# Convert Decimal Time Values

## Contents

Introduction ..... 1
Create properties ..... 2
Create conversion method ..... 3
Method parameter ..... 4
Get Integer Portion ..... 5
Get Decimal ..... 8
Get Conversion ..... 11
Get integer portion of conversion ..... 13
Use Method ..... 16
Get minutes ..... 16
Show hours and minutes ..... 19
Get seconds ..... 22
Show seconds ..... 24
Test ..... 26
Feedback ..... 28

## Introduction

Suppose we get a time value in decimals, say, 1.23 hours, we may want to express the time in hours, minutes and seconds. The conversion is straight ward. The decimal portion of hours multiply 60 is the minutes. The decimal portion of minutes multiply 60 is the seconds. The decimal portion of a number can be retrieved by subtracting its integer portion. The calculation is as following.

Given a number V, we may do following conversions.

Vi = int(V) is the integer portion of V. For example, if V=1.23 then Vi = 1.

Ve = V – Vi is the decimal portion of V. For example, if V=1.23 then Ve = 0.23.

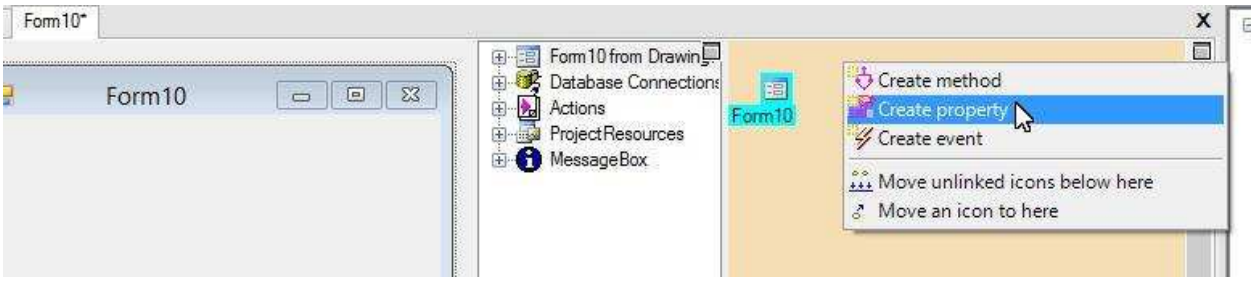
C = 60 * Ve is fraction portion. If V is hours then C is the minutes-portion. If V is minutes then C is seconds-portion. If V=1.23 then C=13.8

Ci = int(C) is the integer portion of C. If V = 1.23 then Ci = 13

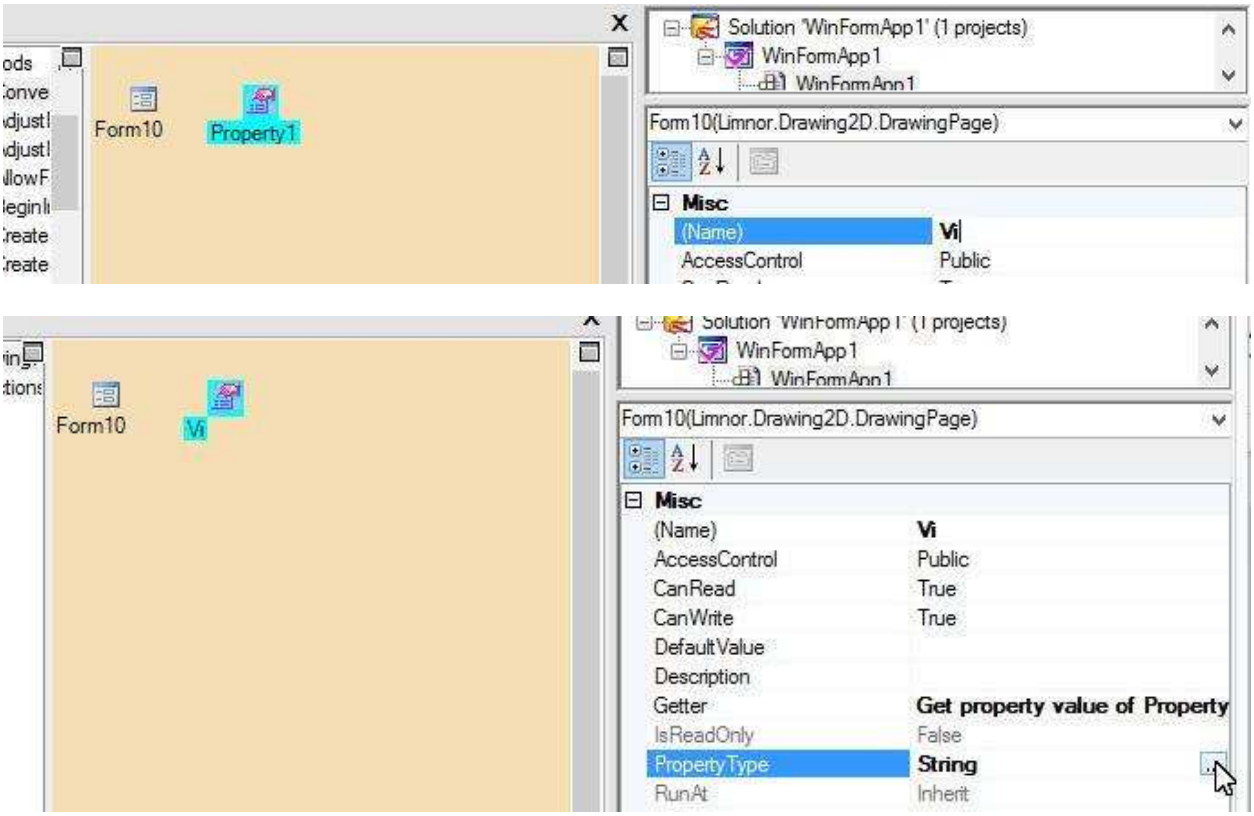
Below we use a method to do the calculations. This method accepts a parameter V and produces Vi, Ve, C and Ci as given above.

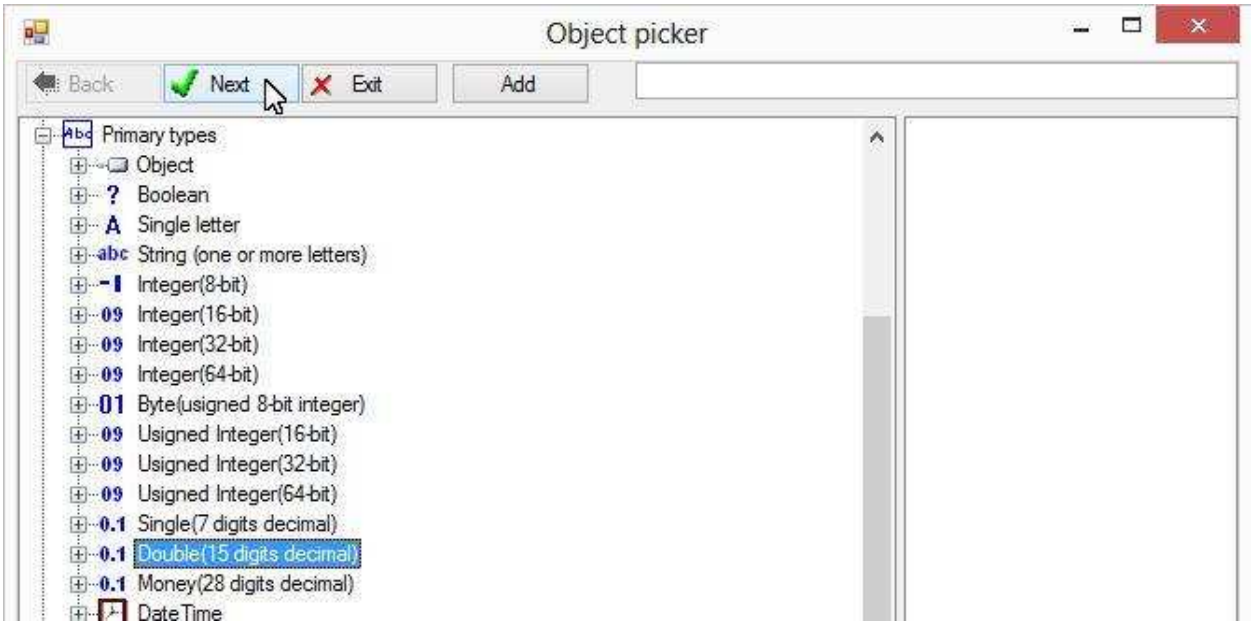
## Create properties

Create a property to hold Vi:

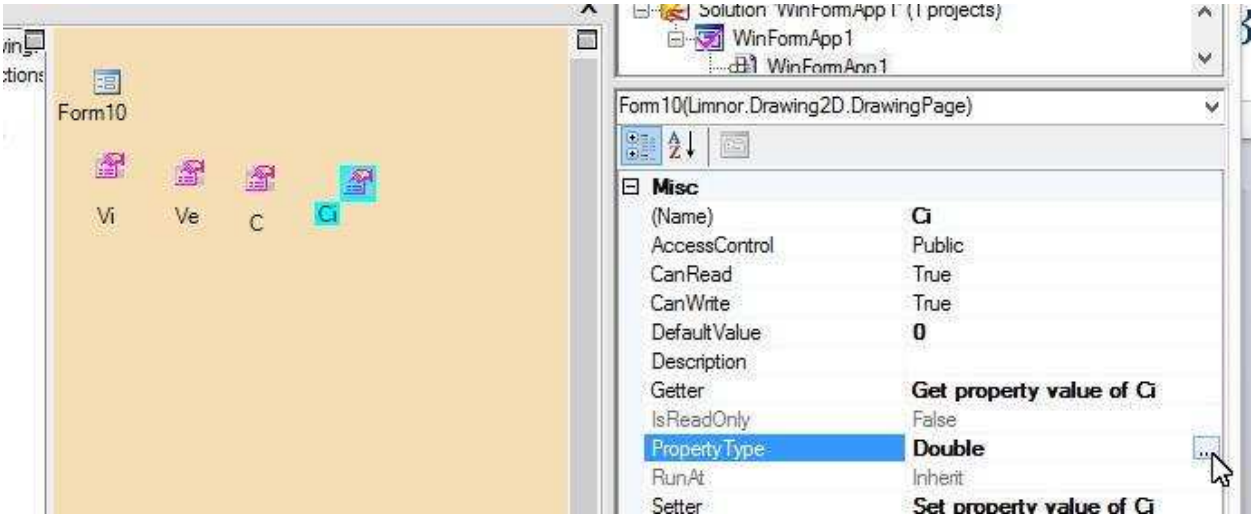



Rename the new property to Vi and change its data type to Double:



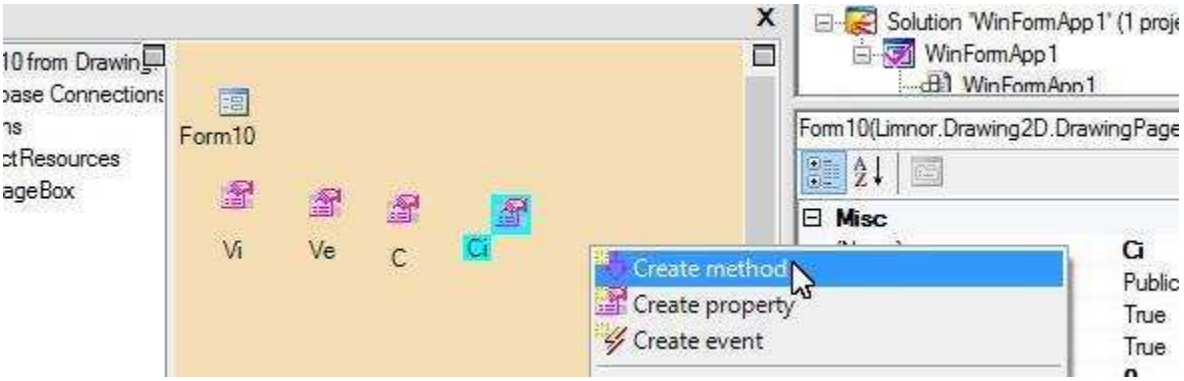



Also create properties for Ve, C and Ci:




## Create conversion method

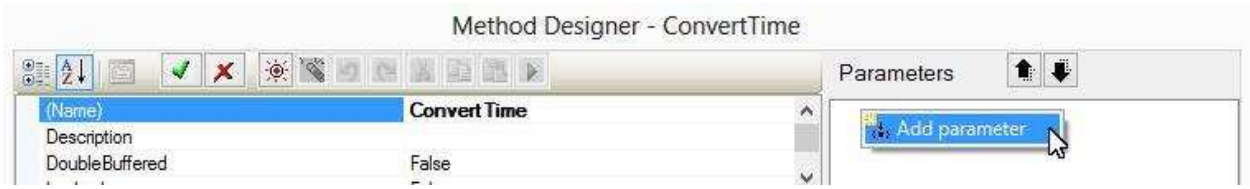
Create a new method to do calculations specified previously.



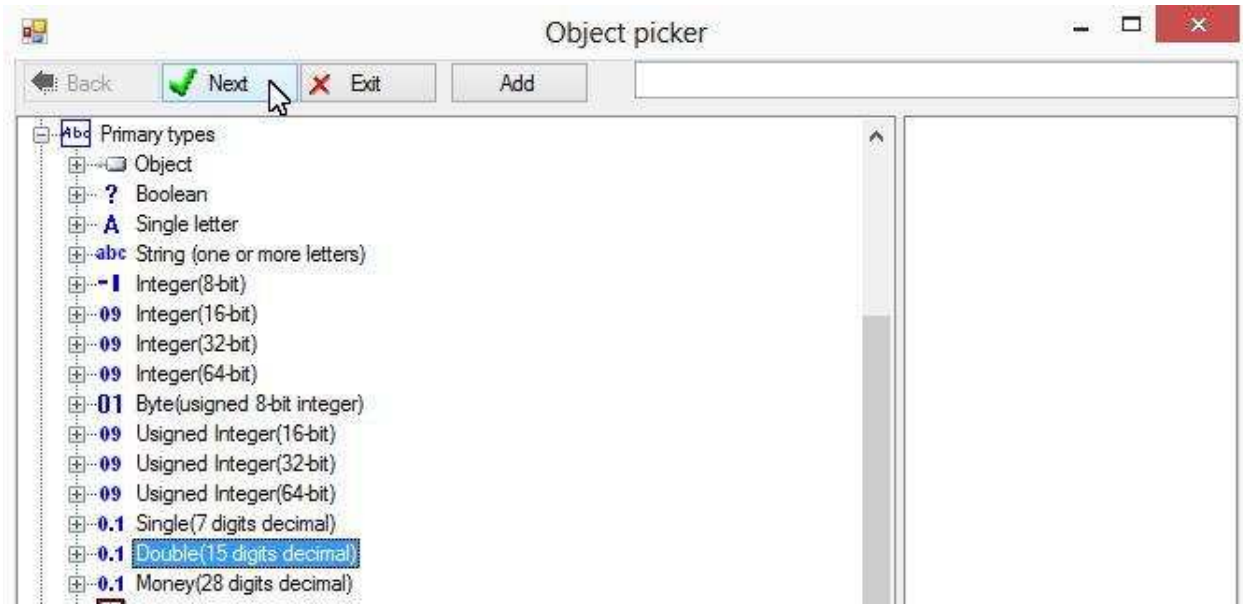

Rename it to ConvertTime:

