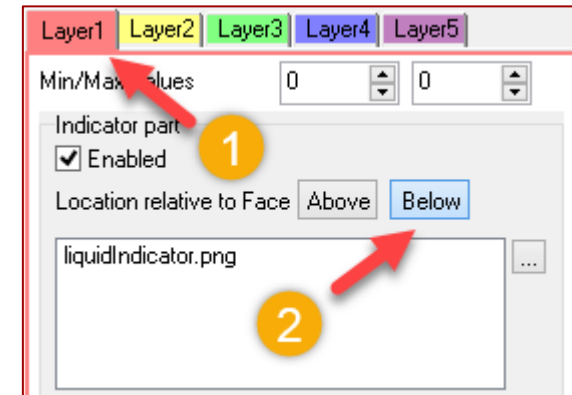


In the working area, the image on Layer 1 is moved 100 pixels downwards.



Location of Layer 1 with Respect to the Base Image

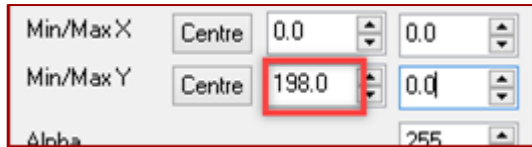
At this point the image on Layer 1 is on top of the base image. Put the image on Layer 1 behind the base image by following the procedure below.



The working area is now updated.



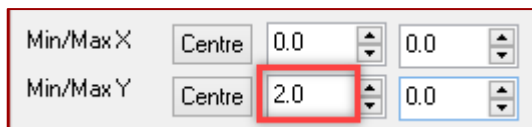
Move the image on Layer 1 further down by setting the value of the **Min Y** parameter as shown below.



The working area is now updated.



Move the image on Layer 1 back up by setting the value of the **Min Y** parameter as shown below.



The working area is now updated.



Therefore, the location of the image on Layer 1 along the y axis when the tank is empty is **198**. On the other hand, the y location is **2** when the tank is full. An offset of a few pixels can be added or subtracted from these y positions if desired.

The Need for Multiple Frames

At this point the widget has only one frame. This frame shows the image on Layer 1 offset by a certain number of pixels from its original position. Also, on top of Layer 1 is the base image. Our next objective is to be able to move the image on Layer 1 up and down the working area. Each state of the widget while the image on Layer 1 moves will be represented by a frame. The widget, therefore, has to have multiple frames.

To illustrate, let's say we want to use 101 frames to represent the different states of the widget as the image on Layer 1 is moved. The output will look like the following:

1st frame (frame index 0):



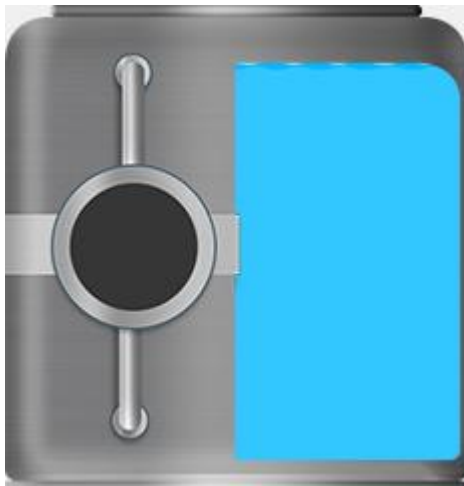
34th frame (frame index 33):



67th frame (frame index 66):



101st frame (frame index 100):



To achieve this, the values of the minimum and maximum Y parameters should be set to the values indicated below.



Set the Minimum and Maximum Values

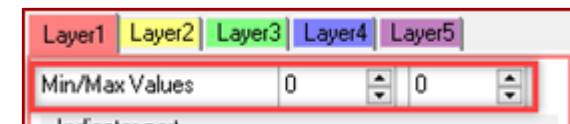
Furthermore, let's say we want to associate a range of values to the different frames of the widget. Each value will correspond to a unique frame. To illustrate:

Frame Index	Associated Value
0	0
33	33
66	66
N - 1	N - 1

In the table above, **N** is the total number of frames. The index number always starts at **0**. The associated value, on the other hand, can start with any number. Hence,

Frame Index	Associated Value
0	Min Value
N - 1	Max Value

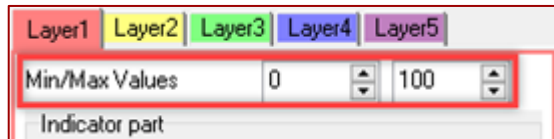
For a layer containing a manipulated image, the user needs to set the **Min** and **Max Values** parameters.



