



Smart Widgets: Horizontal Slider

DOCUMENT DATE: 9th MAY 2020
DOCUMENT REVISION: 1.0



Description

This application note shows how to create custom horizontal slider for Picaso, Diablo16 and Pixxi 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 Pixxi28/44
 - o Picaso
- [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.

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.....	3
<i>Add Smart Slider to Project.....</i>	<i>4</i>
<i>Open Smart Widgets Editor.....</i>	<i>4</i>
<i>Open Smart Widgets Editor.....</i>	<i>5</i>
<i>Adding Main Indicator</i>	<i>6</i>
<i>Setting Up Main Indicator Parameters.....</i>	<i>7</i>
<i>Adding Secondary Indicator</i>	<i>9</i>
<i>Setting up the Smart Slider</i>	<i>11</i>
<i>Adding Gauge for Output</i>	<i>12</i>
Run the Program.....	13
Proprietary Information	14
Disclaimer of Warranties & Limitation of Liability	14

Application Overview

The Smart Widgets Editor 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 any of the following application notes:

- [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 any of the following application notes:

- [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

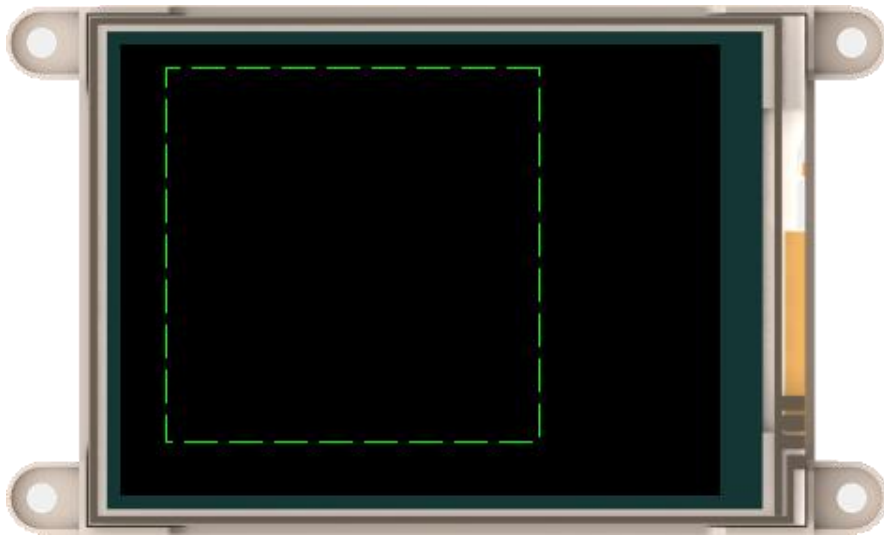
For this application, gen4-uLCD-32DT will be used for the project. Same procedure is applicable for any Picaso, Diablo16 and Pixxi touch screen displays.

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.



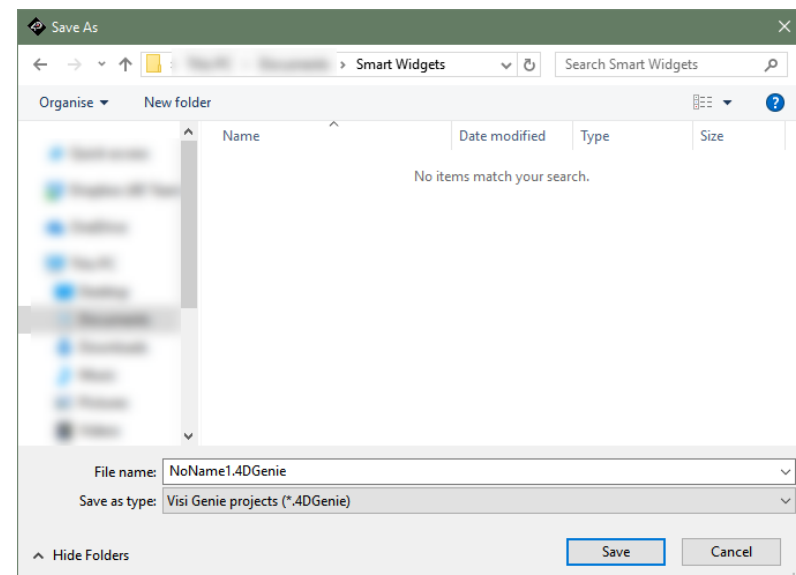
As displayed on the image, the widget appears empty when placed in the WYSIWYG.

