How to Suppress Key Press

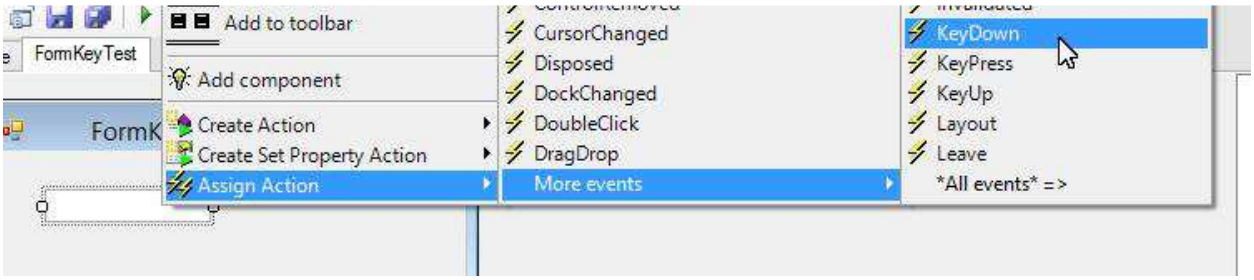
Contents

Scenario

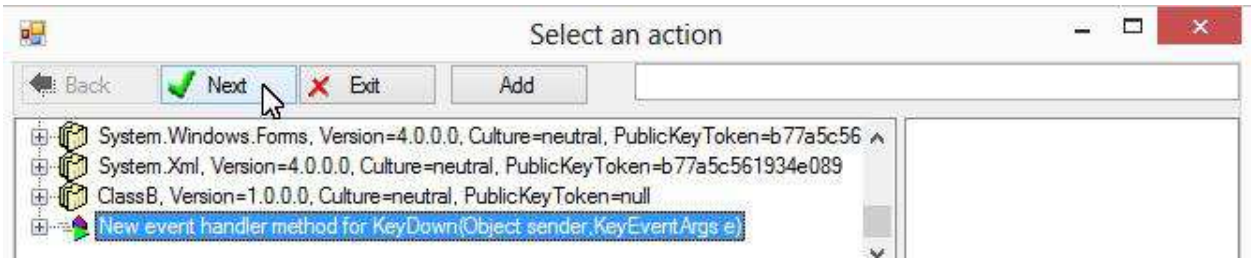
Suppose you use a text box for getting user inputs and you want to limit what values the user is allowed to enter. One way to do it is by handling KeyDown event and use SuppressKeyPress to suppress unwanted key presses. Below is such an example.

Event Handler Method

Right-click a text box, choose “Assign Action”, choose “KeyDown”:



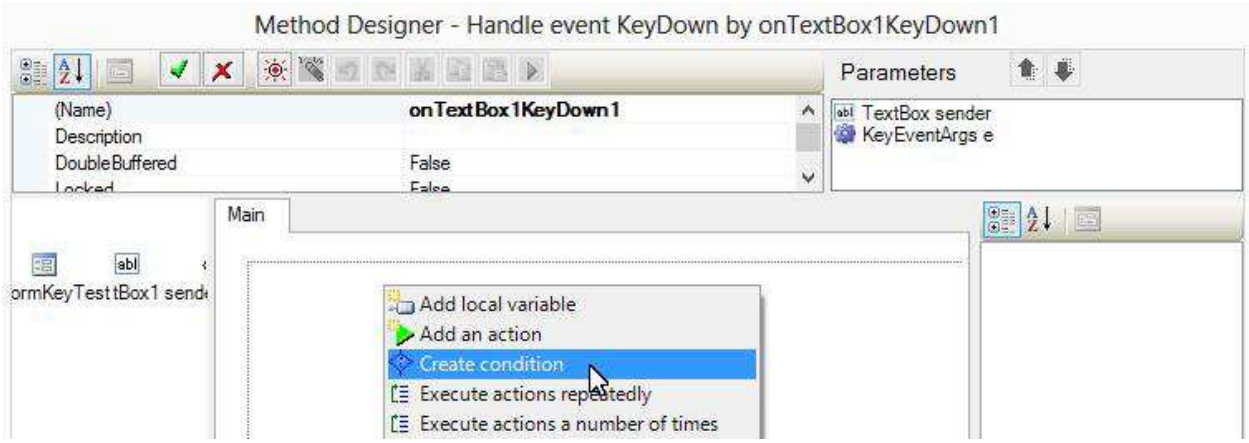
Select “New event handler method for KeyDown” and click Next:



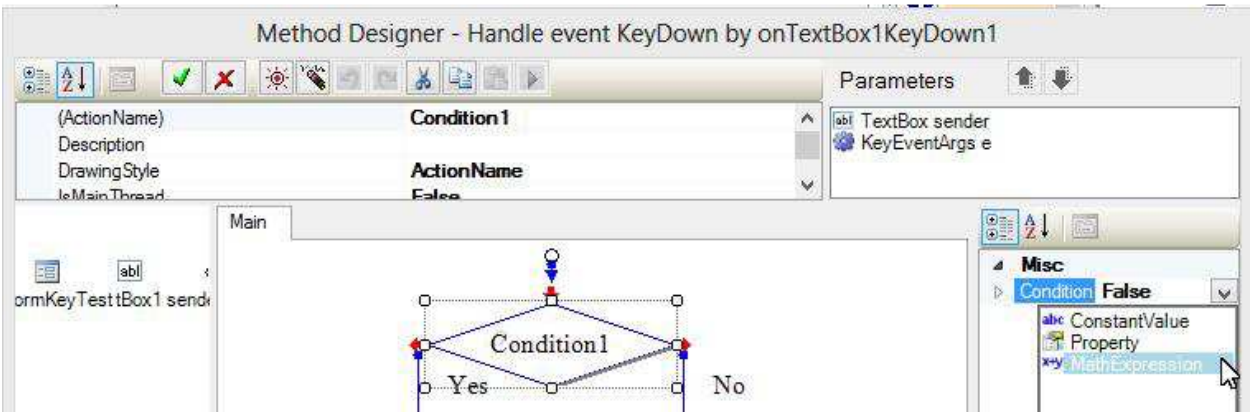
A method editor appears for editing the new event handler method.

Check keyboard value

Add a Condition action to check allowed keyboard value:



Use “Math Expression” to specify allowed keyboard values:

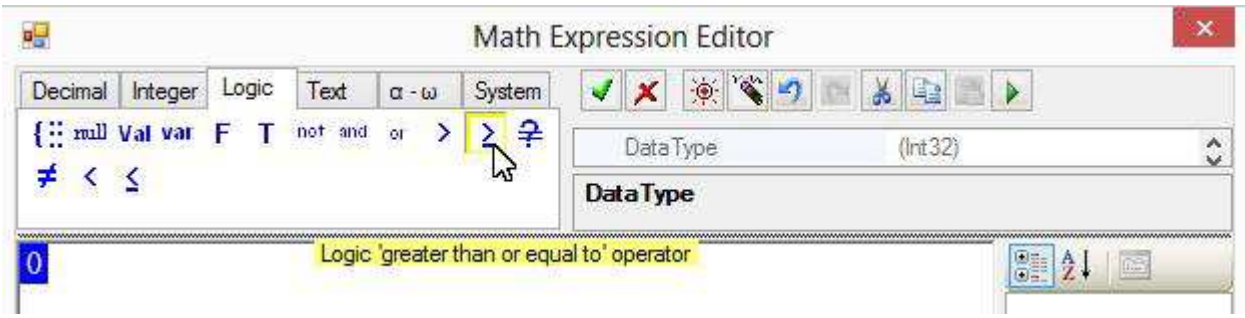


An expression editor appears for building an expression to be used as the condition.

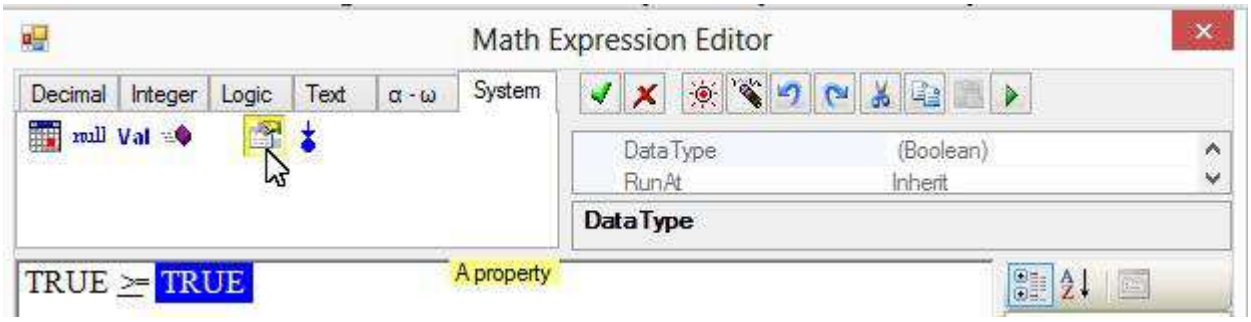
Suppose we want to allow only following characters: from 0 to 9, “(“, “)”, and “-“. Let’s create an expression to allow these characters.