Smart Widgets Editor then computes for the total number of frames, generates the frames, then assigns the index number for each frame. The values of the **Min** and **Max Values** parameters are also used by Smart

Widgets Editor when displaying the numeric part of the widget. This is discussed in the section [Add a Numeric Part to Layer 1](#).

Therefore, to make Smart Widgets Editor generate 101 frames, input the values indicated below to the **Min/Max Values** fields of Layer 1.



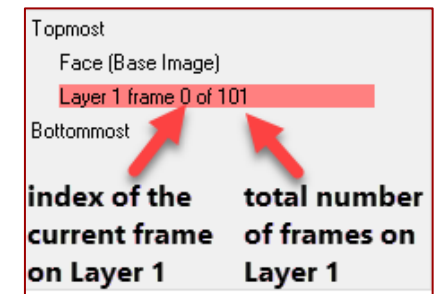
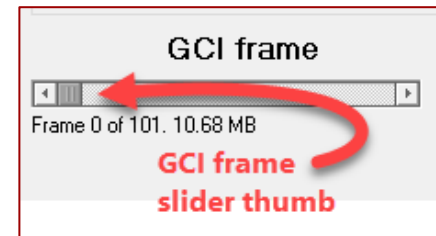
Below is a table showing four parameters of Layer 1 at this point.

Associated Value	Frame Index	X	Y
0 (Min Value)	0	0 (min X)	198 (Min Y)
100(Max Value)	100	0 (max X)	2 (Max Y)

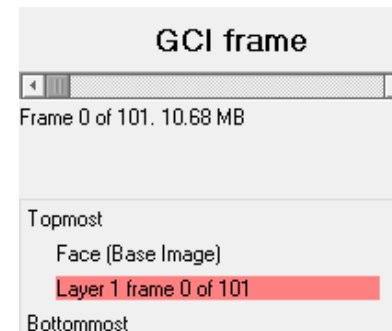
As can be seen in the table, the y location of the image on Layer 1 at frame index **0** or layer value **0** is **198** pixels. The y location is **2** pixels at frame index **100** or layer value **100**. The **x** parameters remain constant. Essentially, therefore, the image on Layer 1 moves vertically from **y = 198 pixels** up to **y = 2 pixels** as the frame index and layer value increase. Smart Widgets Editor then generates all of the frames needed to satisfy this behavior of the image on Layer 1.

Check the Generated Frames

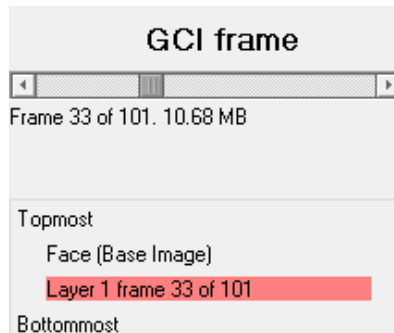
To see the generated frames, move the thumb of the GCI frame slider indicated in the image below. The layer guide section also shows the index of the current frame and the total number of frames in a layer.



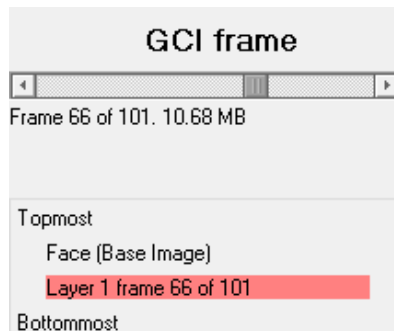
1st frame (frame index 0):



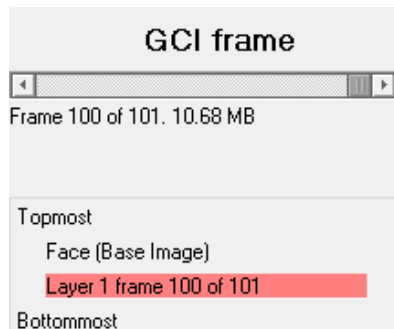
34th frame (frame index 33):



67th frame (frame index 66):

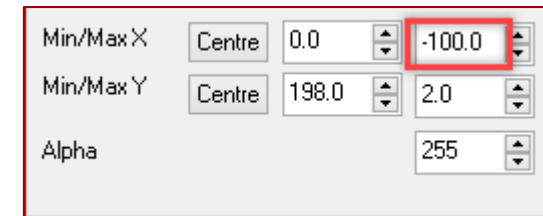


101st frame (frame index 100):



Move the Image on Layer 1 Diagonally

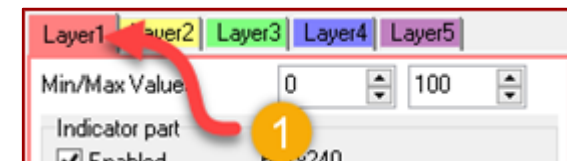
To further improve the animation effect of the liquid part, set the **Max X** parameter to the value indicated below.

