

Calculator Sample Help

Sample Description: [Calculator](#)

Points of Interest

[Calculation Overview](#)

[Summary of Functions](#)

Controls

Button

Label

For Help on Help, Press F1

Calculator

The Calculator application lets a user enter a numeric value by using the Mouse to click on various number buttons. Once a number is entered, many different calculations can be performed.

The purpose of this sample is to show how various numeric calculations are handled in Envelop. The Calculator sample works like most desktop calculators on the market. You can use the number keys to enter a desired number. Once a number is entered, you can click an arithmetic function button such as (+) for Add. Then you can enter a second number followed by the Equals (=) button. The results will be displayed in the Label control at the top of the form.

The Calculator handles both positive and negative numbers. In addition, when working with the Trig functions, numbers can be entered in Degrees or Radians.

Calculation Overview

To understand how calculations are performed, you need to know about the additional properties that have been added to the form. These properties are:

<u>Property</u>	<u>Data Type</u>	<u>Used To...</u>
ComputeStack	Double	Compute the results of calculation.
InputStack	String	Store the input. Same as Display.
Operation	String	Record the numeric operation.
ClearFlag	Integer	Indicate the entry should be cleared.

As numbers are entered, they are processed in a Sub procedure named **AppendNumber**. This procedure determines if it is working with positive or negative values, and appends number input to the **InputStack** property. Once input is appended, a Sub procedure named **UpdateDisplay** is called to copy the information from the InputStack property to the label named **lblResults**. Basically, lblResults.Caption contains the same information as the InputStack property.

When a function button, such as Add (+) is clicked, the Operation property is updated to display the name of the function, in this case the word "Add." Once this is done, a Sub procedure **InputStack2ComputeStack** is called. This method takes the value (Val) of the InputStack and copies it as a double-precision number to the **ComputeStack** property.

When the Equals (=) button is clicked, the following method is executed:

```
Sub btnEquals_Click()  
    If Operation == "" Then Exit Sub  
    ' We will comment out the following line for now since it chokes  
    ' if you step over it in debug mode  
    If ComputeStack == 0 Then Exit Sub  
    ' This guys is suppose to carry out the desire function  
    Select Case Operation  
        Case "Add"  
            lblResults.Caption = ComputeStack + Val(InputStack)  
        Case "Subtract"  
            lblResults.Caption = ComputeStack - Val(InputStack)  
        Case "Multiply"  
            lblResults.Caption = ComputeStack * Val(InputStack)  
        Case "Divide"  
            If Val(InputStack) == 0 Then  
                MsgBox.Message("Calculation Error", "Can't divide by zero!")  
                Exit Sub  
            Else  
                lblResults.Caption = ComputeStack / Val(InputStack)  
            End If  
        End Select  
    InputStack = lblResults.Caption  
    ' Reset the compute stack  
    ComputeStack = 0  
    ' Set the clear display flag  
    ClearFlag = 1  
End Sub
```

Basically, this method performs a calculation between the ComputeStack and InputStack properties that corresponds with the name in the Operation property. Once the calculation is performed, the result is placed back on the InputStack and the display is updated.

Trig functions are handled differently. They are designed to act directly on the InputStack and immediately update the display. The following method is executed when the Sin Trig function button is clicked:

```
Sub btnSin_Click()  
    Dim pi As String  
    Dim radians As String  
    pi = 4 * atn(1)  
    radians = (2 * pi) / 360  
    If InputStack == "" Or InputStack == "0" Then Exit Sub  
    If UNITS == "Degrees" Then  
        InputStack = sin(Val(InputStack) * radians)  
    Else
```

```
        InputStack = sin(Val(InputStack))
    End If
    UpdateDisplay
    ClearFlag = 1
End Sub
```

This code shows you how a calculation is performed directly on the contents of the InputStack property. Depending on whether the current UNITS mode is Degrees or Radians, a different calculation is performed. The result is placed back on the InputStack and the display is then updated.

Summary of Functions

This Calculator sample supports the following functions:

Function	Description
+	Adds two numbers together
-	Subtracts one number from another
*	Multiplies one number by another
/	Divides one number by another
%	Divides the InputStack by 100 for percentage
Pi	Enters a value for "pi"
Ln x	Logarithm
e	Natural Log
Tan	Computes the tangent of a number
aTan	Computes the arc-tangent of a number
Sin	Computes the sin of a number
Cos	Computes the cosine of a number
Sqrt	Computes the square-root of a number
x2	Computes the square of a number
1/x	Computes the reciprocal of a number
Sto	Stores a number for later recall
Rcl	Recalls a stored number for calculation
C	Clears both the InputStack and the ComputeStack
Ce	Clears only the InputStack
0-9	Enters numbers 0-9
BS	Enters a backspace
=	Performs an arithmetic calculation between the InputStack and the ComputeStack