Pay attention to which elements are highlighted when doing expression editing.

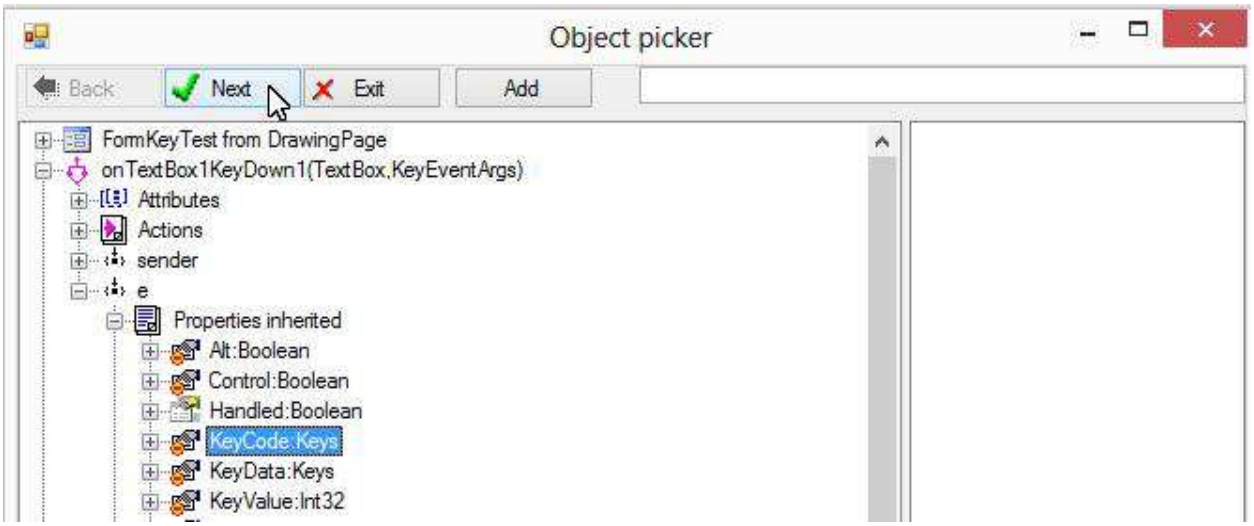
First, create a range by using a >= :



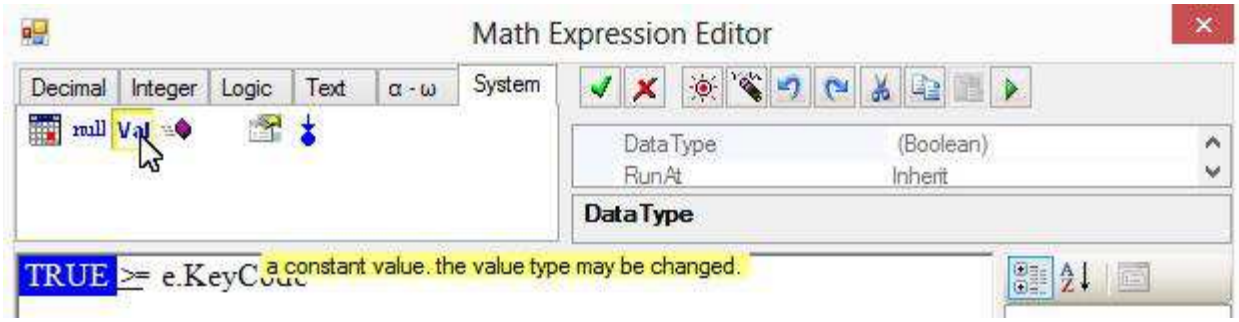
Select Property icon to use key pressed:



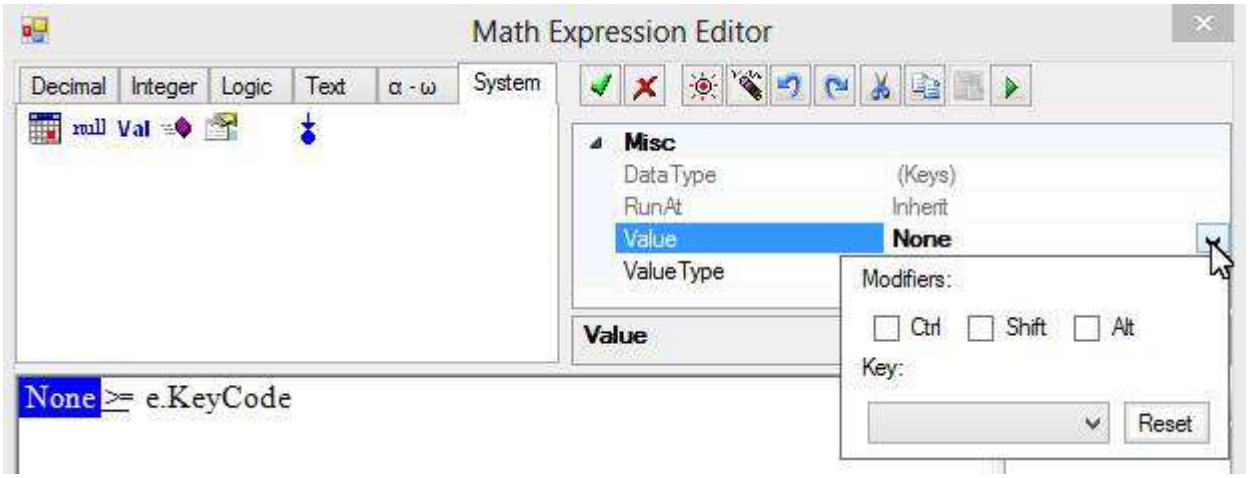
Choose KeyCode of event parameter e:

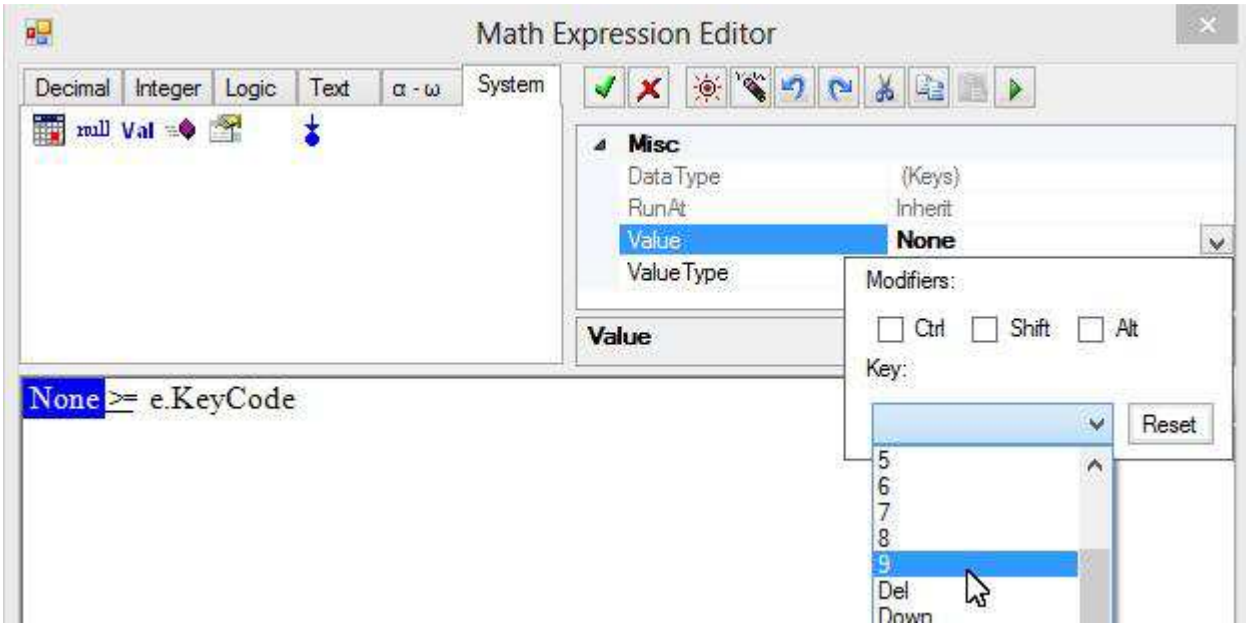


Select Constant Value icon so that we may specify “9”:

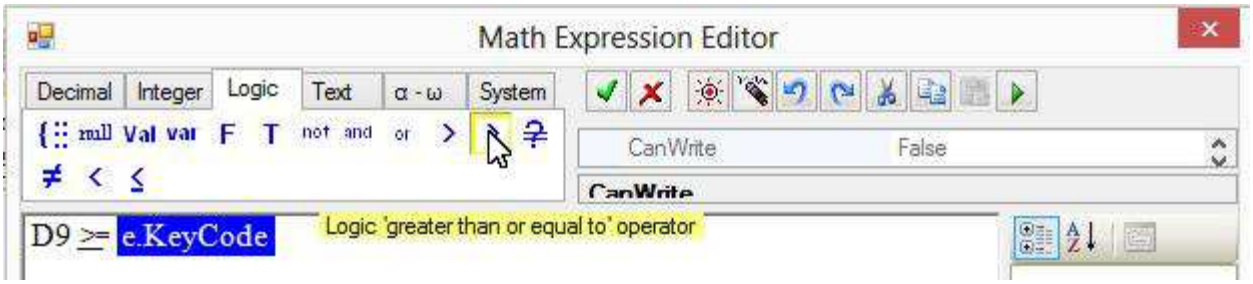


Specify key code for “9”:

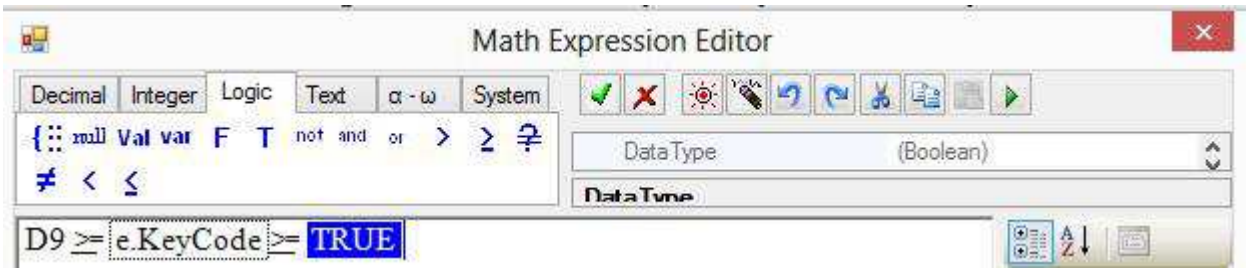




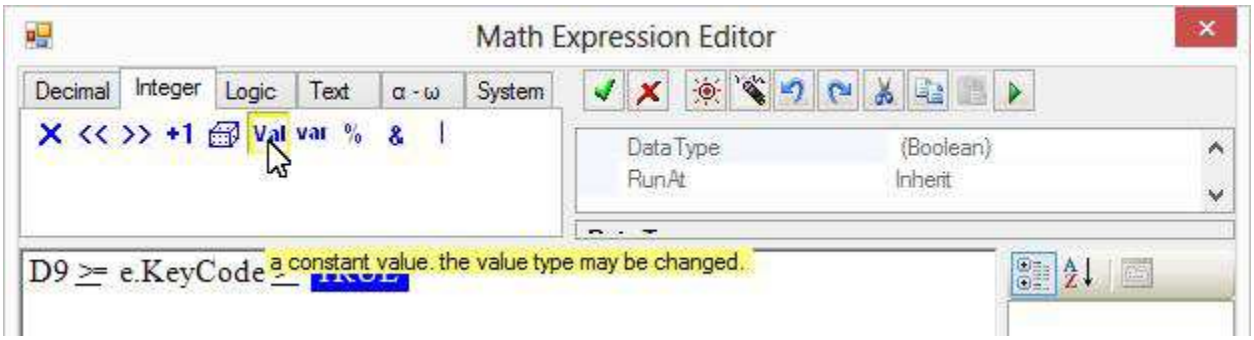
Select element e.KeyCode so that we may specify another >=:



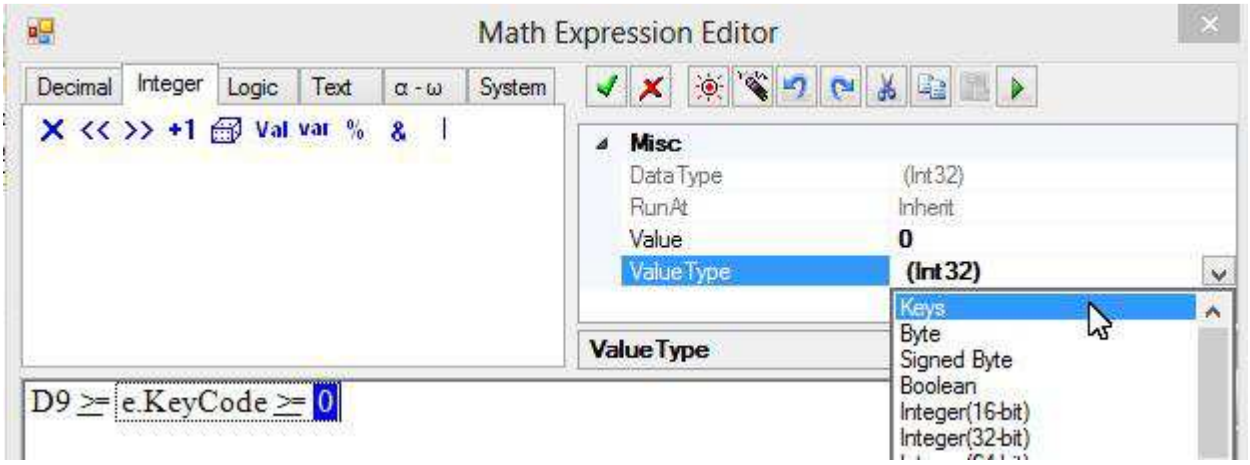
Select the element to be compared:



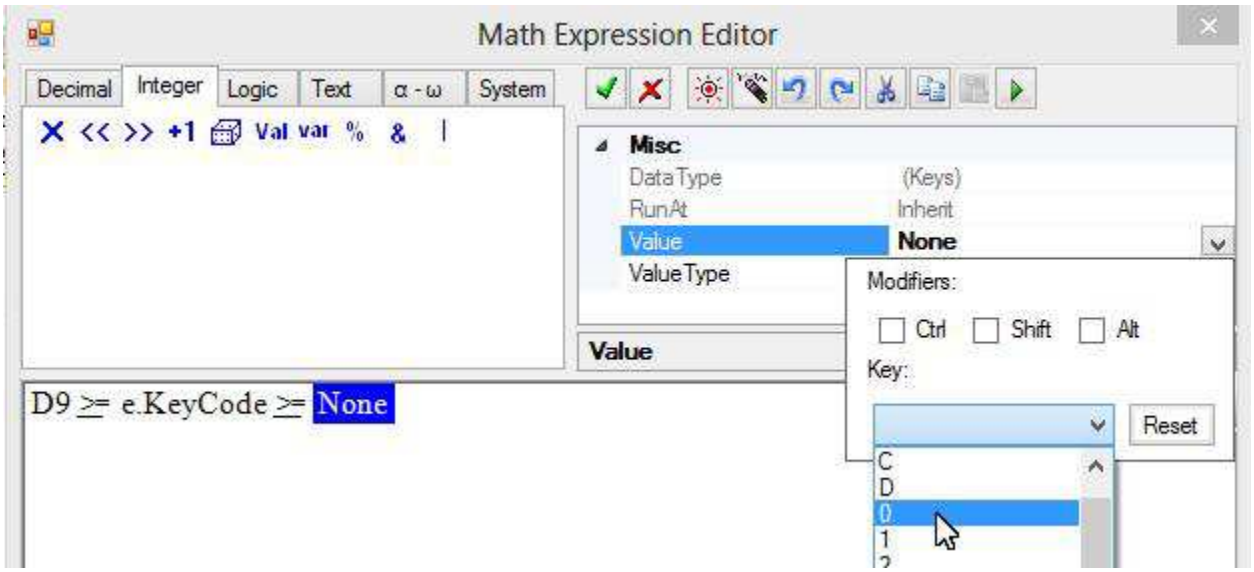
Click Constant Value icon so that we may specify “0”:



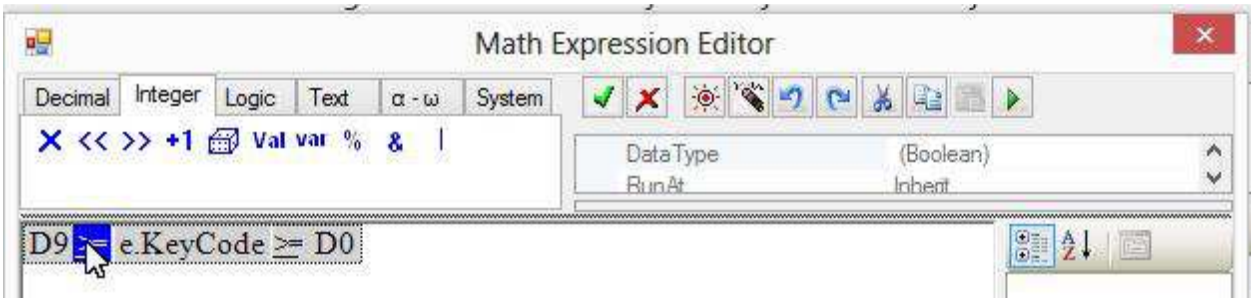
We need to change the value type to Keys:



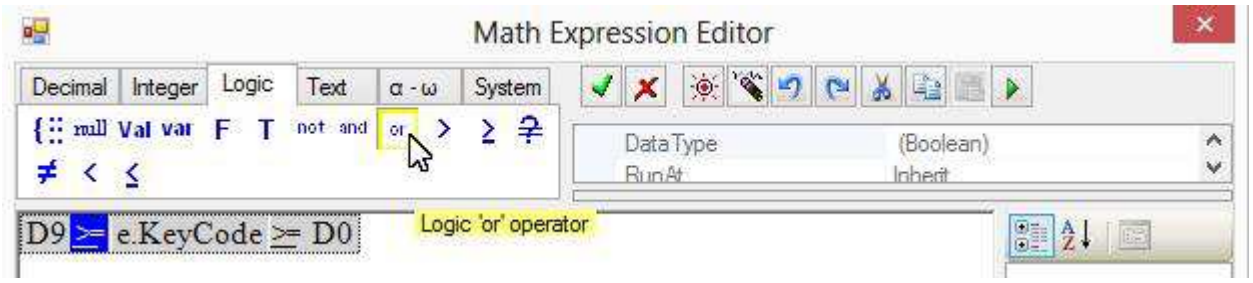
Specify key 0:



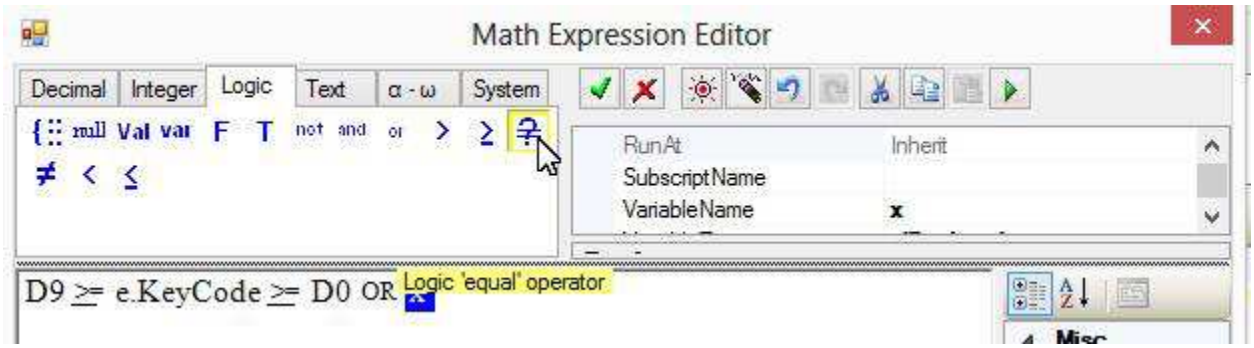
Now the expression is built for key code between 9 and 0. We want to add another condition to allow "(", this condition is an OR to the existing conditions. To add an OR to the existing expression, we must select the whole expression first, that is, the whole expression must be highlighted:



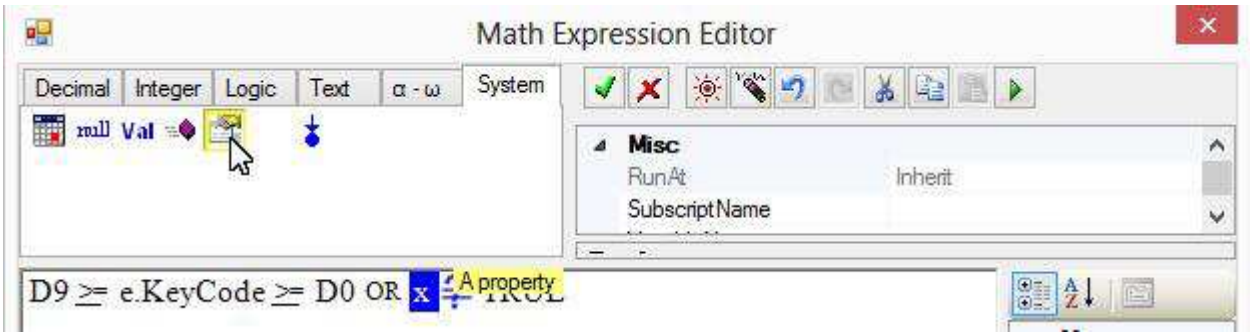
Click OR icon:



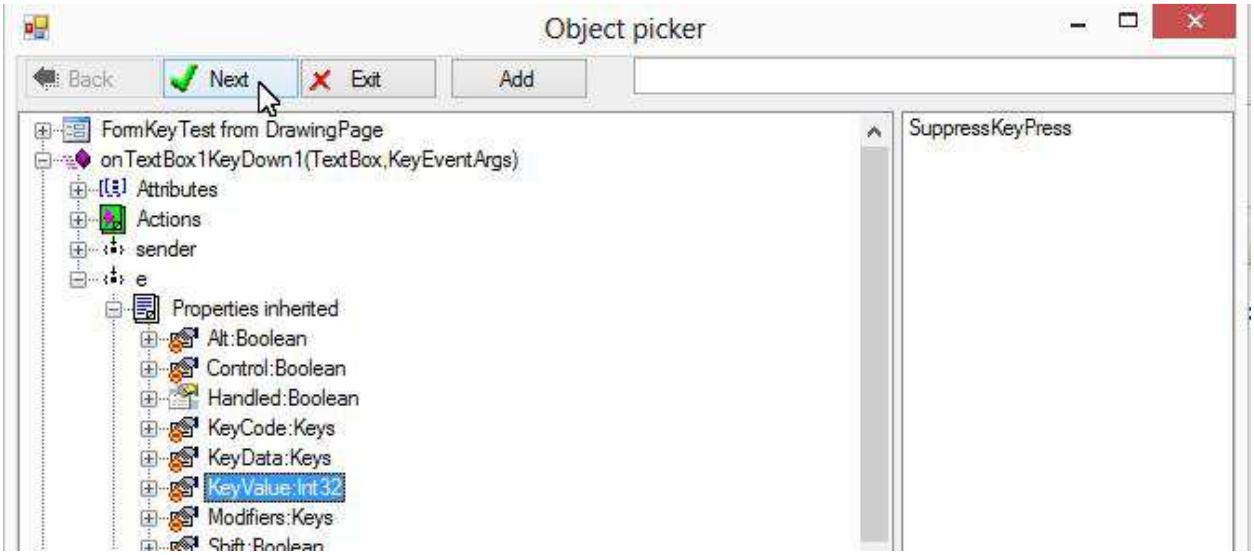
Click x after OR and click =?= icon to add an equality checking condition:



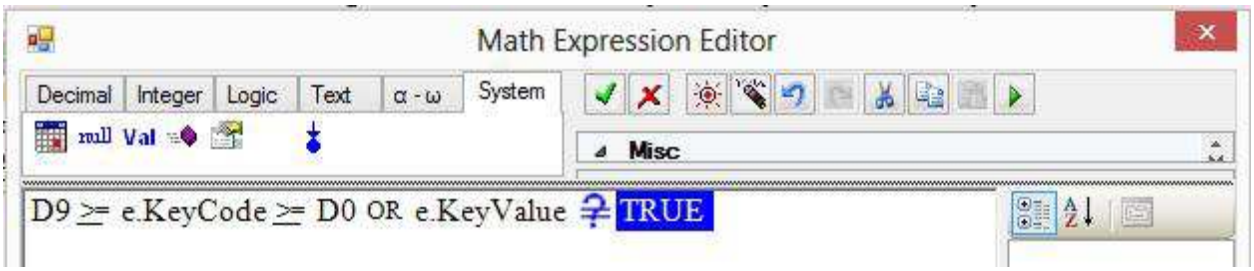
Highlight x and click Property icon to choose key value:

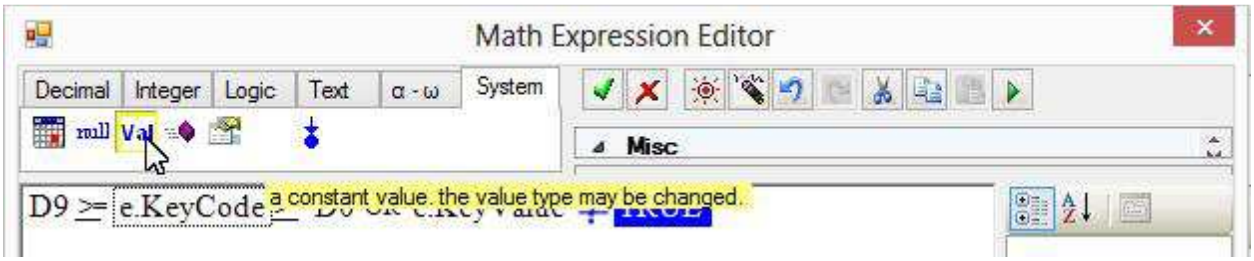


Select KeyValue which is ASCII code of the key pressed:

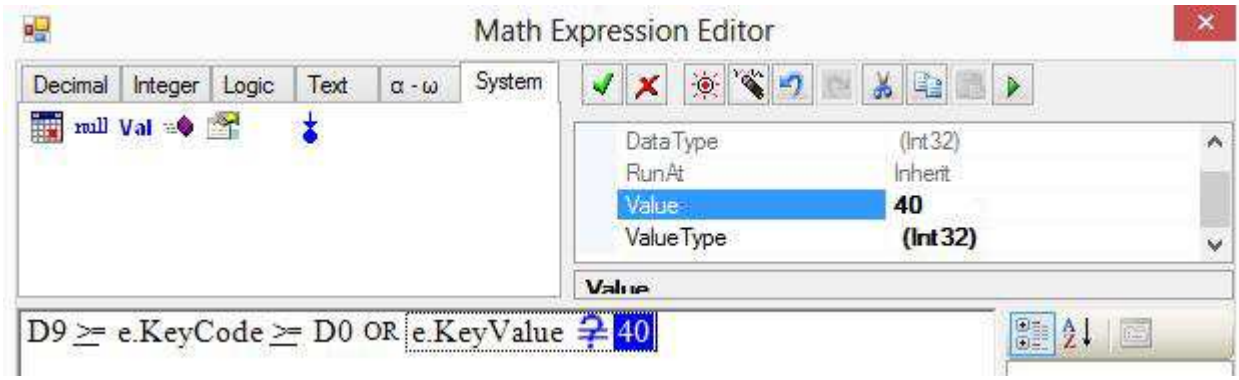


Select the value to be compared and click Constant Value icon:

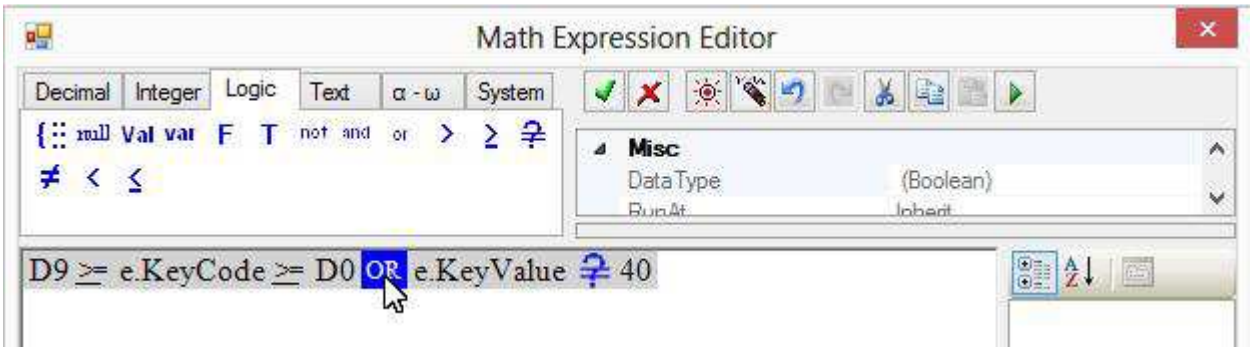


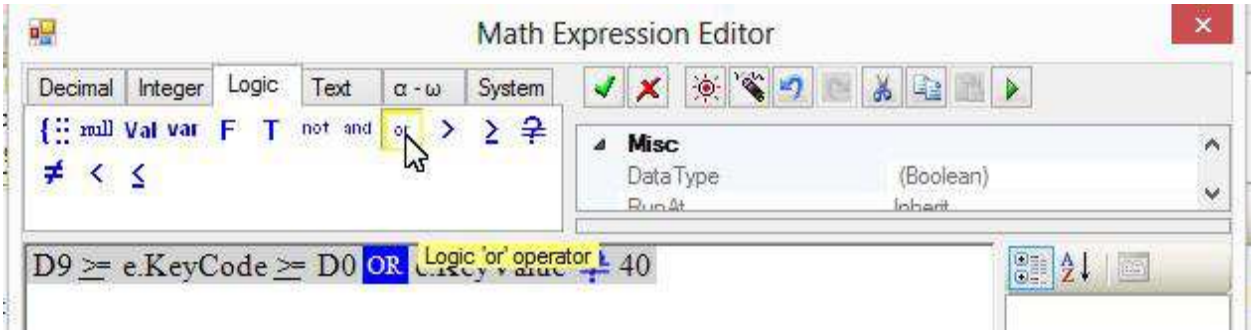


Specify 40 because ASCII code for "(" is 40:

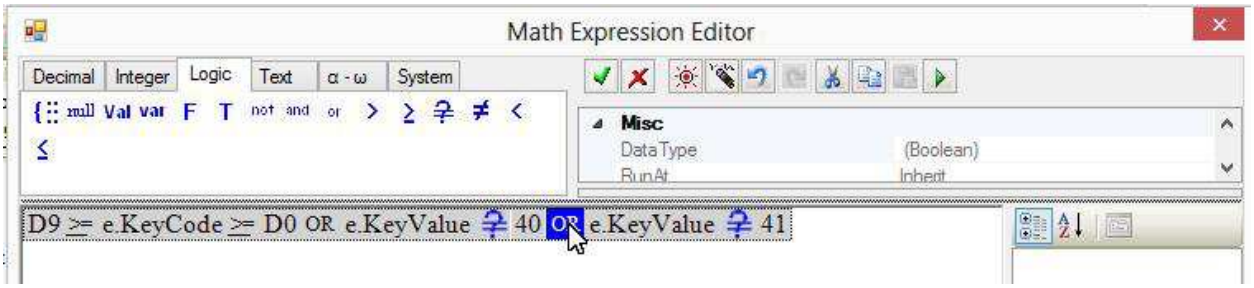


Add another OR by highlighting the whole expression:

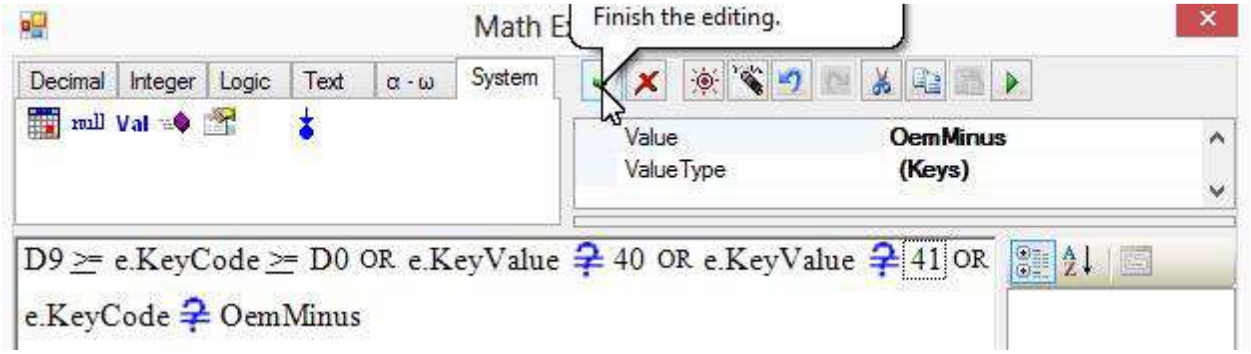




Add another condition for checking ")", the ASCII code for ")" is 41.