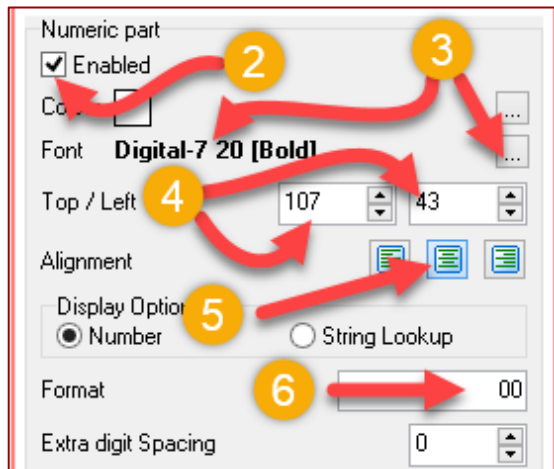


This moves the image on Layer 1 to the left (from $x = 0$ pixels to $x = -100$ pixels) as the frame index and layer value increase. Since the movement has x and y components, it is therefore a diagonal motion.

Add a Numeric Part to Layer 1

As mentioned earlier, there is a range of values that is associated to a layer, and the layer can display these values on the corresponding frames. These values can be shown by adding a numerical part to the layer. To add a numerical part to Layer 1, follow the procedure shown below.





Other fonts can be used if necessary. The numerical part on Layer 1 can now be seen in the working area.

1st frame (frame index 0):



34th frame (frame index 33):



67th frame (frame index 66):

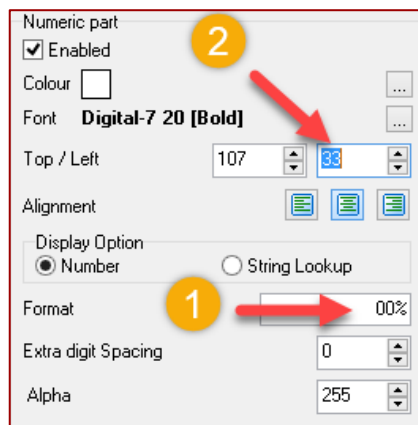


101st frame (frame index 100):



Add Static Characters to a Numerical Part

It is also possible to display static characters on the numerical part of a layer. This is implemented by inserting the characters to the **Format** field of the Numeric part section. For instance, to insert a percent symbol to the numerical part on Layer 1, the settings below could be used.



The output would then look like the following.

1st frame (frame index 0):



34th frame (frame index 33):



67th frame (frame index 66):



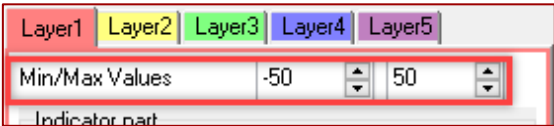
101st frame (frame index 100):



For more information on the different properties of the numerical part of a layer, refer to the document **Smart Widgets Editor User Guide**.

Display Negative Values

In some applications it is necessary for a widget to display negative values. To illustrate this on Layer 1, simply change the values of the **Min/Max Values** parameters to those indicated below.



Note that with the above **Min/Max Values** parameter values, the total number of frames is still 101. The table below shows the frame index numbers and the associated values.

Frame Index	Associated Value
0	-50 (Min Value)
33	-17
66	16
100	50 (Max Value)

The numerical part on Layer 1 is now updated.

