

# DIAMView

Window Program&Event Program

Ruby

2020/05/01



- The concepts of event
- Left button event
- Right button event
- Mouse event
- Window operation event
- Value input event
- Sliding input event
- Rotation input event
- Window program event
- Control event
- Keyboard

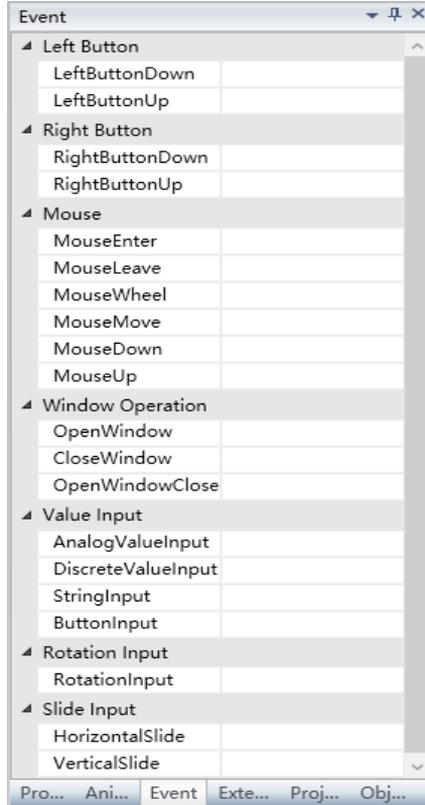
In this chapter, you will learn .....

... more about event

... more about ten event types of DIAView

- The concepts of event
- Left button event
- Right button event
- Mouse event
- Window operation event
- Value input event
- Sliding input event
- Rotation input event
- Window program event
- Control event
- Keyboard

Events are operations that can be recognized and responded by graphical objects, and are divided into system events and user events. The events in the DIAView are generally user events, that is, the user's operations on various graphical objects on the window, and then drive the graphical controls to perform a certain function.

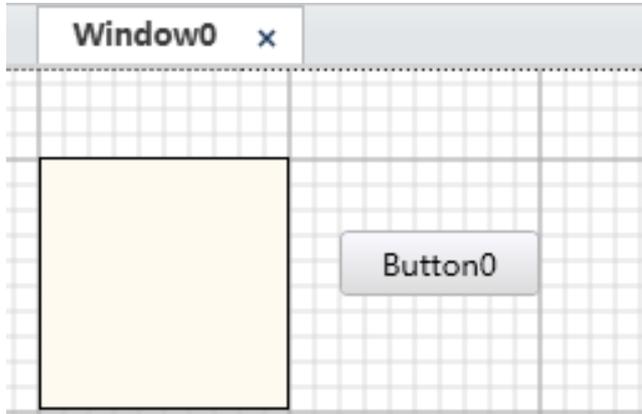


- The concepts of event
- **Left button event**
- Right button event
- Mouse event
- Window operation event
- Value input event
- Sliding input event
- Rotation input event
- Window program event
- Control event
- Keyboard

➤ Left Button Down event example:

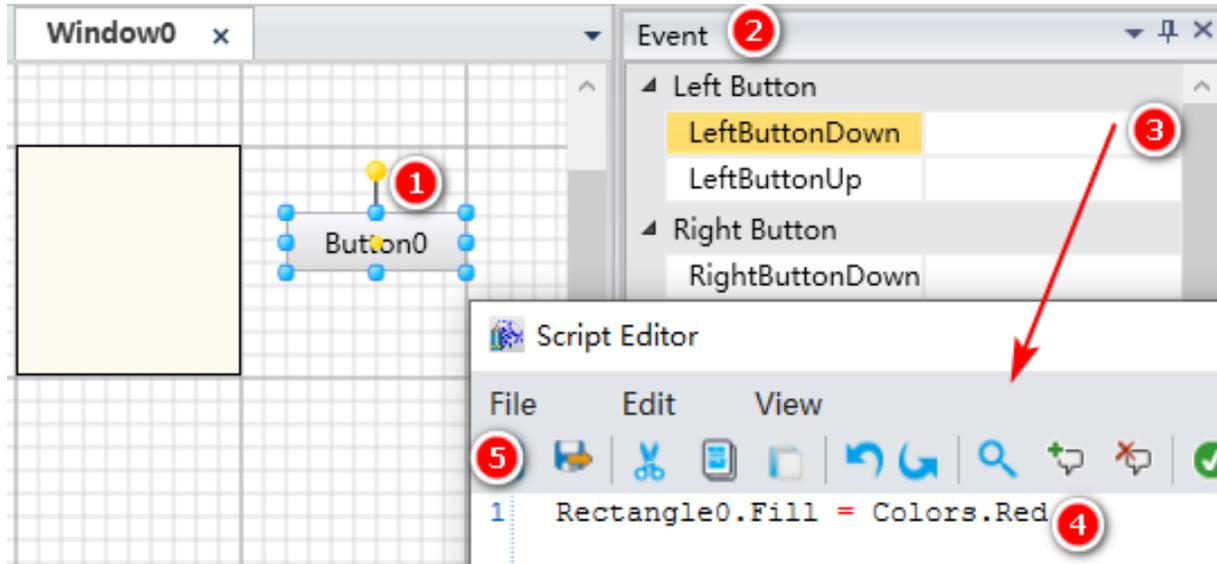
Create a red rectangle with left button down event

(1) Create a Rectangle0 and a Button0 in the Window0



# Left Button Event—Left Button Down

(2) Configure the left button down event of the Button0

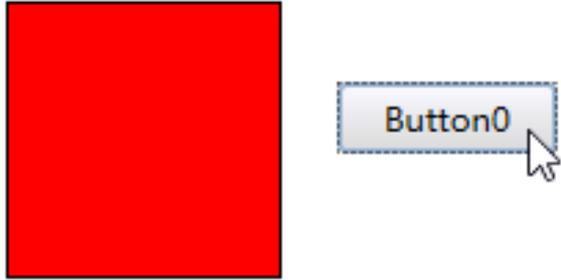


✘Refer to the section "10.2 Left button event" in user manual.



## Left Button Event—Left Button Down

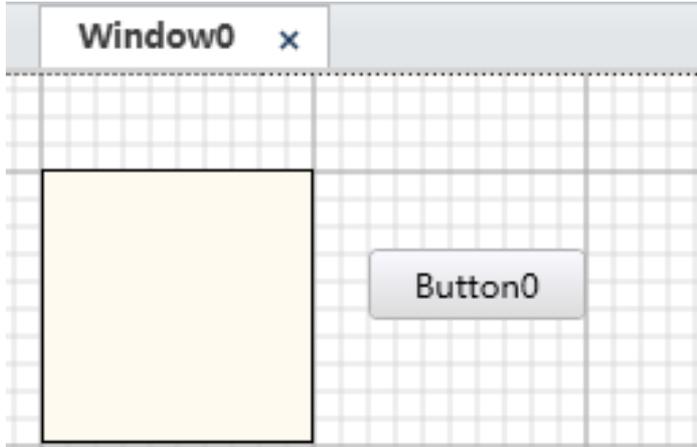
(3) Run the Window0, left click on “Button0”, then the Rectangle0 turns red



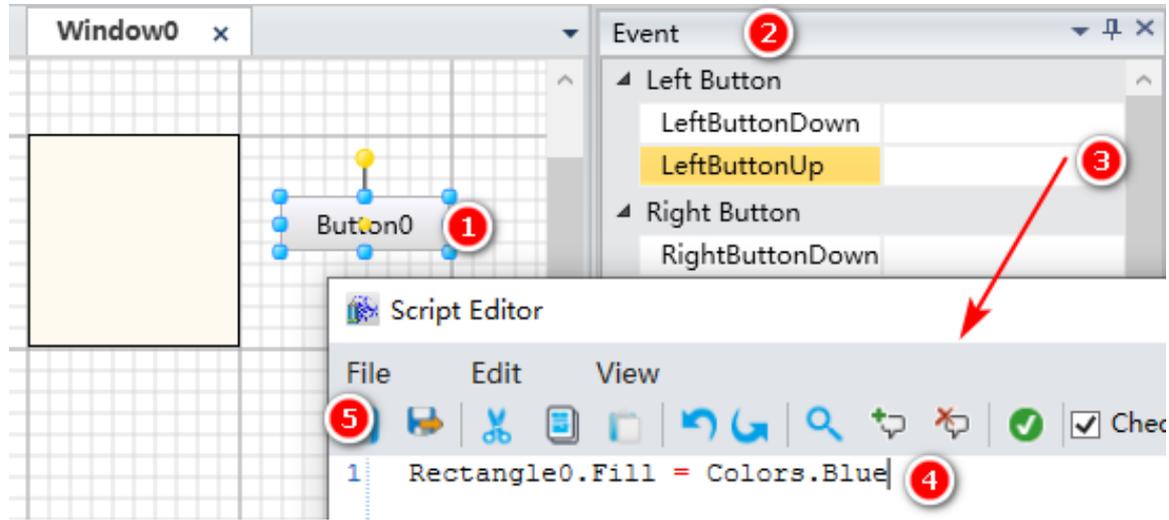
➤ Left Button Up event example:

Create a blue rectangle with left button up event :

(1) Create a Rectangle0 and a Button0 in the Window0



(2) Configure the left button up event of the Button0

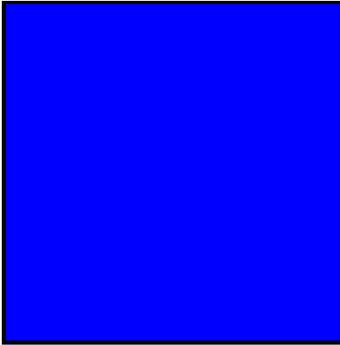


※Refer to the section “10.2 Left button event” in user manual.



## Left Button Event—Left Button Up

(3) Run the Window0, left click on “Button0”, when the mouse is pressed, the colour of the Rectangle0 does not change, then release the mouse, the colour of the Rectangle0 turns blue

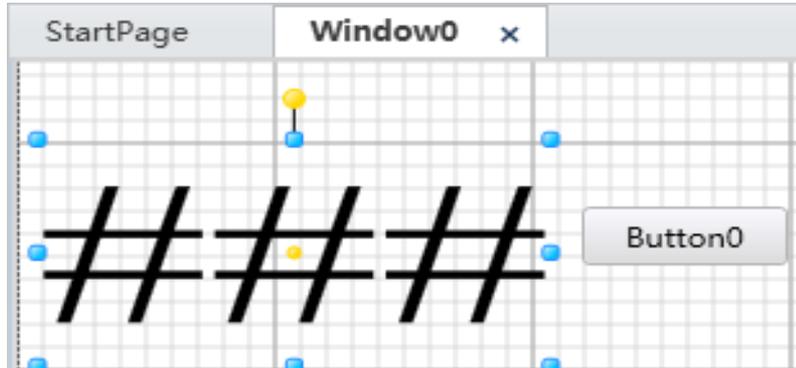


- The concepts of event
- Left button event
- **Right button event**
- Mouse event
- Window operation event
- Value input event
- Sliding input event
- Rotation input event
- Window program event
- Control event
- Keyboard

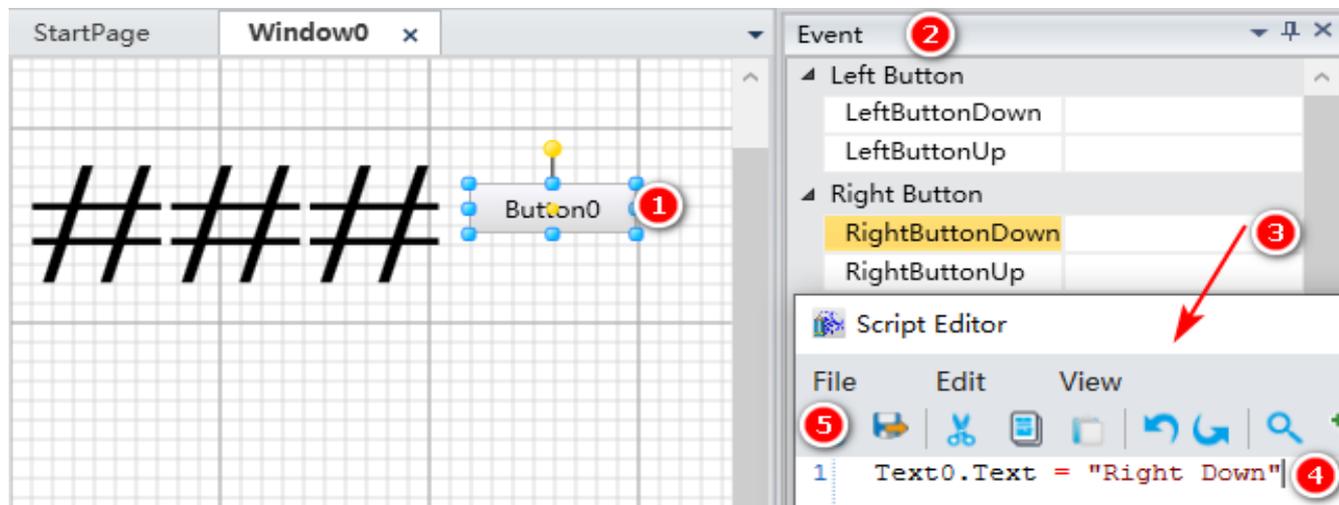
➤ Right Button Down event example:

Create a text and set its content with right button down event :

(1) Create a Text0 and a Button0 in the Window0



(2) Configure the right button down event of the Button0



The screenshot shows a software development environment with a grid workspace and a right-hand panel. In the workspace, three hash symbols (#) are on the left, and a button labeled "Button0" is on the right, marked with a red circle 1. The right-hand panel has an "Event" section (marked with a red circle 2) containing a tree view with "Left Button" and "Right Button" categories. Under "Right Button", "RightButtonDown" is selected and highlighted in yellow (marked with a red circle 3). Below the event list is a "Script Editor" window (marked with a red circle 5) containing a single line of code: `Text0.Text = "Right Down"` (marked with a red circle 4). A red arrow points from the "RightButtonDown" event to the script editor.

※Refer to the section "10.3 Right button event" in user manual.



