



## Smart Widgets: Slide Menu

DOCUMENT DATE: **27<sup>th</sup> FEBRUARY 2020**  
DOCUMENT REVISION: **1.2**

[WWW.4DSYSTEMS.COM.AU](http://WWW.4DSYSTEMS.COM.AU)



Description

This application note shows how to create custom slide menu for Picaso and Diablo16 touch screen 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:

- SlideMenu.zip

**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 ..... 3**

**Create a New Project ..... 3**

**Design the Project..... 4**

*Add a Smart Slider Object.....4*

*Open the Smart Widgets Editor Tool.....5*

*Add a Face or Base Image.....6*

*Add a Manipulated Image to Layer 1.....7*

*Initial and Final Positions of Layer1 .....8*

*Add Userimages Object .....9*

*Use Userimage as Button ..... 10*

*Add Destination Forms..... 11*

**Run the Program.....11**

**Proprietary Information .....12**

**Disclaimer of Warranties & Limitation of Liability .....12**

## 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**

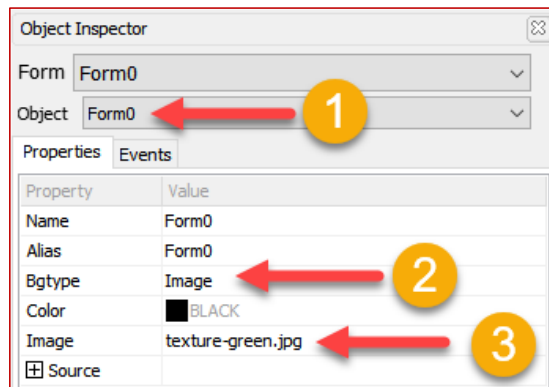
## Create a New Project

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-43DT will be used for the project. Same procedure is applicable for any Picaso and Diablo16 displays. First, set the background image of Form0.



## Add a Smart Slider Object

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




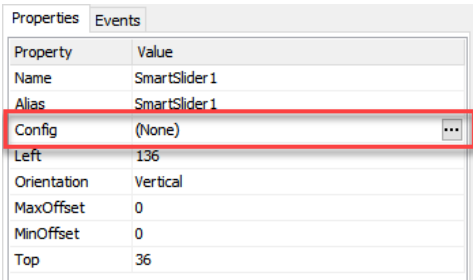
Simply click on this icon  to select it. Then place it on the WYSIWYG area.



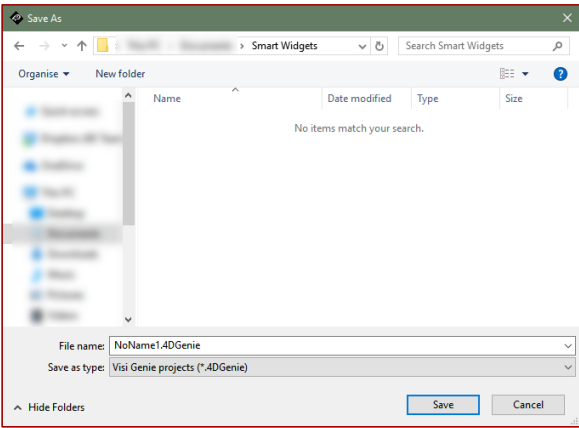
As displayed on the previous image, the widget appears empty when placed in the WYSIWYG area.

Open the Smart Widgets Editor Tool

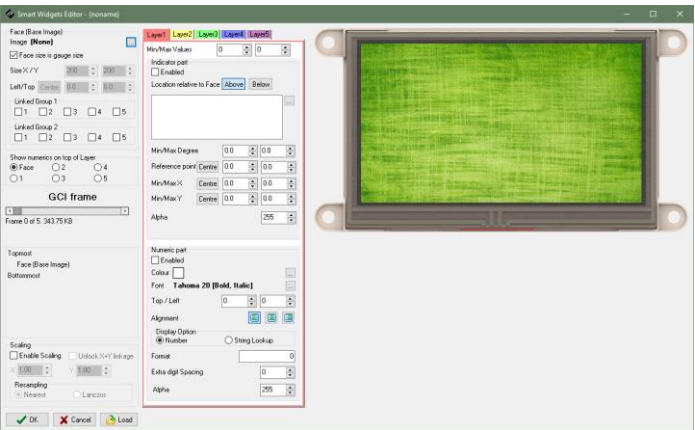
Open the Smart Widgets Editor tool by clicking on  of **Config** in the Object Inspector Properties tab.