1st frame (frame index 0):



67th frame (frame index 66):



34th frame (frame index 33):

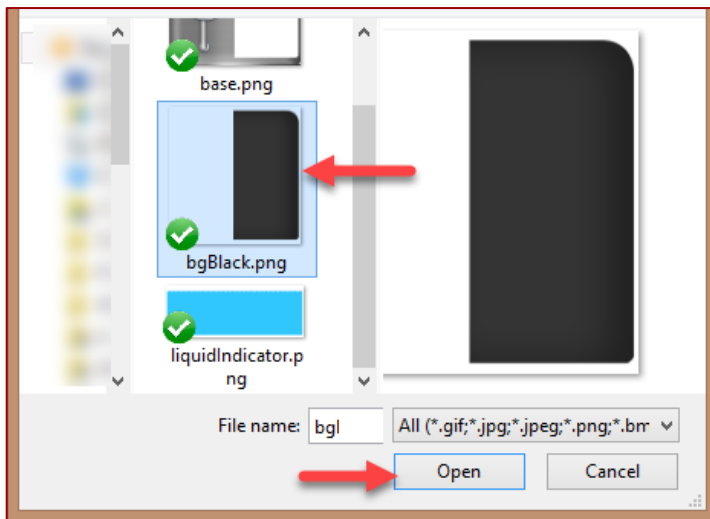


101st frame (frame index 100):

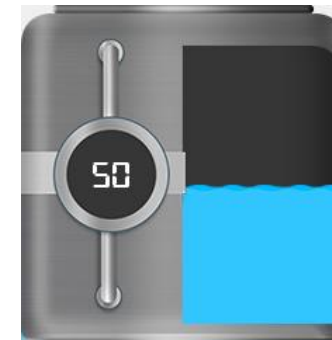


Add a Background Image to Layer 2

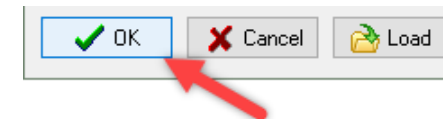
A background image (to cover the transparent part of the base image) can be added to Layer 2. To do this, follow the procedure below.



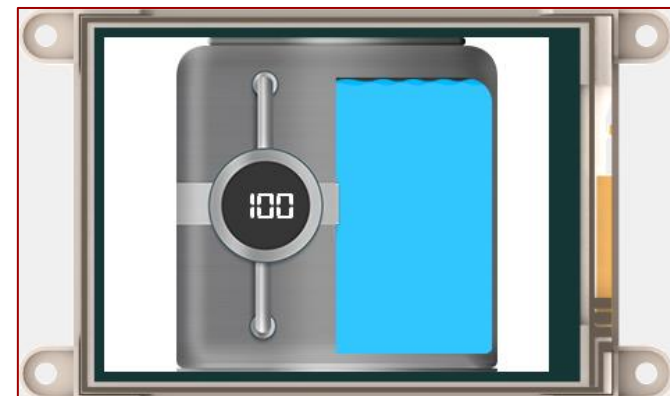
The working area is now updated.



After successfully configuring the smart gauge, press **OK**.



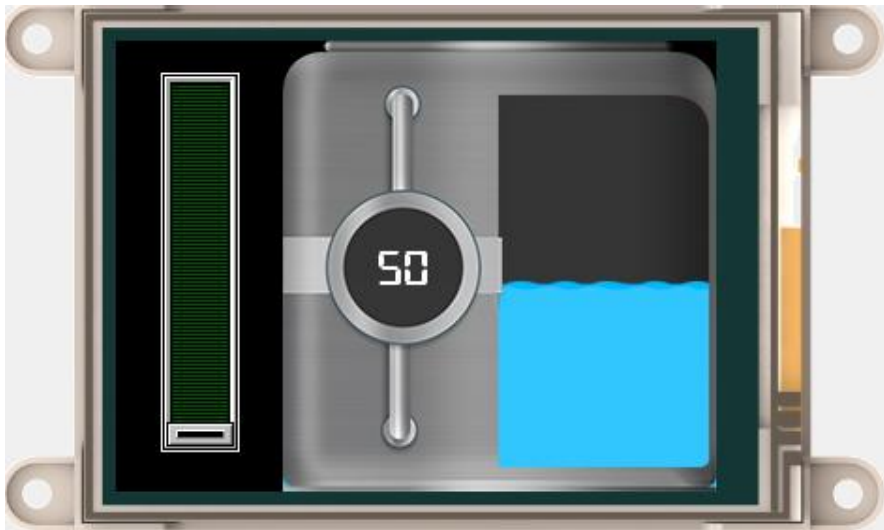
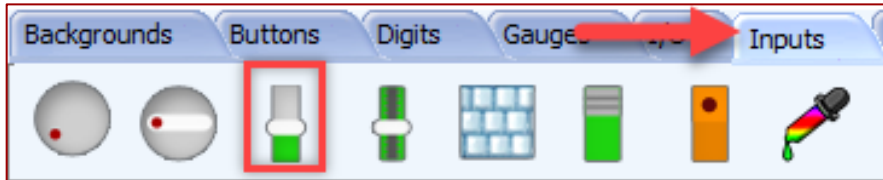
The smart widget will now appear on the WYSIWYG area of the project.