## Right Button Event—Right Button Down

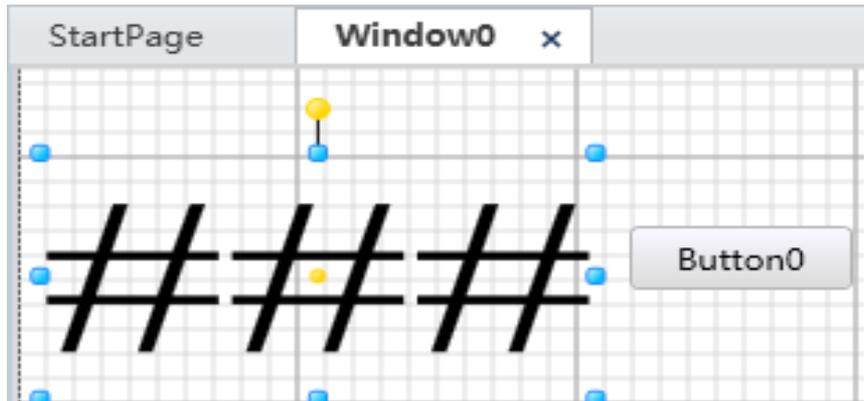
(3) Run the Window0, right click on "Button0", then the content of Text0 becomes "Right Down"

Right Down 

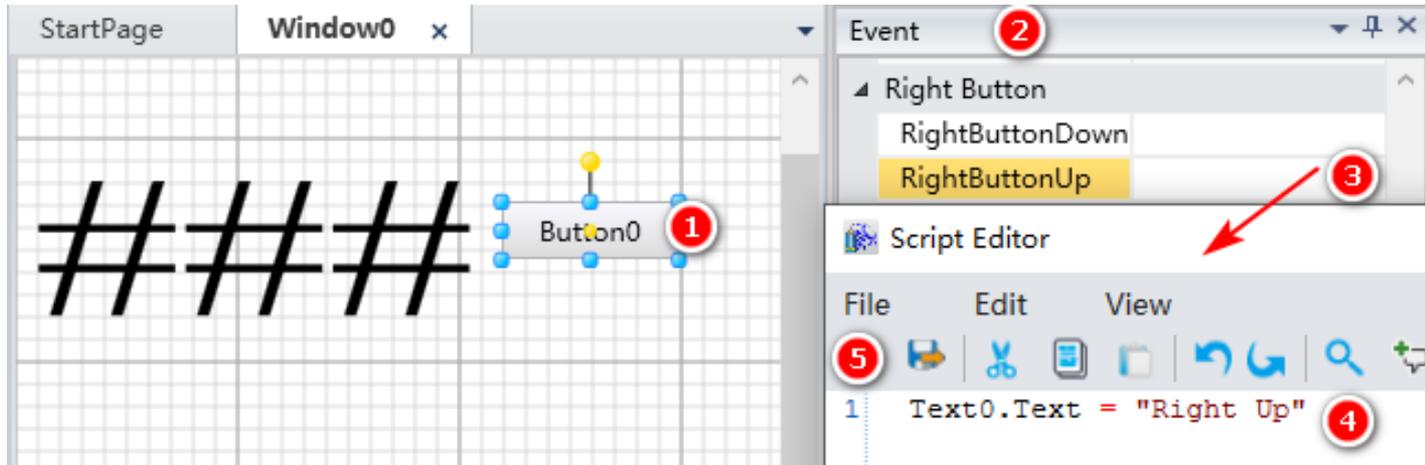
➤ Right Button Up event example:

Create a text and set its content with right button up event :

(1) Create a Text0 and a Button0 in the Window0



(2) Configure the right button up event of the Button0



The screenshot displays a software development environment with a grid-based workspace on the left and a right-hand sidebar. In the workspace, three hash symbols (#) are visible, followed by a button labeled "Button0" (marked with a red circle 1). The sidebar contains an "Event" panel (marked with a red circle 2) showing a tree view with "Right Button" expanded, and "RightButtonUp" selected (marked with a red circle 3). Below the event panel is a "Script Editor" window (marked with a red circle 5) with a menu bar (File, Edit, View) and a toolbar. The script editor contains the code: `1 Text0.Text = "Right Up"` (marked with a red circle 4). A red arrow points from the "RightButtonUp" event to the script editor.

※Refer to the section "10.3 Right button event" in user manual.



## Right Button Event—Right Button Up

(3) Run the Window0, right click on “Button0”, when the mouse is pressed, the content of the Text0 does not change, then release the mouse, the content of the Text0 becomes “Right Up”

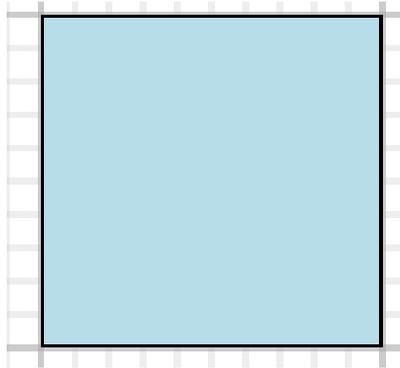
Right Up 

- The concepts of event
- Left button event
- Right button event
- **Mouse event**
- Window operation event
- Value input event
- Sliding input event
- Rotation input event
- Window program event
- Control event
- Keyboard

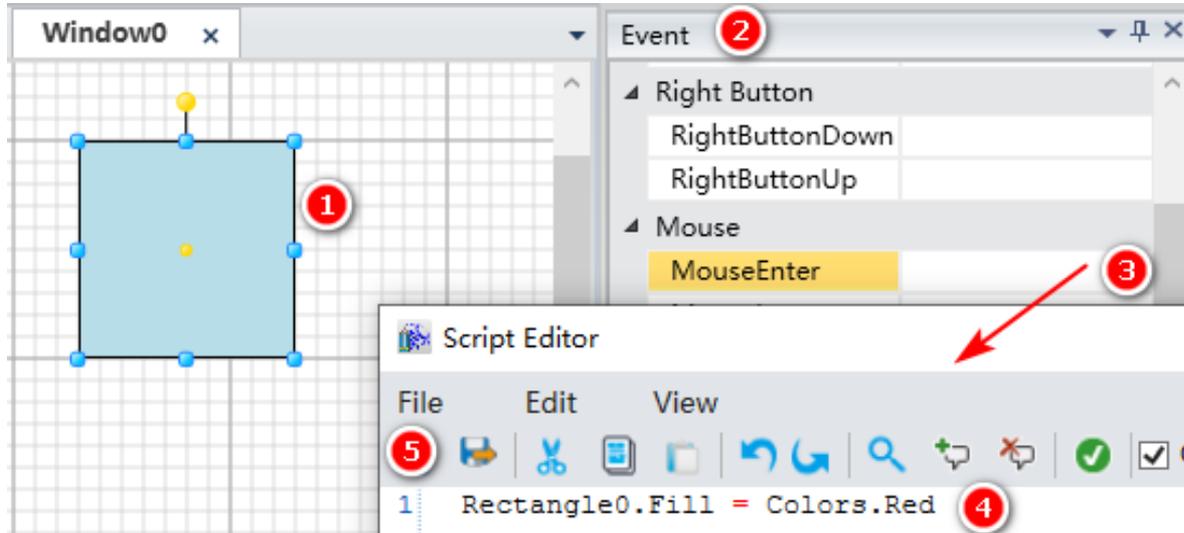
➤ **Mouse Enter event** example:

Create a rectangle and set its color with mouse enter event :

(1) Create a Rectangle0 in the Window0



(2) Configure mouse enter event of the Rectangle0

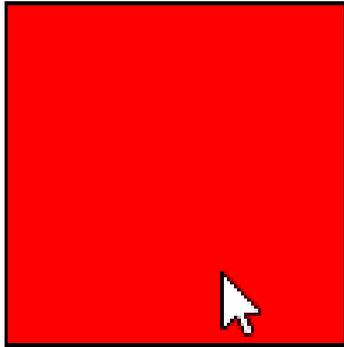


✘Refer to the section "10.4 Mouse event" in user manual.



## Mouse Event—Mouse Enter

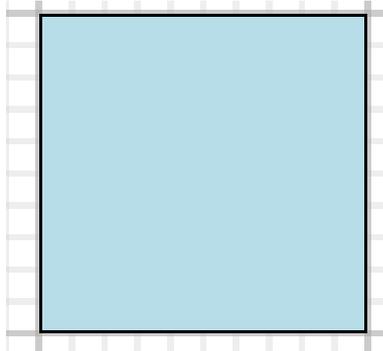
(3) Run the Window0, move mouse, when mouse enter the Rectangle0, then the Rectangle0 turns red



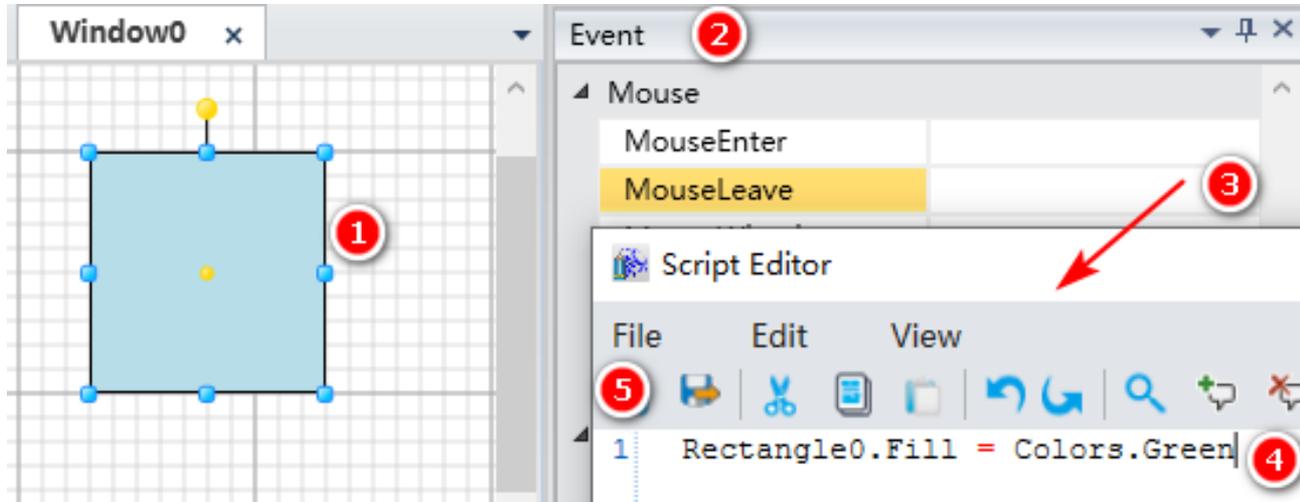
➤ **Mouse Leave event** example:

Create a rectangle and set its color with mouse leave event :

(1) Create a Rectangle0 in the Window0

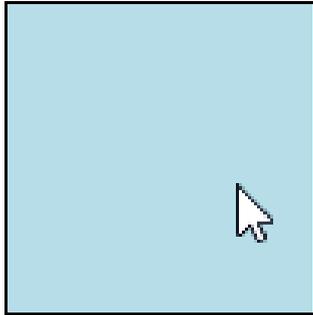


(2) Configure mouse leave event of the Rectangle0

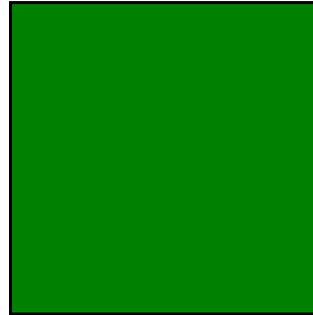


✘Refer to the section “10.4 Mouse event” in user manual.

(3) Run the Window0, move mouse, when mouse enter the Rectangle0, the color of Rectangle0 does not change ; when the mouse leave then Rectangle0, the Rectangle0 turns green



1

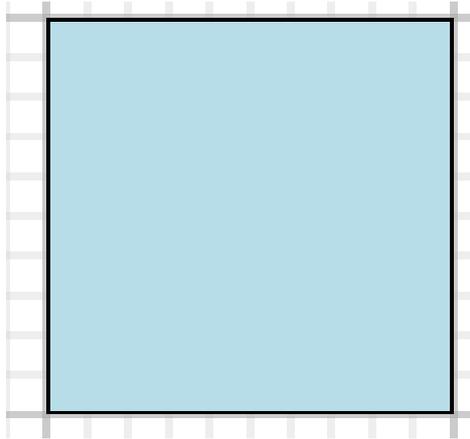


2

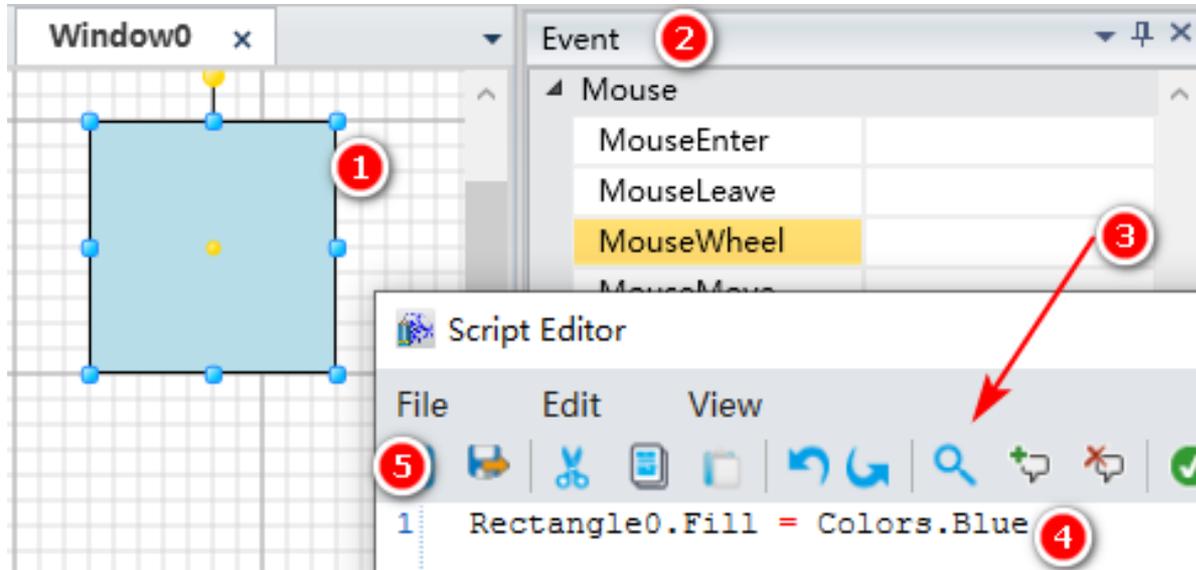
➤ **Mouse Wheel event** example:

Create a rectangle and set its color with mouse wheel event :

(1) Create a Rectangle0 in the Window0

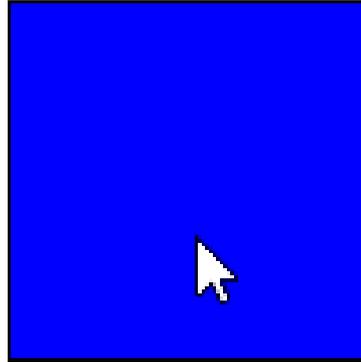


(2) Configure mouse wheel event of the Rectangle0



※Refer to the section “10.4 Mouse event” in user manual.

