APPLICATION NOTES

4D SYSTEMS
TURNING TECHNOLOGY INTO ART

ViSi Form Switch

DOCUMENT DATE: **22nd April 2019**
DOCUMENT REVISION: **1.1**

Description

This application note is intended to demonstrating to the user on how to switch forms in ViSi.

Before getting started, the following are required:

- Any of the following 4D Picaso display modules:

uLCD-24PTU gen4-uLCD-24PT	uLCD-28PTU gen4-uLCD-28PT	uVGA-III gen4-uLCD-32PT

  and other superseded modules which support the ViSi environments.

- The target module can also be a Diablo16 display

gen4-uLCD-24D Series gen4-uLCD-35D Series gen4-uLCD-70D Series uLCD-35DT	gen4-uLCD-28D Series gen4-uLCD-43D Series uLCD-43D Series	gen4-uLCD-32D Series gen4-uLCD-50D Series uLCD-70DT

Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor.

- 4D Programming Cable / µUSB-PA5/µUSB-PA5-II
  for non-gen4 displays (uLCD-xxx)
- 4D Programming Cable & gen4-IB / gen4-PA / 4D-UPA,
  for gen-4 displays (gen4-uLCD-xxx)
- micro-SD (µSD) memory card
- Workshop 4 IDE (installed according to the installation document)
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Application Overview

The main focus of this application note is to demonstrate a method on how to switch forms in ViSi. There are many factors to consider in switching from one form to another. In this application documentation, a ViSi based project that includes two forms with different objects and background color was introduced.

Setup Procedure

For instructions on how to launch Workshop 4, how to open a **ViSi** project, and how to change the target display, kindly refer to the section "**Setup Procedure**" of the application note

ViSi Getting Started - First Project for Picaso and Diablo16

Create a New Project

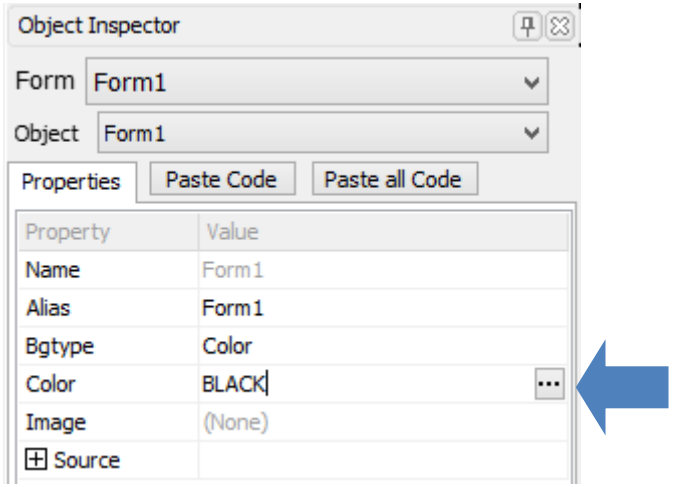
For instructions on how to create a new **ViSi** project, please refer to the section "**Create a New Project**" of the application note