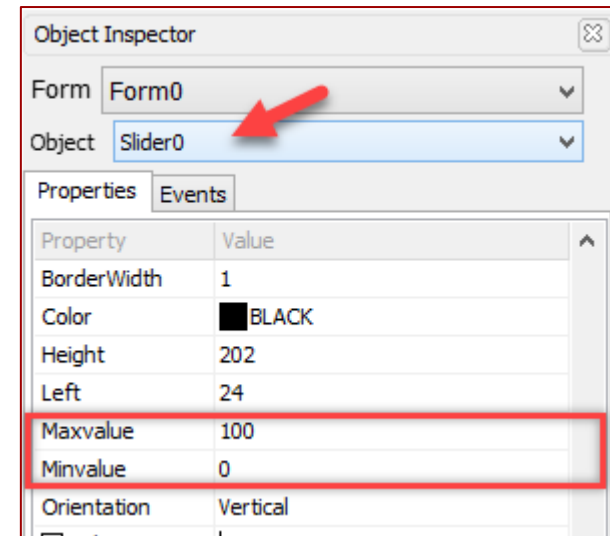
Set the background color of the form back to black by following the procedure shown earlier.

Add and Configure an Input Object

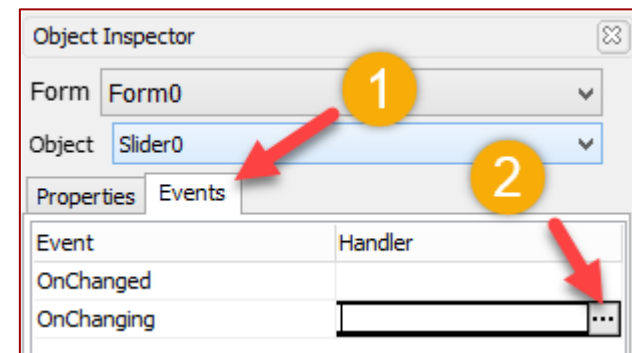
Input objects such as sliders can be used to control information that will be displayed on gauge objects. Add a slider to the project.

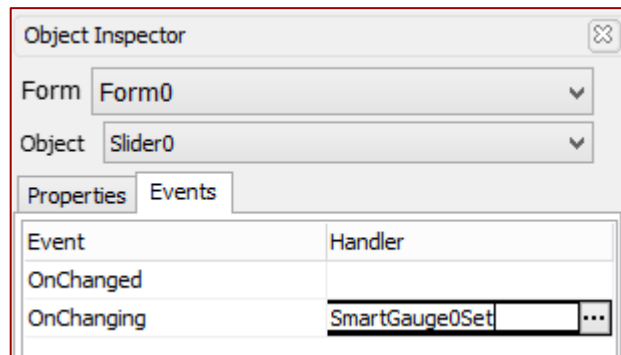
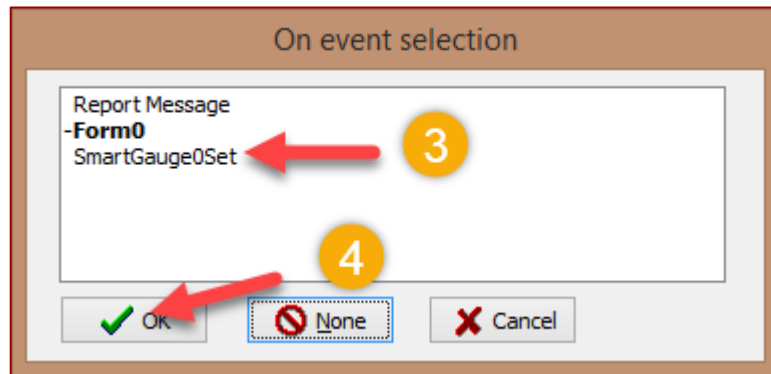


Make sure that the **Maxvalue** and **Minvalue** parameters of the slider are set as shown below.



Configure the OnChanging event handler of the slider.





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

- [ViSi-Genie Getting Started - First Project for Diablo16 Display Modules](#)
- [ViSi-Genie Getting Started - First Project for Picaso Displays](#)
- [ViSi-Genie Getting Started - First Project for Pixxi Display Modules](#)

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.