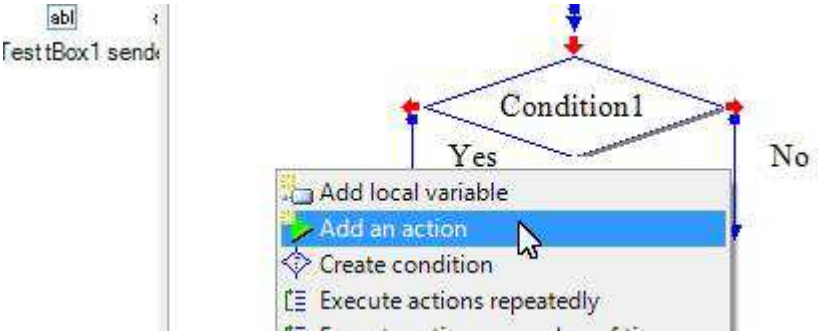


Add another condition for KeyCode equal to "-". That is all we need for the expression:

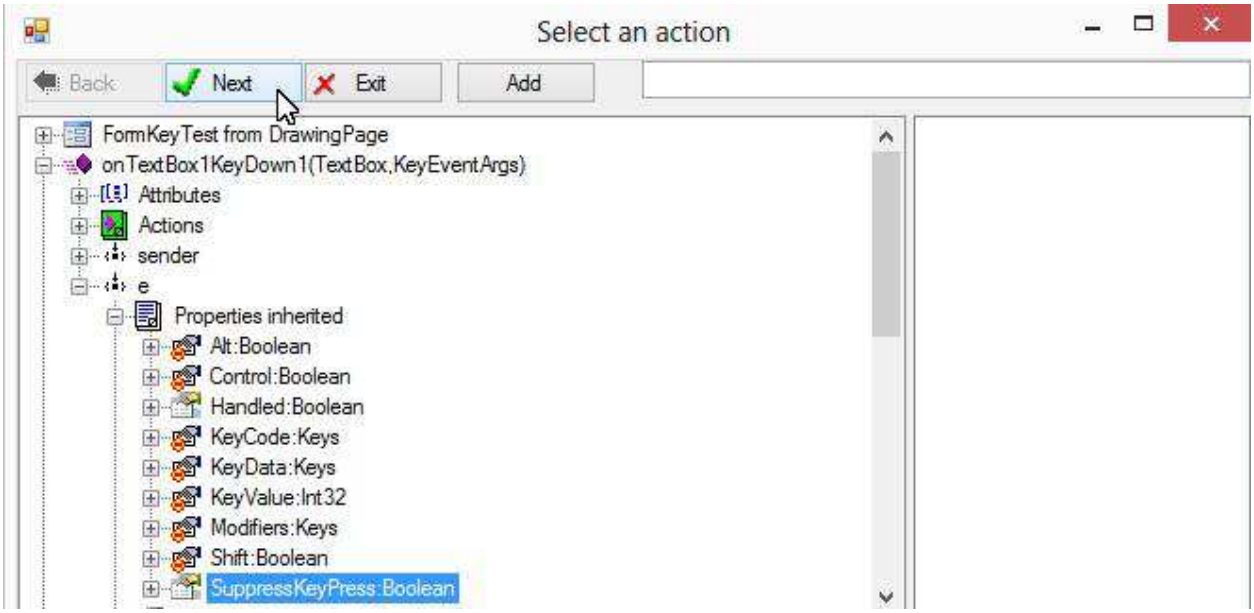


Action for suppress key press

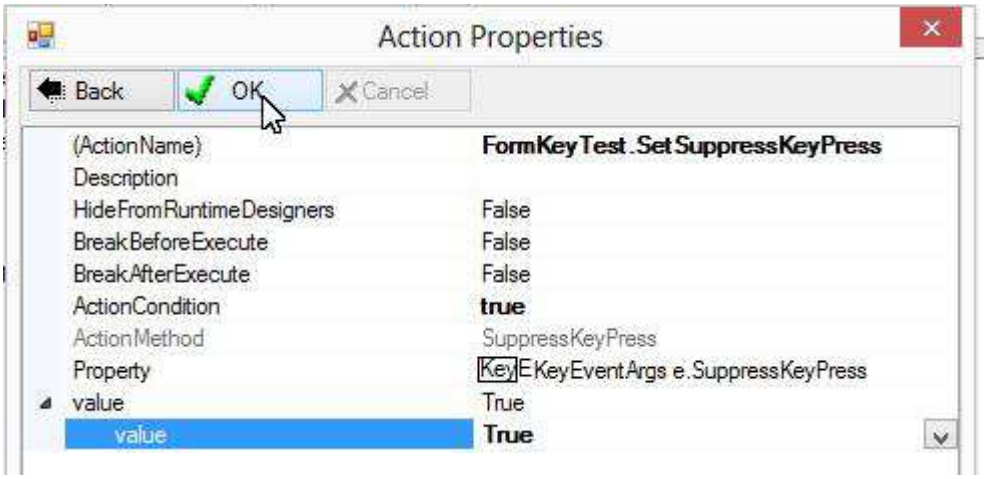
We need to add an action to suppress key press:



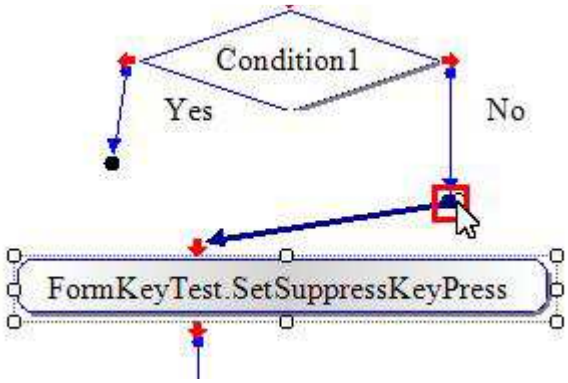
Select SuppressKeyPress property of event parameter e:



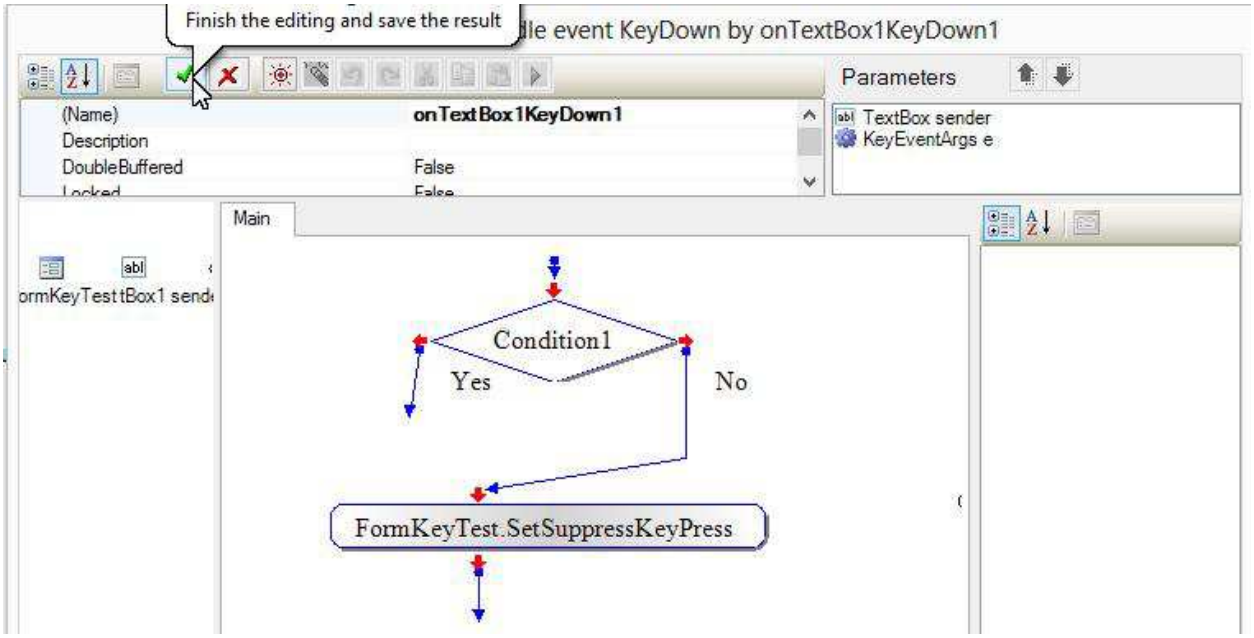
Specify True for “value”, click OK:



The new action appears. Link the new action to No port of the condition action:

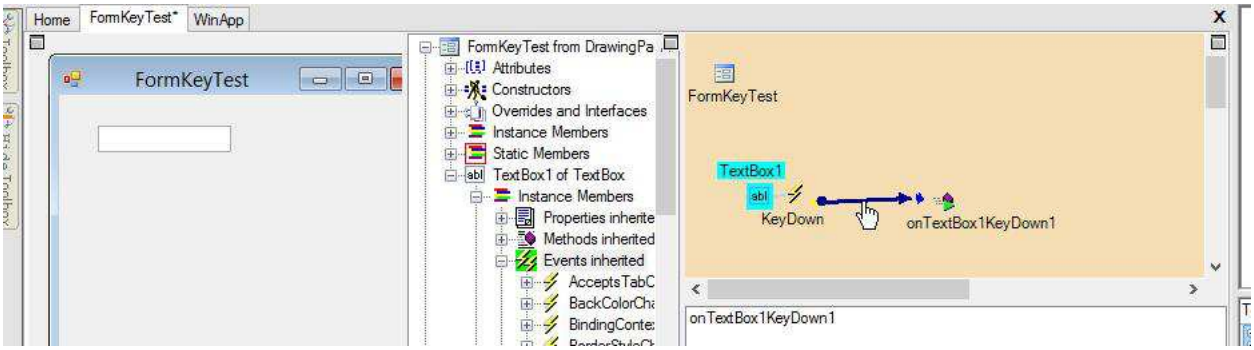


This is our event handler method:



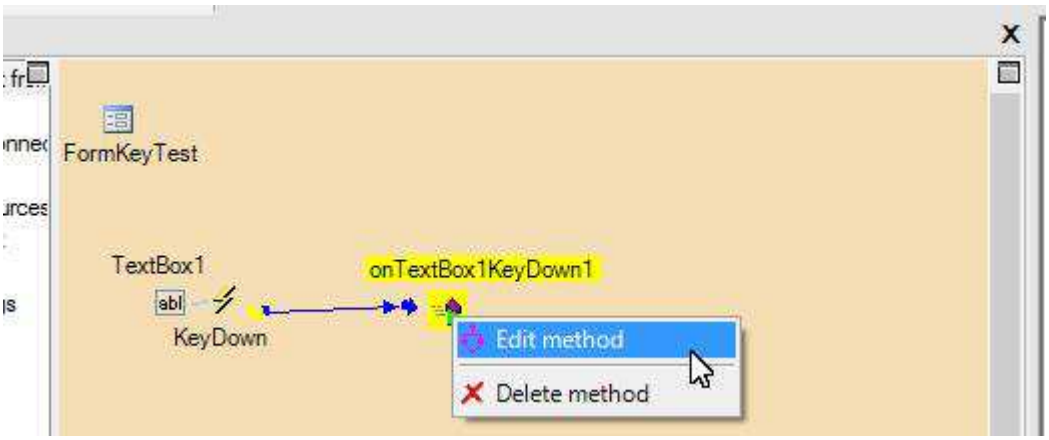
This event handler suppresses the key press if the key is not allowed.

In the event map, the event handler method is linked to KeyDown event of the text box:

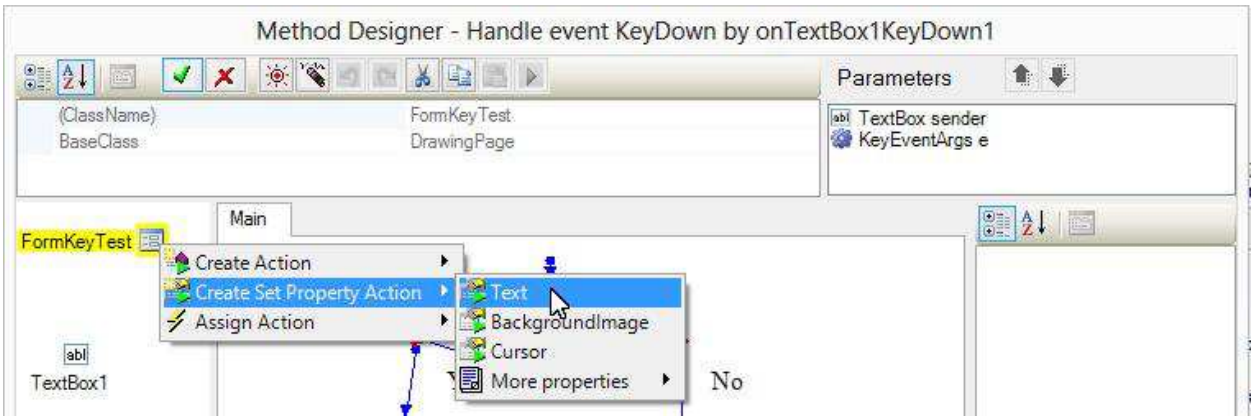


Examine Key Value

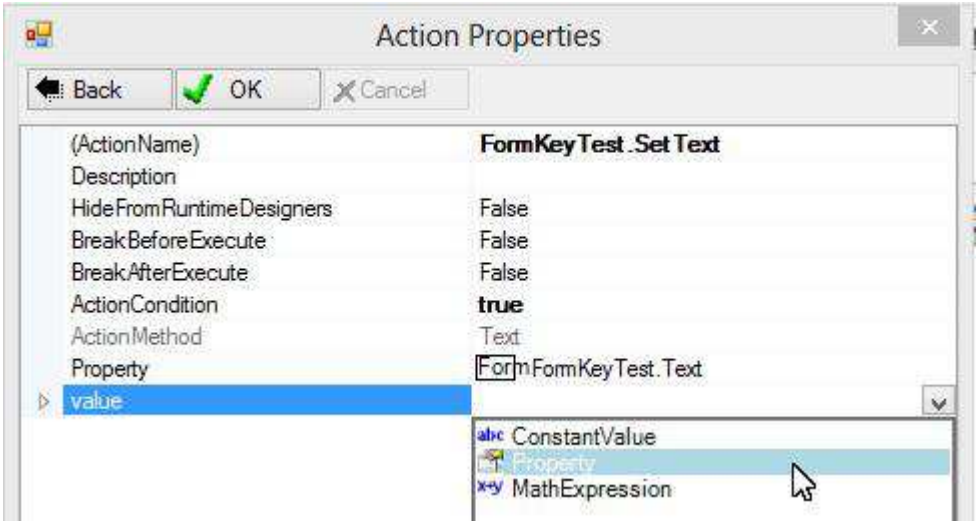
If you are not sure what are the integer values for certain keyboard keys then you may add an action in the event handler method to show key value. Let's do it by editing the event handler method:



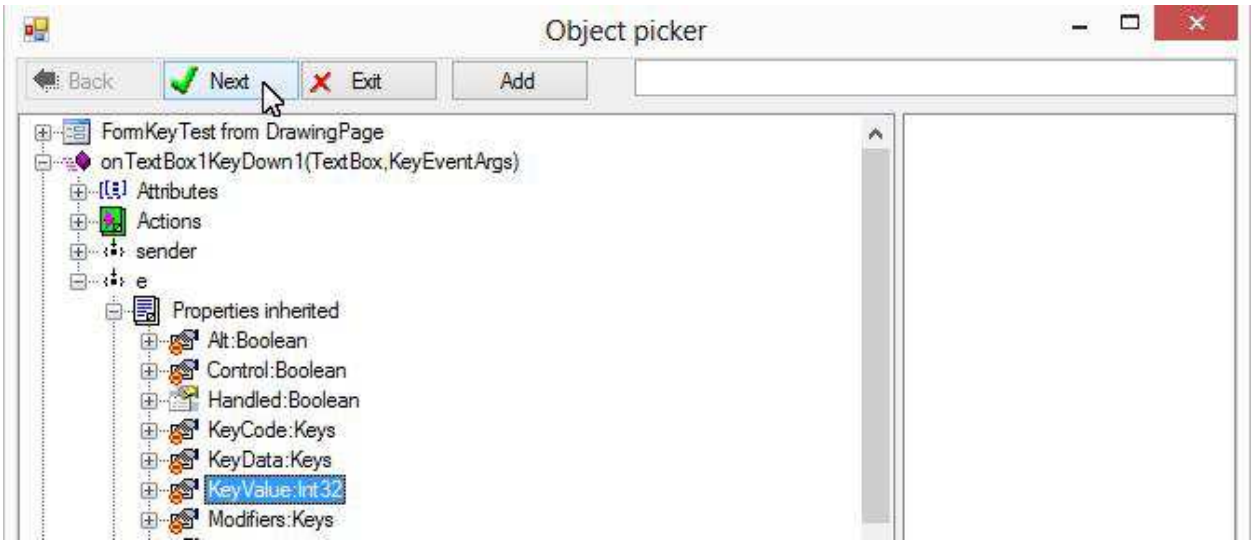
Suppose we want to show key value on the title bar of the form. We create an action to set Text property of the form:



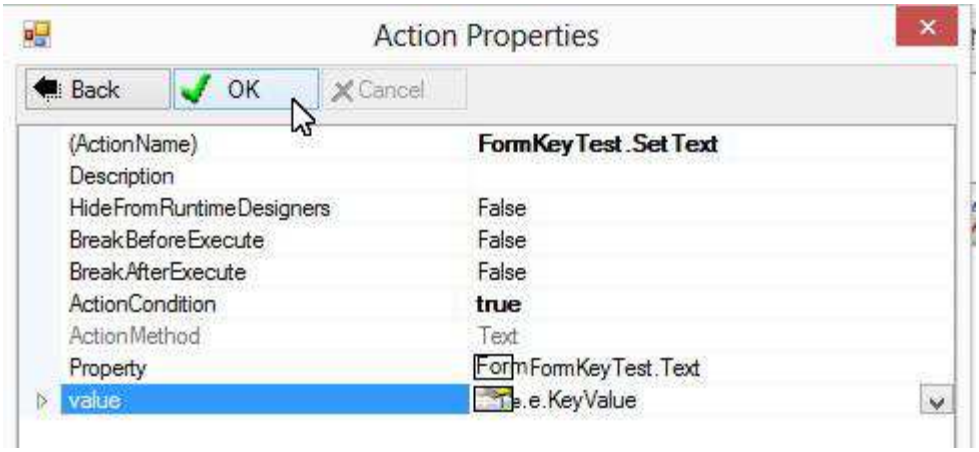
Select Property for “value” of the action:



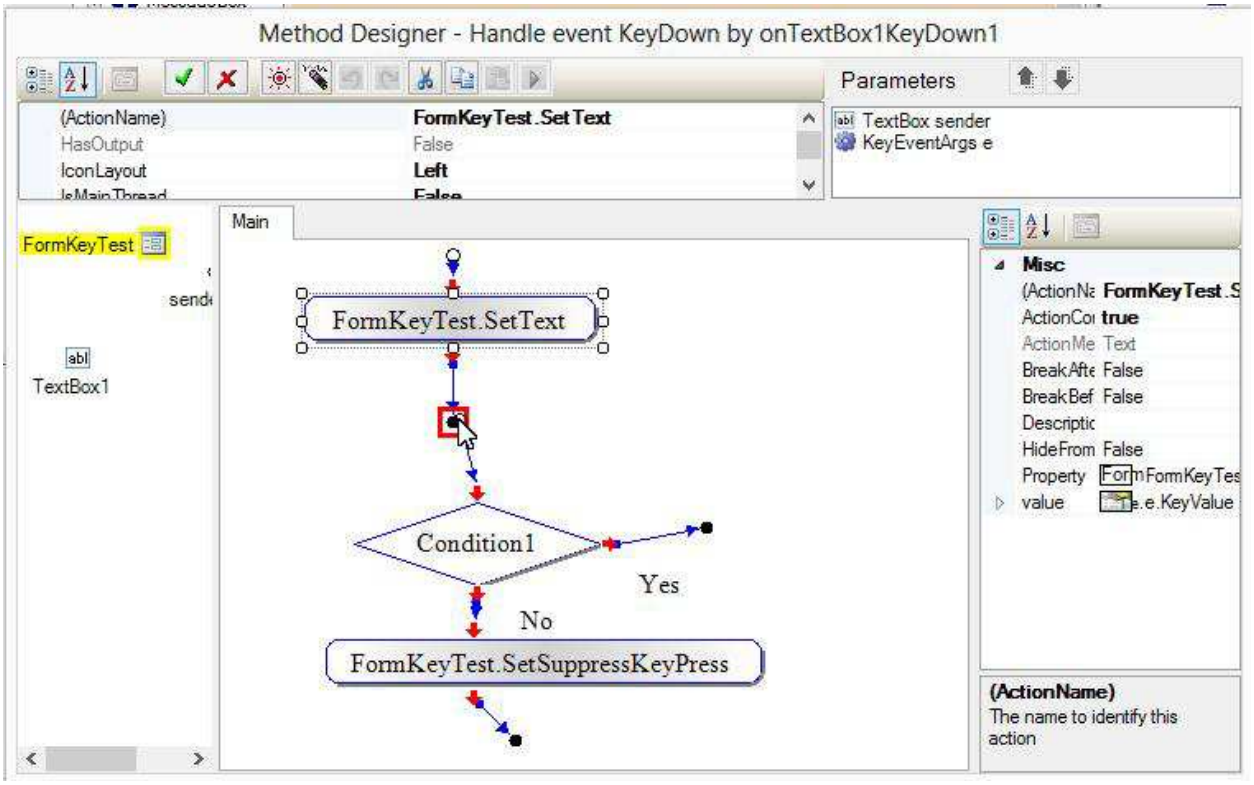
Select e.KeyValue:

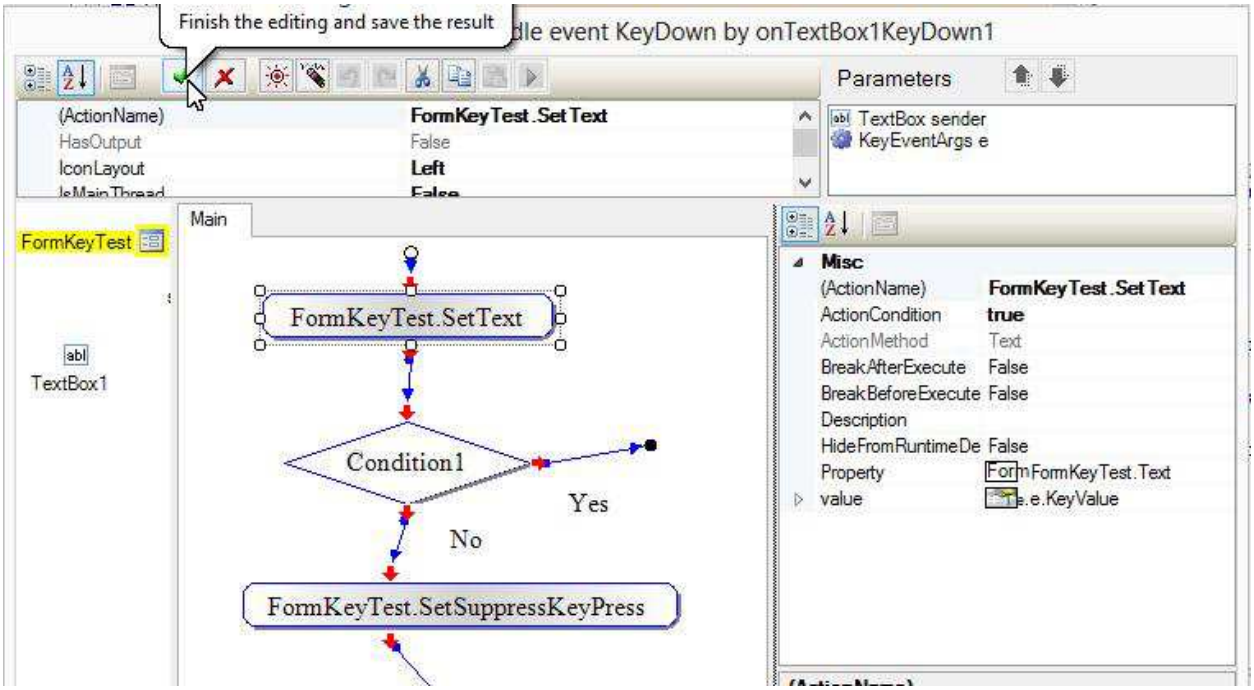


Click OK to create the action:



The new action appears in the method editor. Link it to the condition action:





Feedback

Please send your feedback to support@limnor.com