ViSi Getting Started - First Project for Picaso and Diablo16

Design the Project

Change Background Color of Form1

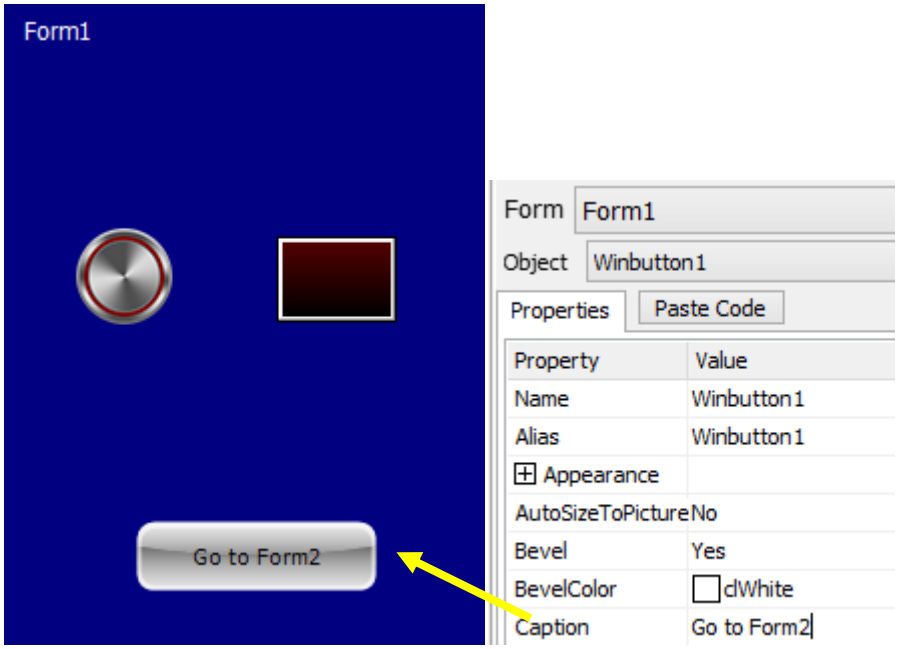
To change the background color of form1, In the properties tab, select Bgtype as Color and then click on the [...] button of the Color property to select a color.



In the sample project attached, NAVY is the background color of form 1. Changing the background color of form1 is important, this will be explained further.

Add a Fancy Buttons in Form1

It is shown below how to add a Fancy button/Winbutton.

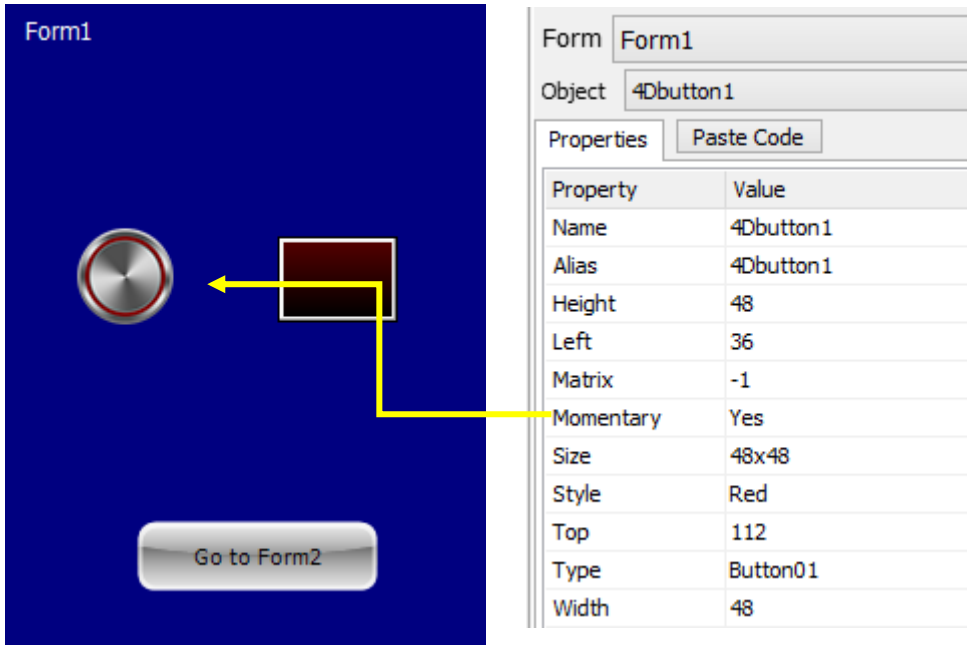


The object Winbutton1 will be used to go to Form2.

To know more about adding Fancy Buttons/Winbuttons in ViSi refer to the application note ViSi Winbuttons.

Add a 4Dbutton in Form1

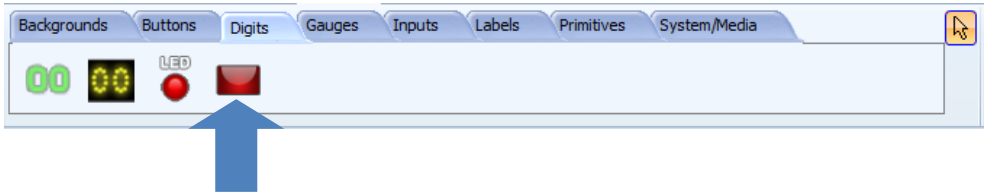
It is shown below how to add a 4Dbutton.