Open Smart Widgets Editor

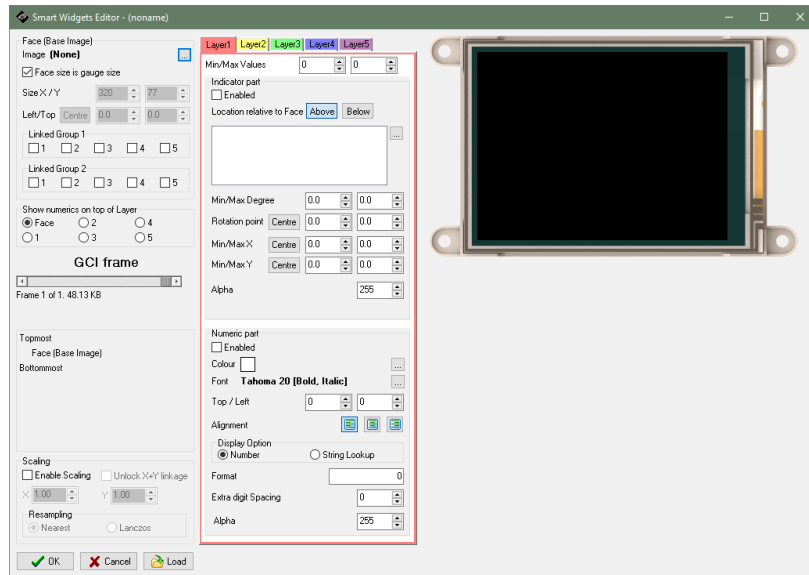
Open the Smart Widgets Editor tool by clicking on  of **Config** in the Object Inspector Properties tab.

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.




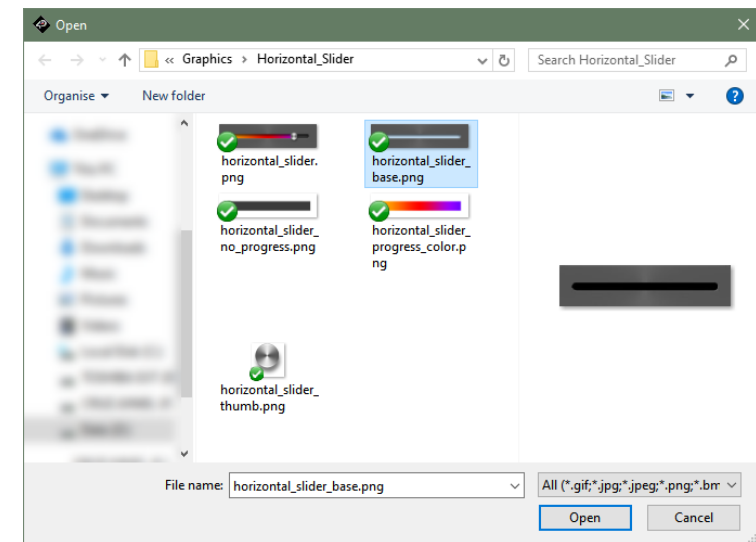
As shown in the image, this tool has a lot of parts. The next steps will focus only on the minimum tool functionalities required to make a basic slider. For detailed discussion on how each part works, please refer to the **Smart Widgets Editor User Guide**.

Open Smart Widgets Editor

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.



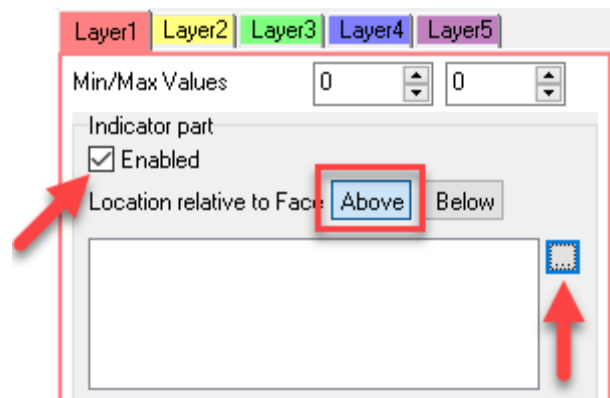
Any part of any image used in a smart widget will not be displayed if outside this area.

