



Smart Widgets: Circular Progress Bar

DOCUMENT DATE: **20th FEBRUARY 2020**
DOCUMENT REVISION: **1.4**

WWW.4DSYSTEMS.COM.AU



Description

This application note shows how to create custom circular progress bar for Picaso, Diablo16 and Pixxi display modules.

Before getting started, the following are required:

Hardware

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

Software

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

This application note comes with one (1) ViSi-Genie project:

- C_ProgressBar_R_1_03.4DGenie

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

Content

Description 2

Content 2

Application Overview..... 3

Setup Procedure 4

Create a New Project 4

Design the Project..... 4

Add a Smart Gauge Object.....5

Open the Smart Widgets Editor Tool5

Add a Face or Base Image.....7

Add a Manipulated Image to Layer 18

Rotation Point of an Image9

Location of an Image in the Working Area..... 10

Change the Rotation Point of the Image on Layer 1 10

Center the Reference Point on the Working Area..... 11

Set the Min and Max Angles of Rotation 12

The Need for Multiple Frames 13

Set the Minimum and Maximum Values 14

Check the Generated Frames 15

Technique for Showing and Hiding Widget Parts 16

Add a New Layer and Swap Contents..... 19

Add Two Static Images to Layer 1.....20

Link Layer 1 and Layer 221

Add a Background Image to Layer 3.....22

Rearrange the Layers with Respect to the Base Image23

Add a Numeric Part to Layer 224

Add a Special Character to a Numerical Part.....26

Display Negative Values.....27

Add and Configure an Input Object29

Run the Program 30

Proprietary Information 31

Disclaimer of Warranties & Limitation of Liability 31

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 circular progress bar 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**

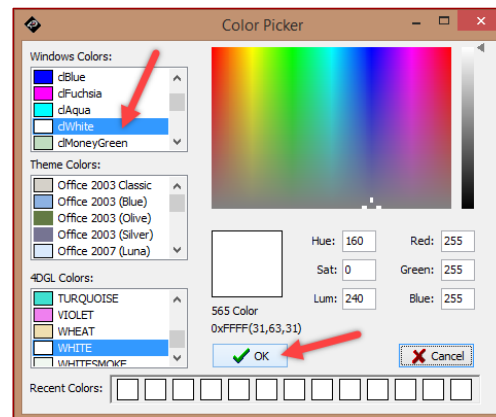
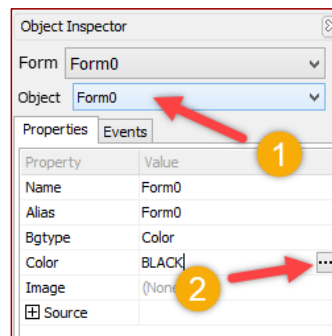
Create a New Project

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**

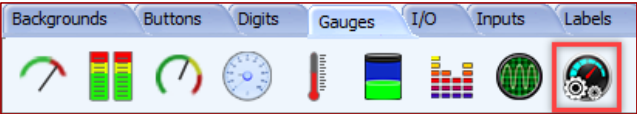
Design the Project


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

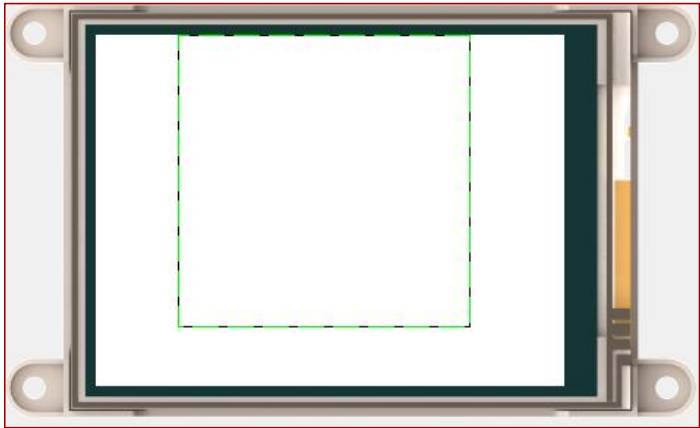


Add a Smart Gauge Object

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




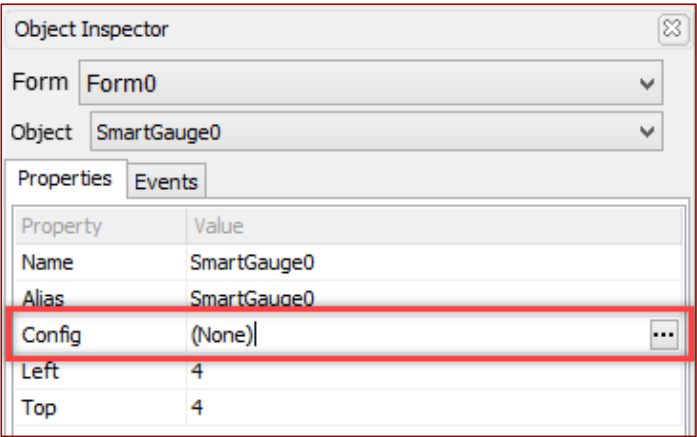
Simply click on this icon  to select it. Then place it on the WYSIWYG area.