## Method parameter

Add a parameter to accept value to be converted:



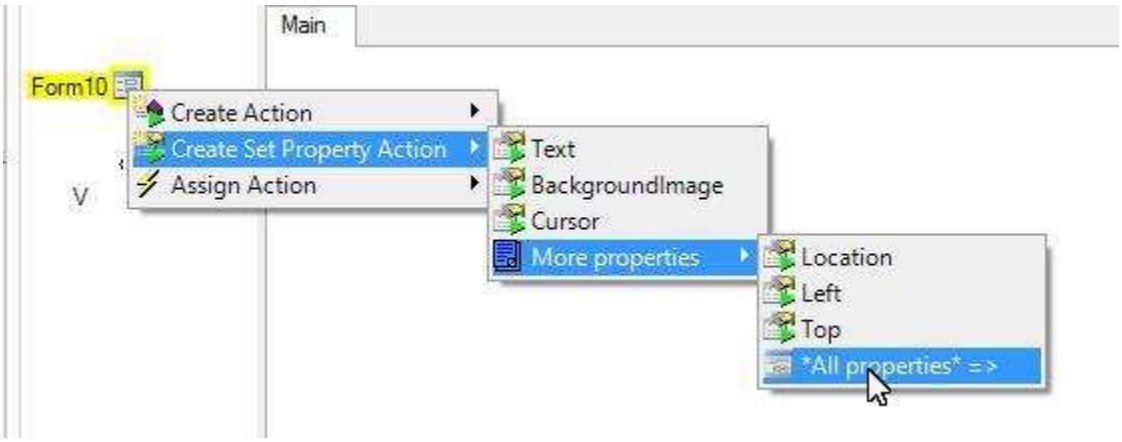

Select parameter type:

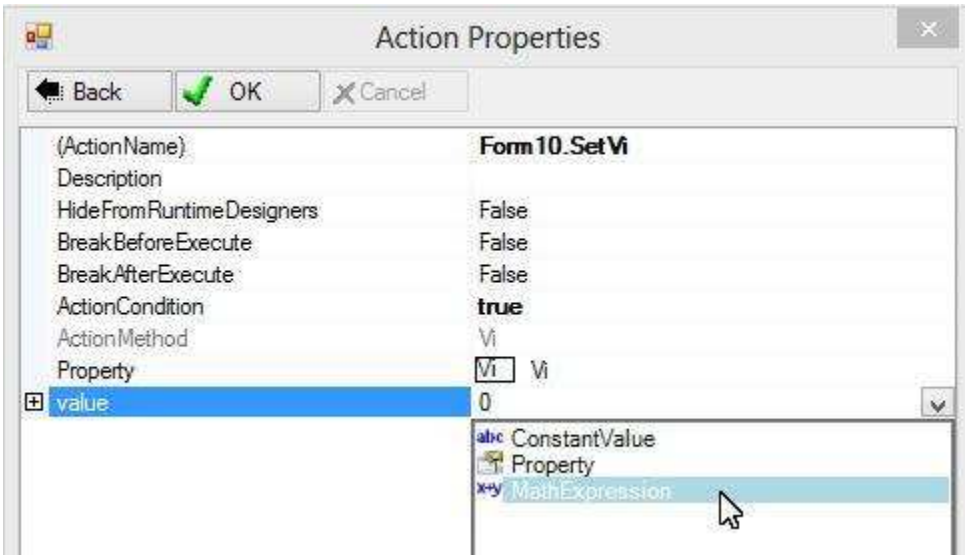


Rename the parameter if we want:

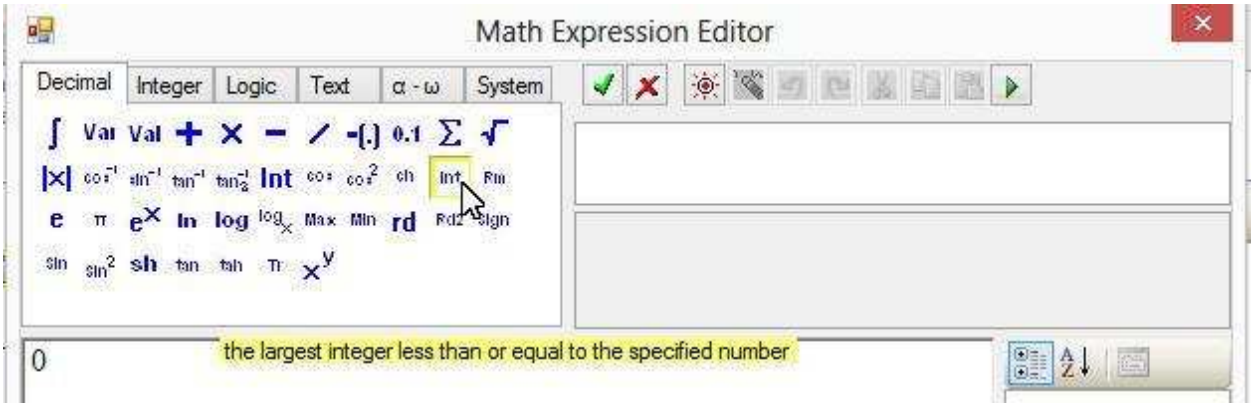
## Get Integer Portion

Create an action to set Vi:

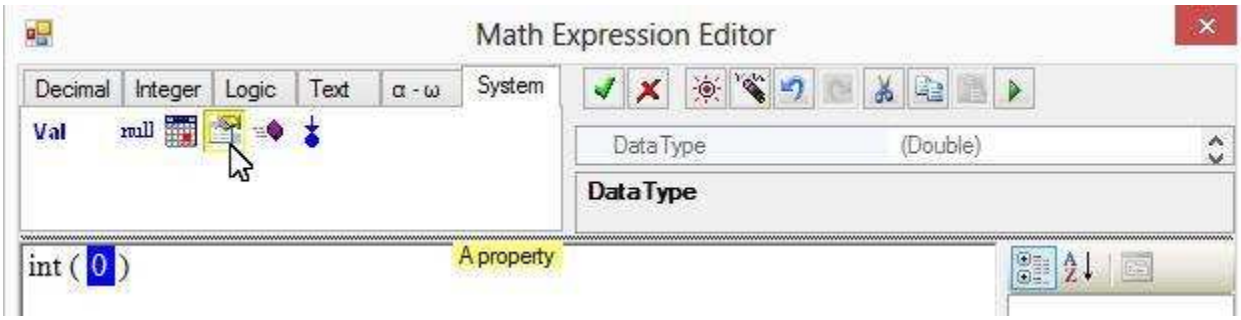




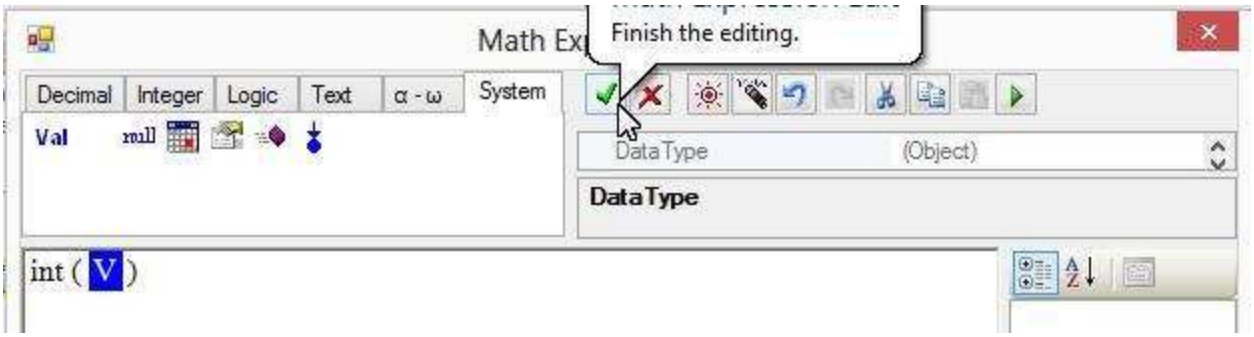

Select operator int:

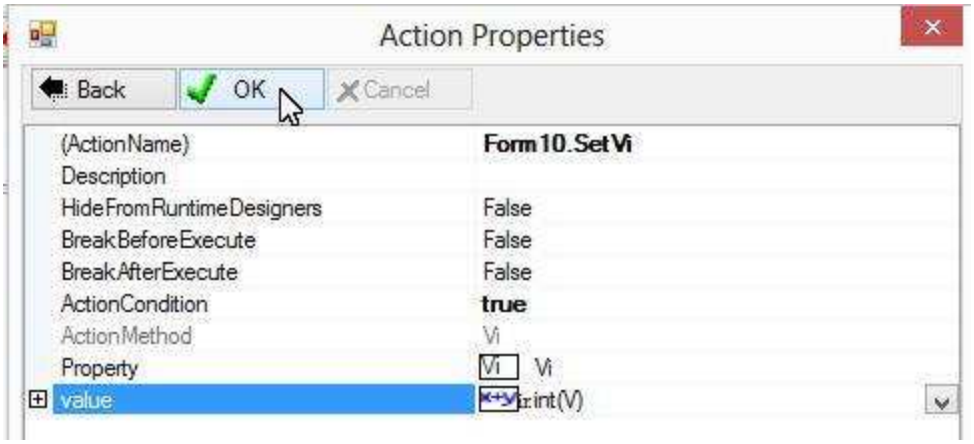



Select method parameter for the operator:

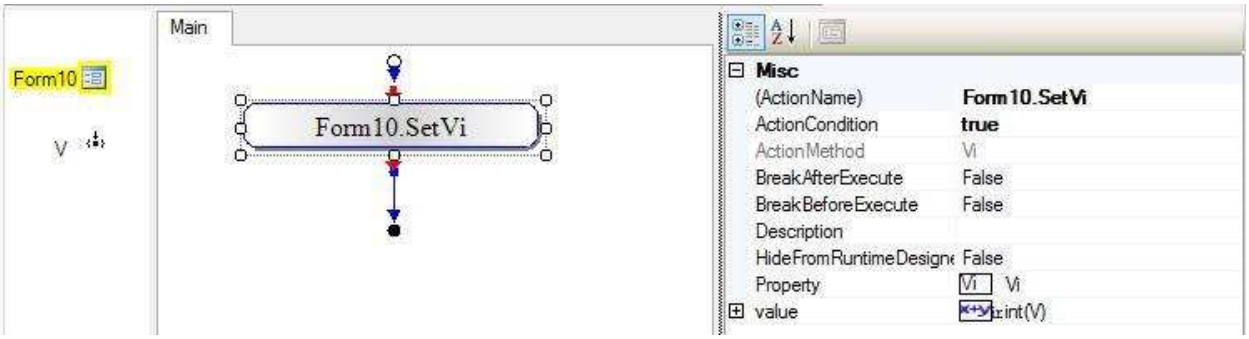


That is how we should calculate Vi



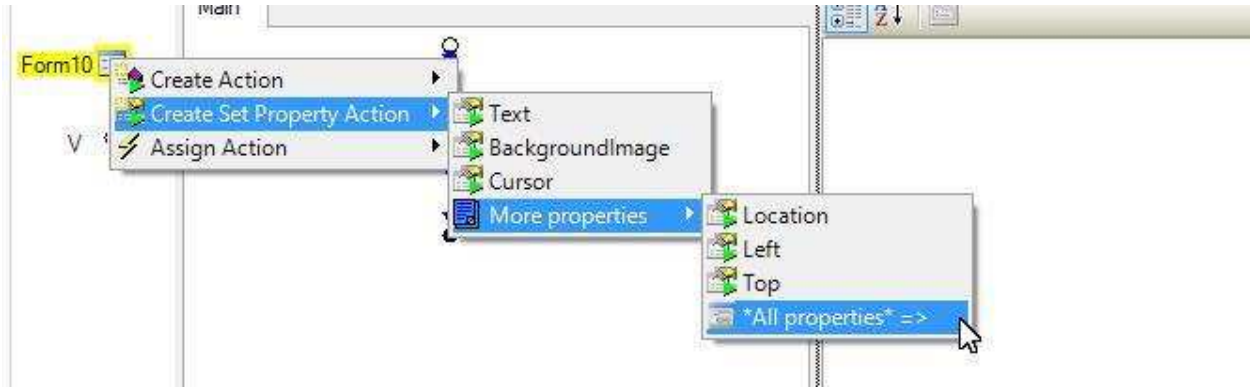



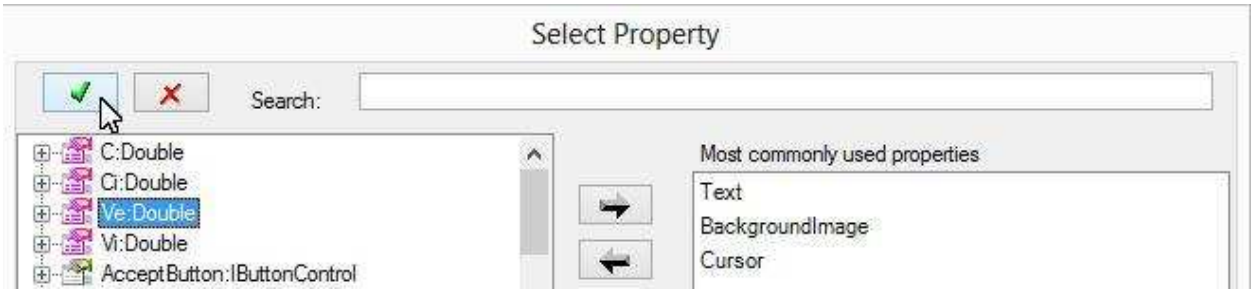

The action appears:

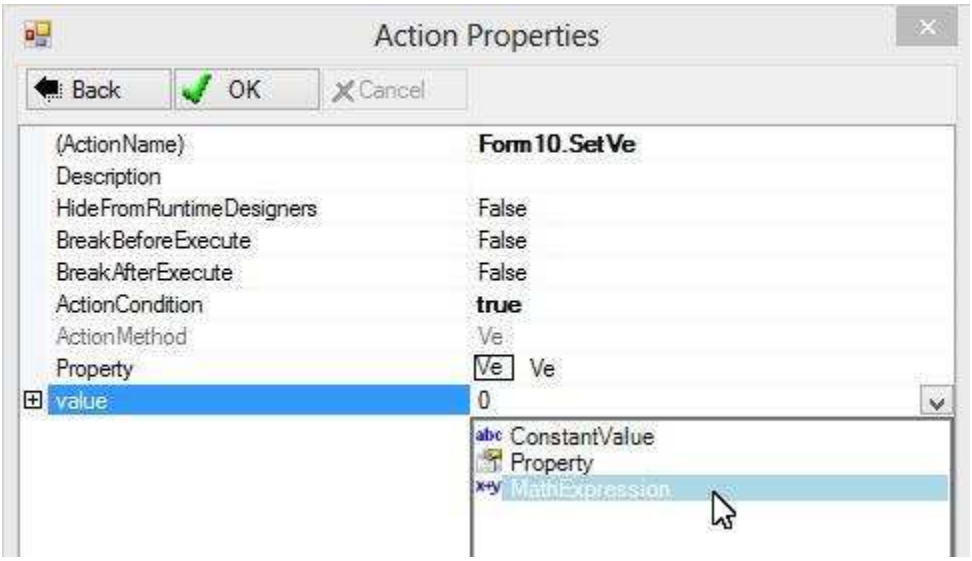


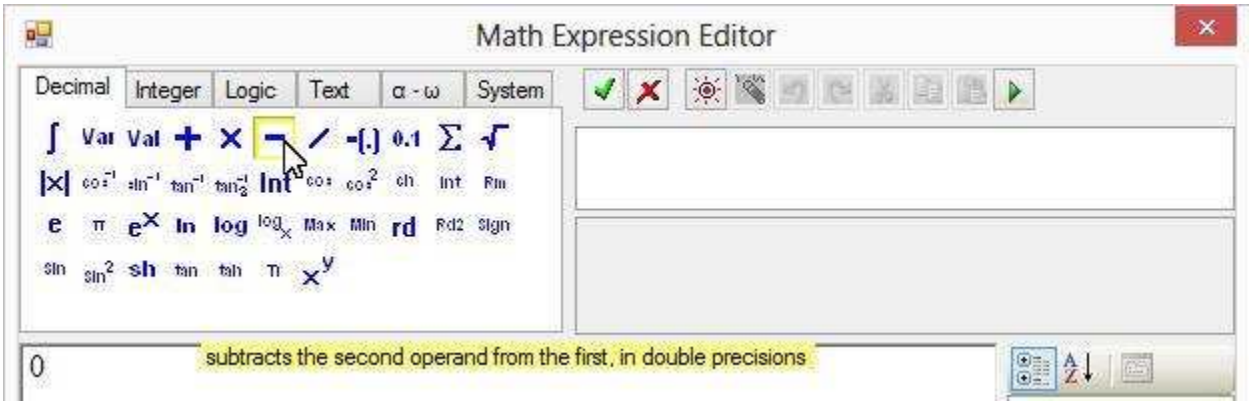
## Get Decimal

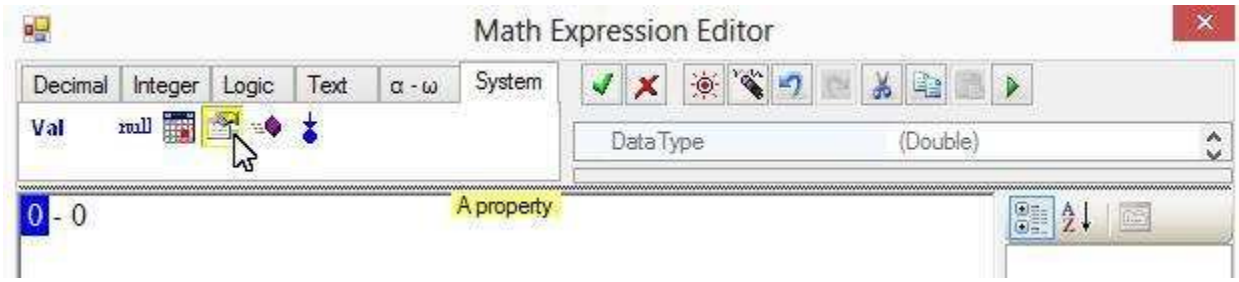
Create an action to set Ve.



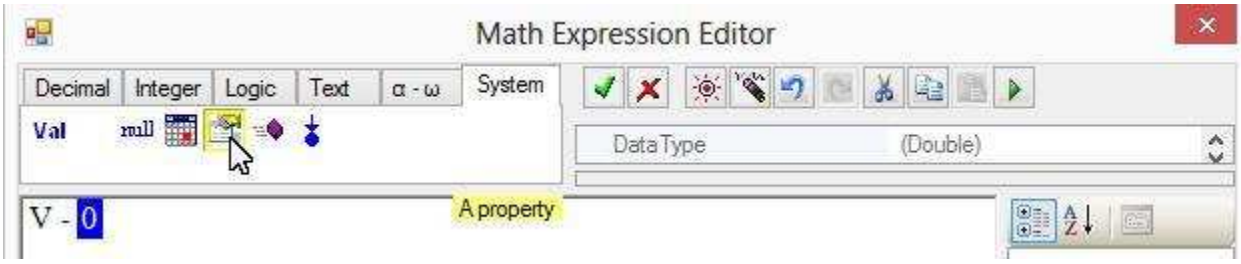



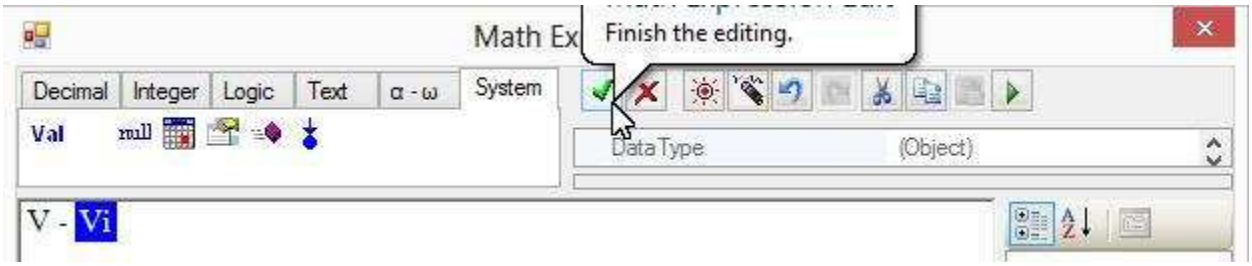



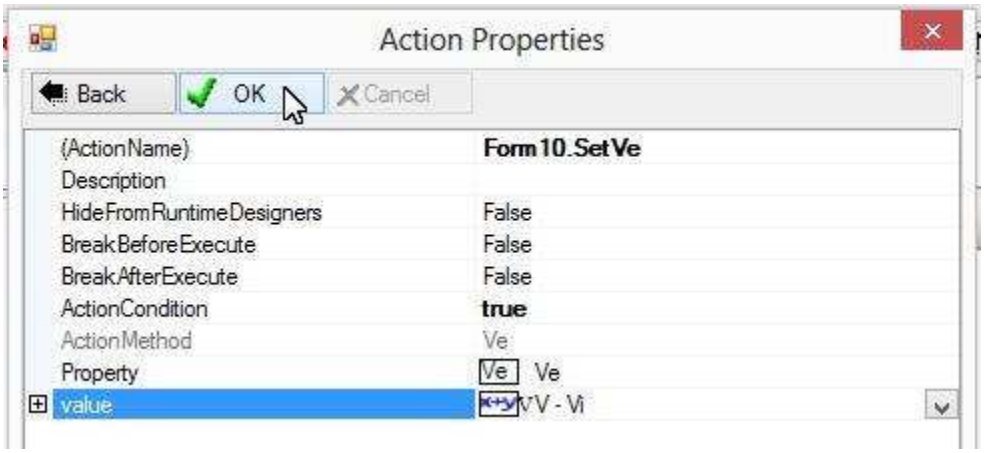

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
0
subtracts the second operand from the first, in double precisions


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(Double)
0 - 0
A property

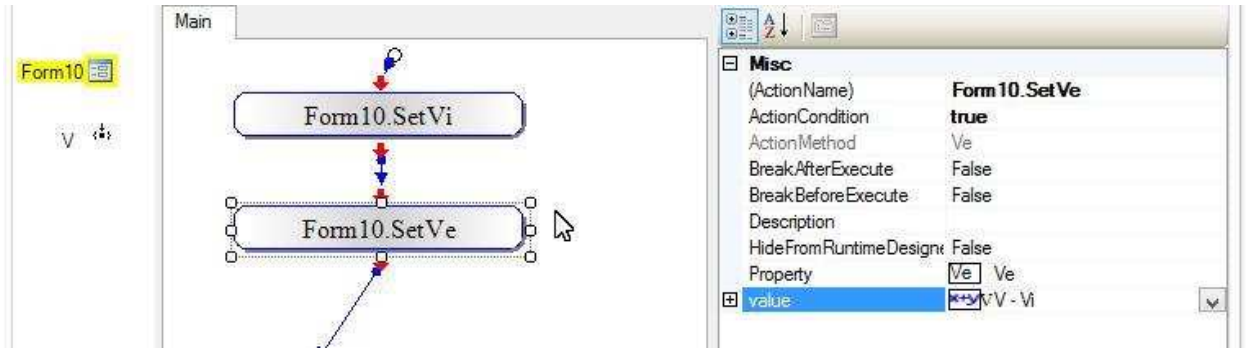
Object picker
Back
Next
Exit
Add
Form10 from DrawingPage
ConvertTime(Double)
Attributes
Actions
Primary types


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(Double)
V - 0
A property



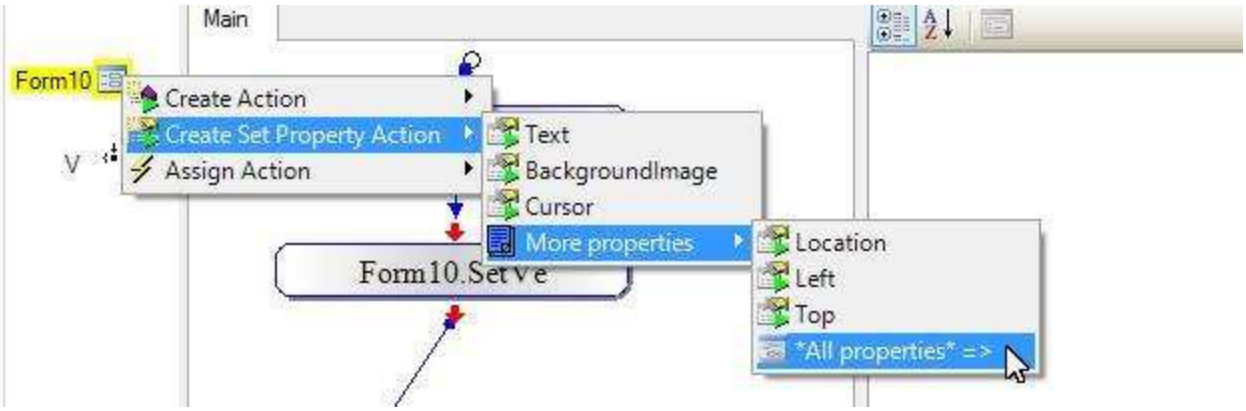



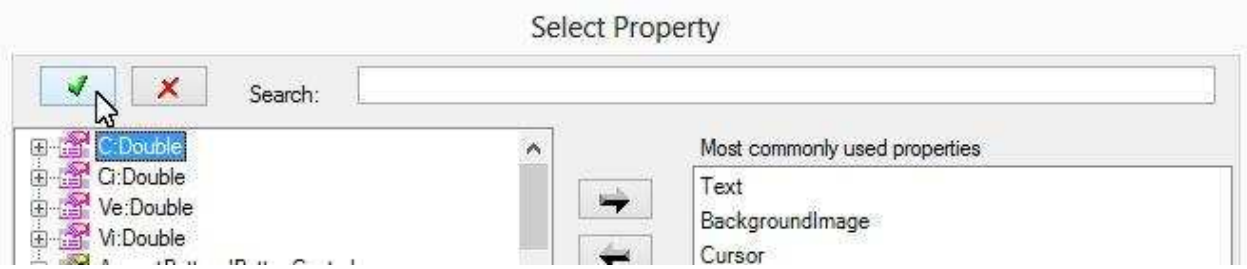

The action appears. Link it to the last action:

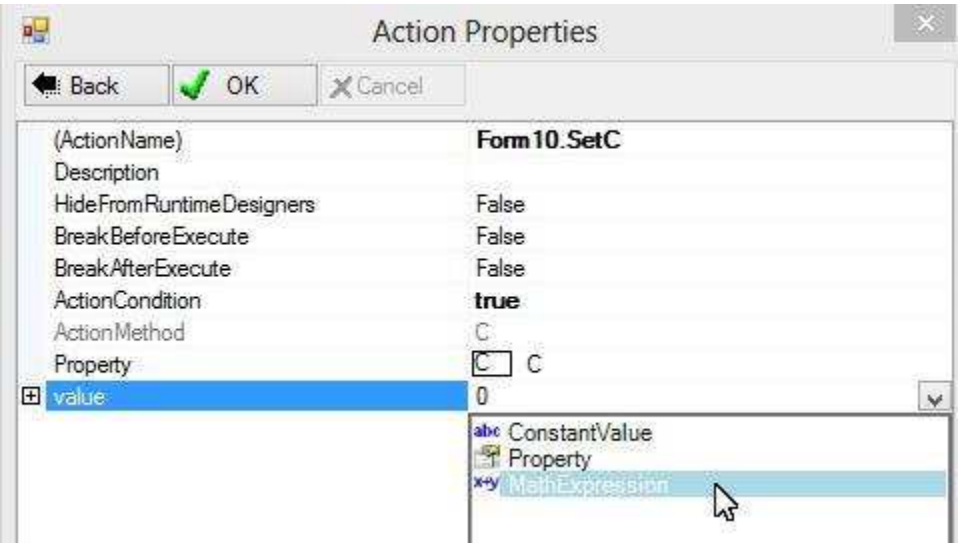


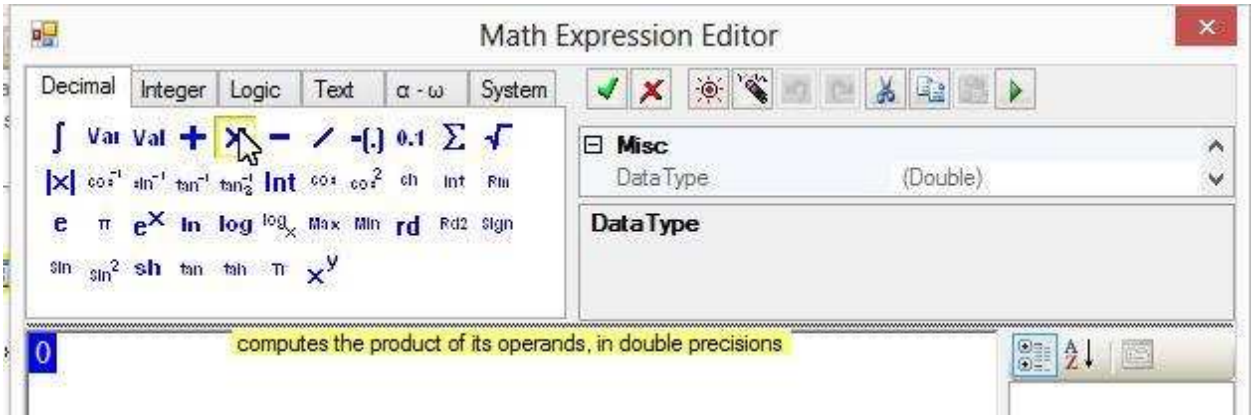
## Get Conversion

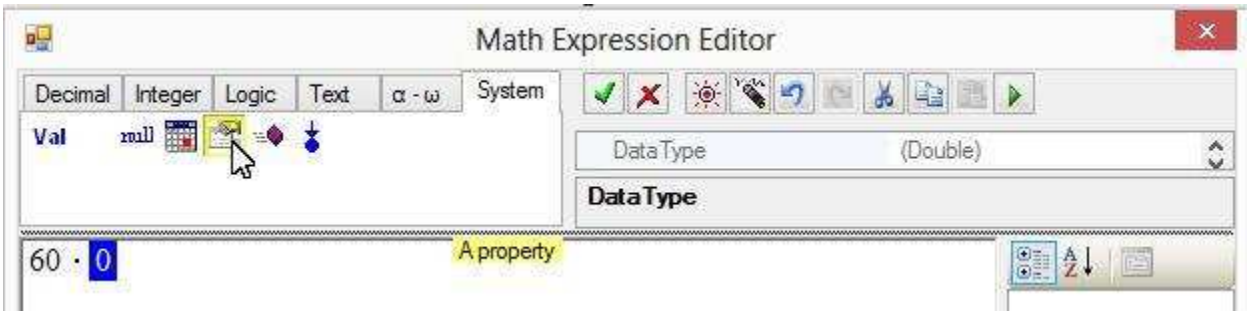
Create an action to calculate C=60 * Ve:

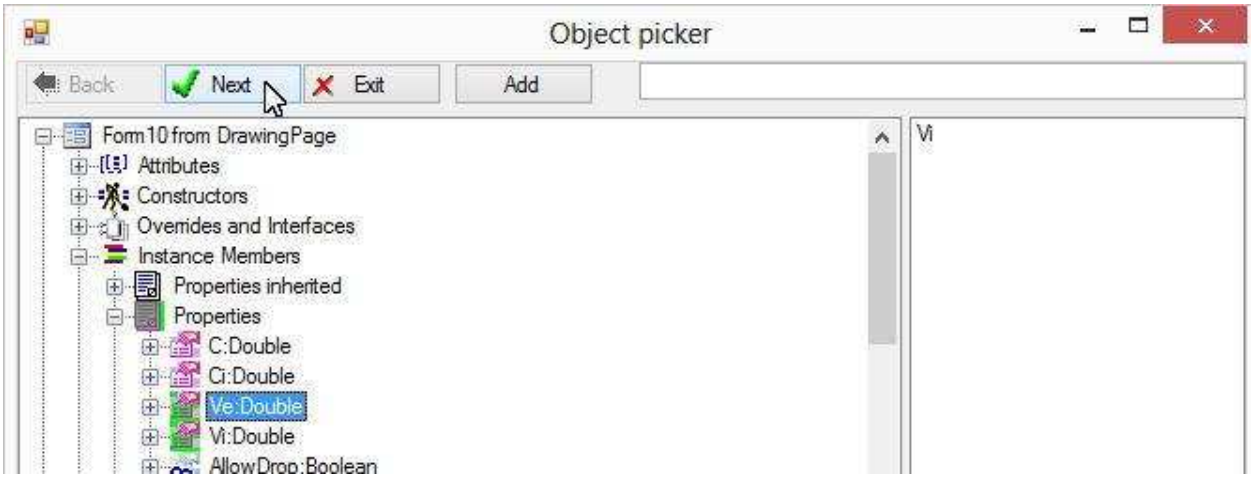


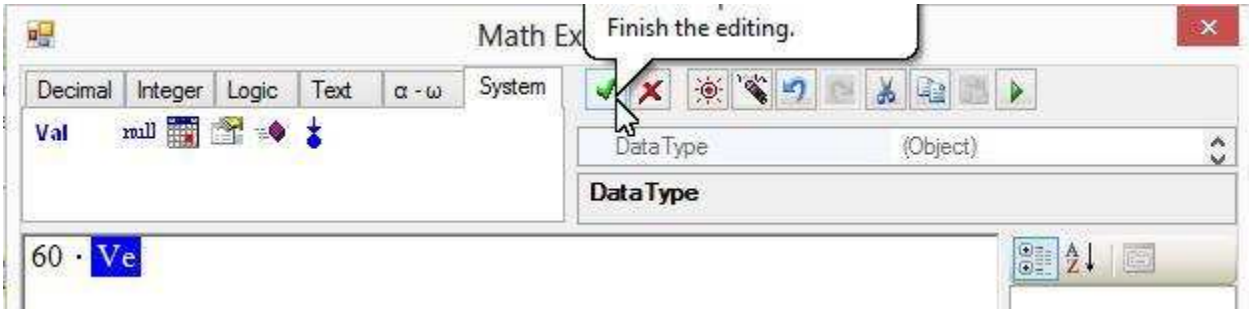



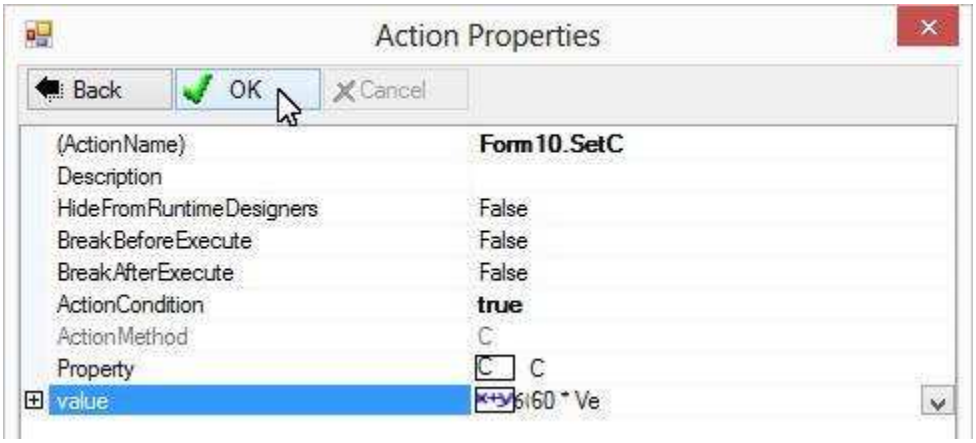




Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Misc
DataType
(Double)
DataType
0
computes the product of its operands, in double precisions

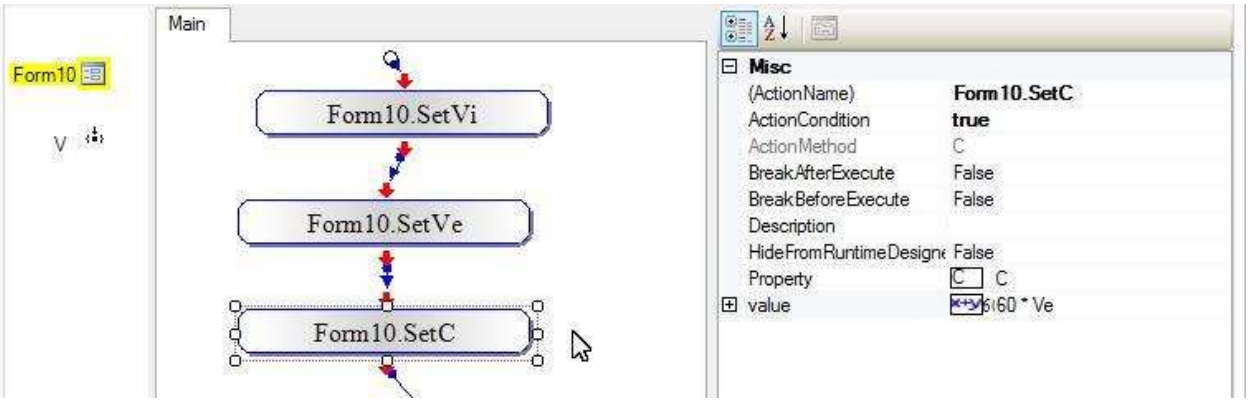

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(Double)
DataType
60 · 0
A property


Object picker
Back
Next
Exit
Add
Form10 from DrawingPage
Attributes
Constructors
Overrides and Interfaces
Instance Members
Properties inherited
Properties
C:Double
Ci:Double
Ve:Double
Vi:Double
AllowDrop:Boolean
Vi


Math Ex
Finish the editing.
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(Object)
DataType
60 · Ve

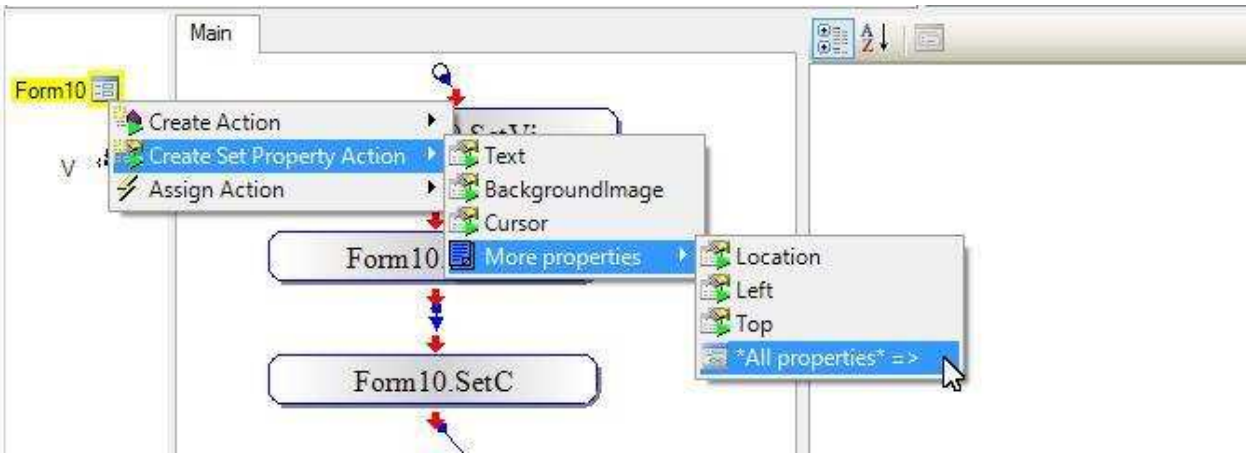


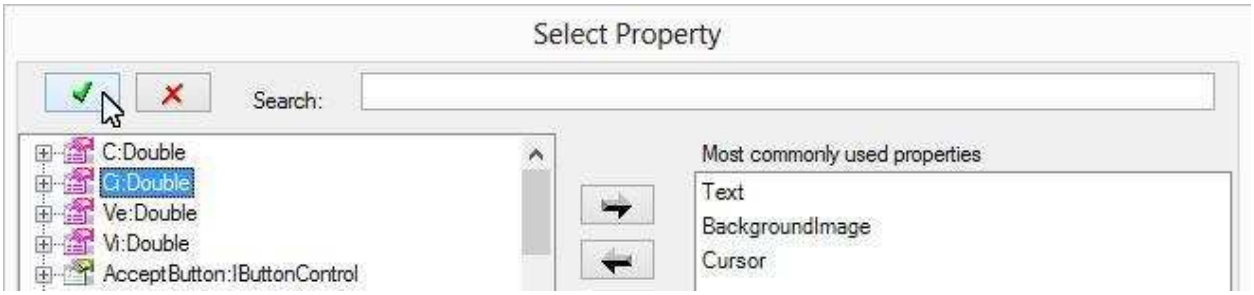

The action appears. Link it to the last action.

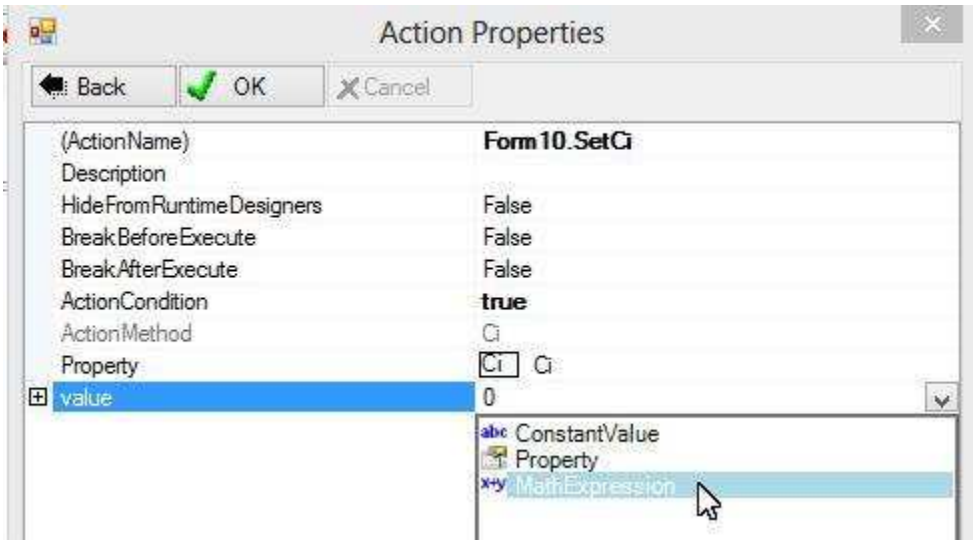


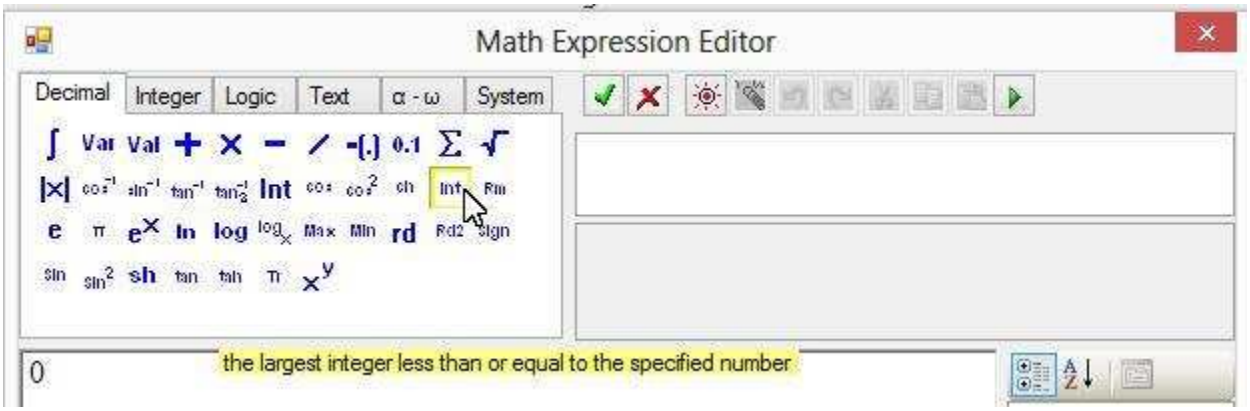

## Get integer portion of conversion

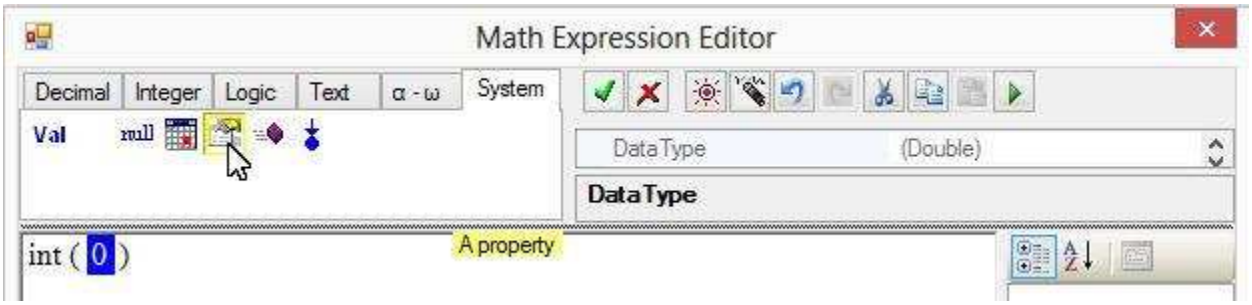
Create an action to get Ci.