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://4dsystems.com.au/4DLS-MN-SE-UG.html>

Add a Face or Base Image

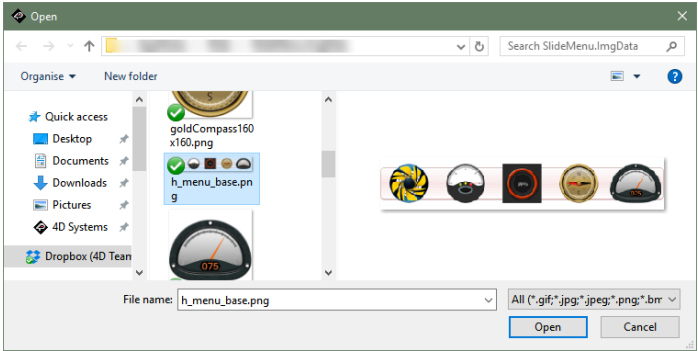
The first step when creating a smart widget is to select the face (base image). The face or base image is a static part of the widget. Click [...] to select an image.



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.



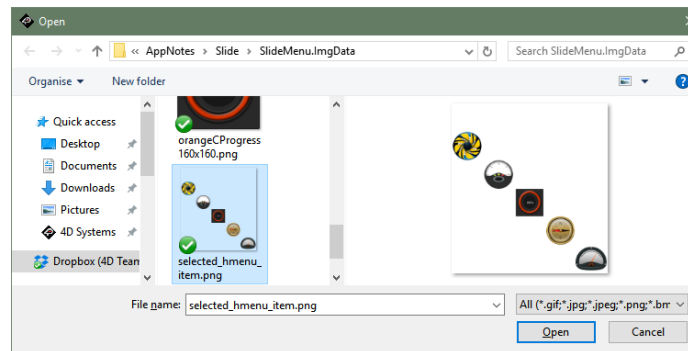
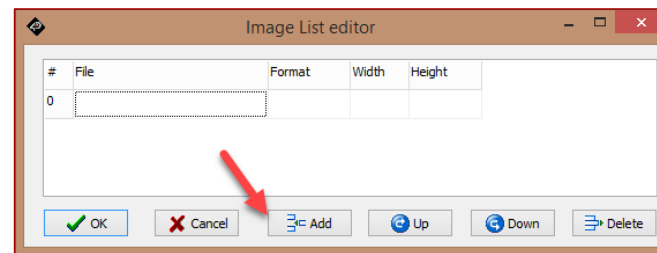
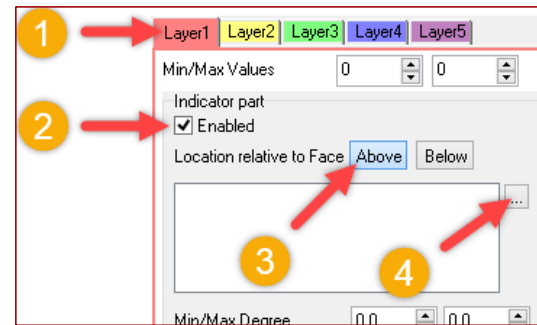
Any part of any image used in a smart widget will not be displayed if outside the working area.



### Add a Manipulated Image to Layer 1

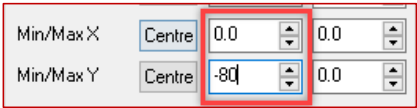
An image is needed to be used as a moving part that fills up the transparent part of the widget. To implement this, we need a layer containing a manipulated image. We will use Layer 1 for this purpose.