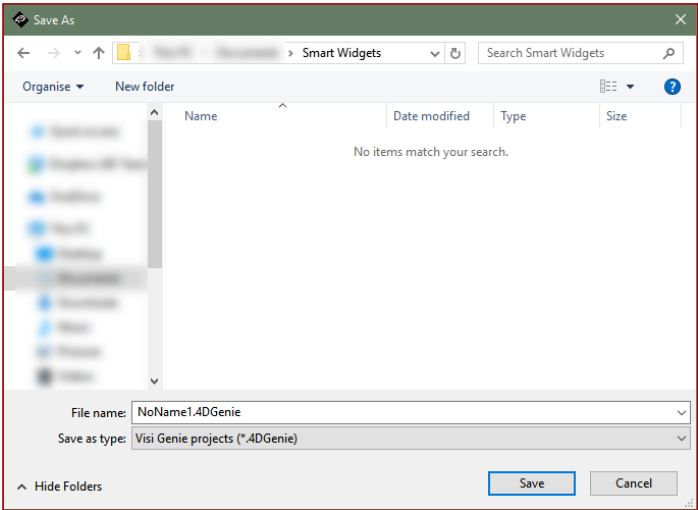
As displayed on the previous image, the widget appears empty when placed in the WYSIWYG area.

Open the Smart Widgets Editor Tool

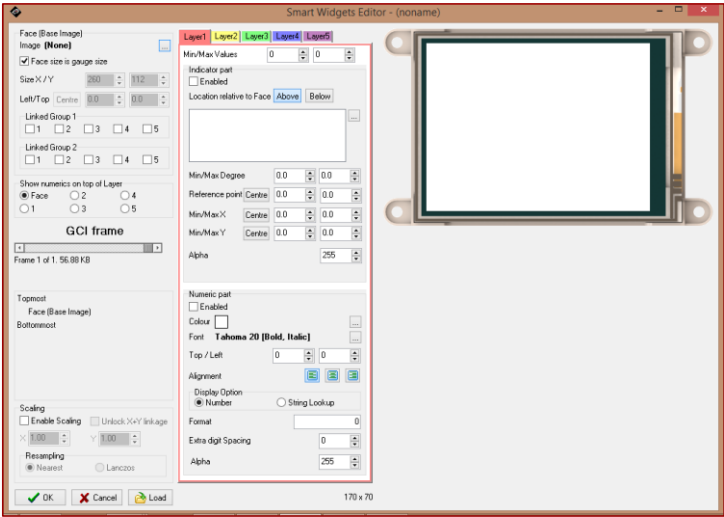
Open the Smart Widgets Editor tool by clicking on  of **Config** in the Object Inspector Properties tab.



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.



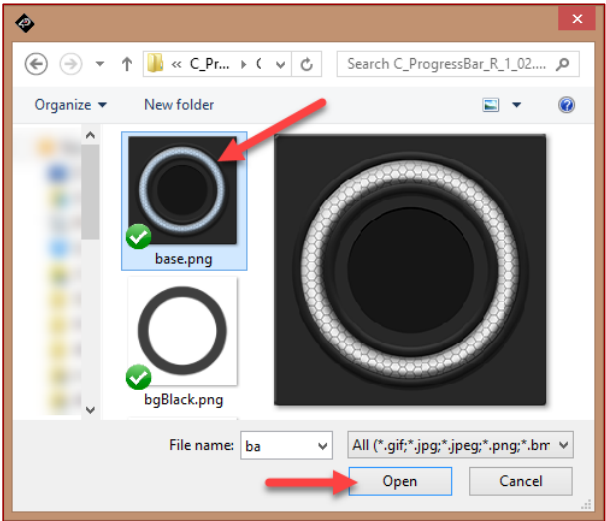
As shown in the image, this tool has a lot of parts. The next steps will focus only on the minimum tool functionalities required to make a basic circular progress bar.

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

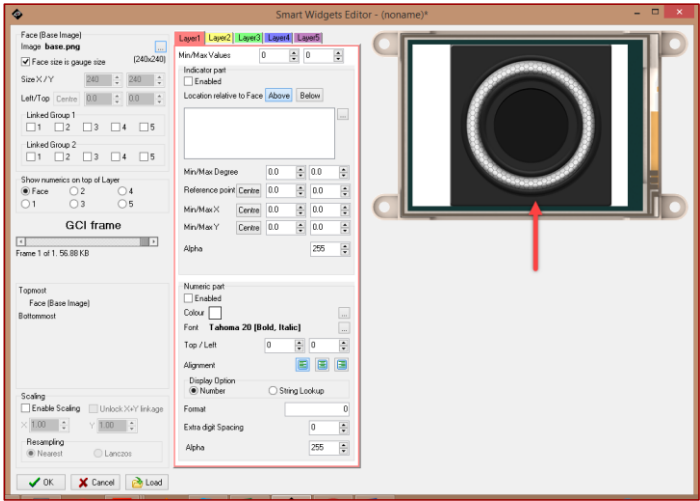
Commented [A1]: <http://4dlabs.com.au/4DLS-MN-SE-UG.html>
Commented [A2]: <http://4dlabs.com.au/4DLS-MN-SE-UG.html>

Add a Face or Base Image

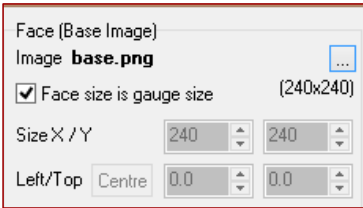
The first step when creating a smart widget is to select the face (base image). The face or base image is a static part of the widget. Click [...] to select an image.