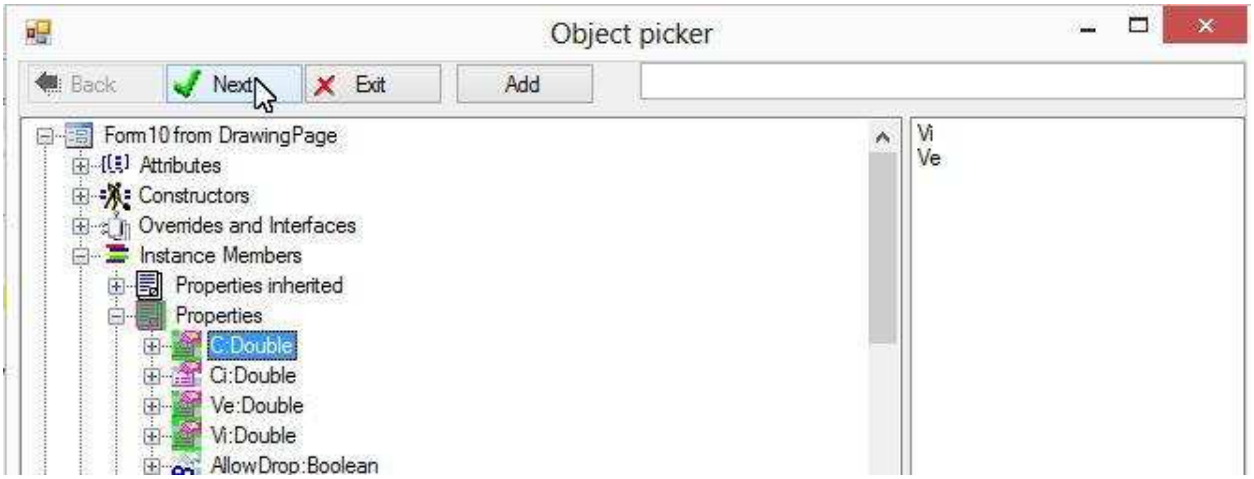


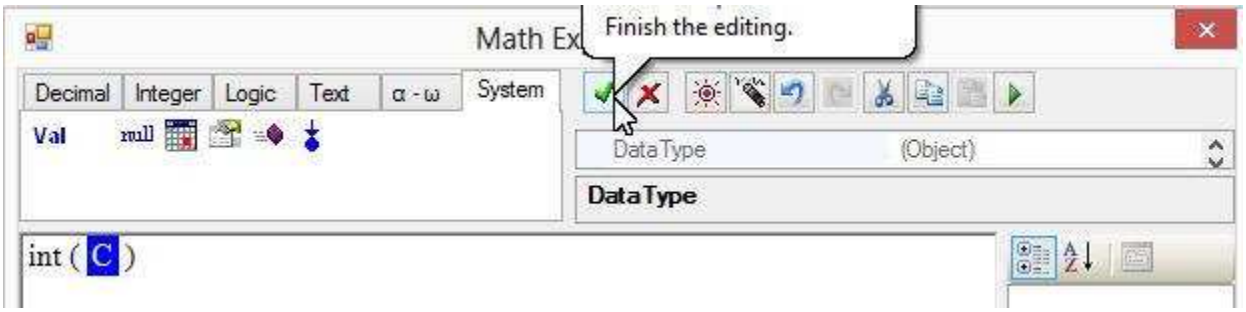

Select Property
Search:
C:Double
Ci:Double
Ve:Double
Vi:Double
AcceptButton:IButtonControl
Most commonly used properties
Text
BackgroundImage
Cursor

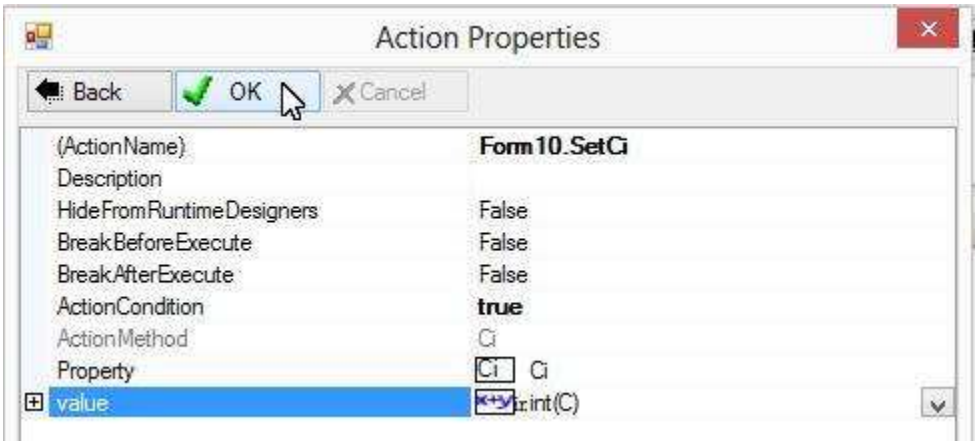

Action Properties
Back
OK
Cancel
(ActionName)
Form10.SetCi
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Ci
Property
Ci
value
0
ConstantValue
Property
MathExpression


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
0
the largest integer less than or equal to the specified number

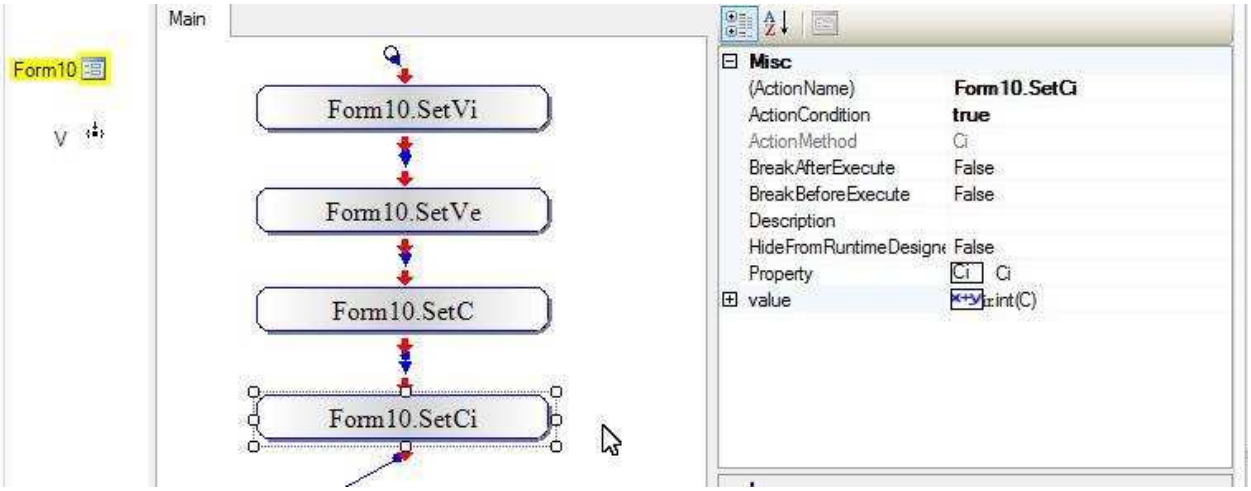

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
DataType
(Double)
DataType
int ( 0 )
A property



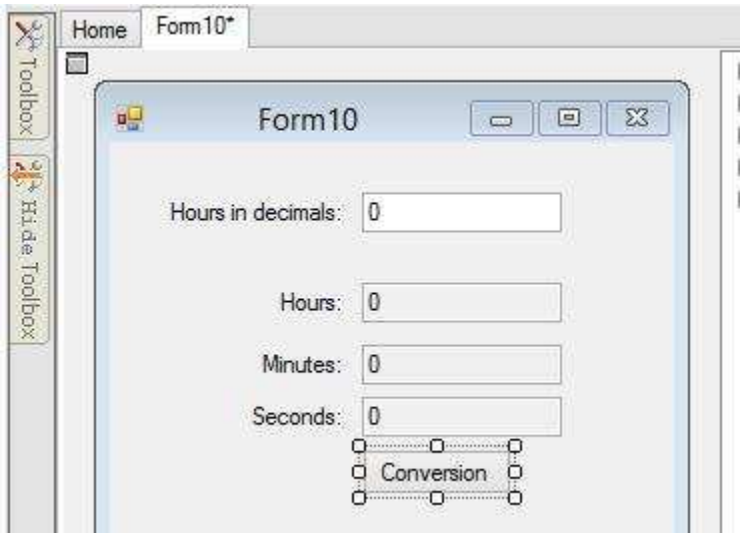

The action appears. Link it to the last action:




That is all for this method.

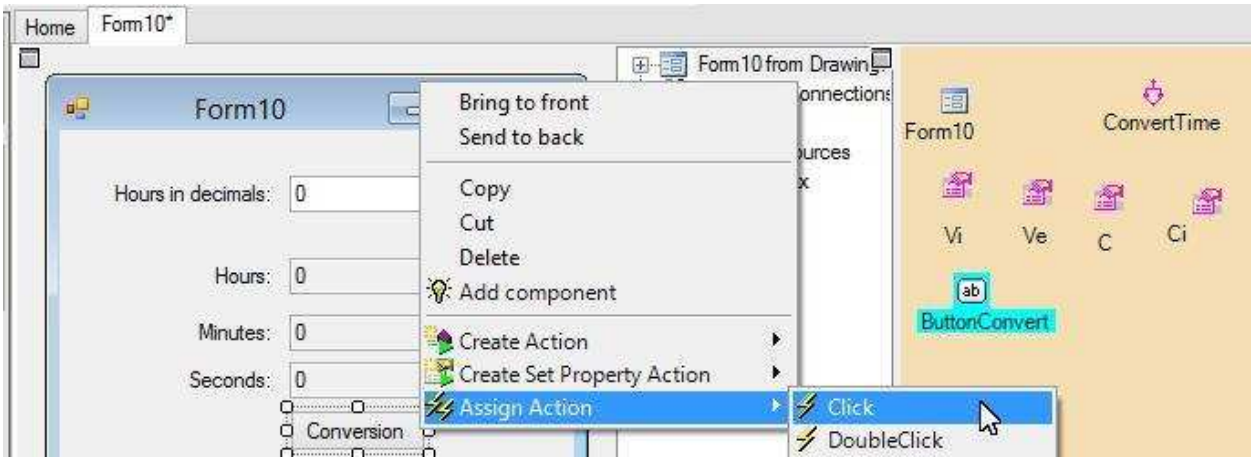
## Use Method

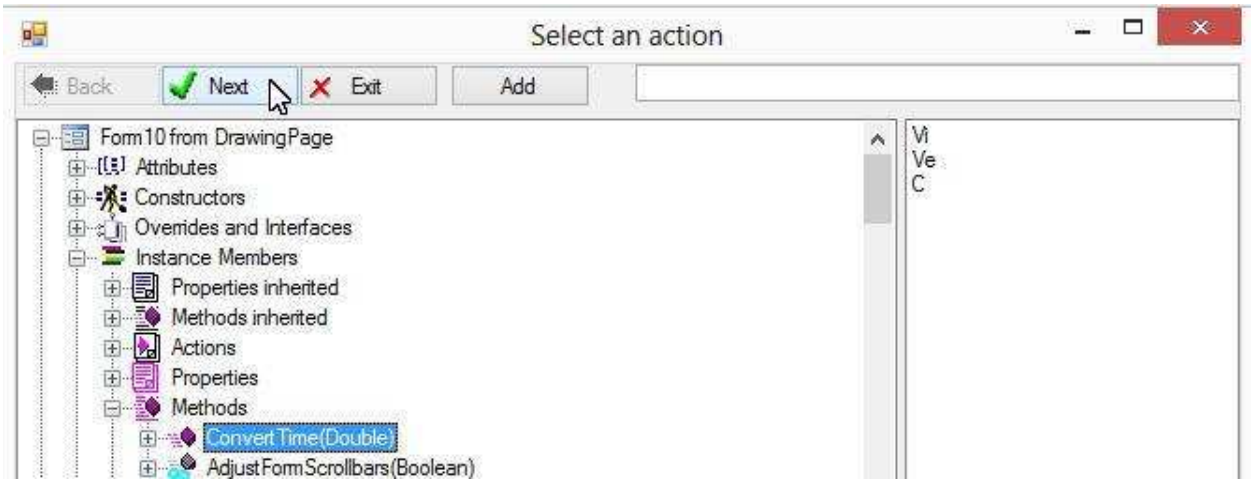
We use a text box (TextBoxNumber) for entering hours in decimals. Use 3 text boxes to show converted hours, minutes and seconds.



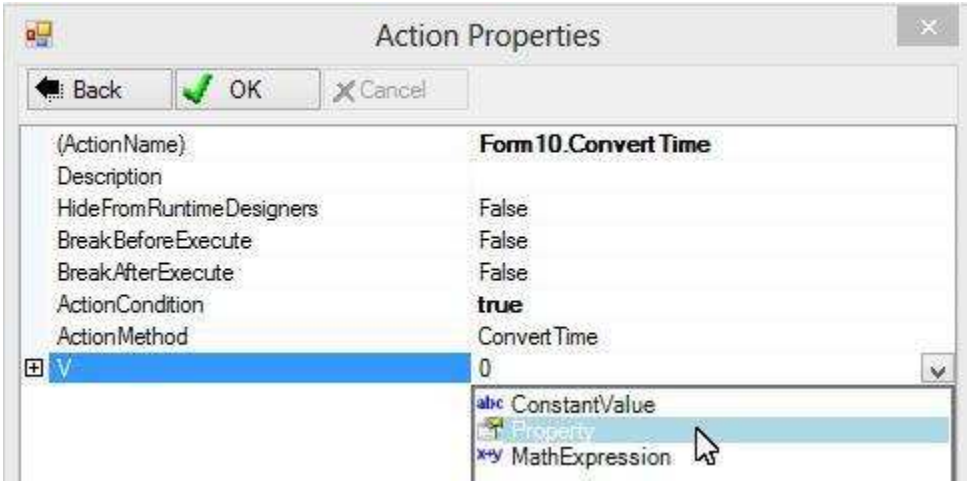

### Get minutes

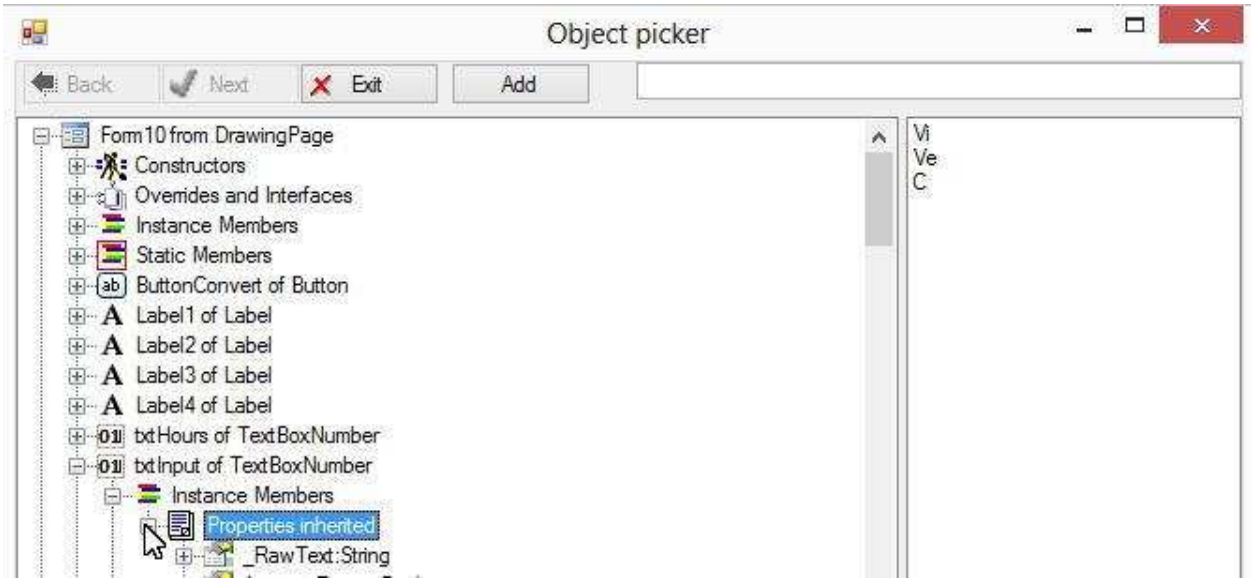
Call the method to do conversion.