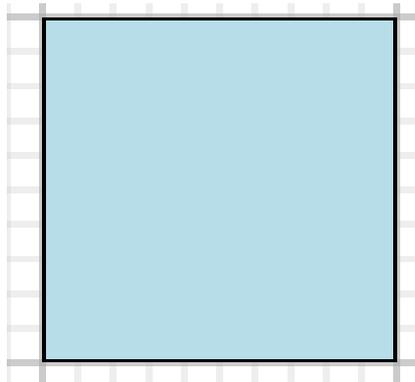
(3) Run the Window0, move mouse, when the mouse is on the Rectangle0, rolling the mouse, the Rectangle0 becomes blue



➤ **Mouse Move event** example:

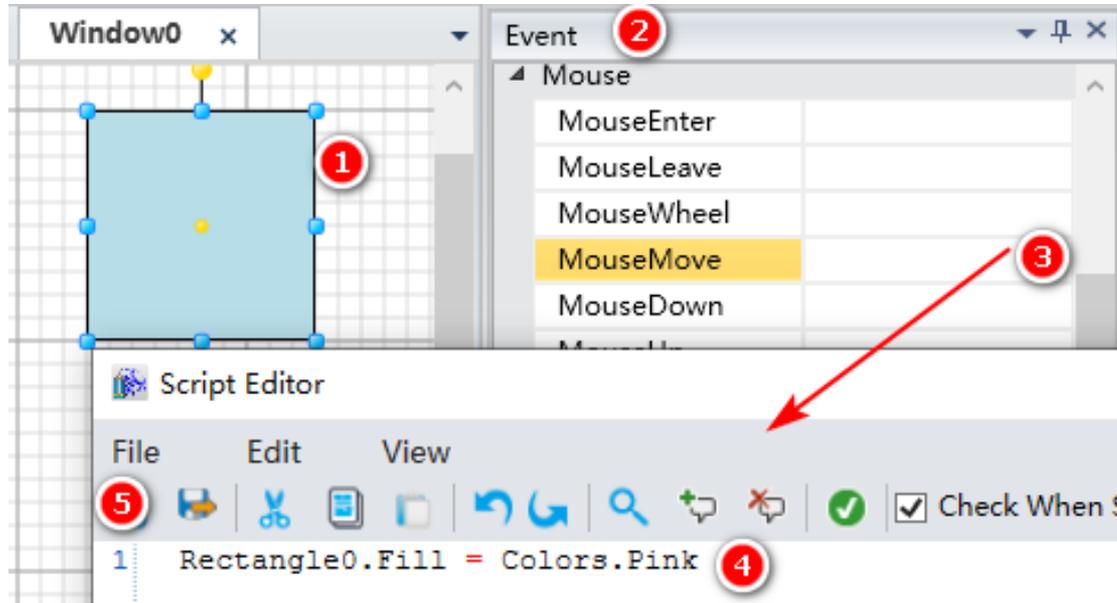
Create a rectangle and set its color with mouse move event :

(1) Create a Rectangle0 in the Window0

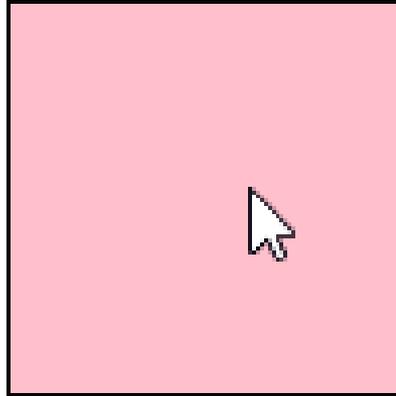


※Refer to the section “10.4 Mouse event” in user manual.

(2) Configure mouse move event of the Rectangle0



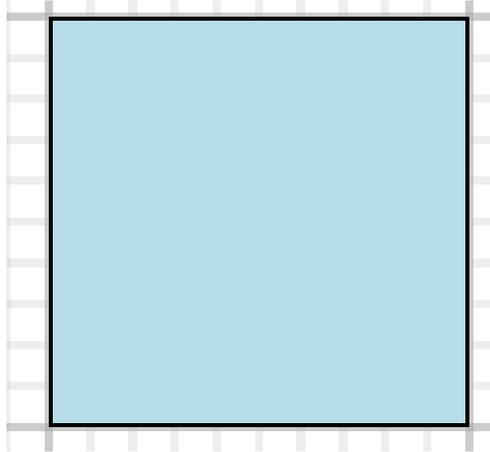
(3) Run the Window0, move mouse, when the mouse is on the Rectangle0, move the mouse, the Rectangle0 becomes pink



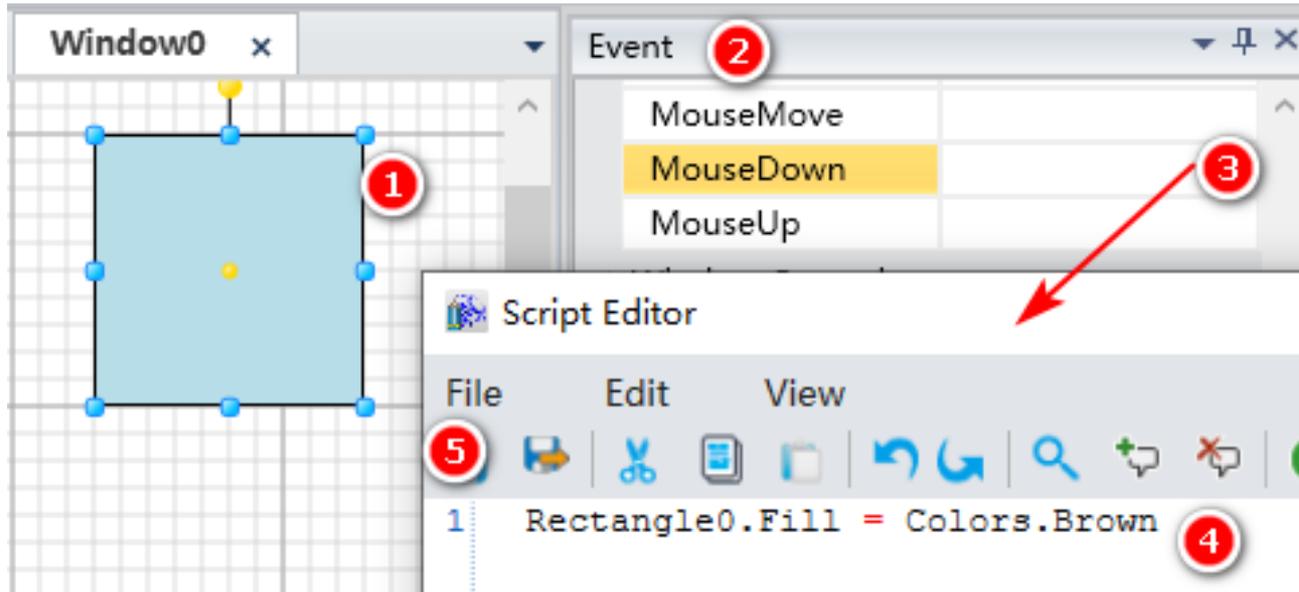
➤ **Mouse Down event** example:

Create a rectangle and set its color with mouse down event :

(1) Create a Rectangle0 in the Window0



(2) Configure mouse down event of the Rectangle0

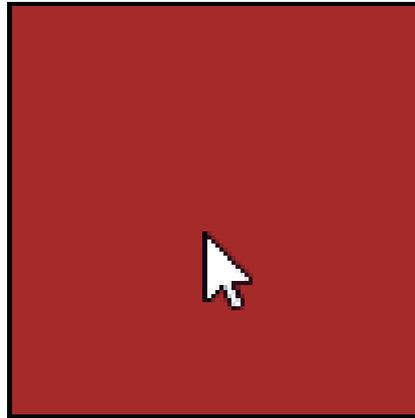


※Refer to the section "10.4 Mouse event" in user manual.



## Mouse Event—Mouse Down

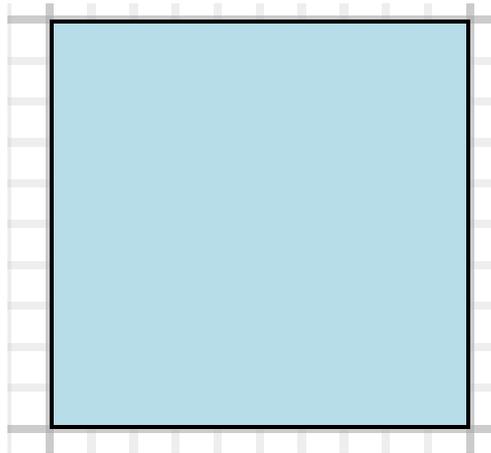
(3) Run the Window0, move mouse, when the mouse is on the Rectangle0, press the left or right mouse button, the Rectangle0 becomes brown



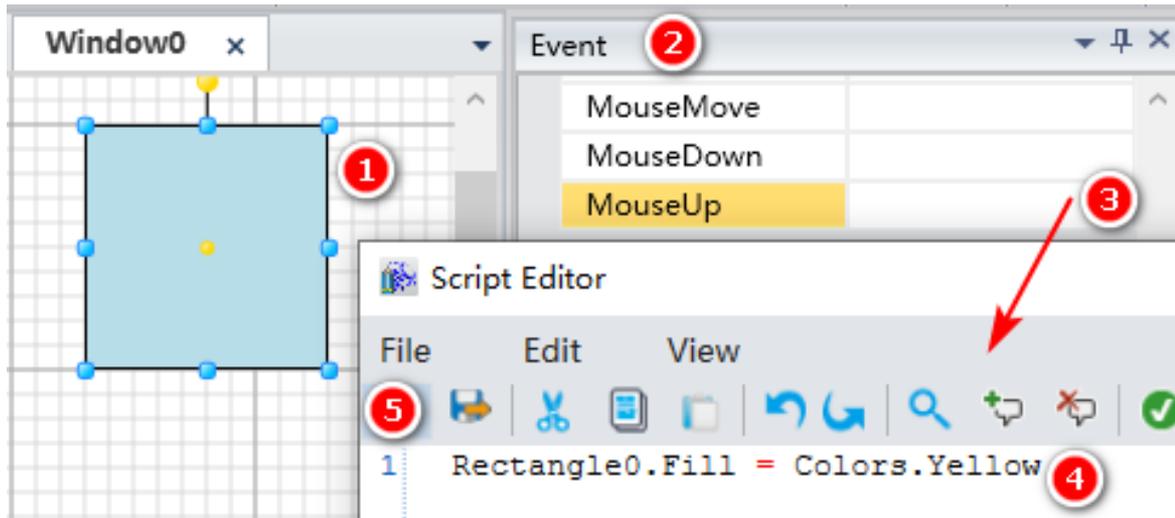
➤ **Mouse Up event** example:

Create a rectangle and set its color with mouse up event :

(1) Create a Rectangle0 in the Window0

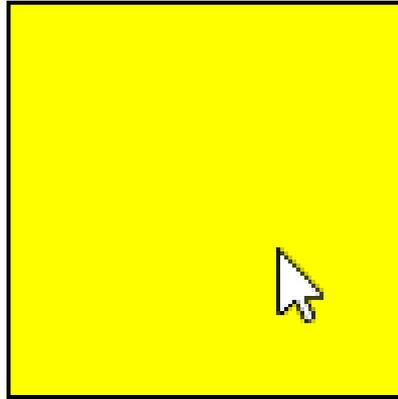


(2) Configure mouse up of the Rectangle0



※Refer to the section "10.4 Mouse event" in user manual.

(3) Run the Window0, move mouse, when the mouse is on the Rectangle0, press the left or right mouse button , the colour of Rectangle0 does not change , then release the mouse, Rectangle0 becomes yellow



- The concepts of event
- Left button event
- Right button event
- Mouse event
- **Window operation event**
- Value input event
- Sliding input event
- Rotation input event
- Window program event
- Control event
- Keyboard

➤ Open Window event example

(1) Create two windows(Window0,Window1) in the project



※Refer to the section “7.2.1.1 Add window” in user manual.



# Window Operation Event—Open Window

(2) Set the properties of the two windows

Window0

Property

Base

Name Window0

Title

GridDisplay

Background  ...

Layout

Size 250 . 500

Location 0 . 0

Property Animation Event Extension Project Object

Window0

Window1

Property

Base

Name Window1

Title

GridDisplay

Background  ...