Adding Main Indicator

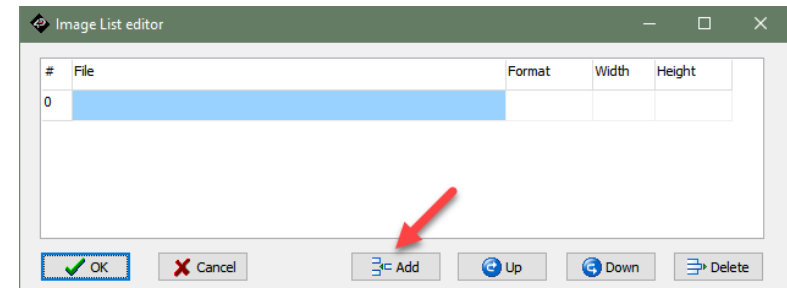
After selecting the image, a slider needs a thumb and an optional indicator.

In this project, Layer 1 will manipulate the thumb while Layer 2 will manipulate the other indicator. Another layer (Layer 3) will contain a still image.

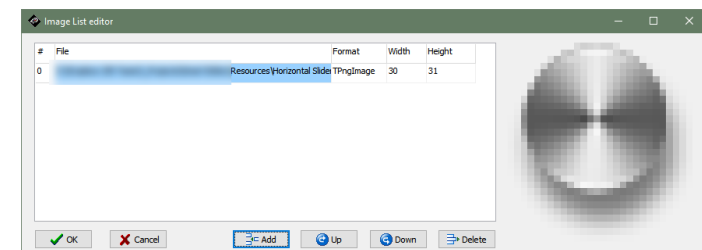
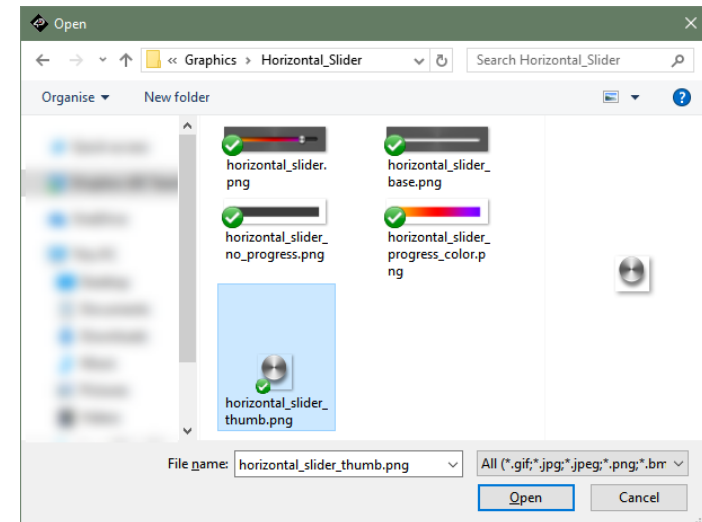
Enable Layer 1 and select the image.



Obviously the thumb should be above the base image.



Select the image and click **Open**



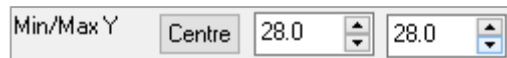
You'll see the knob at the top left of the base image.



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

Setting Up Main Indicator Parameters

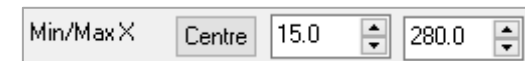
First, we want the knob to be centered vertically all the time.



For this slider, the **Min/Max Y** needs to be both at 28. This will make the Y position of the knob constant from minimum value to maximum value.

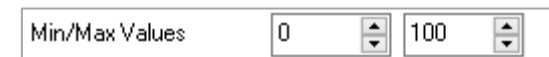


Now, to create frames that moves the thumb horizontally, an initial X position (**Min X**) and final X position (**Max X**) must both be specified.



In this case, the thumb moves from 15 to 280.

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 15 to 280 X positions.