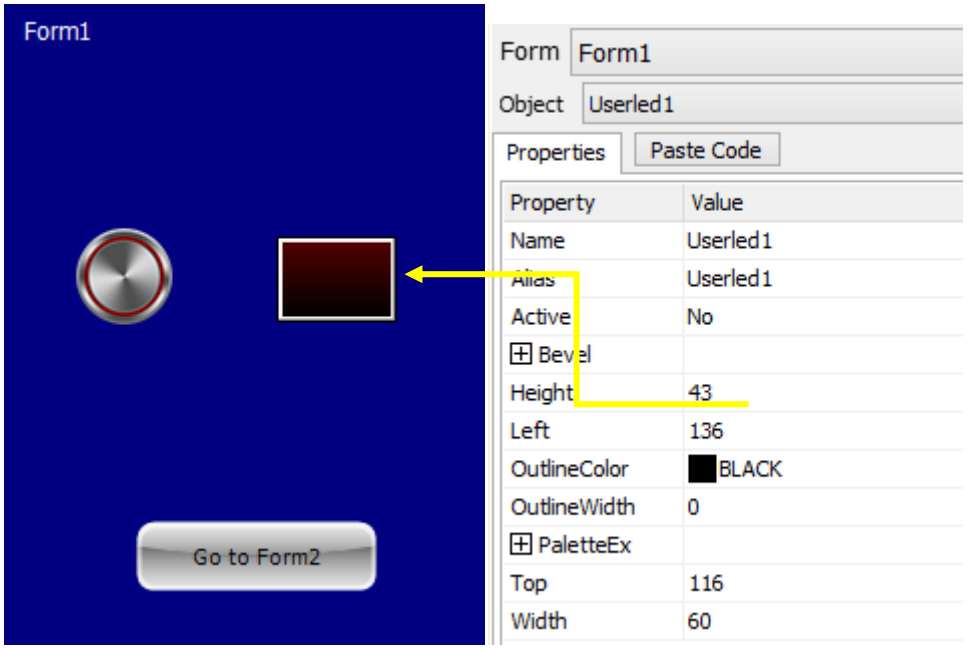


The object 4Dbutton1 is used for Form1 to have another touch objects besides Winbutton1. This also turns the ON the Userled1.

Add a Userled in Form1

It is shown below how to add a Userled object.