Layout

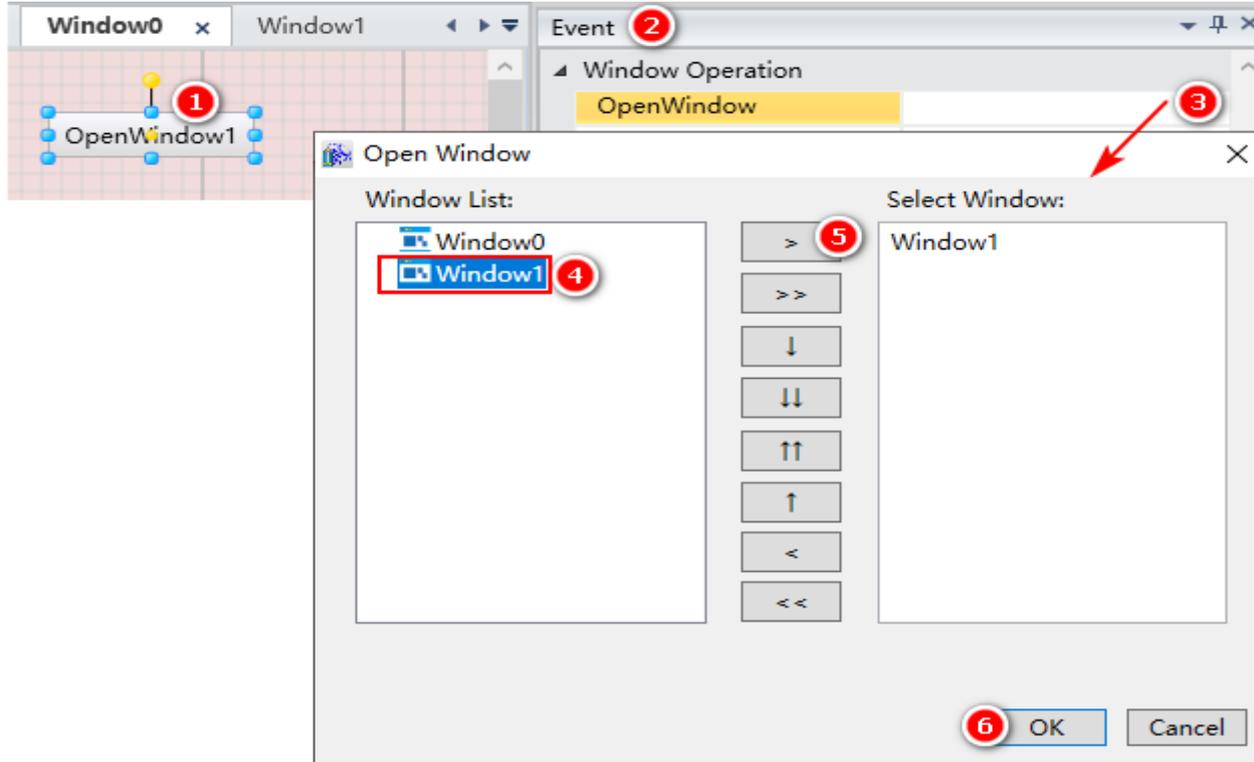
Size 500 . 250

Location 0 . 0

Property Animation Event Extension Project Object

Window1

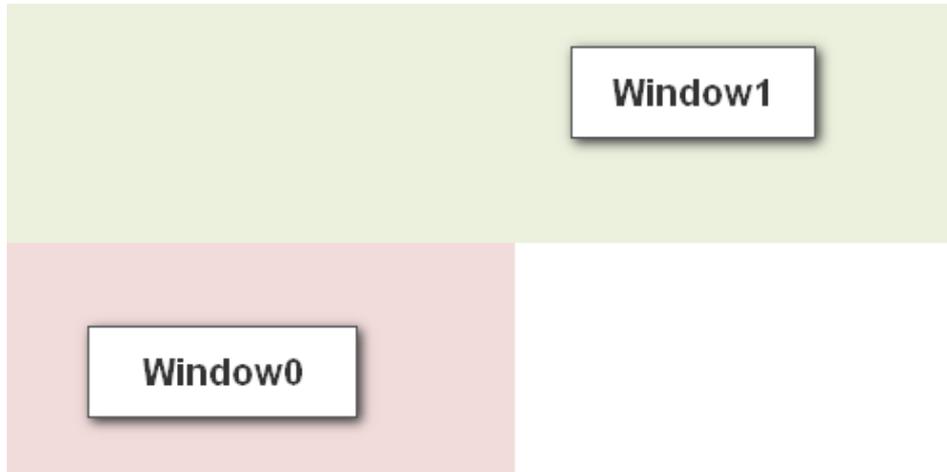
(3) Create a button in the Window0, configure its open window event





# Window Operation Event—Open Window

(4) Only run the Window0 default. Then click the “OpenWindow1” button, the Window1 is opened . The two windows are all running



➤ Close Window event example

(1) Create two windows(Window0,Window1) in the project



※Refer to the section “7.2.1.1 Add window” in user manual.



# Window Operation Event—Close Window

(2) Set the properties of the two windows

Window0

Property	Value
Name	Window0
Title	
GridDisplay	<input checked="" type="checkbox"/>
Background	(Red swatch)
Layout Size	250 . 500
Location	0 . 0

Window0

Window1

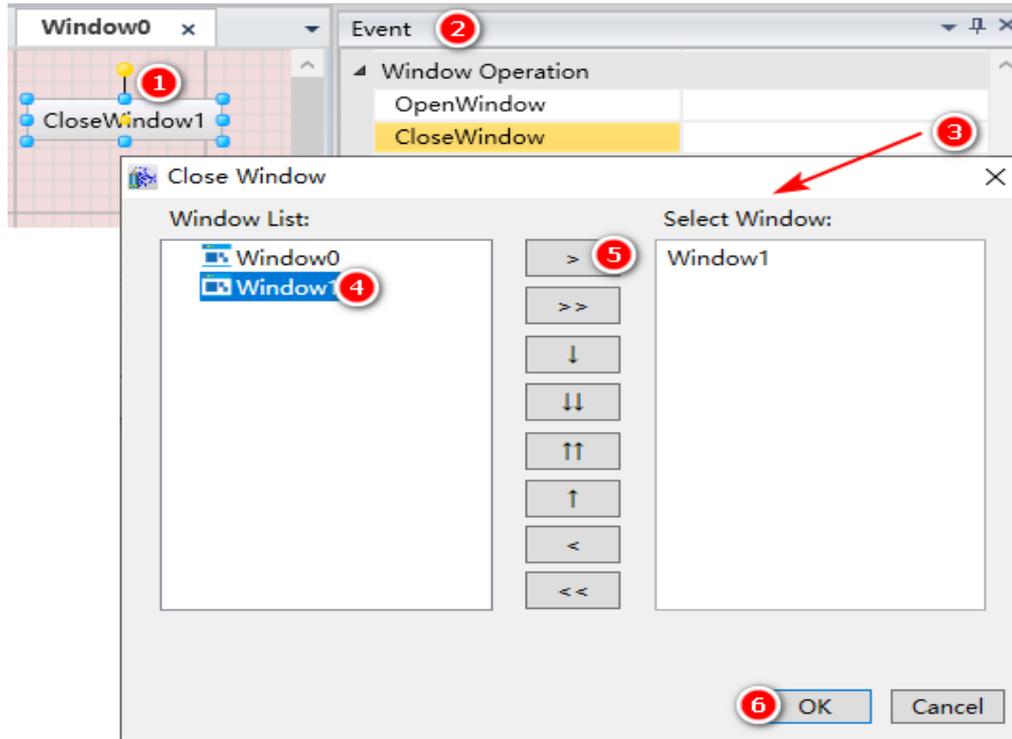
Property	Value
Name	Window1
Title	
GridDisplay	<input checked="" type="checkbox"/>
Background	(Green swatch)
Layout Size	500 . 250
Location	0 . 0

Window1



# Window Operation Event—Close Window

(3) Create a button in the Window0, configure its close window event



(4) Run the two windows default. Then click the “CloseWindow1” button, the Window1 is closed . Only Window0 is running

# Window Operation Event—Open Window And Close Others



## ➤ Open Window And Close Others event example

(1) Create two windows(Window0,Window1) in the project



※Refer to the section "7.2.1.1 Add window" in user manual.

# Window Operation Event—Open Window And Close Others



(2) Set the properties of the two windows

Window0 **1**

Property

Base

Name	Window0
Title	
GridDisplay	<input checked="" type="checkbox"/>
Background	<b>3</b> [Red Swatch] ...

Layout

Size	<b>4</b> [250] . [500]
Location	[0] . [0]

**2** Property Animation Event Extension Project Object

Window0

Window1 **1**

Property

Base

Name	Window1
Title	
GridDisplay	<input checked="" type="checkbox"/>
Background	<b>3</b> [Green Swatch] ...

Layout

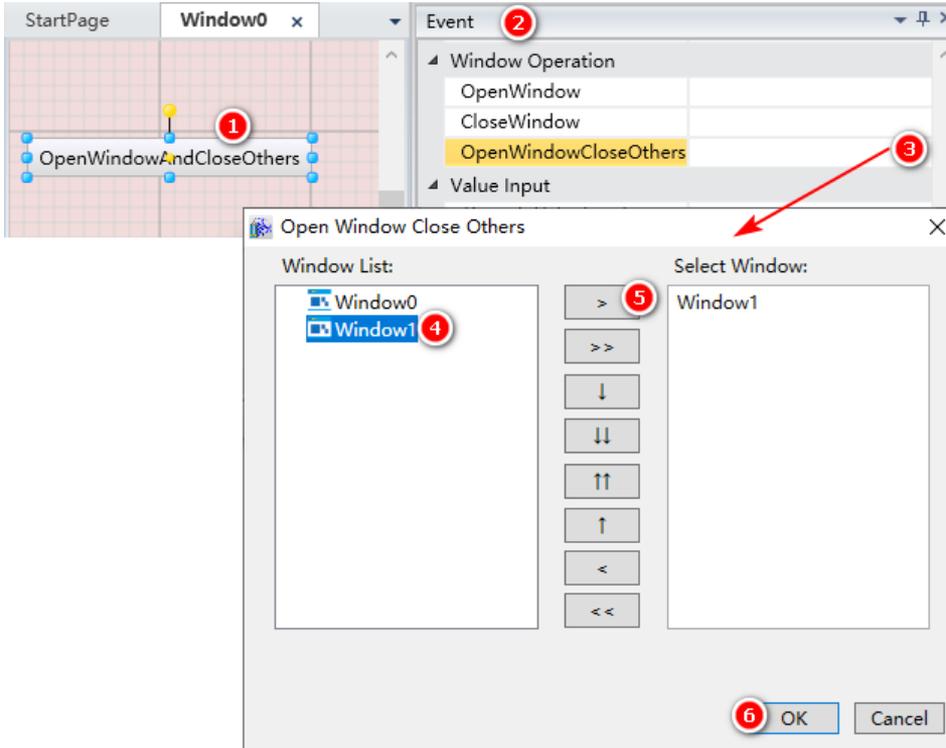
Size	<b>4</b> [500] . [250]
Location	[0] . [0]

**2** Property Animation Event Extension Project Object

Window1

# Window Operation Event—Open Window And Close Others

(3) Create a button in the Window0, configure its open window and close others event



(4) Only run the Window0 default. Then click the “OpenWindowAndCloseOthers” button, the Window1 is opened, Window0 is Closed. Only Window1 is running

- The concepts of event
- Left button event
- Right button event
- Mouse event
- Window operation event
- Value input event
- **Sliding input event**
- Rotation input event
- Window program event
- Control event
- Keyboard

➤ Horizontal Sliding event example

Create a text and move it horizontally and display the distance of moving :

(1) Create a analog variable: Variable0



	Name	Variable Type	Initial Value	Retentive Value
1	Variable0	Analog	0	

EventProject  
IO Device  
Variable Dictionary  
Window  
Window0

※Refer to the section "6.3 Variables" in user manual.



# Sliding Input Event—Horizontal Sliding

(2) Create a Text0 and a Line0 in the Window0. Configure event and animation of the Text0

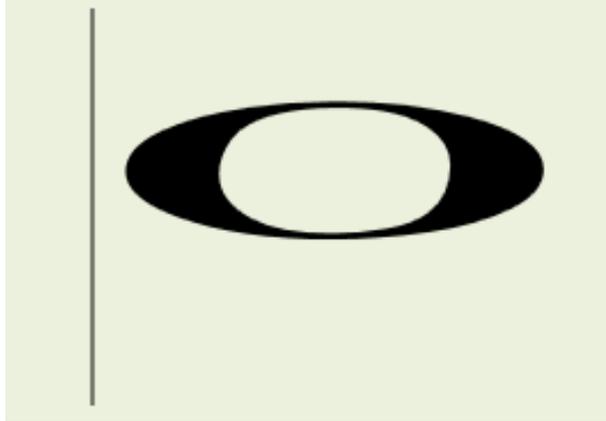
The screenshot shows the 'Event' panel on the right with 'HorizontalSlide' selected. A red circle '2' is around the 'Event' header. The 'Horizontal Slide' dialog box is open, showing 'Variable: Var.Variable0' with a red circle '5' around the text box. A red circle '4' is around the '...' button. The 'OperateMode' dropdown is set to 'OnceWrite' with a red circle '6' around it. Red circles '1' and '3' are also present on the main interface.

Horizontal Slide Event

The screenshot shows the 'Animation' panel on the right with 'AnalogValueDisplay' selected. A red circle '2' is around the 'Animation' header. The 'Analog Value Display' dialog box is open, showing 'Expression: Var.Variable0' with a red circle '5' around the text box. A red circle '4' is around the '...' button. The 'DecimalDigits' dropdown is set to '0' with a red circle '6' around it. Red circles '1' and '3' are also present on the main interface.

Analog Value Display Animation

(3) Run the Window0. The initial display is as follows



(4) Drag Text0 to the right with the mouse. During the dragging process, the content of Text0 remains unchanged(Figure1). When the mouse is released, the content of Text0 changes, showing the distance of dragging.(Figure2)

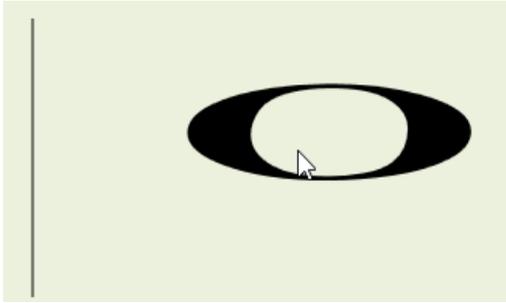


Figure1



Figure2

## ➤ Vertical Sliding event example

Create a text and move it vertically and display the distance of moving :

(1) Create a analog variable: Variable0



The screenshot shows the software interface with a toolbar at the top containing buttons for Add, Insert, Delete, Batch Add, Import, and Export. Below the toolbar is a table with the following data:

	Name	Variable Type	Initial Value	Retentive Value	M
1	Variable0	Analog	0		

On the right side, a tree view shows the project structure: EventProject, IO Device, Variable Dictionary (circled with a red '1'), Window, and Window0. A red arrow points from the Variable Dictionary icon to the 'Retentive Value' column of the table.