At the initial X position at 15:



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



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

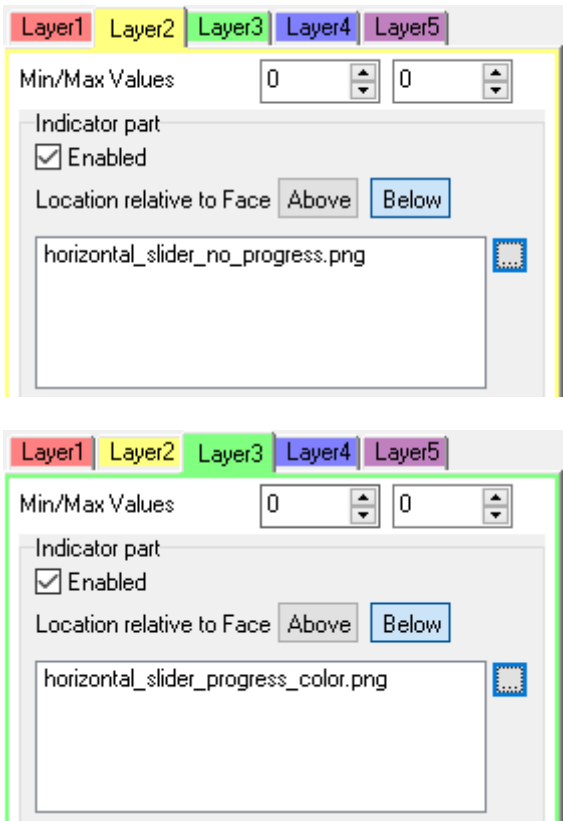


At this state, this slider is already sufficient to be used in the project. However, in this application note project, additional layers will be used to for better design.

Adding Secondary Indicator

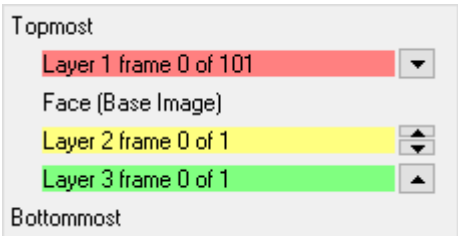
Next, add another indicator for the slider. This time with two additional layers.

Enable and select the images the same way as Layer 1.



Both Layers will be below the Face (base image)

Layer 1	horizontal_slider_thumb.png
Face	horizontal_slider_base.png
Layer 2	horizontal_slider_no_progress.png
Layer 3	horizontal_slider_progress_color.png



Layer 3 will serve as a stationary layer which will be covered by Layer 2.

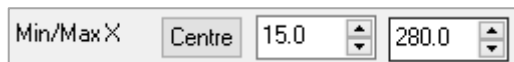


Layer 2 is the gray area covering the middle track.

To check Layer 3, temporarily disable Layer 2.

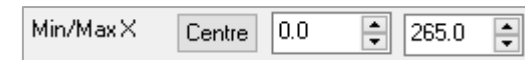


Layer 2 is simply covering Layer 3, therefore if Layer 2 is moved, part of Layer 3 will be shown.

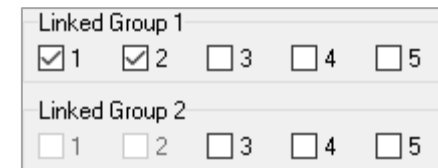


Remember that the thumb moves from the above X position. That is a 265-pixel displacement.

For Layer 2, it should have the same displacement (**MaxX – MinY**) 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 will be the output:

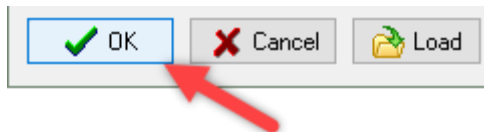
- At 26th frame (index 25)



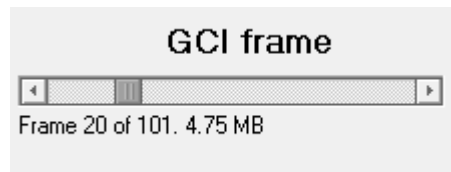
- At 101st frame (last frame / index 100)



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



The slider can then be tested at any desired position by using the GCI frame slider below.