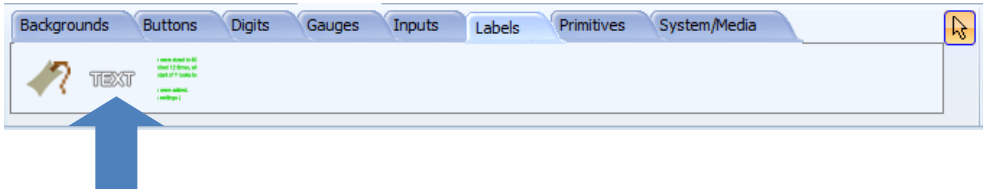


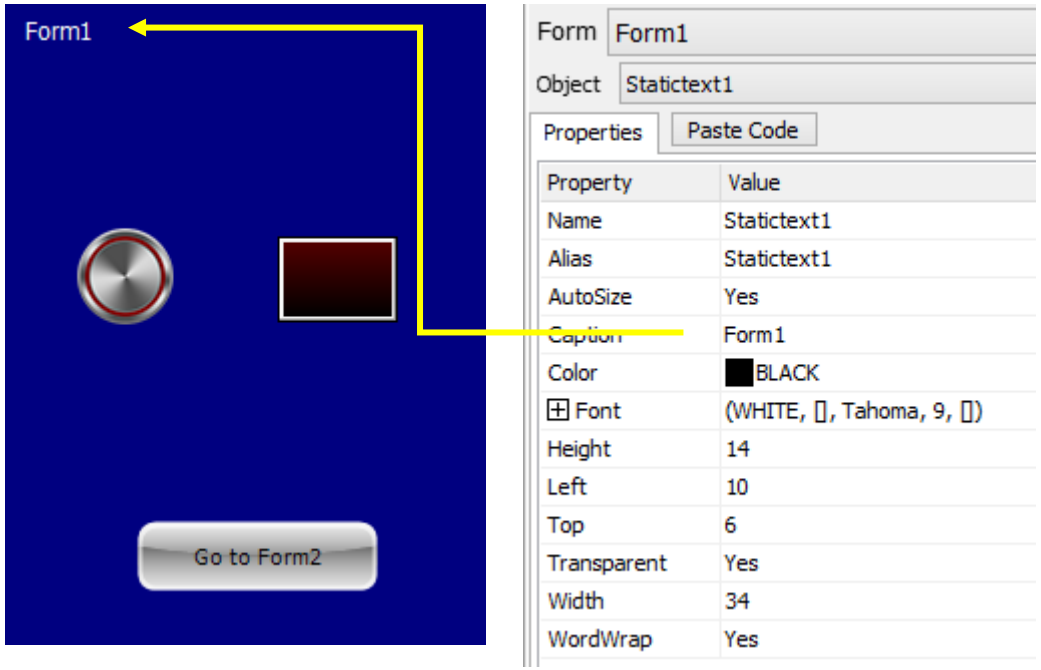
The object Userled1 is used an output for the 4Dbutton1. If 4Dbutton is pressed, Userled1 will turn ON, else it will turn OFF.

To know more about adding Userled in ViSi refer to the application note ViSi LED and Custom Digits.

Add a Statictext in Form1

It is shown below how to add a Statictext object.





The object Statictext1 is used to indicate that the current form is Form1.

Add Second Form

To add another form, click on the 'Form' object as shown below.

Automatically a new form will be added to WYSIWYG. This is Form2.

Form2 contains a background color of CRIMSON, 4Dbutton, Cool Gauge, Winbutton and a Statictext. If 4Dbutton2 is pressed the Coolgauge1 will start to move from 1 to 100, else if 4Dbutton2 is not pressed then Coolgauge1 will stay at 0 value. Winbutton2 object is used to go back to Form1. Statictext2 is used to indicate that the current form is Form2.

The Program

Function main()

```
func main()

    var reply;
    putstr("Mounting...\n");
    if (!(file_Mount()))
        while(!(file_Mount()))
            putstr("Drive not mounted...");
            pause(200);
            gfx_Cls();
            pause(200);
        wend
    endif

    hndl := file_LoadImageControl("FORMSW~1.dat", "FORMSW~1.gci", 1);

    touch_Set(TOUCH_ENABLE); //Enable the touch screen
    reply := Form1();  //run Form1 initially, get return value
    repeat
        if(reply == iWinbutton1)
            reply := Form2();    //run Form2
        else if(reply == iWinbutton2)
            reply := Form1();    //run Form1
        endif
    forever
endfunc
```

The main loop in the sample mounts the uSD card first. After that it will enable the touch screen. The variable reply is used to determine which Form will run. The object Winbutton1 is located at Form1, and if that button is pressed then Form2 will run. Same thing goes for Winbutton2.

Function Form1()

```
func Form1()
    var _state, _n, _returnValue;
    gfx_Cls();

    // Form1 1.1 generated 7/16/2015 9:07:12 AM
    gfx_BGcolour(NAVY) ;
    gfx_Cls() ;

    // Statictext1 1.0 generated 7/8/2015 3:32:29 PM
    img_Show(hndl,iStatictext1) ;

    // Userled1 1.0 generated 7/8/2015 3:32:45 PM
    img_Show(hndl,iUserled1) ;  // show initialy, if require
    img_SetWord(hndl, iUserled1, IMAGE_INDEX, 0) ;      // wi
    img_Show(hndl,iUserled1) ;

    // 4Dbutton1 1.0 generated 7/8/2015 3:32:35 PM
    img_ClearAttributes(hndl, i4Dbutton1, I_TOUCH_DISABLE);
    img_Show(hndl, i4Dbutton1);  // show button, only do thi
    img_SetWord(hndl, i4Dbutton1, IMAGE_INDEX, 0); // where
    img_Show(hndl,i4Dbutton1) ;

    // Winbutton1 1.0 generated 7/8/2015 3:32:40 PM
    img_ClearAttributes(hndl, iWinbutton1, I_TOUCH_DISABLE);
    img_Show(hndl, iWinbutton1);  // show button, only do th
    img_SetWord(hndl, iWinbutton1, IMAGE_INDEX, 0); // where
    img_Show(hndl,iWinbutton1) ;
```