※Refer to the section "6.3 Variables" in user manual.



# Sliding Input Event—Vertical Sliding

(2) Create a Text0 and a Line0 in the Window0. Configure event and animation of the Text0

Window0 x Variable Dictionary

Event 2

- OpenWindowCloseOthers
- Value Input
  - AnalogValueInput
  - DiscreteValueInput
  - StringInput
  - ButtonInput
- Rotation Input
  - RotationInput
- Slide Input
  - HorizontalSlide
  - VerticalSlide**

Vertical Slide

Variable: Var.Variable0 5

Top Location: 0.00 Bottom Location: 100.00

VariableValue: 0.00 VariableValue: 100.00

Operate Mode: SequentialWrite 6

OK Cancel 7

Vertical Slide Event

Window0 x Vari

Animation 2

- Visibility
  - Visibility
  - Blink
- Value Display
  - TextDisplay
  - AnalogValueDisplay**

Analog Value Display

Expression: Var.Variable0 5

Digit

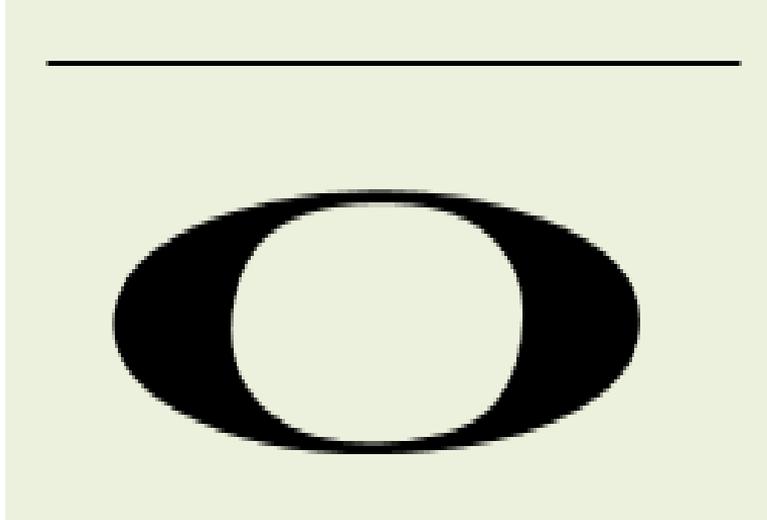
IntegerDigits: 1 DecimalDigits: 0 6

Scientific Notation Thousands Separator

OK Cancel 7

Analog Value Display Animation

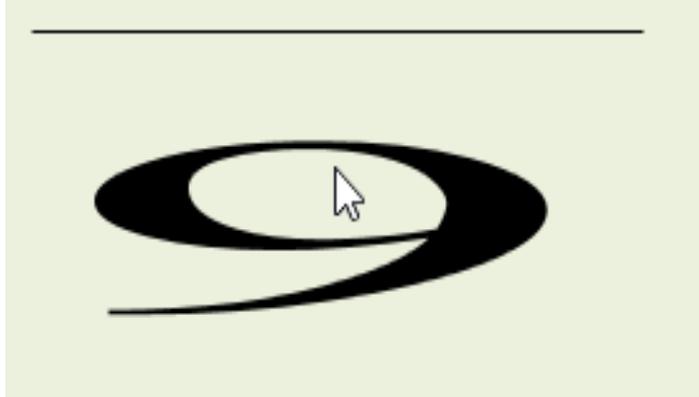
(3) Run the Window0. The initial display is as follows





## Sliding Input Event—Vertical Sliding

(4) Drag the Text0 down with the mouse. During the dragging process, the content of Text0 changes as the drag distance changes, that is, Text0 displays the drag distance in real time

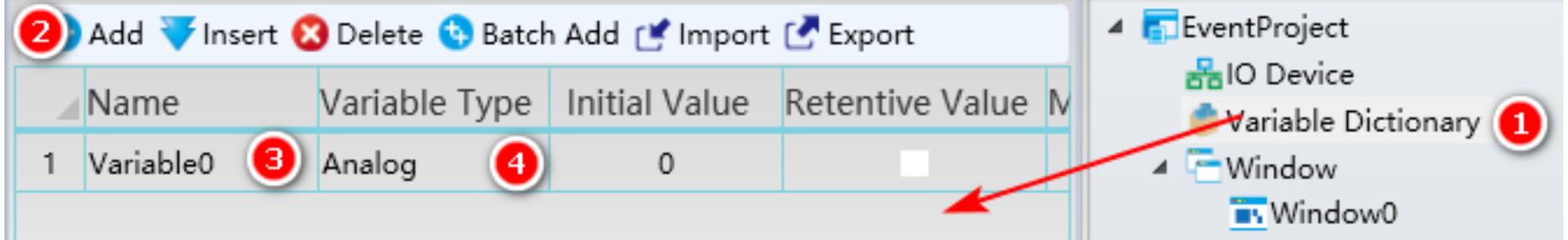


- The concepts of event
- Left button event
- Right button event
- Mouse event
- Window operation event
- **Value input event**
- Sliding input event
- Rotation input event
- Window program event
- Control event
- Keyboard

➤ Analog Value Input event example

Input a analog value in the text with the analog value input event :

(1) Create a analog variable: Variable0



	Name	Variable Type	Initial Value	Retentive Value	M
1	Variable0	Analog	0		

EventProject  
IO Device  
Variable Dictionary  
Window  
Window0

※Refer to the section "6.3 Variables" in user manual.

(2) Create a Text0 in the Window0. Configure event and animation of the Text0

The screenshot shows the 'Event' configuration window for 'Analog Value Input' in a software environment. The window is titled 'Event' and has a red circle '2' next to its title. The 'Value Input' section is expanded, and 'AnalogValueInput' is selected, highlighted with a yellow background. A red circle '3' is next to this selection. Below the event list, the 'Analog Value Input' dialog box is open. It has a 'Variable' field containing 'Var.Variable0' with a red circle '5' next to it. To the right of the field is a button with three dots and a red circle '4', and a 'Clear' button. Below the variable field are 'Value Range' settings: 'MinimumValue' set to '0.00' and 'MaximumValue' set to '10000.00'. At the bottom, there are 'Input Panel Size' options: 'Default' (selected), 'Adapt', and 'Full Screen'. At the very bottom are 'OK' and 'Cancel' buttons, with a red circle '6' next to the 'OK' button. A red arrow points from the 'AnalogValueInput' selection in the event list to the dialog box.

Analog Value Input event

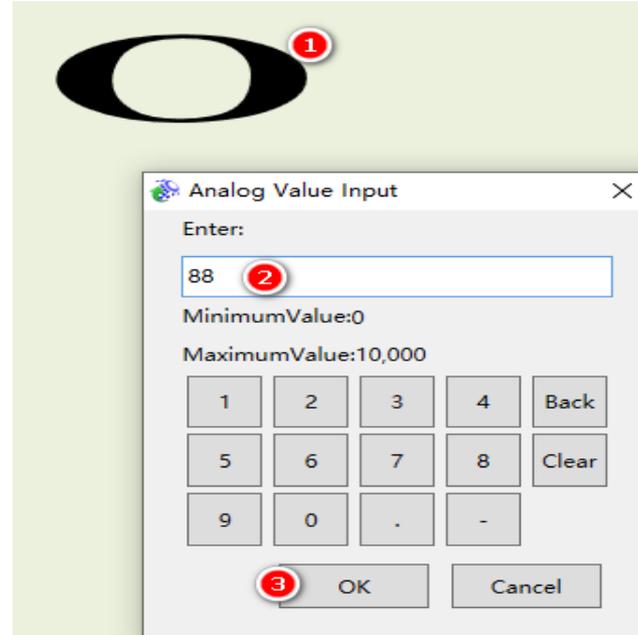
The screenshot shows the 'Animation' configuration window for 'Analog Value Display' in a software environment. The window is titled 'Animation' and has a red circle '2' next to its title. The 'Value Display' section is expanded, and 'AnalogValueDisplay' is selected, highlighted with a yellow background. A red circle '3' is next to this selection. Below the animation list, the 'Analog Value Display' dialog box is open. It has an 'Expression' field containing 'Var.Variable0' with a red circle '5' next to it. To the right of the field is a button with three dots and a red circle '4', and a 'Clear' button. Below the expression field are 'Digit' settings: 'IntegerDigits' set to '1' and 'DecimalDigits' set to '0', which is highlighted with a red box and a red circle '6'. At the bottom, there are checkboxes for 'Scientific Notation' and 'Thousands Separator'. At the very bottom are 'OK' and 'Cancel' buttons, with a red circle '7' next to the 'OK' button. A red arrow points from the 'AnalogValueDisplay' selection in the animation list to the dialog box.

Analog Value Display Animation



# Value Input Event—Analog Value Input

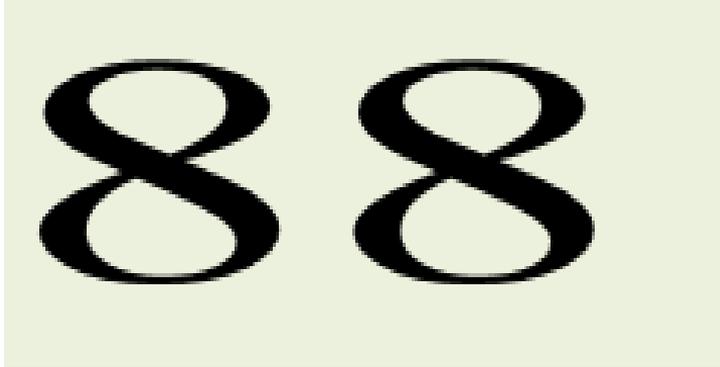
(3) Run the Window0. Click Text0 to input 88 in the Text0





## Value Input Event—Analog Value Input

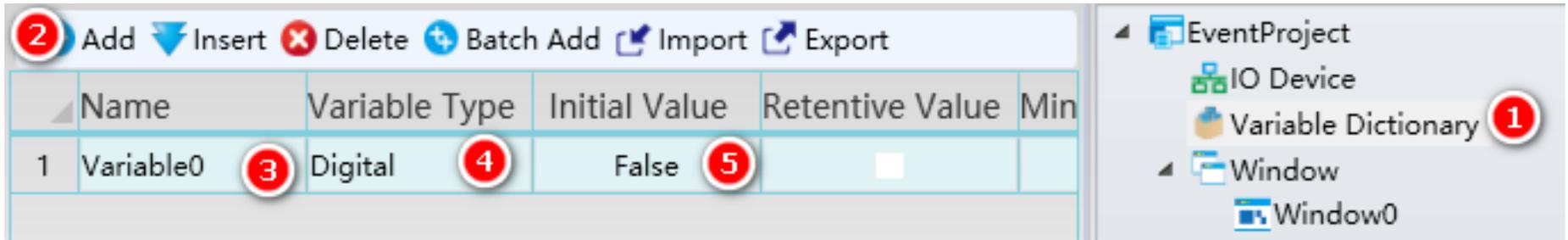
(4) The Text0 displays 88.



## ➤ Discrete Value Input event example

Input a discrete value in the text with the discrete value input event :

(1) Create a digital variable: Variable0



The screenshot shows a software interface with a table of variables and a project tree on the right. The table has columns for Name, Variable Type, Initial Value, Retentive Value, and Min. The first row contains 'Variable0', 'Digital', 'False', and a checkbox. The project tree on the right shows a hierarchy: EventProject > IO Device > Variable Dictionary (1) > Window > Window0.

	Name	Variable Type	Initial Value	Retentive Value	Min
1	Variable0	Digital	False	<input type="checkbox"/>	

※Refer to the section "6.3 Variables" in user manual.

(2) Create a Text0 in the Window0. Configure event and animation of the Text0

The screenshot shows the 'Event' configuration window for a 'Discrete Value Input' event. The background shows a window with three '#' symbols. The event list includes 'CloseWindow', 'OpenWindowCloseOthers', and 'Value Input', with 'DiscreteValueInput' selected. The 'Discrete Value Input' dialog box is open, showing the following configuration:

- Variable: Var.Variable0
- Button Content: SetTrue: True, SetFalse: False
- Buttons: OK, Cancel

Red numbered callouts (1-8) highlight specific elements: 1 (background symbols), 2 (Event window title), 3 (DiscreteValueInput in list), 4 (More options button), 5 (Variable field), 6 (SetTrue value), 7 (SetFalse value), and 8 (OK button).

Analog Value Input event

The screenshot shows the 'Animation' configuration window for a 'Discrete Value Display' animation. The background shows a window with three '#' symbols. The animation list includes 'Value Display', 'TextDisplay', 'AnalogValueDisplay', 'AnalogValueStringDisplay', and 'DiscreteValueDisplay', with 'DiscreteValueDisplay' selected. The 'Discrete Value Display' dialog box is open, showing the following configuration:

- Expression: Var.Variable0
- When True, DisplayString: True
- When False, DisplayString: False
- Buttons: OK, Cancel

Red numbered callouts (1-8) highlight specific elements: 1 (background symbols), 2 (Animation window title), 3 (DiscreteValueDisplay in list), 4 (More options button), 5 (Expression field), 6 (When True, DisplayString value), 7 (When False, DisplayString value), and 8 (OK button).