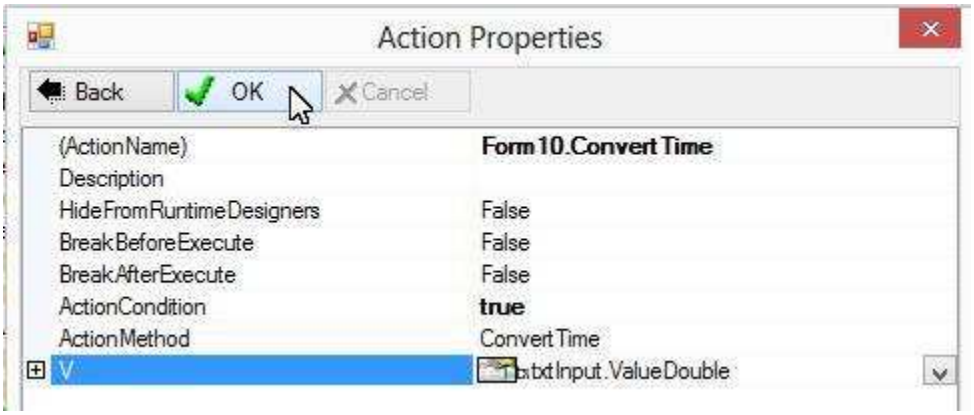


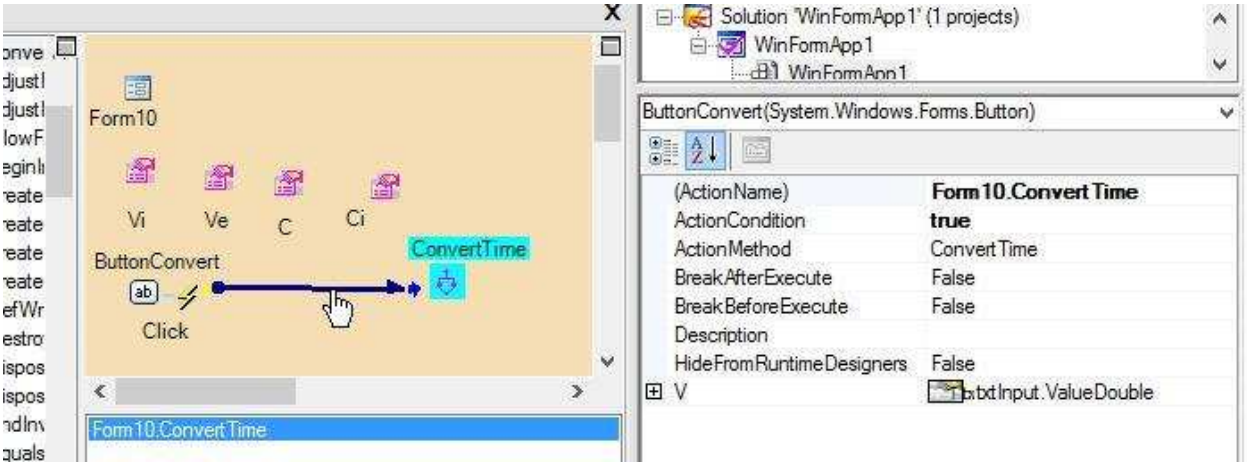

Pass the input to the method:





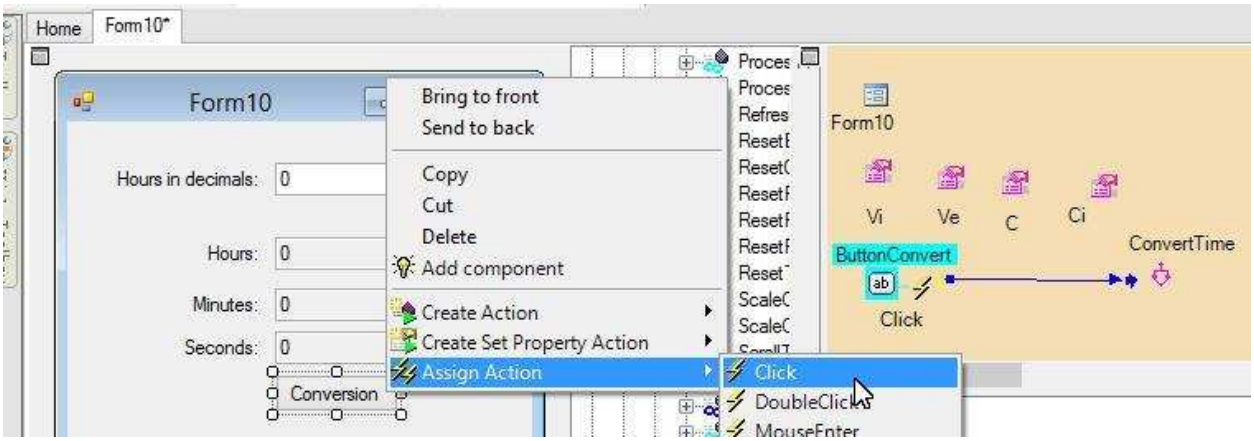


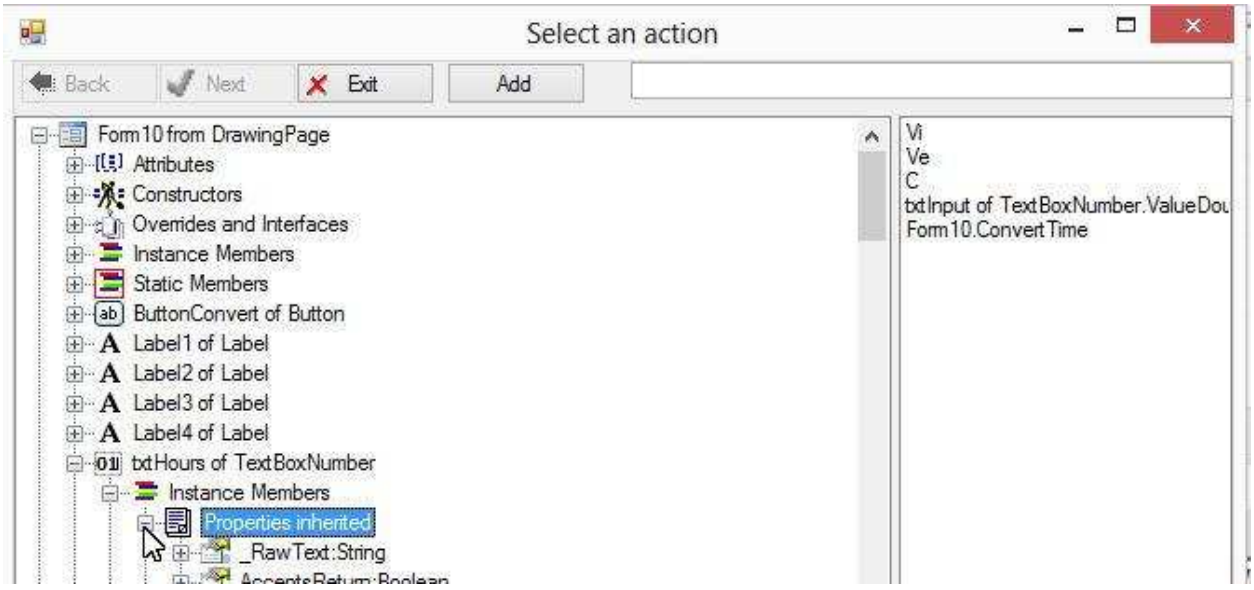

The action is created and assigned to the button:

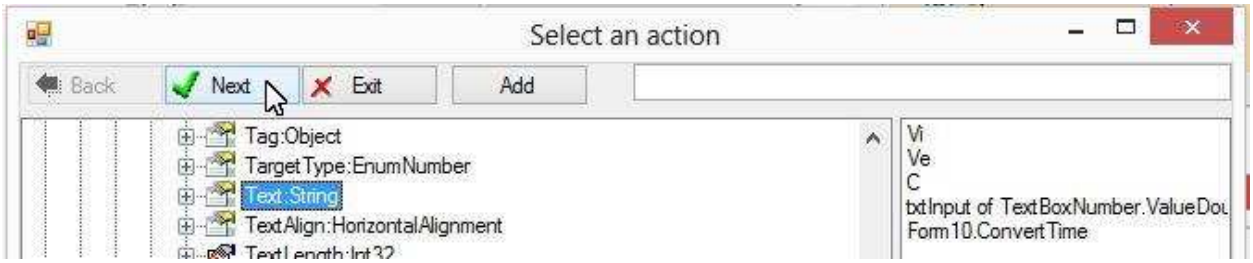


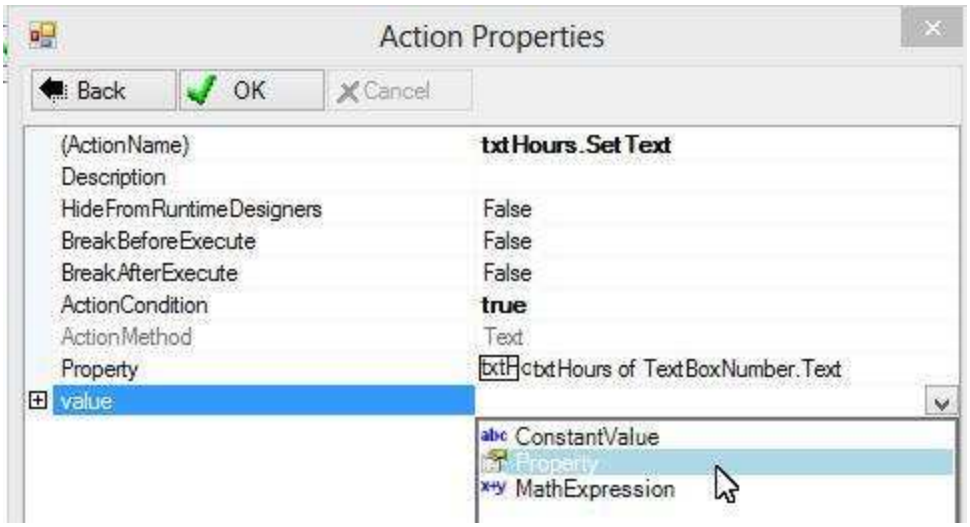

## Show hours and minutes

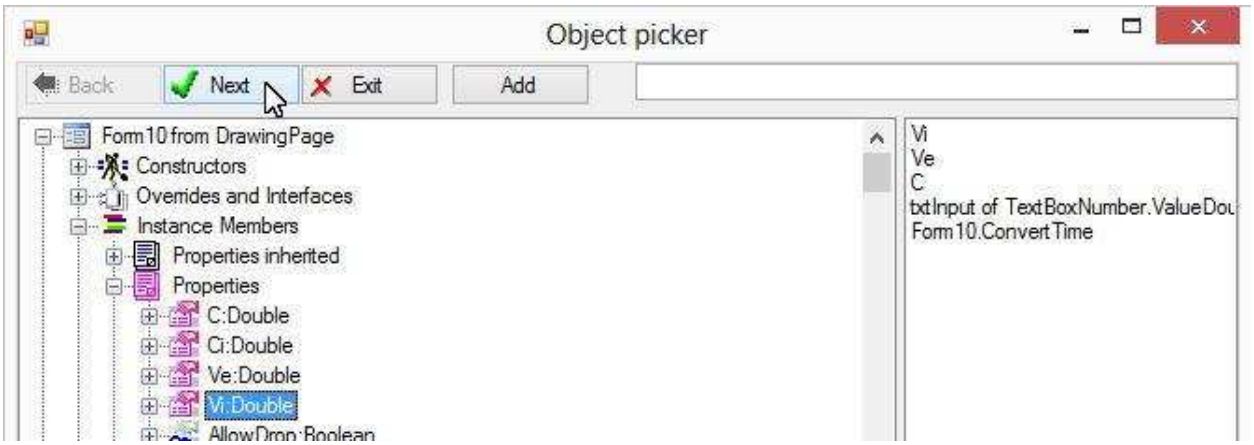
After executing the action, Vi is the integer portion of hours; Ci is the integer portion of the minutes. We may show them in the text boxes:

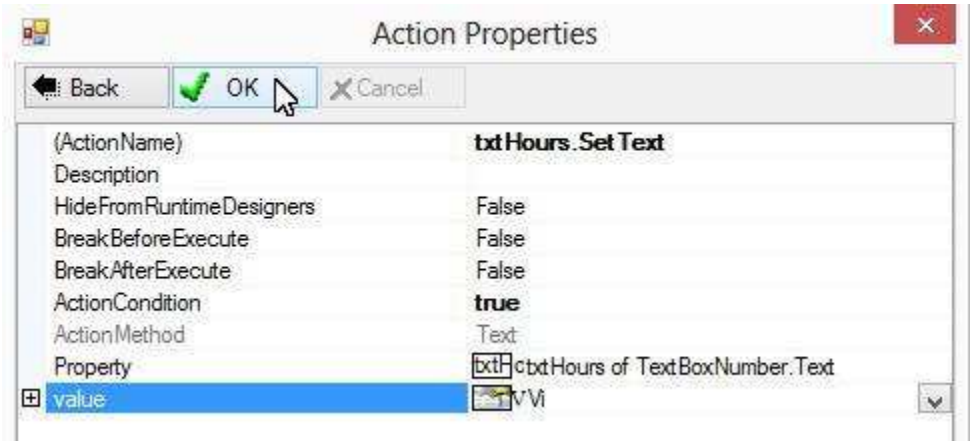


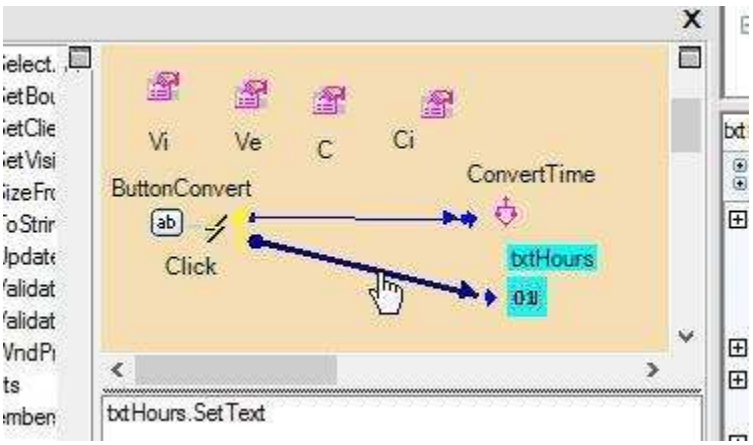

Select an action
Back
Next
Exit
Add
Tag:Object
TargetType:EnumNumber
Text:String
TextAlign:HorizontalAlignment
Vi
Ve
C
txtInput of TextBoxNumber.ValueDou
Form10.ConvertTime