This part of Form1() function initially displays the objects in Form1. This includes background color, Statictext1, Userled1, 4Dbutton1, and Winbutton1.

```
repeat

    _state := touch_Get(TOUCH_STATUS);              // get t
    _n := img_Touched(hndl,-1) ;

    if(_state == TOUCH_PRESSED)
        if(_n == i4Dbutton1)

            img_SetWord(hndl, i4Dbutton1, IMAGE_INDEX, 1); //
            img_Show(hndl,i4Dbutton1) ;

            img_SetWord(hndl, iUserled1, IMAGE_INDEX, 1) ;
            img_Show(hndl,iUserled1) ;
        endif

        if(_n == iWinbutton1)

            img_SetWord(hndl, iWinbutton1, IMAGE_INDEX, 1);
            img_Show(hndl,iWinbutton1) ;

        endif
    endif
```

The part of the loop of Form1() function is shown in the image above. The code above gets touch status and image touched. If 4Dbutton1 is pressed it will change the state of 4Dbutton1 and Userled1 to value of 1. if Winbutton1 is pressed then it will change the state if Winbutton1.

```
    if(_state == TOUCH_RELEASED)
        if(_n == i4Dbutton1)
            img_SetWord(hndl, i4Dbutton1, IMAGE_INDEX, 0);
            img_Show(hndl,i4Dbutton1) ;
            img_SetWord(hndl, iUserled1, IMAGE_INDEX, 0) ;
            img_Show(hndl,iUserled1) ;
        endif

        if(_n == iWinbutton1)
            img_SetWord(hndl, iWinbutton1, IMAGE_INDEX, 0);
            img_Show(hndl,iWinbutton1) ;
            _returnValue := iWinbutton1;
            goto exit;
        endif
    endif

forever
```

The second part of the loop of form1() function is shown above. If 4Dbutton1 is released then it will change the state of 4Dbutton1 and Userled1 to value of 0. If iWinbutton1 is pressed then the variable '_returnValue' will be equal to iWinbutton1 and then will go to the exit label which will be discussed further.

```
exit:
    pause(200);

    img_SetAttributes(hndl, i4Dbutton1, I_TOUCH_DISABLE);
    img_SetAttributes(hndl, iWinbutton1, I_TOUCH_DISABLE);
    img_SetAttributes(hndl, Form1, I_TOUCH_DISABLE);



    return _returnValue;
endfunc
```

This part of the program will occur if Winbutton1 is released. This means that it will break from the loop of Form1. The function img_SetAttributes(handle, index, value) SETS one or more bits in the IMAGE_FLAGS field of an image control entry. If the 'value' is set to 'I_TOUCH_DISABLE', then the index such as 4Dbutton1, Winbutton1 and Form1 will have their touch disabled. This is necessary for switching to Form2. If the touch of the inputs objects and background of form is not disabled, then in Form2 the objects that are not disabled will still be detected causing errors in the functionality of the program.

Function Form2()

```
func Form2()
    var _state, _n, _returnValue,x;
    gfx_Cls();
    x := 0;

    // Form2 1.1 generated 7/16/2015 9:08:08 AM
    gfx_BGcolour(CRIMSON) ;
    gfx_Cls() ;

    // Statictext2 1.0 generated 7/8/2015 3:54:30 PM
    img_Show(hndl,iStatictext2) ;

    // Coolgauge1 1.0 generated 7/8/2015 3:54:32 PM
    img_SetWord(hndl, iCoolgauge1, IMAGE_INDEX, 0) ; // wher
    img_Show(hndl,iCoolgauge1) ;

    // 4Dbutton2 1.0 generated 7/8/2015 3:54:36 PM
    img_ClearAttributes(hndl, i4Dbutton2, I_TOUCH_DISABLE);
    img_Show(hndl, i4Dbutton2);  // show button, only do thi
    img_SetWord(hndl, i4Dbutton2, IMAGE_INDEX, 0); // where
    img_Show(hndl,i4Dbutton2) ;

    // Winbutton2 1.0 generated 7/8/2015 3:54:38 PM
    img_ClearAttributes(hndl, iWinbutton2, I_TOUCH_DISABLE);
    img_Show(hndl, iWinbutton2);  // show button, only do th
    img_SetWord(hndl, iWinbutton2, IMAGE_INDEX, 0); // where
    img_Show(hndl,iWinbutton2) ;
```