Discrete Value Display Animation

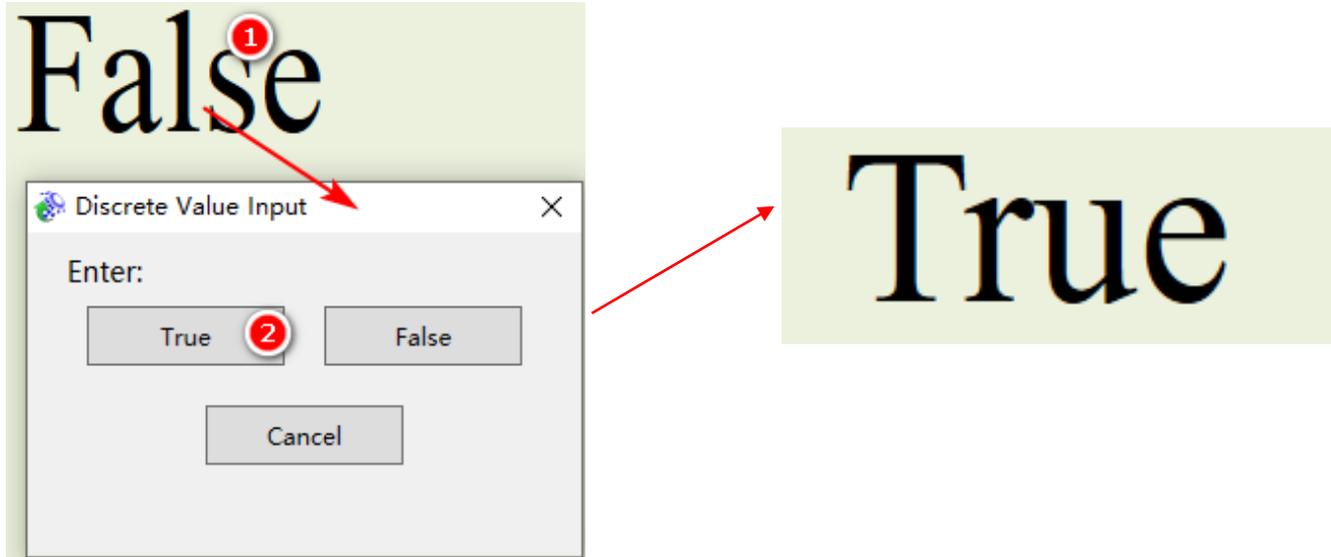


## Value Input Event—Discrete Value Input

(3) Run the Window0. The Text0 displays “False” default

False

(4) Click the Text0 to input True in it, then the Text0 displays “True”



## ➤ String Input event example

Input a string in the text with the string input event :

(1) Create a string variable: Variable0



	Name	Variable Type	Initial Value	Retentive Value	Minimum Value
1	Variable0	String	StringTest		N

- EventProject
  - IO Device
  - Variable Dictionary (1)
  - Window
    - Window0

✘ Refer to the section "6.3 Variables" in user manual.

(2) Create a Text0 in the Window0. Configure event and animation of the Text0

The screenshot shows the 'String Input' configuration dialog. The background window 'Window0' contains three '#' symbols, with a red circle '1' highlighting one of them. The 'Event' panel on the right has 'StringInput' selected, with a red circle '2' above it and a red arrow '3' pointing to the dialog. The dialog has a 'Variable' field containing 'Var.Variable0' with a red circle '5' next to it, and a 'Clear' button with a red circle '4' next to it. There is a 'Show Password' checkbox and an 'Input Panel Size' section with radio buttons for 'Default' (selected), 'Adapt', and 'Full Screen'. At the bottom are 'OK' (with red circle '6') and 'Cancel' buttons.

String Input event

The screenshot shows the 'Text Display' configuration dialog. The background window 'Window0' contains three '#' symbols, with a red circle '1' highlighting one of them. The 'Animation' panel on the right has 'TextDisplay' selected, with a red circle '2' above it and a red arrow '3' pointing to the dialog. The dialog has an 'Expression' field containing 'Var.Variable0' with a red circle '5' next to it, and a 'Clear' button with a red circle '4' next to it. At the bottom are 'OK' (with red circle '6') and 'Cancel' buttons.

Text Display Animation



## Value Input Event—String Input

(3) Run the Window0. The Text0 displays “StringTest” default

StringTest

(4) Click the Text0 to input Good in it, then the Text0 displays “Good”



Good

## ➤ Button Input event example1

Set value of a variable with the button input event :

(1) Create a string variable: Variable0



The screenshot displays the software interface for creating a variable. On the left, a table lists the variable 'Variable0' with an initial value of 0. On the right, a project tree shows the 'Variable Dictionary' folder selected. Red circles and arrows highlight the steps: 1. Selecting 'Variable Dictionary' in the project tree, 2. Clicking the 'Add' button, 3. Entering 'Variable0' in the 'Name' field, and 4. Selecting 'Analog' in the 'Variable Type' field.

	Name	Variable Type	Initial Value	Retentive Value	M
1	Variable0	Analog	0		

- EventProject
  - IO Device
  - Variable Dictionary (1)
  - Window
    - Window0

※Refer to the section "6.3 Variables" in user manual.

# Value Input Event—Button Input

(2) Create a Text0 in the Window0. Configure event and animation of the Text0

Window0 x Variable Dictionary Event 2

- OpenWindowCloseC
- Value Input
  - AnalogValueInput
  - DiscreteValueInput
  - StringInput
  - ButtonInput

Button Input

Variable: Var.Variable0 5

Clear 4

InputValue: 5.00 6

ValueType: Set Value 7

OK 8 Cancel

Button Input event

Window0 x Vari Animation 2

- Visibility
  - Visibility
  - Blink
- Value Display
  - TextDisplay
  - AnalogValueDisplay

Analog Value Display

Expression: Var.Variable0 5

Clear 4

Digit

IntegerDigits: 1

DecimalDigits: 0 6

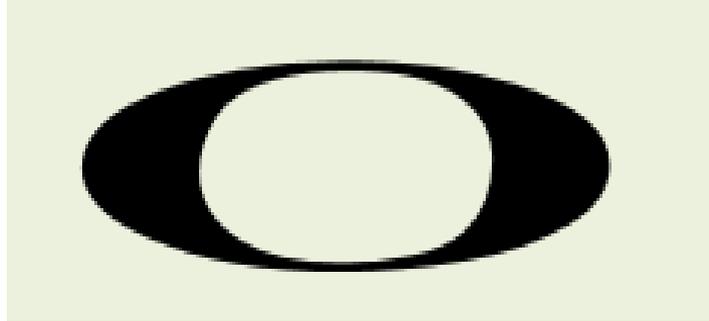
Scientific Notation

Thousands Separator

OK 7 Cancel

Analog Value Display Animation

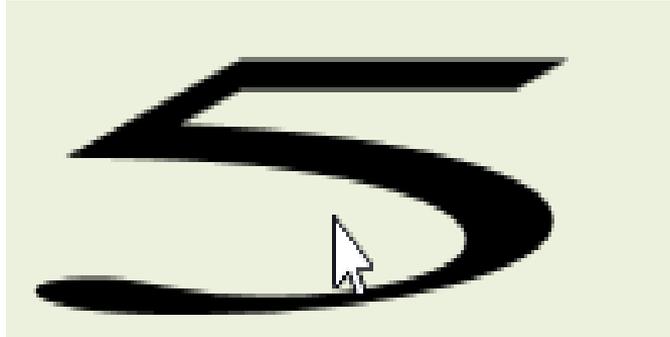
(3) Run the Window0. The Text0 displays “0” default





## Value Input Event—Button Input

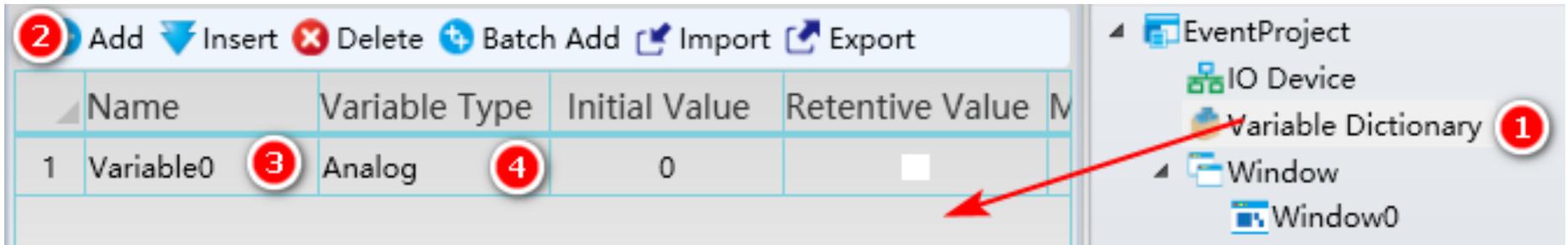
(4) Click the Text0, then the Text0 displays “5”



## ➤ Button Input event example2

Set value of a variable by increasing 5 each time:

(1) Create a analog variable: Variable0



The screenshot shows a software interface with a table of variables and a project tree on the right. The table has columns for Name, Variable Type, Initial Value, and Retentive Value. The first row shows 'Variable0' with an 'Analog' type and an initial value of '0'. The project tree on the right shows a hierarchy: EventProject > IO Device > Variable Dictionary > Window > Window0. A red arrow points from the 'Variable Dictionary' node in the tree to the 'Variable0' row in the table. Red circles with numbers 1 through 4 highlight specific elements: 1 on the 'Variable Dictionary' node, 2 on the 'Add' button, 3 on the 'Variable0' cell, and 4 on the 'Analog' cell.

	Name	Variable Type	Initial Value	Retentive Value
1	Variable0	Analog	0	

EventProject  
IO Device  
Variable Dictionary  
Window  
Window0

※Refer to the section "6.3 Variables" in user manual.

(2) Create a Text0 in the Window0. Configure event and animation of the Text0

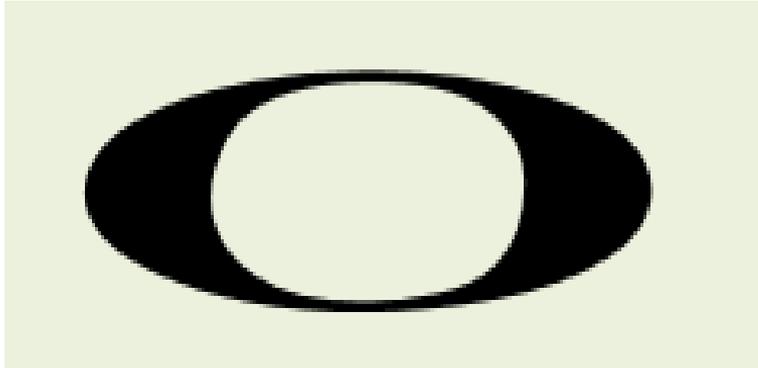
This screenshot shows the configuration of a Button Input event for a text object in a window. The main window displays three '#' characters. A red circle '1' is placed over the first character. An 'Event' panel is open, showing a list of events with 'ButtonInput' selected and highlighted in yellow. A red circle '2' is placed over the 'Event' panel title. A red arrow points from the 'ButtonInput' event to a 'Button Input' dialog box. In this dialog box, the 'Variable' field contains 'Var.Variable0' with a red circle '5' next to it. A red circle '4' is placed over the selection button to the right of the variable field. The 'InputValue' field contains '5.00' with a red circle '6' next to it. The 'ValueType' dropdown is set to 'Increase' and is highlighted with a red box, with a red circle '7' next to it. At the bottom of the dialog box, the 'OK' button is highlighted with a red circle '8'.

Button Input event

This screenshot shows the configuration of an Analog Value Display animation for a text object in a window. The main window displays three '#' characters. A red circle '1' is placed over the first character. An 'Animation' panel is open, showing a list of animation types with 'AnalogValueDisplay' selected and highlighted in yellow. A red circle '2' is placed over the 'Animation' panel title. A red arrow points from the 'AnalogValueDisplay' animation to an 'Analog Value Display' dialog box. In this dialog box, the 'Expression' field contains 'Var.Variable0' with a red circle '5' next to it. A red circle '4' is placed over the selection button to the right of the expression field. The 'Digit' section has 'IntegerDigits' set to '1' and 'DecimalDigits' set to '0', with a red box around the '0' and a red circle '6' next to it. At the bottom of the dialog box, the 'OK' button is highlighted with a red circle '7'.

Analog Value Display Animation

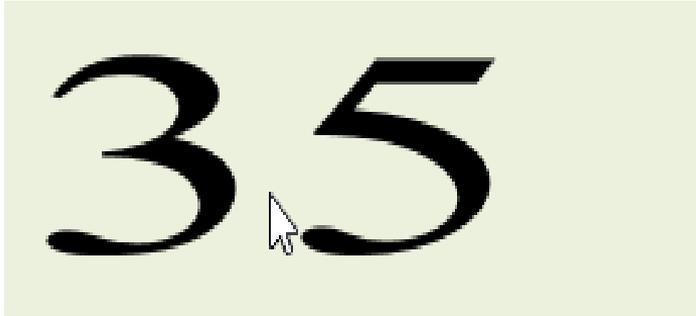
(3) Run the Window0. The Text0 displays “0” default





## Value Input Event—Button Input

(4) Click the Text0 multiple times, each time the display value of the Text0 is incremented by 5

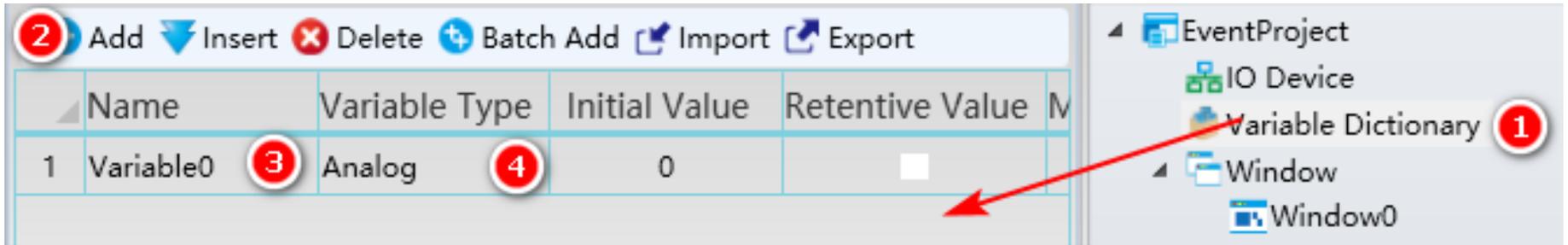


- The concepts of event
- Left button event
- Right button event
- Mouse event
- Window operation event
- Value input event
- Sliding input event
- **Rotation input event**
- Window program event
- Control event
- Keyboard

## ➤ Rotation Input event example

Create a rectangle that can be rotated:

(1) Create an analog variable: Variable0



	Name	Variable Type	Initial Value	Retentive Value	M
1	Variable0	Analog	0		

EventProject  
IO Device  
Variable Dictionary  
Window  
Window0

※Refer to the section "6.3 Variables" in user manual.