Action Properties
Back
OK
Cancel
(ActionName)
txtHours.SetText
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Text
Property
txtHoctxtHours of TextBoxNumber.Text
value
ConstantValue
Property
MathExpression

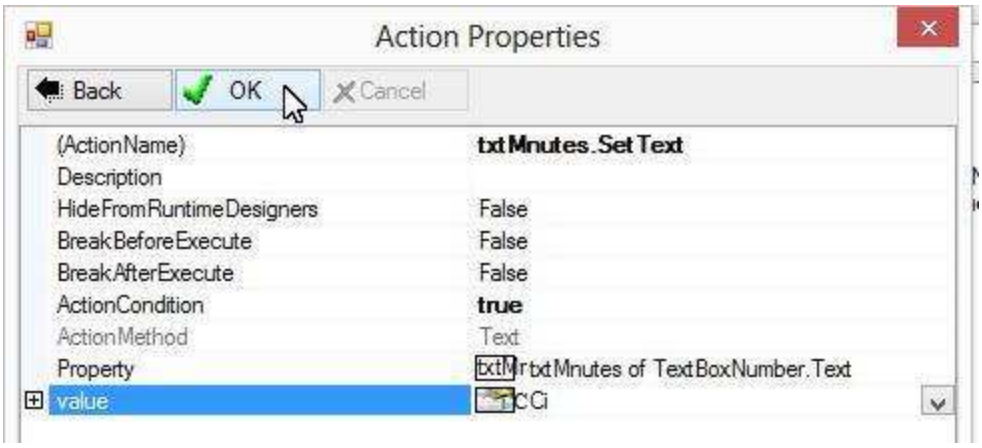

Object picker
Back
Next
Exit
Add
Form10 from DrawingPage
Constructors
Overrides and Interfaces
Instance Members
Properties inherited
Properties
C:Double
Ci:Double
Ve:Double
Vi:Double
AllowDrop:Boolean
Vi
Ve
C
txtInput of TextBoxNumber.ValueDou
Form10.ConvertTime

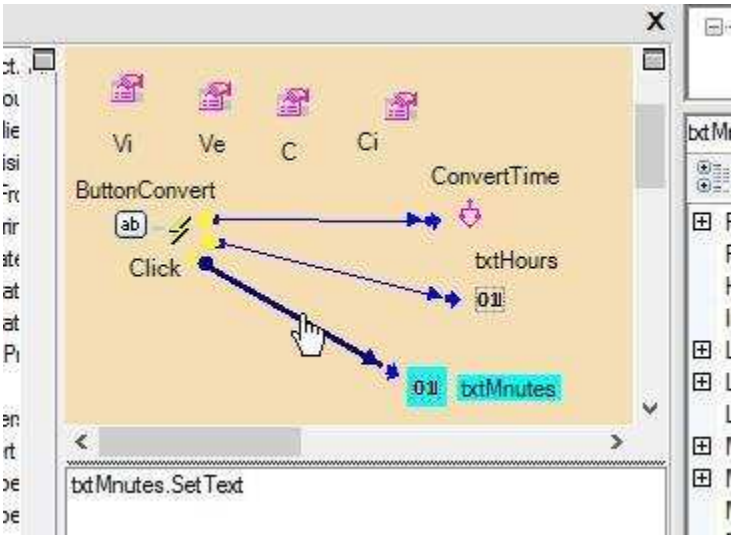



The action is created and assigned to the button:



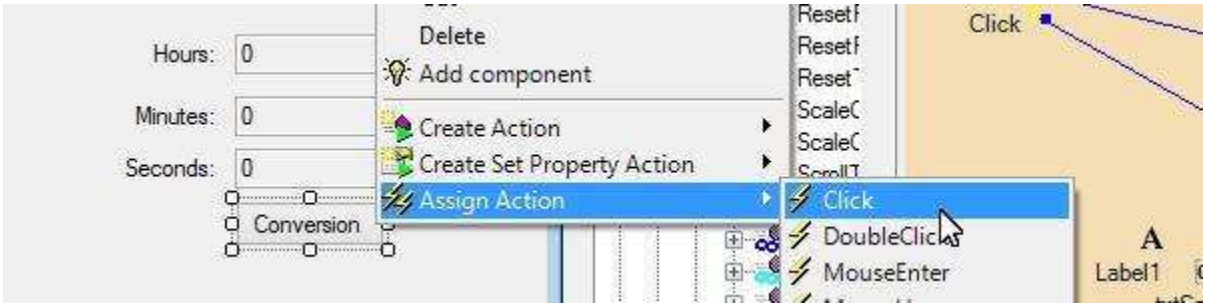

In the same way we show minutes represented by Ci:

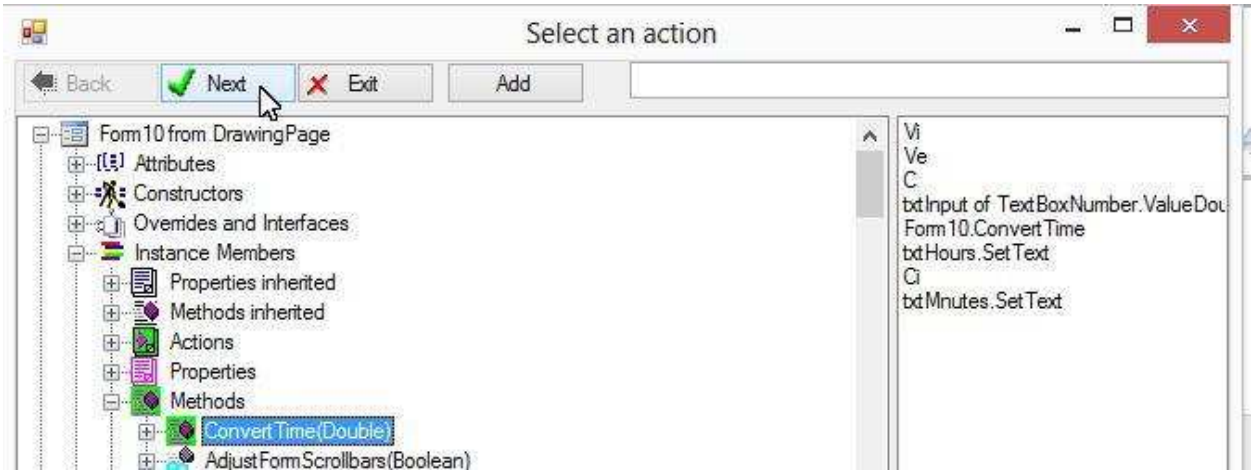




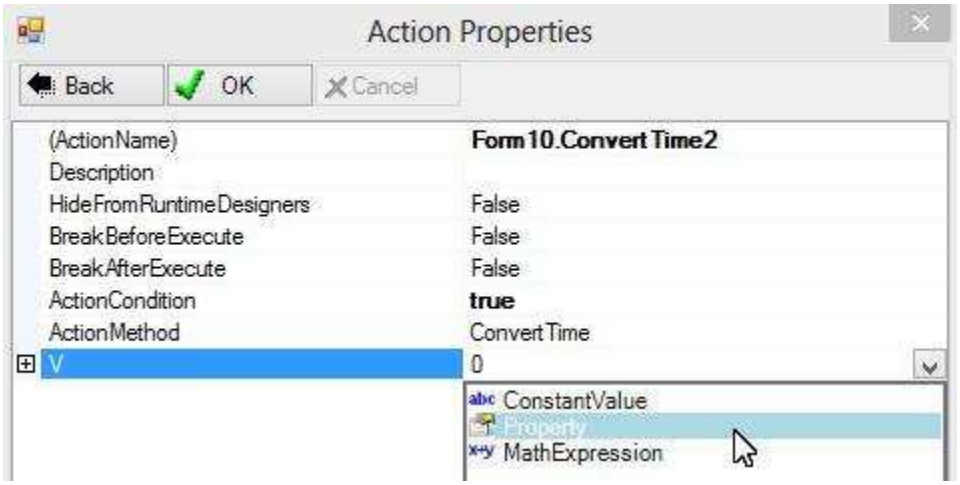

## Get seconds

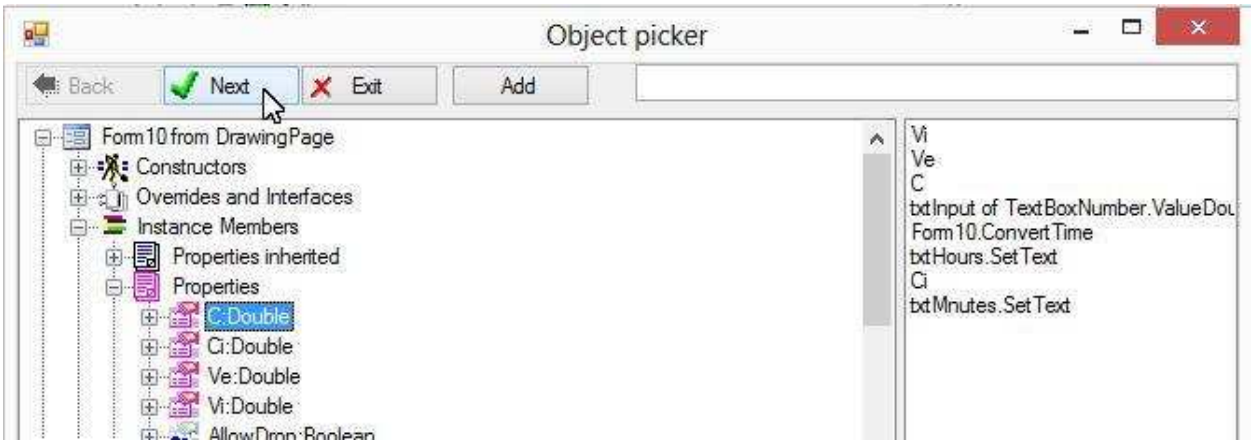
After executing the method for hours, C is the minutes. We may execute the method again for C to get seconds.

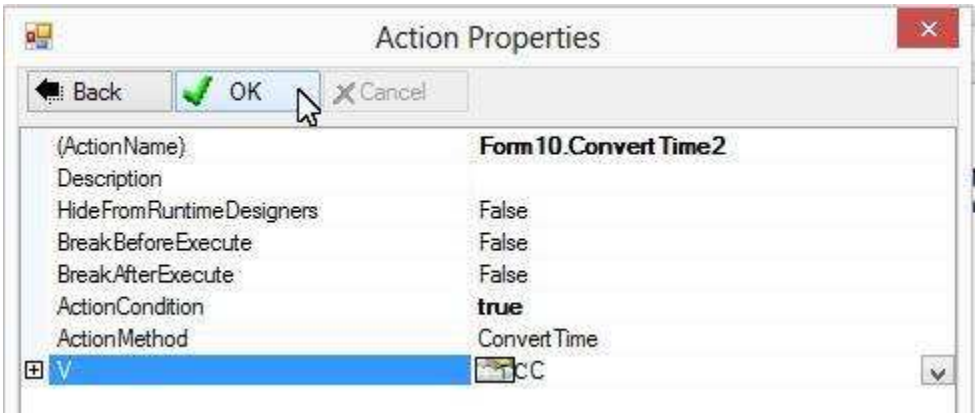


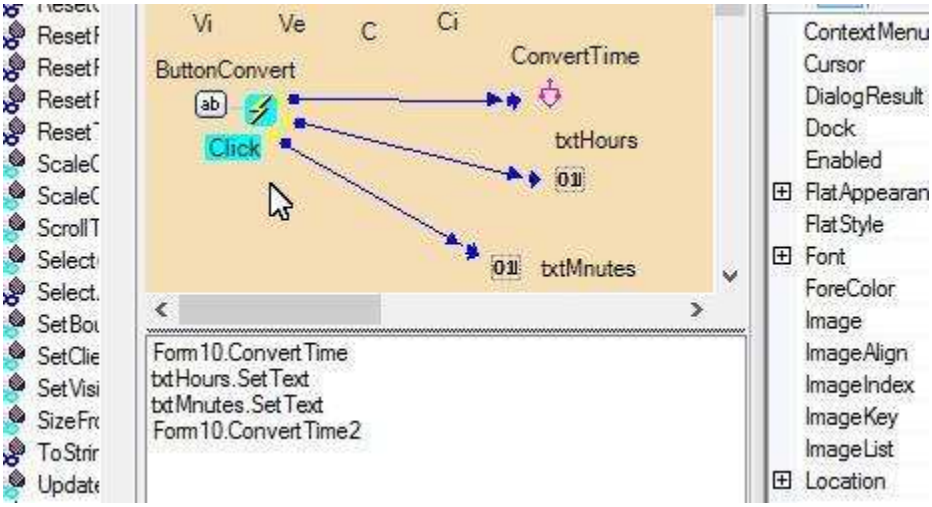

This time, pass C to the method:



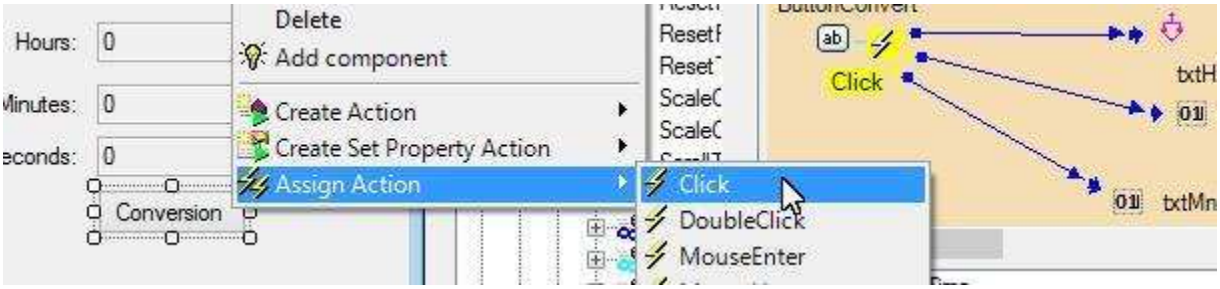



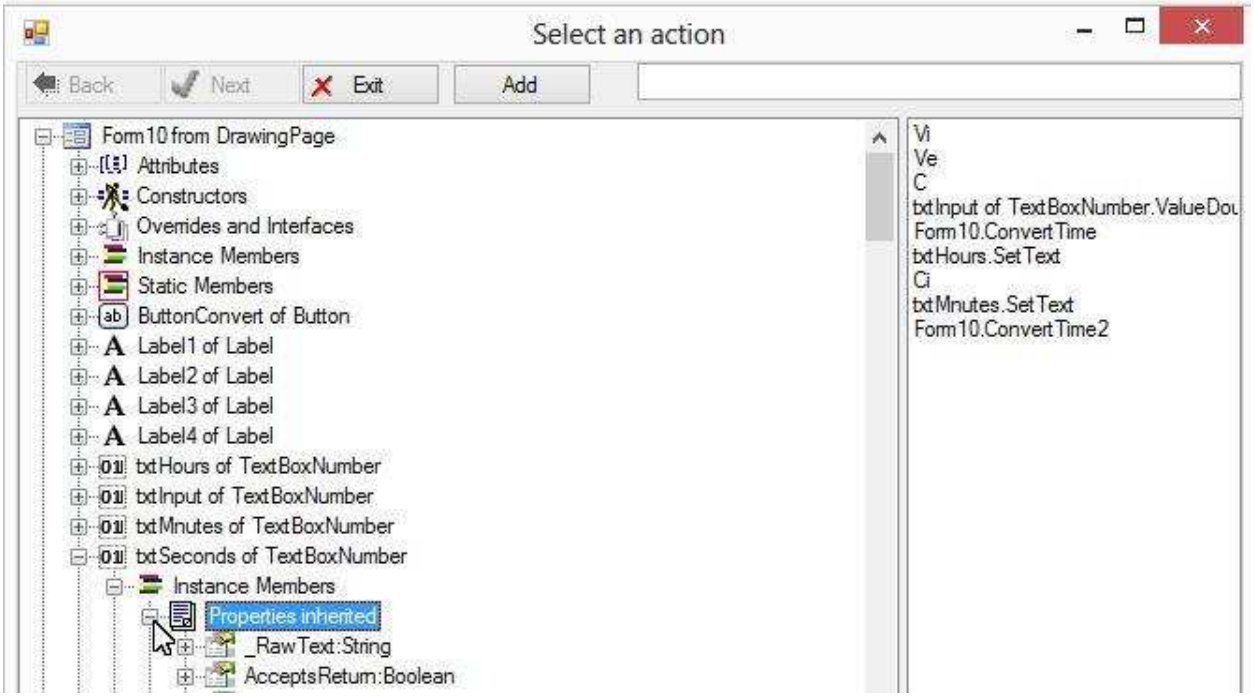

The action is created and assigned to the button:

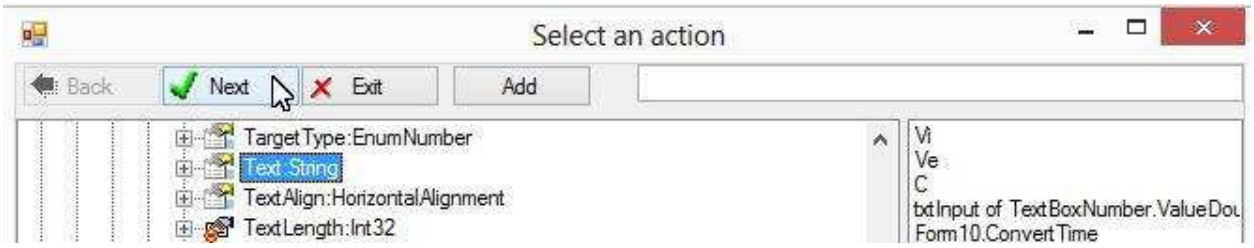


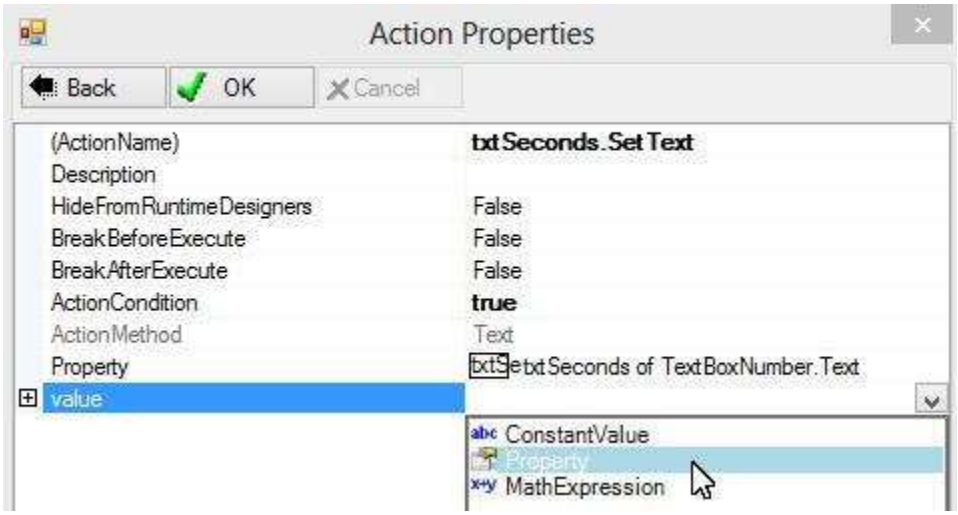

## Show seconds

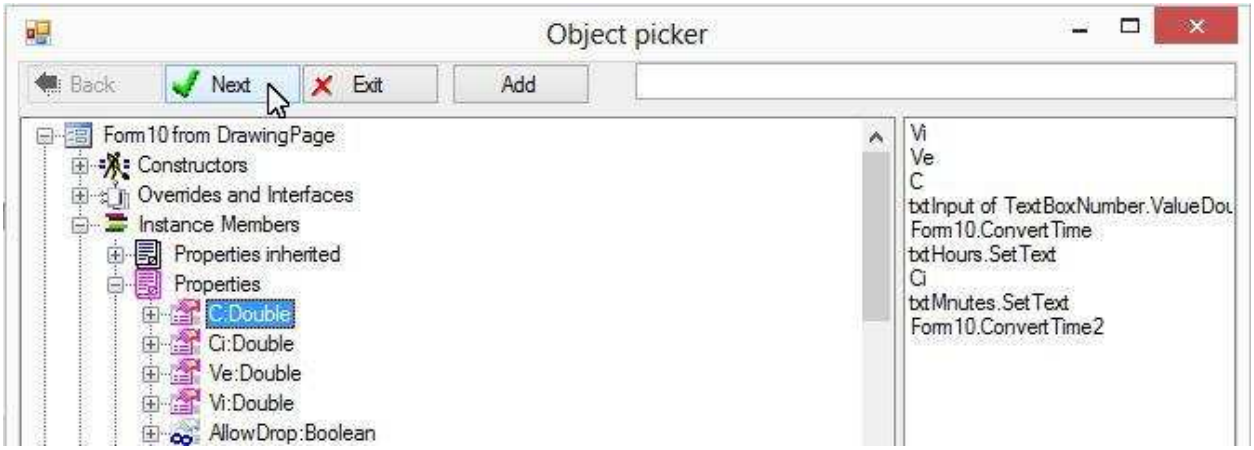
After executing the method, C is the seconds. Let's display it in the text box for seconds.

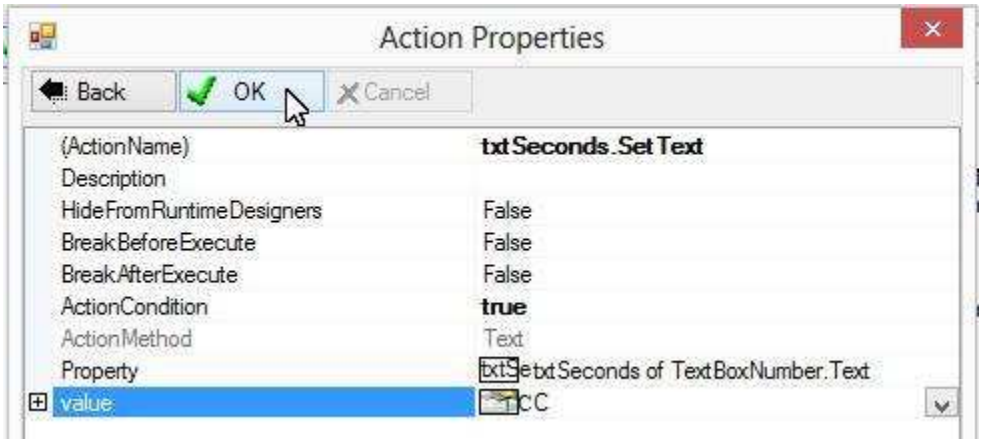


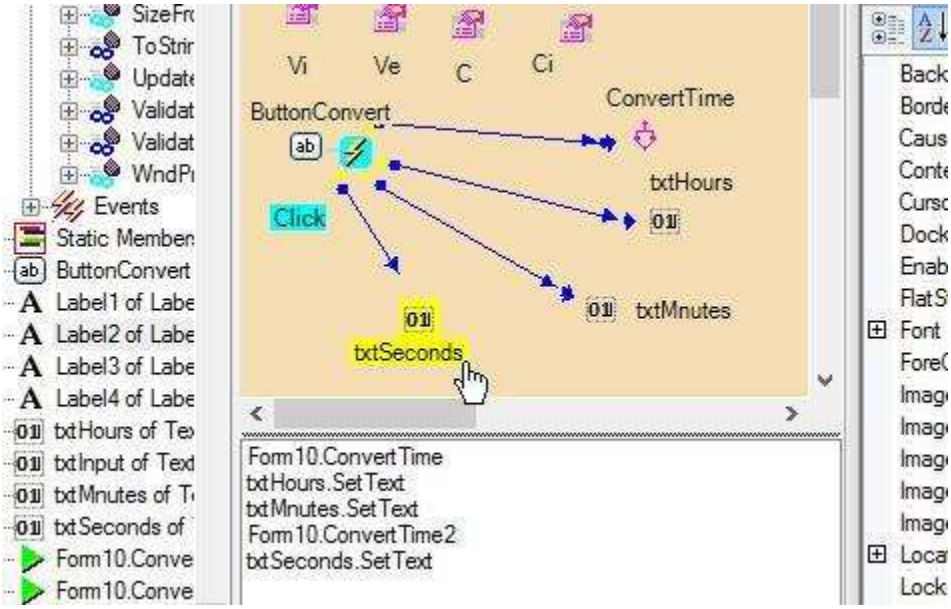

Select an action
Back
Next
Exit
Add
TargetType:EnumNumber
Text:String
TextAlign:HorizontalAlignment
TextLength:Int32
Vi
Ve
C
txtInput of TextBoxNumber.ValueDou
Form10.ConvertTime


Action Properties
Back
OK
Cancel
(ActionName)
txtSeconds.SetText
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Text
Property
txtSetxtSeconds of TextBoxNumber.Text
value
ConstantValue
Property
MathExpression


Object picker
Back
Next
Exit
Add
Form10 from DrawingPage
Constructors
Overrides and Interfaces
Instance Members
Properties inherited
Properties
C:Double
Ci:Double
Ve:Double
Vi:Double
AllowDrop:Boolean
Vi
Ve
C
txtInput of TextBoxNumber.ValueDou
Form10.ConvertTime
txtHours.SetText
Ci
txtMnutes.SetText
Form10.ConvertTime2

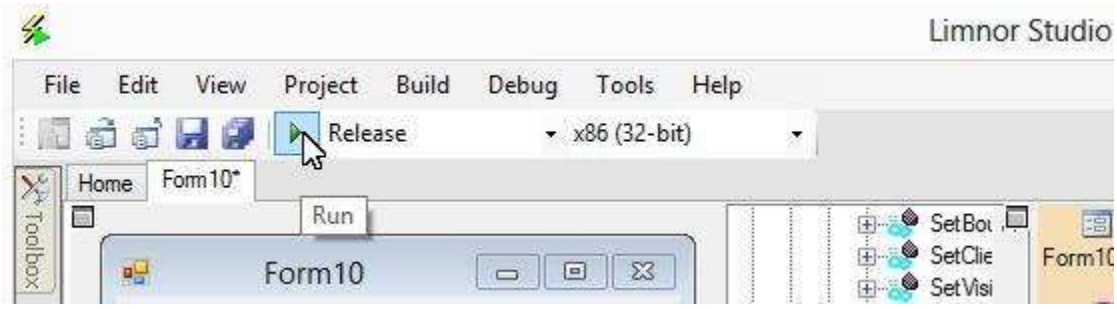



The action is created and assigned to the button:

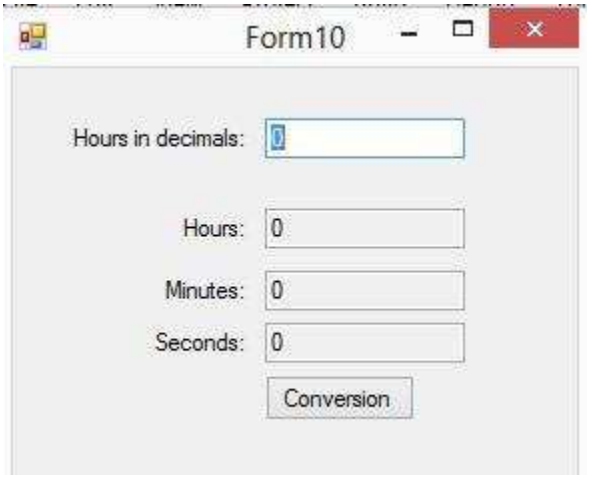



## Test

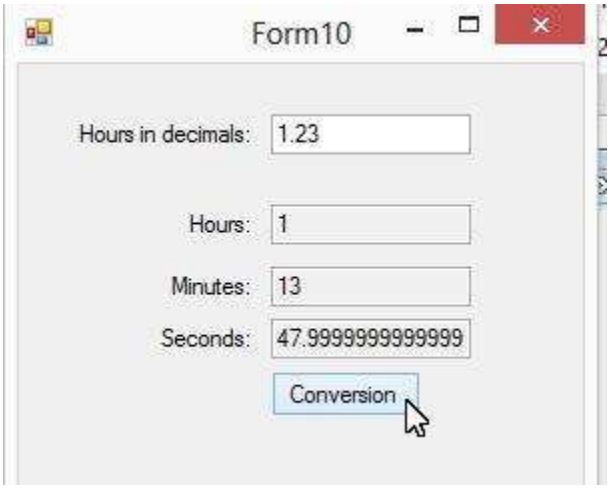
Run the project:



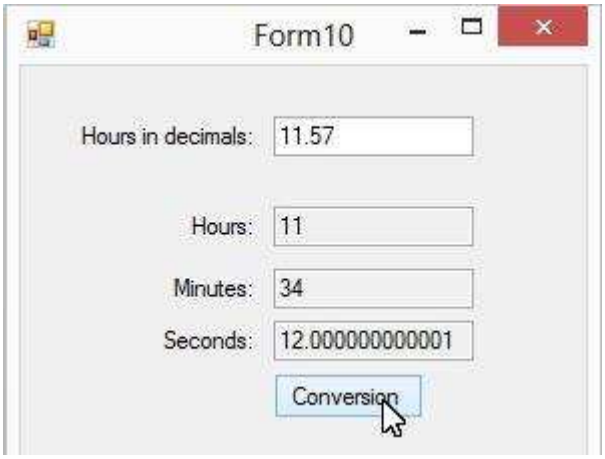

The form appears.




Enter a value for hours and click the button.




Enter another value and click the button to get another conversion:



## Feedback

You may send your feedback to support@limnor.com.