Setting up the Smart Slider

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

Property	Value
Name	SmartSlider0
Alias	SmartSlider0
Config	(None)
Left	0
Orientation	Vertical
MaxOffset	0
MinOffset	0
Top	163

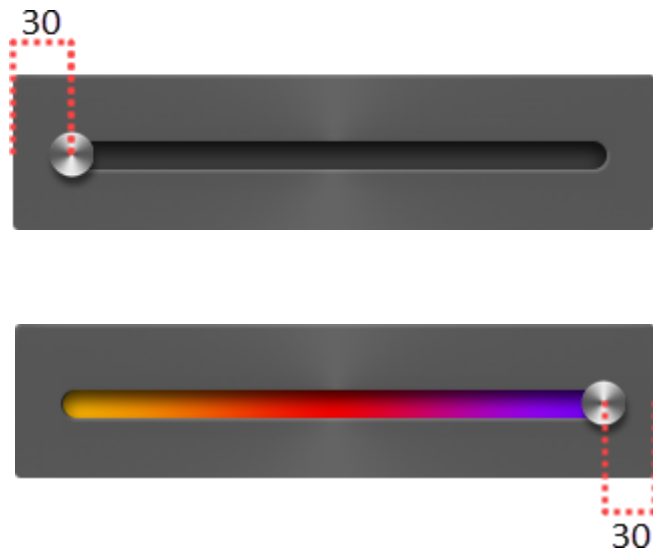
On the Object Inspector's Properties tab, **Orientation**, **MaxOffset** and **MinOffset** needs to be provided by the user.

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

As for the other two, this project uses:

Property	Value
Name	SmartSlider0
Alias	SmartSlider0
Config	SmartSlider0
Left	0
Orientation	Horizontal
MaxOffset	30
MinOffset	30
Top	163

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.

Adding Gauge for Output

Gauges are great options to present information. Add a Coolgauge to the project.

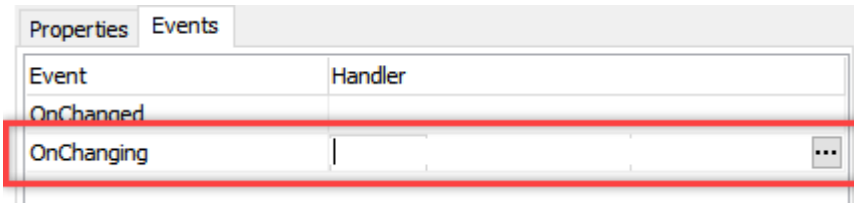


Simply click on this icon  to select it. Then place it on the WYSIWYG area.

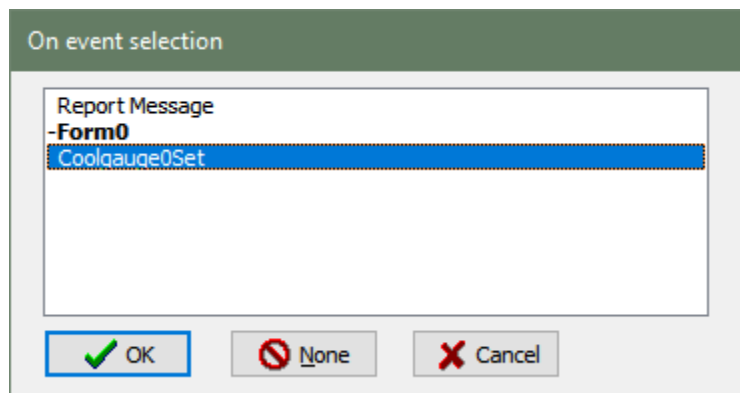


It is important that the Coolgauge has the same number of frames as the slider. With values 0 to 100 for the Coolgauge, they have same number of frames.

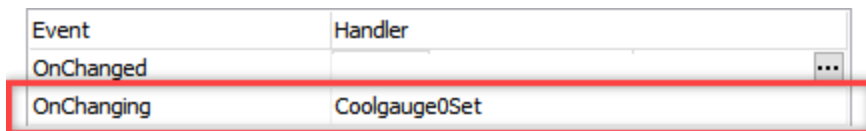
Then for the Slider, under Events tab of the Object Inspector:



Set the event for **OnChanging**:



Select **Coolgauge0Set** 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

- [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.