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



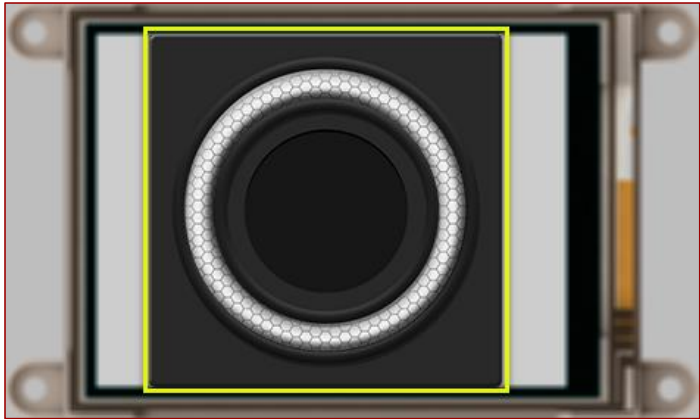
Note that the base image has a transparent part. Also, note that, by default, the area occupied by the base image is the working area.



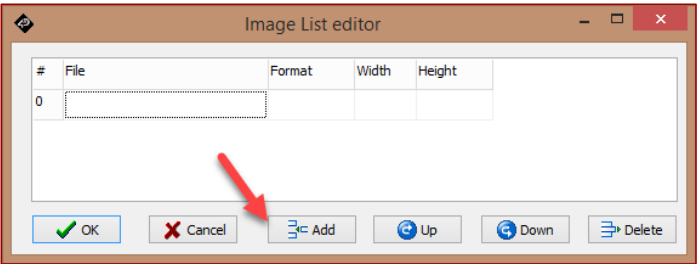
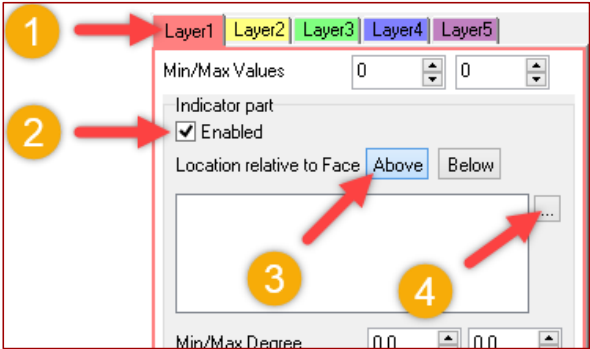
Add a Manipulated Image to Layer 1

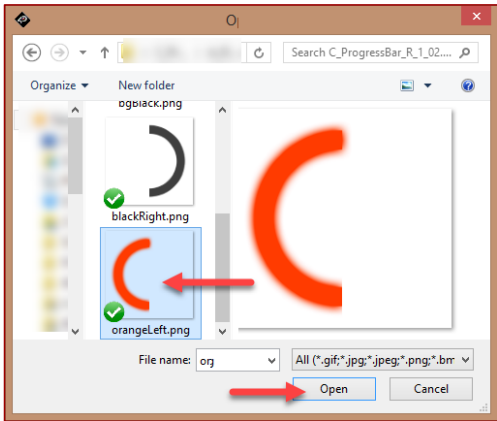
An image is needed to be used as a moving part that fills up the transparent part of the widget. To implement this, we need a layer containing a manipulated image. We will use Layer 1 for this purpose.

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



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



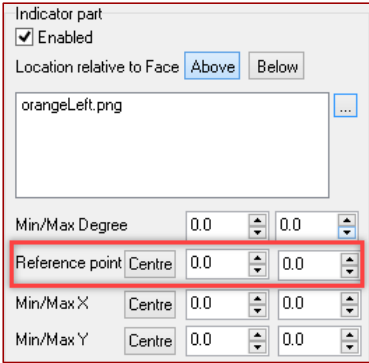


The image on Layer 1 now appears in the working area.



Rotation Point of an Image

The rotation point of an image is a point on the image about which it is rotated. The coordinates of this point are set using the *Reference point* parameters.

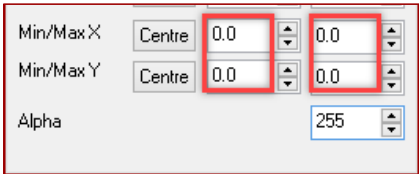


Furthermore, these coordinates are relative to the top-left corner of the image. Therefore, as can be seen above, the default reference and rotation point of an image is also its top-left corner.



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.

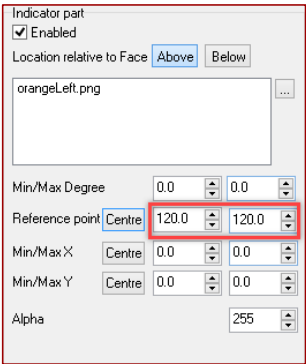


These coordinates are relative to the top-left corner of the working area. Smart Widgets Editor uses the reference point of an image when it places the image in the working area. As can be seen above, the values of the *Min/Max X and Y* parameters are set to 0 by default. This means that, by default,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

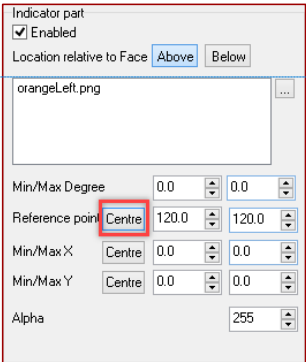
For a detailed discussion on reference points and their location in the working area, refer to the document [Smart Widgets Editor User Guide](#).