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



Initial and Final Positions of Layer1

Set the initial position of the manipulated image using MinX and MinY.



Notice that the first icon is bigger than it used to be.



Before setting the final position, it is important to take note that the layer should first have a range of values.

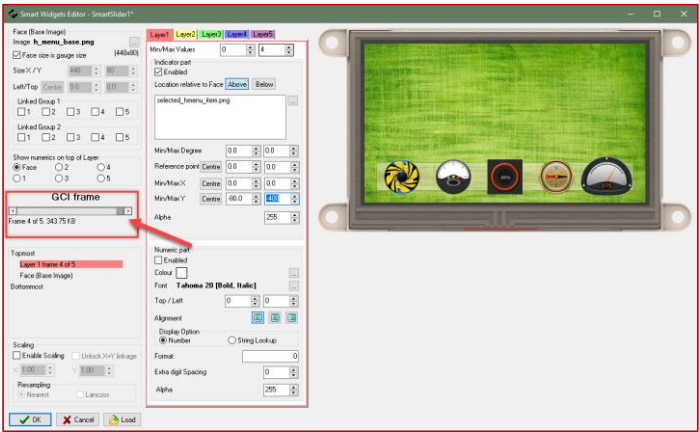


In this case, since there are 5 possible outcomes for the menu, the Min/Max Values is set to 0 to 4.

Set the initial position of the manipulated image using MaxX and MaxY.



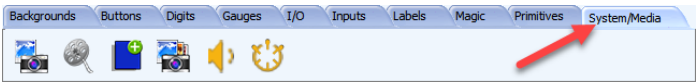
You can check the frames of the slider menu using the GCI frame slider on the left part of the editor.





Add Userimages Object

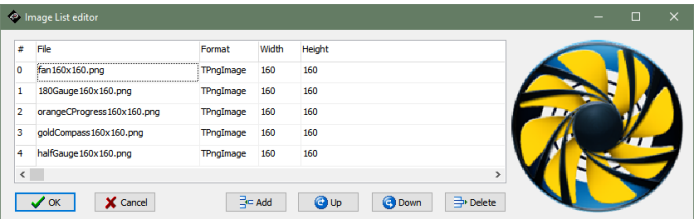
The slide menu designed previously was only for selecting an item from the menu. We can link this Slider to a Userimages object.



Add a Userimages object to the project.



Add the icons/images that represents the item in the slide menu.



Be sure that the Slide Menu and the Userimages object follow the same order.

## Use Userimage as Button

Userimages are non-input objects by default. For this step, 4DGL code will be added to the project.



Touch functionality should be enabled for the Userimages object when currently at the form containing it and disabled if not. For that add a **Magic Code** to the project.



Set the insert point to be **PostActivateForm**

Property	Value
Name	MagicCode0
Alias	MagicCode0
Code	MagicCode0.inc
InsertPoint	PostActivateForm

Open the editor by clicking on  of the row **Code**

Then add the following lines of code.

```
if (CurrentForm == 0)
    img_ClearAttributes(hndl, iUserimages0, I_TOUCH_DISABLE);
else
    img_SetAttributes(hndl, iUserimages0, I_TOUCH_DISABLE);
endif
```

Finally, we need to program what it will do when a touch event occurs. Add **Magic Touch**, **Magic Release** and **Magic Move** objects to the project.

Then add the following lines of code to each of the three.

```
if (ImageTouched == iUserimages0)
    ImageTouched := -1;
endif
```

Open Magic Touch object add these lines of code inside the **if** condition.

```
var form;
form := img_GetWord(hndl, iUserimages0, IMAGE_INDEX) + 1;
ActivateForm(form);
```

### Add Destination Forms

The previous code would read the current frame of the Userimages object then navigate to the corresponding form. Since the Userimages object has 5 frames, the project should have 5 additional forms as destinations.



In this appnote project, the destination forms only included a Statictext to show the current form number and a Winbutton to navigate back to the Initial form.

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.

### 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.