# Rotation Input Event—Rotation Input

(2) Create a Rectangle0 and a Text0 in the Window0. Configure rotation input event of Rectangle0 and analog value display animation of Text0

Window0 x Variable Dictionary

Event 2

- OpenWindowCloseOthers
- Value Input
  - AnalogValueInput
  - DiscreteValueInput
  - StringInput
  - ButtonInput
- Rotation Input
- Slide Input
  - HorizontalSlide

Rotation Input

Variable: Var.Variable0 5

Anti-clockwise Angle: 0.00 VariableValue: 0.00

Clockwise Angle: 100.00 VariableValue: 100.00

OperateMode: SequentialWrite 6

OK Cancel 7

Button Input event

Window0 x Vari

Animation 2

- Visibility
  - Visibility
  - Blink
- Value Display
  - TextDisplay
  - AnalogValueDisplay

Analog Value Display

Expression: Var.Variable0 5

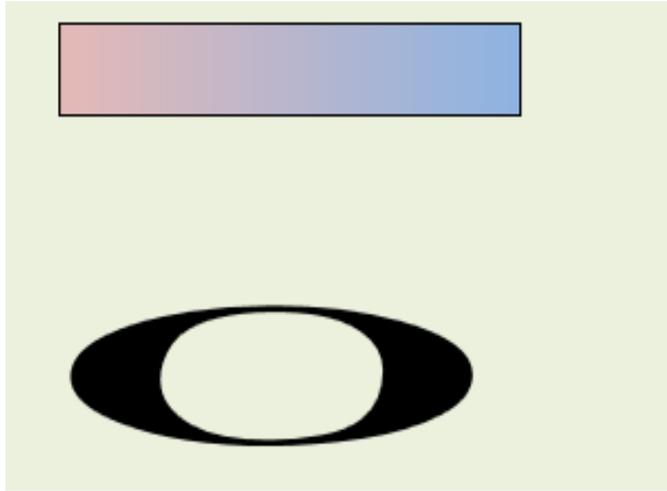
Digit IntegerDigits: 1 DecimalDigits: 0 6

Scientific Notation Thousands Separator

OK Cancel 7

Analog Value Display Animation

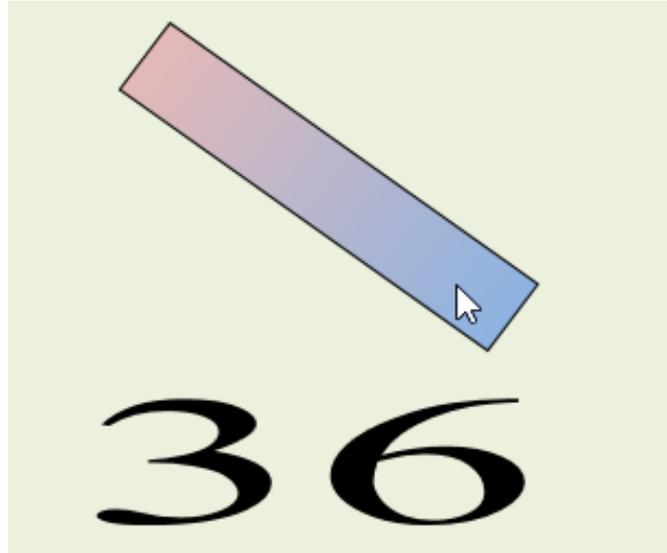
(3) Run the Window0. The initial display is as follows





## Rotation Input Event—Rotation Input

(4) Drag the Rectangle0 clockwise with the mouse. During the dragging process, the content of Text0 changes as the rotation angle changes, that is, Text0 displays the rotation angle of the Recatangle0 in real time

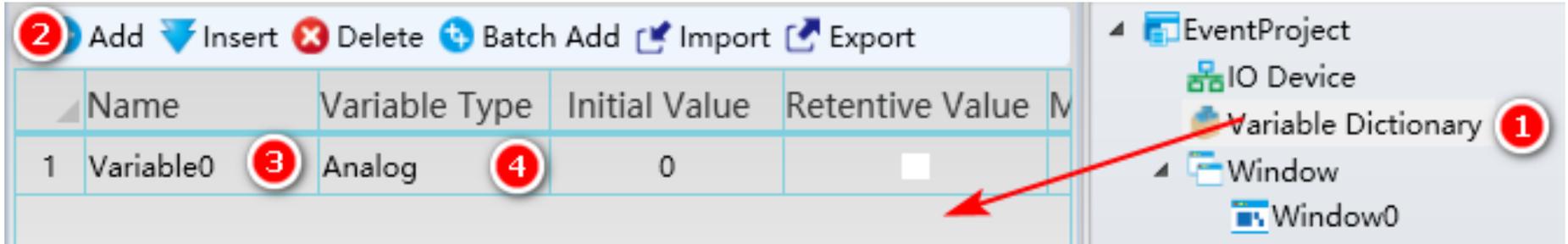


- The concepts of event
- Left button event
- Right button event
- Mouse event
- Window operation event
- Value input event
- Sliding input event
- Rotation input event
- **Window program event**
- Control event
- Keyboard

## ➤ Window Program event example

Create a window program to be executed at runtime:

(1) Create an analog variable: Variable0



The screenshot displays the software interface for configuring variables. On the left, a table lists the variables, and on the right, a project tree shows the hierarchy.

	Name	Variable Type	Initial Value	Retentive Value	M
1	Variable0	Analog	0	<input type="checkbox"/>	

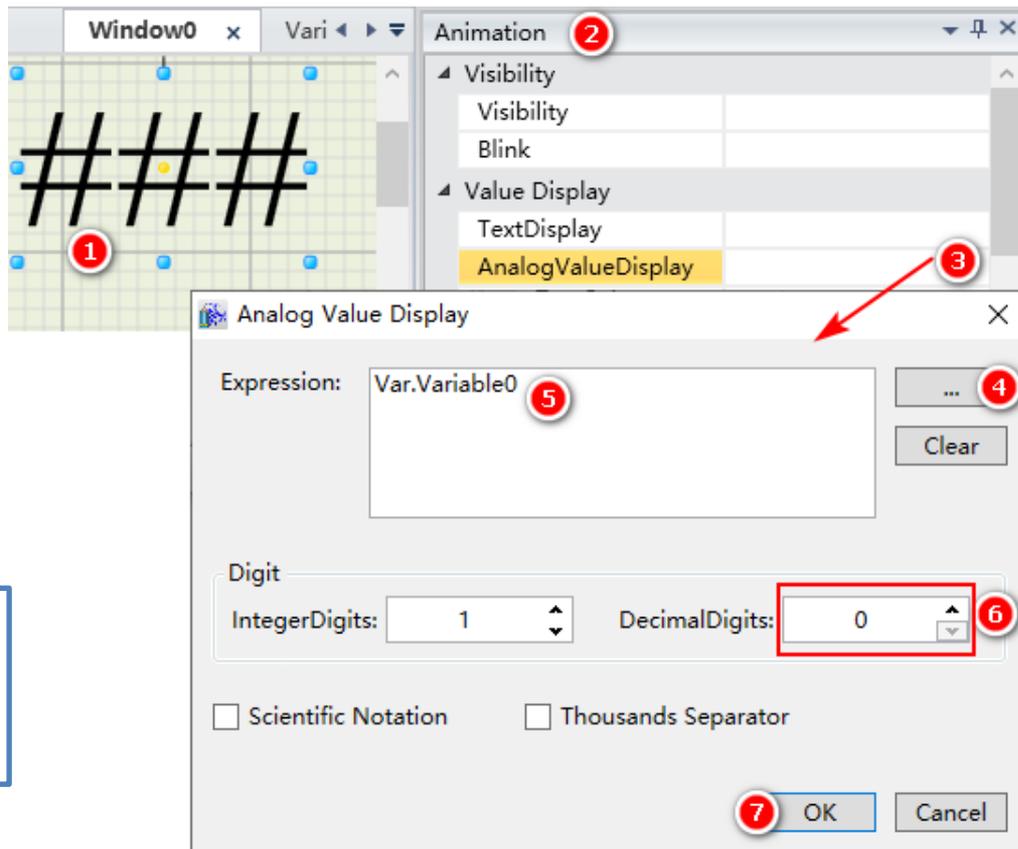
The project tree on the right shows the following structure:

- EventProject
  - IO Device
  - Variable Dictionary (1)
  - Window
    - Window0

Red circles with numbers 1, 2, 3, and 4 highlight specific elements: 1 points to the Variable Dictionary in the tree, 2 points to the 'Add' button, 3 points to the 'Variable0' cell in the table, and 4 points to the 'Analog' cell in the table. A red arrow points from the Variable Dictionary in the tree to the 'Variable0' row in the table.

※Refer to the section "6.3 Variables" in user manual.

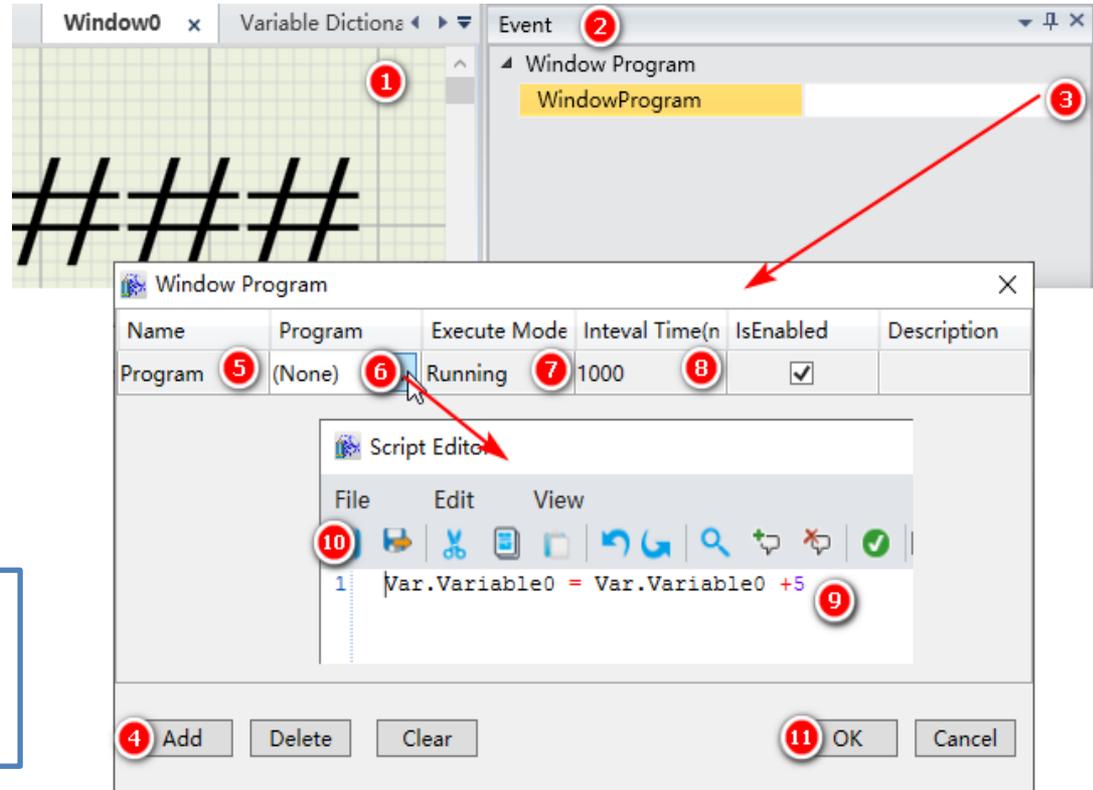
(2) Create a Text0 in the Window0. The analog value display animation of the Text0 associated Variable0



※ Refer to the section “9.9 Value display animation” in user manual.

# Window Program Event—Window Program

(3) Create a window program in the Window0 (Refer to the section “10.9 Window program event” in user manual)



- ① Click any blank space in Window0
- ② Open event window
- ⑤ Name the window program
- ⑦⑧ Set to execute every second at run time

※ Refer to the section “10.9 Window program event” in user manual.



## Window Program Event—Window Program

(4) Run the Window0. The Text0 initially displays 0, and the displayed value increases by 5 per second

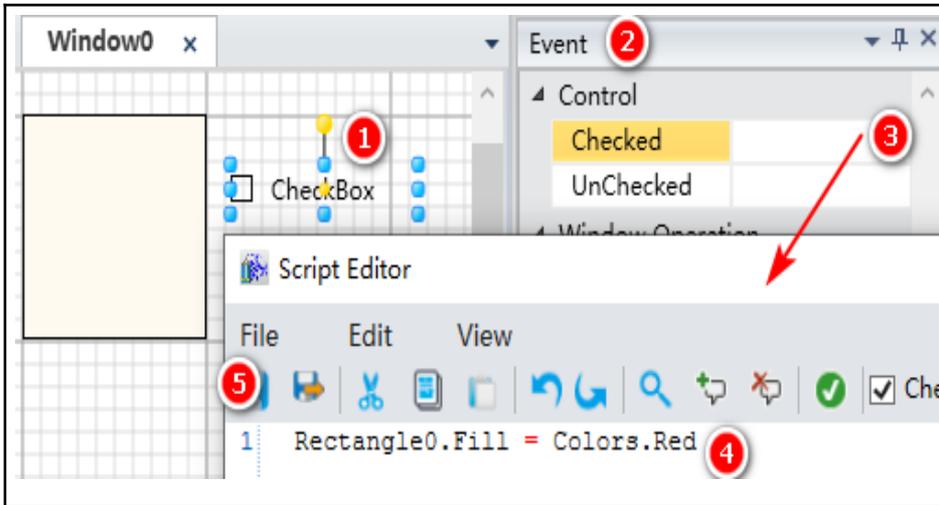
A light green rectangular box containing the number '35' in a large, black, serif font.