Change the Rotation Point of the Image on Layer 1

Obviously, the image on Layer 1 needs to be rotated at its center. For this to happen, set the *Reference point* coordinates to the values indicated below.

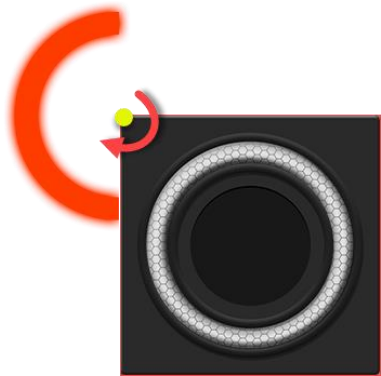


This can also be achieved by clicking on the *Centre* button of the *Reference point* parameters.



Commented [A3]: <http://4dlabs.com.au/4DLS-MN-SE-UG.html>
Commented [A4]: <http://4dlabs.com.au/4DLS-MN-SE-UG.html>

Now note that the image on Layer 1 disappeared from the working area. This is because it is currently outside the working area. To illustrate:



Center the Reference Point on the Working Area

Now move the location of the reference point of the image on Layer 1 to the center of the working area by setting the **Min/Max X and Y** parameters to the values indicated below.

Reference point	Centre	120.0	120.0
Min/Max X	Centre	120.0	120.0
Min/Max Y	Centre	120.0	120.0
Alpha		255	

The same can be accomplished by pressing the **Centre** buttons of the **Min/Max X and Y** parameters.

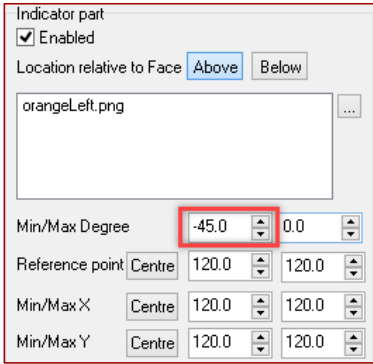
Min/Max X	Centre	120.0	120.0
Min/Max Y	Centre	120.0	120.0
Alpha		255	

In the working area, the image on Layer 1 is now visible.



Set the Min and Max Angles of Rotation

For demonstration purposes, set the minimum angle to **-45**, as shown below.



In the working area, the image on Layer 1 is rotated by 45 degrees, counter-clockwise, from its original position.



Now set the minimum angle to 45, as shown below.



In the working area, the image on Layer 1 is rotated by 45 degrees, clockwise from its original position.



The Need for Multiple Frames

At this point the widget has only one frame. This frame shows the image on Layer 1 rotated by a certain angle from its original position. Also, below Layer 1 is the base image. Our next objective is to be able to rotate the image on Layer 1 starting and ending at a certain position (the reason for this will become apparent later). Each state of the widget while the image on Layer 1 is rotating will be represented by a frame. The widget, therefore, has to have multiple frames.

To illustrate, let's say we want to use 100 frames to represent the different states of the widget as the image on Layer 1 is rotated. The output will look like the following:

1st frame (frame index 0):



34th frame (frame index 33):



67th frame (frame index 66):



100th frame (frame index 99):



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

Min/Max Degree

-180.0

180.0

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.

Layer1Layer2Layer3Layer4Layer5

Min/Max Values

0

0

Indicator part

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 2](#).

Therefore, to make Smart Widgets Editor generate 100 frames, input the values indicated below to the **Min/Max Values** fields of Layer 1.



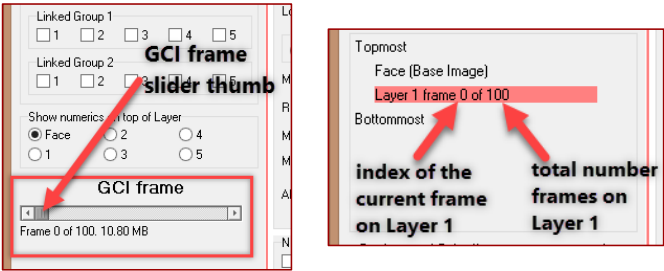
Below is a table showing six parameters of Layer 1 at this point:

Associated Value	Frame Index	Angle of Rotation	Rotation Point	X and Y Positions
0 (Min Value)	0	-180 (min angle)	120	120 (Min X and Min Y)
99(Max Value)	99	180 (max angle)	120	120 (Max X and Max Y)

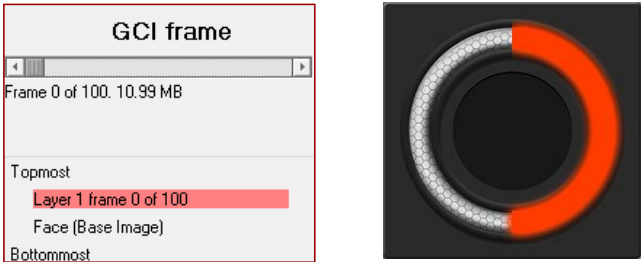
As can be seen in the table, the angle of rotation of the image on Layer 1 at frame index **0** or layer value **0** is **-180** degrees. The angle of rotation is **180** degrees at frame index **99** or layer value **99**. The rotation point is the center of the image. The **Min/Max X** and **Y** parameters remain constant. Essentially, therefore, the image on Layer 1 rotates clockwise from **-180** degrees to **180** degrees at its center as the frame index and layer value increase. Moreover, the rotation point of the image on Layer 1 remains fixed at the center of the working area at all frames. Smart Widgets Editor then generates all of the frames needed to satisfy this behavior of the image on Layer 1.