This part of Form2() function initially clears then displays the objects in Form2. This includes background color, Statictext2, Coolgauge1, 4Dbutton2, and Winbutton2.

```
repeat
    _state := touch_Get(TOUCH_STATUS);                // get touch
    _n := img_Touched(hndl,-1) ;

    if(_state == TOUCH_PRESSED)
        if(_n == i4Dbutton2)

            img_SetWord(hndl, i4Dbutton2, IMAGE_INDEX, 1); // whe
            img_Show(hndl,i4Dbutton2) ;

            while(_n == i4Dbutton2)
                _state := touch_Get(TOUCH_STATUS);
                _n := img_Touched(hndl,-1) ;
                // Coolgauge1 1.0 generated 7/8/2015 3:58:38 PM
                img_SetWord(hndl, iCoolgauge1, IMAGE_INDEX, x) ;
                img_Show(hndl,iCoolgauge1) ;

                if(_state == TOUCH_RELEASED)
                    break;
                endif

                x++;
                if(x == 100)
                    x := 0;
                endif
            wend

        endif
```

The part of the loop of Form2() function is shown in the image above. The code above gets touch status and image touched. If 4Dbutton2 is pressed it will change the state of 4Dbutton1 and moves Coolgauge1 from value of 0 to 100 and then repeats again if still pressed. if Winbutton2 is pressed then it will change the state of Winbutton2.

```
    if(_state == TOUCH_RELEASED)
        if(_n == i4Dbutton2)
            img_SetWord(hndl, i4Dbutton2, IMAGE_INDEX, 0);
            img_Show(hndl,i4Dbutton2) ;
            // Coolgauge1 1.0 generated 7/8/2015 3:59:39 PM
            img_SetWord(hndl, iCoolgauge1, IMAGE_INDEX, x)
            img_Show(hndl,iCoolgauge1) ;

        endif

        if(_n == iWinbutton2)
            img_SetWord(hndl, iWinbutton2, IMAGE_INDEX, 0);
            img_Show(hndl,iWinbutton2) ;
            _returnValue := iWinbutton2;
            goto exit;
        endif
    endif

forever
```

The second part of the loop of form2() function is shown above. If 4Dbutton2 is released then it will change the state of 4Dbutton2 and Coolgauge1 to its last value. If iWinbutton2 is pressed then the variable '_returnValue' will be equal to iWinbutton2 and then will go to the exit label which will be discussed further.

```
exit:
    pause(200);

    img_SetAttributes(hndl, i4Dbutton2, I_TOUCH_DISABLE);
    img_SetAttributes(hndl, iWinbutton2, I_TOUCH_DISABLE);
    img_SetAttributes(hndl, Form2, I_TOUCH_DISABLE);


    return _returnValue;

endfunc
```

This part of the program will occur if Winbutton2 is released. This means that it will break from the loop of Form2. The function img_SetAttributes(handle, index, value) SETS one or more bits in the IMAGE_FLAGS field of an image control entry. If the 'value' is set to 'I_TOUCH_DISABLE', then the index such as 4Dbutton2, Winbutton2 and Form2 will have their touch disabled. This is necessary for switching back to Form1. If the touch of the inputs objects and background of form is not disabled, then in Form1 the objects that are not disabled will still be detected causing errors in the functionality of the program.

Run the Program

For instructions on how to save a **ViSi** project, how to connect the target display to the PC, how to select the program destination (this option is not available for Goldelox displays), and how to compile and download a program, please refer to the section "**Run the Program**" of the application note

ViSi Getting Started - First Project for Picaso and Diablo16

The uLCD-32PTU and/or uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

Proprietary Information

Disclaimer of Warranties & Limitation of Liability