- The concepts of event
- Left button event
- Right button event
- Mouse event
- Window operation event
- Value input event
- Sliding input event
- Rotation input event
- Window program event
- **Control event**
- Keyboard

## ➤ Checked/UnChecked event example

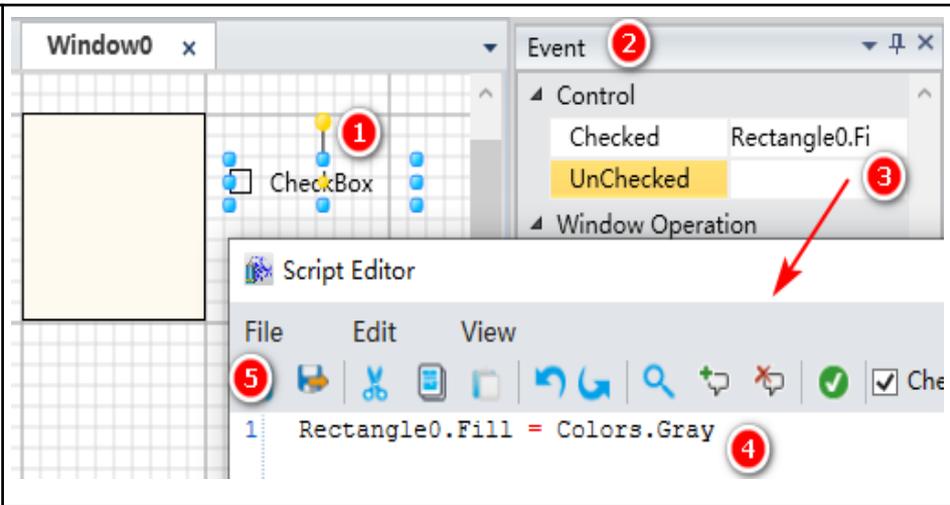
Set rectangle color with the control event of checkbox:

(1) Create a CheckBox0 and a Rectangle0 in the Window0. Configure the control event of CheckBox0



The screenshot shows the IDE interface for 'Window0'. A yellow rectangle is on the left. A checkbox is on the right. The 'Event' window is open, showing the 'Checked' event selected. A red arrow points from the 'Checked' event to the 'Script Editor'. The script editor shows the code: `1: Rectangle0.Fill = Colors.Red`. A red circle with the number 5 is on the 'File' menu, and a red circle with the number 4 is on the code line.

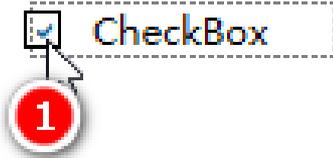
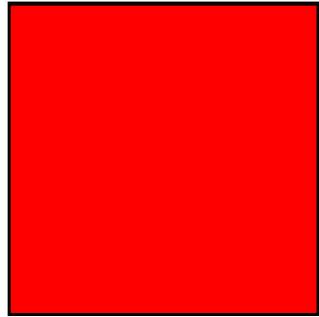
Red



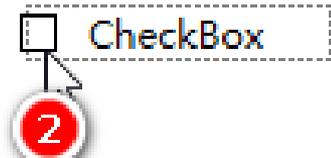
The screenshot shows the IDE interface for 'Window0'. A yellow rectangle is on the left. A checkbox is on the right. The 'Event' window is open, showing the 'UnChecked' event selected. A red arrow points from the 'UnChecked' event to the 'Script Editor'. The script editor shows the code: `1: Rectangle0.Fill = Colors.Gray`. A red circle with the number 5 is on the 'File' menu, and a red circle with the number 4 is on the code line.

Gray

(2) Run the Window0



Red



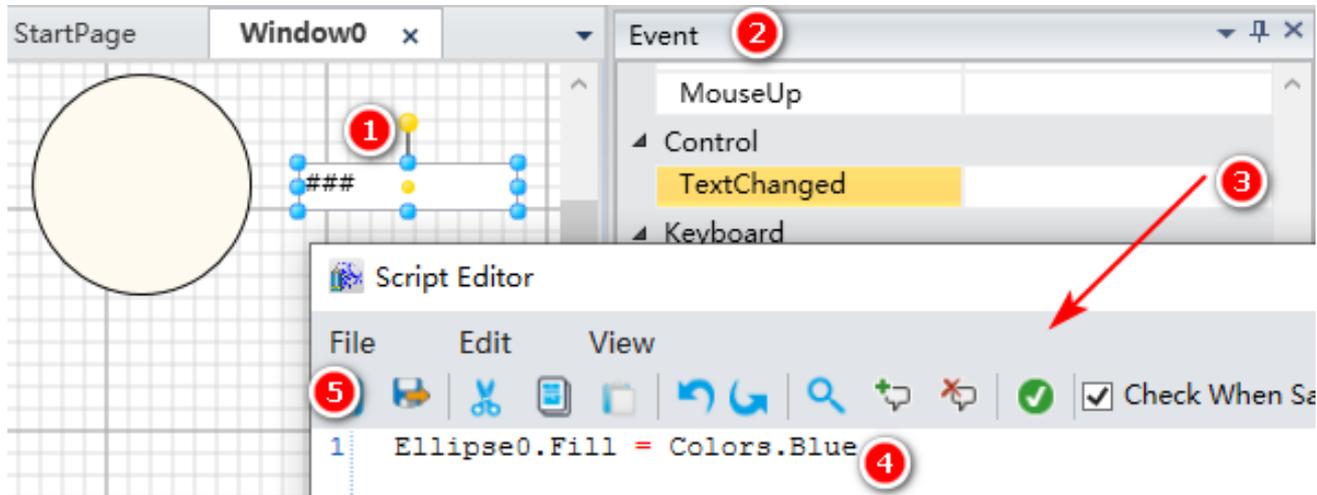
Gray

- ① Check the Checkbox0, the Rectangle0 turns red
- ② Uncheck the Checkbox0, the Rectangle0 turns gray

## ➤ Text Change event example

Set ellipse color with the control event of text box:

(1) Create a Ellipse0 and a TextBox0 in the Window0. Configure the text changed event of TextBox0.



The screenshot displays a software development environment with a grid-based workspace. On the left, a yellow circle (Ellipse0) is visible. In the center, a text box (TextBox0) is connected to a script editor. The script editor shows the following code:

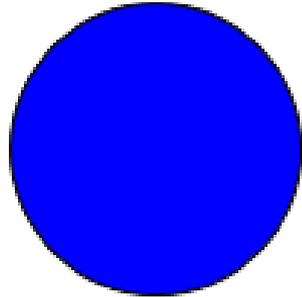
```
1 Ellipse0.Fill = Colors.Blue
```

The Event window on the right shows the following configuration:

- Event: MouseUp
- Control: TextChanged
- Keyboard: (empty)

Red circles with numbers 1 through 5 highlight key elements: 1 (connection point), 2 (Event window), 3 (TextChanged event), 4 (code line), and 5 (Script Editor icon).

(2) Run the Window0. Change the content of TextBox0, then the Ellipse0 turns blue.



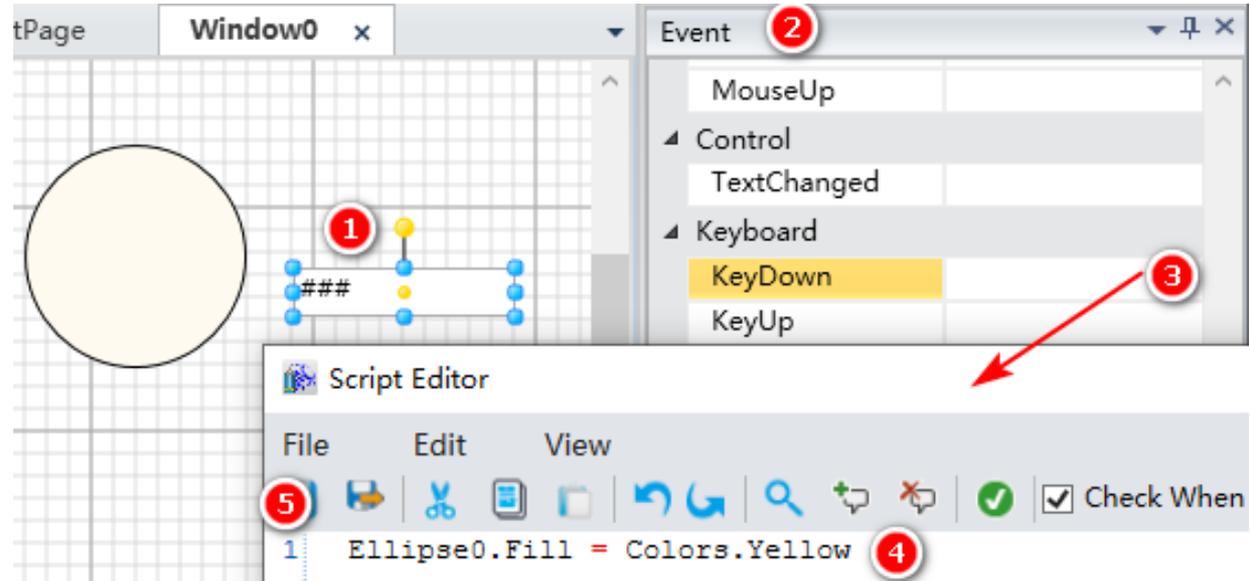
###test

- The concepts of event
- Left button event
- Right button event
- Mouse event
- Window operation event
- Value input event
- Sliding input event
- Rotation input event
- Window program event
- Control event
- **Keyboard**

## ➤ Key Down event example

Set rectangle color with the control event of text box:

(1) Create a Ellipse0 and a TextBox0 in the Window0. Configure the key down event of TextBox0.



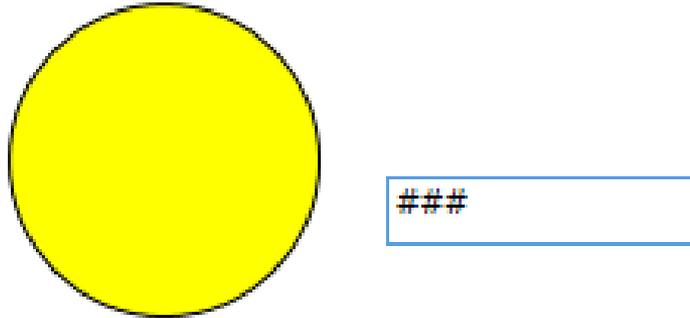
The screenshot displays a software development environment with a grid-based workspace. On the left, a yellow circle (Ellipse0) and a text box (TextBox0) are visible. A red circle '1' is placed over the text box. To the right, an 'Event' panel is open, showing a list of events. The 'Keyboard' category is expanded, and 'KeyDown' is highlighted with a yellow background. A red circle '2' is placed over the 'Event' panel title, and a red circle '3' is placed over the 'KeyDown' event. Below the event panel, a 'Script Editor' window is open, showing a menu bar (File, Edit, View) and a toolbar with icons for file operations and a 'Check When' checkbox. A red circle '5' is placed over the 'File' menu, and a red circle '4' is placed over the script editor. The script editor contains the following code:

```
1 Ellipse0.Fill = Colors.Yellow
```



## Keyboard Event—Key Down

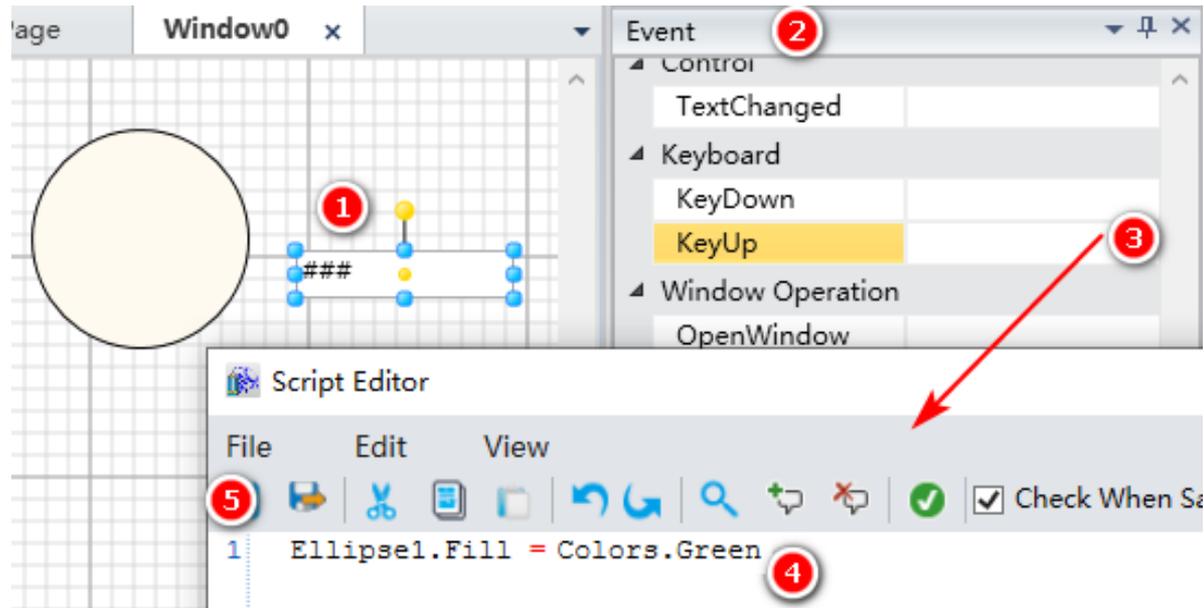
(2) Run the Window0. When mouse focus is in TextBox0, press any key on the keyboard, the Ellipse0 turns yellow



## ➤ Key Up event example

Set rectangle color with the control event of text box:

(1) Create a Ellipse0 and a TextBox0 in the Window0. Configure the key up event of TextBox0.



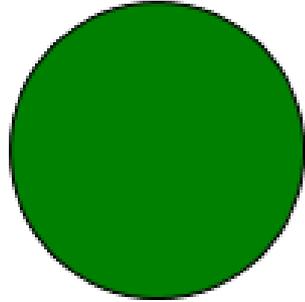
The screenshot illustrates the configuration of a Key Up event in a software development environment. It shows a visual programming interface with a grid and a script editor.

- 1**: A yellow circle (Ellipse0) and a text box (TextBox0) are placed on the grid. A red circle with the number 1 is positioned near the text box.
- 2**: The Event panel on the right shows the Keyboard section expanded, with the KeyUp event selected and highlighted in yellow. A red circle with the number 2 is positioned above the Event panel.
- 3**: A red arrow points from the KeyUp event in the Event panel to the script editor.
- 4**: The script editor shows the following code: `Ellipse1.Fill = Colors.Green`. A red circle with the number 4 is positioned below the code.
- 5**: A red circle with the number 5 is positioned above the script editor's toolbar.



## Keyboard Event—Key Up

(2) Run the Window0. When mouse focus is in TextBox0 , press any key on the keyboard, the colour of the Ellipse0 does not change. When the key is released, the Ellipse0 turns green.



# Smarter. Greener. Together.

To learn more about Delta, please visit  
[www.deltaww.com](http://www.deltaww.com).