Check the Generated Frames

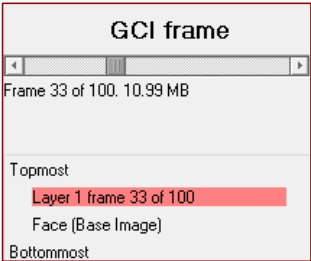
To see the generated frames, move the thumb of the GCI frame slider indicated in the image below. The layer guide section also shows the index of the current frame and the total number of frames in a layer.



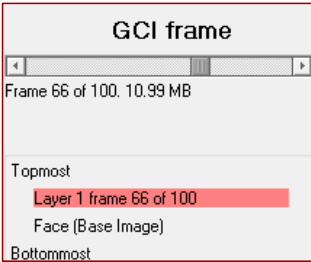
1st frame (frame index 0):



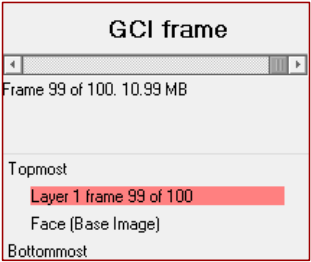
34th frame (frame index 33):



67th frame (frame index 66):



100th frame (frame index 99):



Technique for Showing and Hiding Widget Parts

The technique chosen to animate this widget is to hide the right-half side of the transparent part of the widget as the image on Layer 1 rotates from -180 degrees to 0 degrees. As soon as the image on Layer 1 reaches the 0 degree position, another image is added to cover the transparent part of the left-half side of the widget. At this point the image covering the right-half side of the transparent part of the widget is removed. The logic behind this will become more apparent in the subsequent sections.

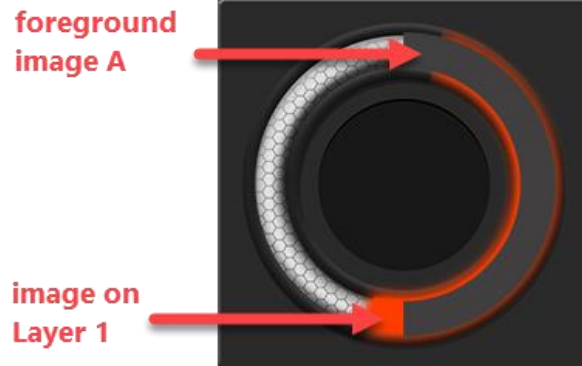
The table below shows how the complete rotation of the image on Layer 1 is divided.

Part	Angle of Rotation of Image on Layer 1	Frame Index Range
First	-180 to -1 degrees	0 to 49
Second	0 to 180 degrees	50 to 99

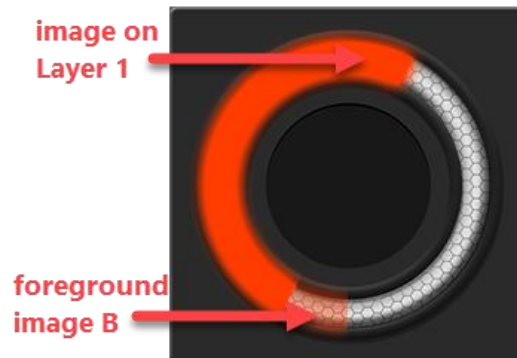
Two foreground images (image A and image B) are then used to cover the right-half and left-half sides of the transparent part of the widget, respectively. These images will be on top of Layer 1, so they will also cover the rotating image on Layer 1.

Part	Angle of Rotation of Image on Layer 1	Frame Index	Foreground Image
First	-180 to -1 degrees	0 to 49	A
Second	0 to 180 degrees	50 to 99	B (image A is removed)

Foreground image A on top of the image on Layer 1:

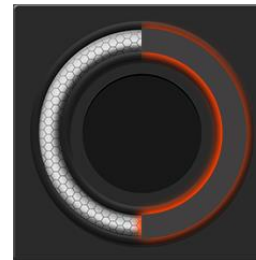


Foreground image B on top of the image on Layer 1 (image A is removed):



Note that foreground image B and the image on Layer 1 are identical. The alpha of foreground image B in the illustration above is reduced so the user can distinguish between the two. The intended outcome is shown by the following sequence of frames.

1st frame, frame index 0 (image on Layer 1 at -180 degrees, foreground image A on top):



34th frame, frame index 33 (image on Layer 1 at -60 degrees, foreground image A on top):



50th frame, frame index 49 (image on Layer 1 at -1 degree, foreground image A on top):



51st frame, frame index 50 (image on Layer 1 at 0 degrees, foreground image B on top, foreground image A is removed):



67th frame, frame index 66 (image on Layer 1 at 60 degrees, foreground image B on top):



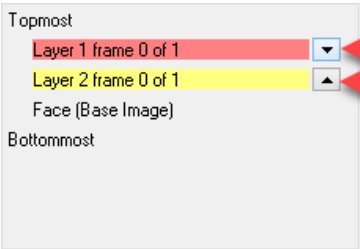
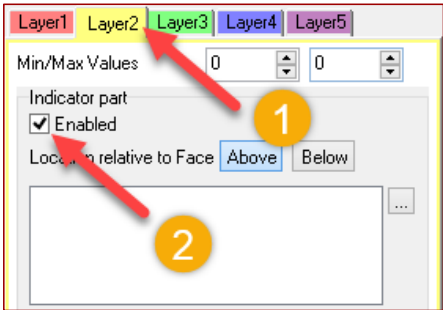
100th frame, frame index 99 (image on Layer 1 at 180 degrees, foreground image B on top):



To achieve the above, the widget has to be able to hide and show specific parts at a certain range of frame indices. This can be implemented using a layer containing static images.

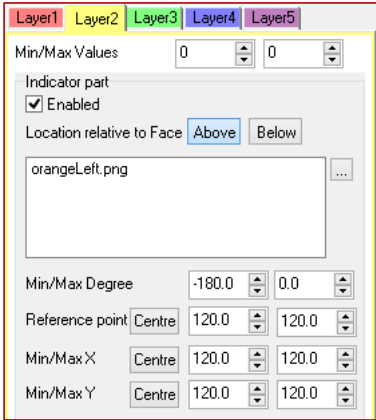
Add a New Layer and Swap Contents

A new layer is now needed on top of Layer 1. This layer will contain the foreground images A and B. However, since Layer 1 is always the topmost layer, its content must be moved to Layer 2. To do this, follow the procedure below.

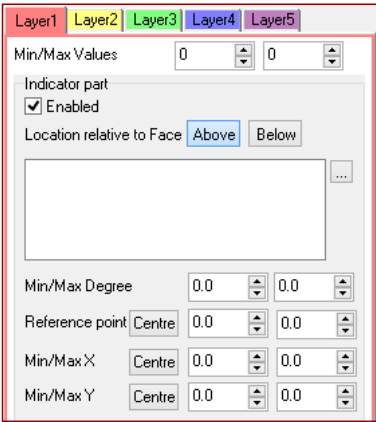


click any of these buttons

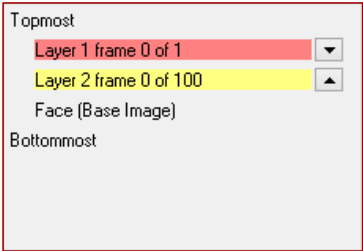
The rotating image originally on Layer 1 is now on Layer 2.



Layer 1 is now an empty layer.

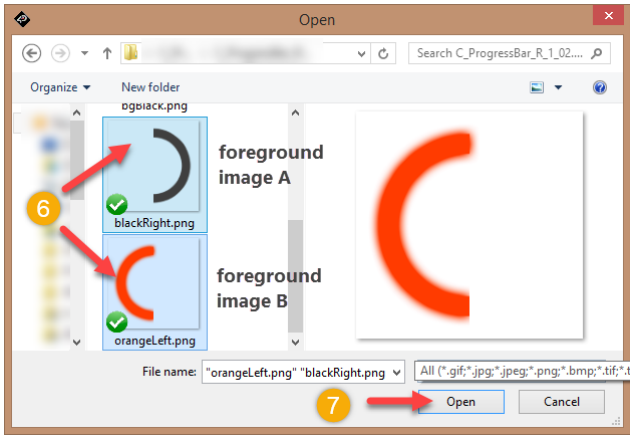
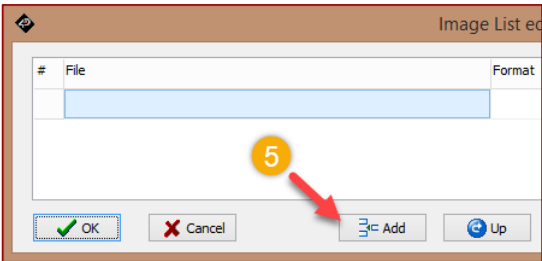
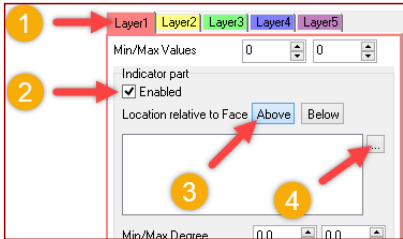


Note also that Layer 1 is on top of Layer 2.

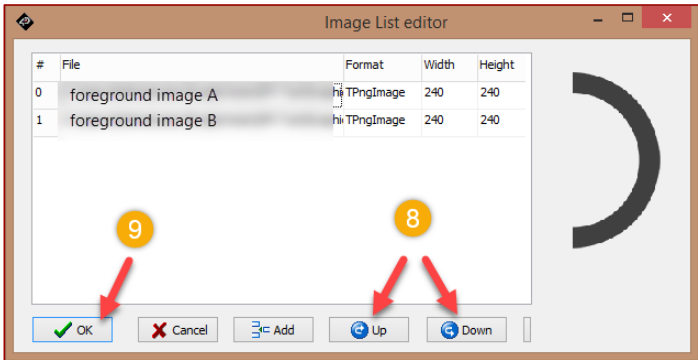


Add Two Static Images to Layer 1

The foreground images A and B are added as static images to Layer 1. The procedure for this is as follows.



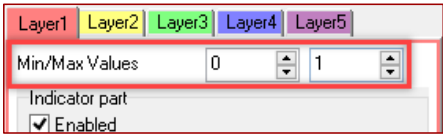
In the image list, make sure that image A is indexed before image B. Click on the **Up** or **Down** button to rearrange the images if necessary.



Foreground image A is now shown in the working area.

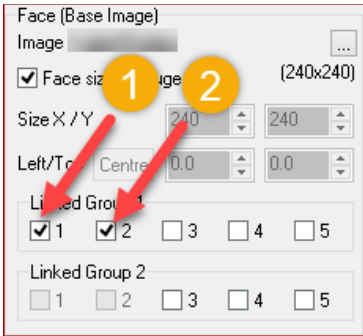


Also, set the Min/Max Values parameters for Layer 1 as follows:



Link Layer 1 and Layer 2

To review, the objective is that foreground image A on Layer 1 must be shown while the image on Layer 2 rotates from **-180** to **-1** degree (frame 0 to 49). Similarly, foreground image B on Layer 1 must be shown while the image on Layer 2 rotates from **0** to **180** degrees (frame 50 to 99). Also, foreground image A is removed from the working area from frame 50 to 99. To implement this, Layer 1 and Layer 2 must be linked. The procedure for this is shown below.

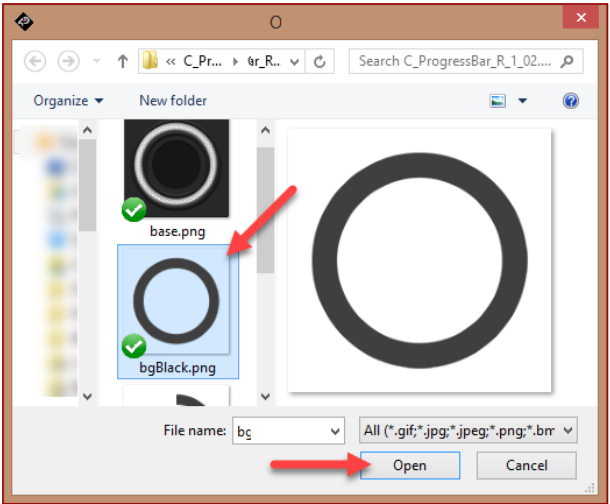
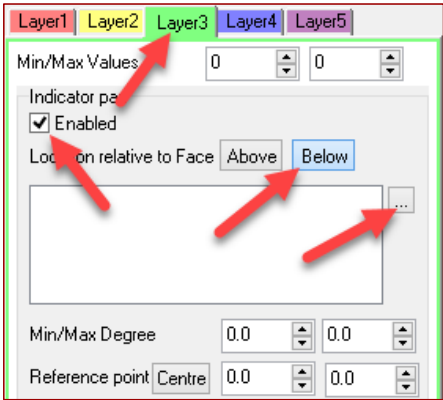


For more information on linking of layers, refer to the [Smart Widgets Editor User Guide](#).

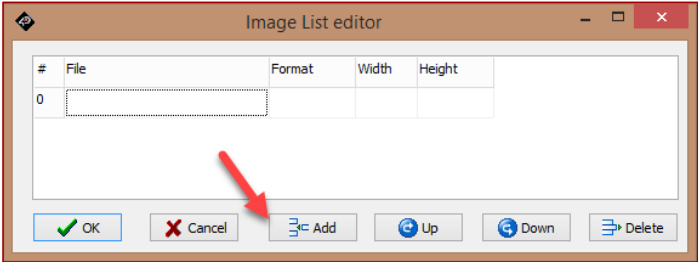
Commented [A5]: <http://4dlabs.com.au/4DLS-MN-SE-UG.html>

Add a Background Image to Layer 3

Let us add a background image to cover the transparent part of the base image that is being filled. This background image will reside on Layer 3. Follow the procedure shown below.

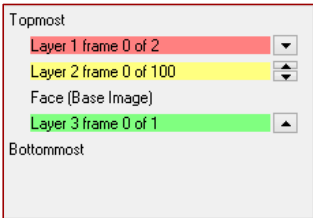


The image on Layer 3 now appears in the working area.

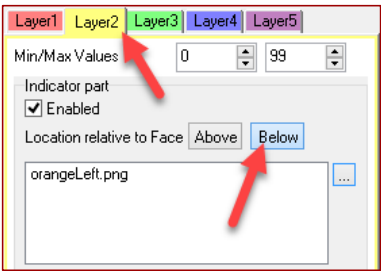
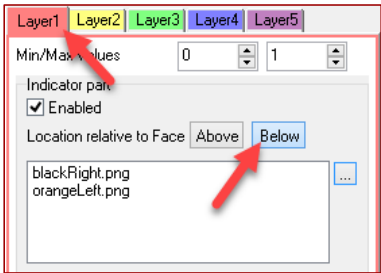


Rearrange the Layers with Respect to the Base Image

At this point, the arrangement of the layers is as follows:



For best aesthetic effect, Layer 1 and Layer 2 must be below the base image. To implement this, follow the procedure below.



The appearance of the widget is now updated.

1st frame (frame index 0):



34th frame (frame index 33):



67th frame (frame index 66):

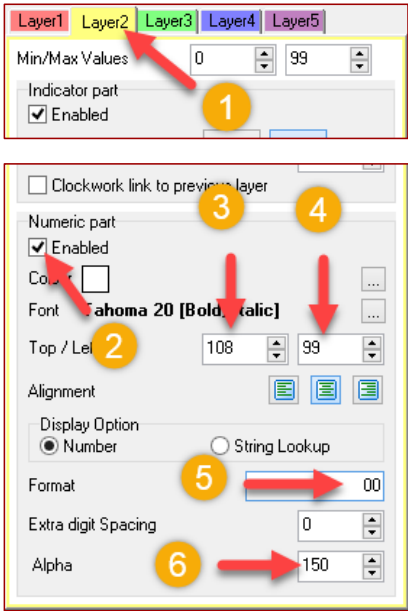


100th frame (frame index 99):



Add a Numeric Part to Layer 2

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 2, follow the procedure shown below.



The numerical part on Layer 2 can now be seen in the working area.

1st frame (frame index 0):



67th frame (frame index 66):



34th frame (frame index 33):

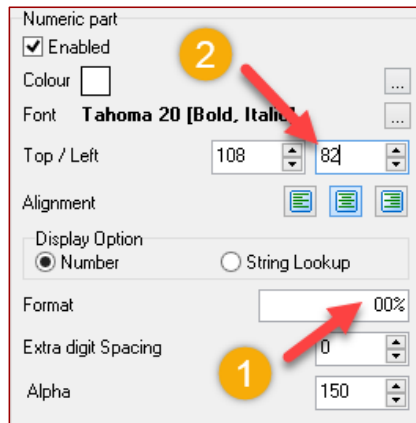


100th frame (frame index 99):



Add a Special Character to a Numerical Part

It is also possible to display a special character on the numerical part of a layer. This is implemented by inserting the special character to the **Format** field of the Numeric part section. For instance, to insert a percent symbol to the numerical part on Layer 2, the settings below could be used.



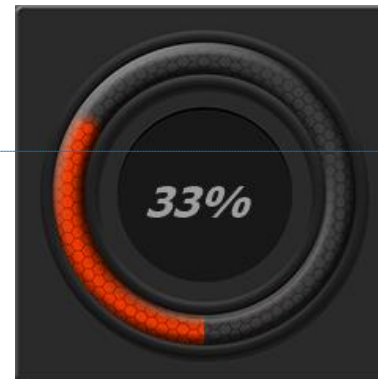
For more information on the different properties of the numerical part of a layer, refer to the document [Smart Widgets Editor User Guide](#).

The output would then look like the following.

1st frame (frame index 0):



34th frame (frame index 33):



Commented [A6]: <http://4dlabs.com.au/4DLS-MN-SE-UG.html>

Commented [A7]: <http://4dlabs.com.au/4DLS-MN-SE-UG.html>

67th frame (frame index 66):

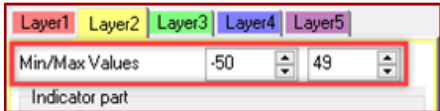


100th frame (frame index 99):



Display Negative Values

In some applications it is necessary for a widget to display negative values. To illustrate this on Layer 2, 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 100. The table below shows the frame index numbers and the associated values.

Frame Index	Associated Value
0	-50 (Min Value)
33	-17
66	16
N - 1	49 (Max Value)

The numerical part on Layer 2 is now updated.

1st frame (frame index 0):



67th frame (frame index 66):



34th frame (frame index 33):



100th frame (frame index 99):



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



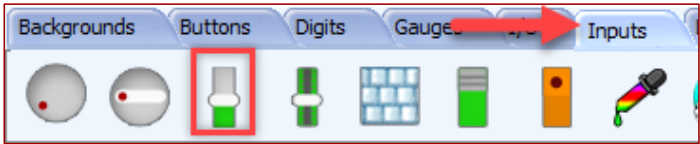
The circular progress bar will now appear on the WYSIWYG area of the project.



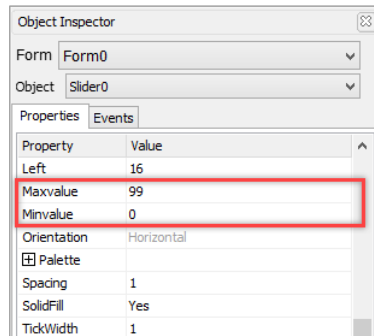
Set the background color of the form back to black by following the procedure shown earlier.

Add and Configure an Input Object

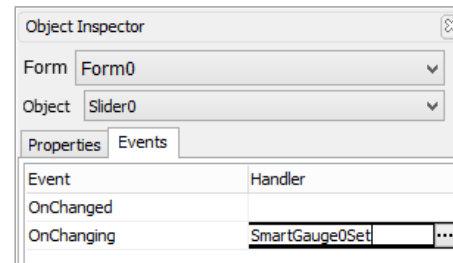
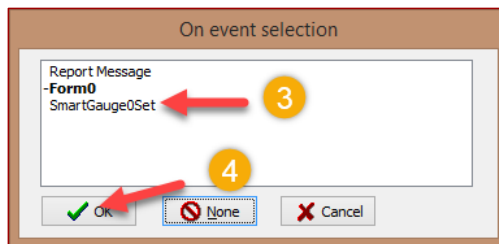
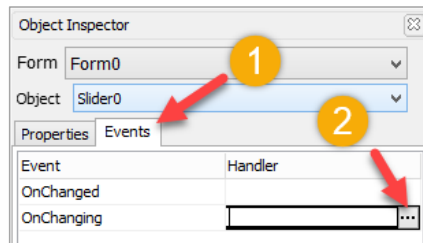
Input objects such as sliders can be used to control information that will be displayed on gauge objects. Add a slider to the project.



Set the **Maxvalue** and **Minvalue** as shown below.



Configure the OnChanging event handler of the slider as shown below.



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.