Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.

Foxborough, MA 02035

Tel: 508-543-8600

Fax: 508-543-1503

E-mail: support@iconics.com

Web: www.iconics.com

© ICONICS, Inc. All rights reserved.

Contents

Introduction
Getting Started
Visual Basic Form Configuration
Entering the VB Code

Introduction

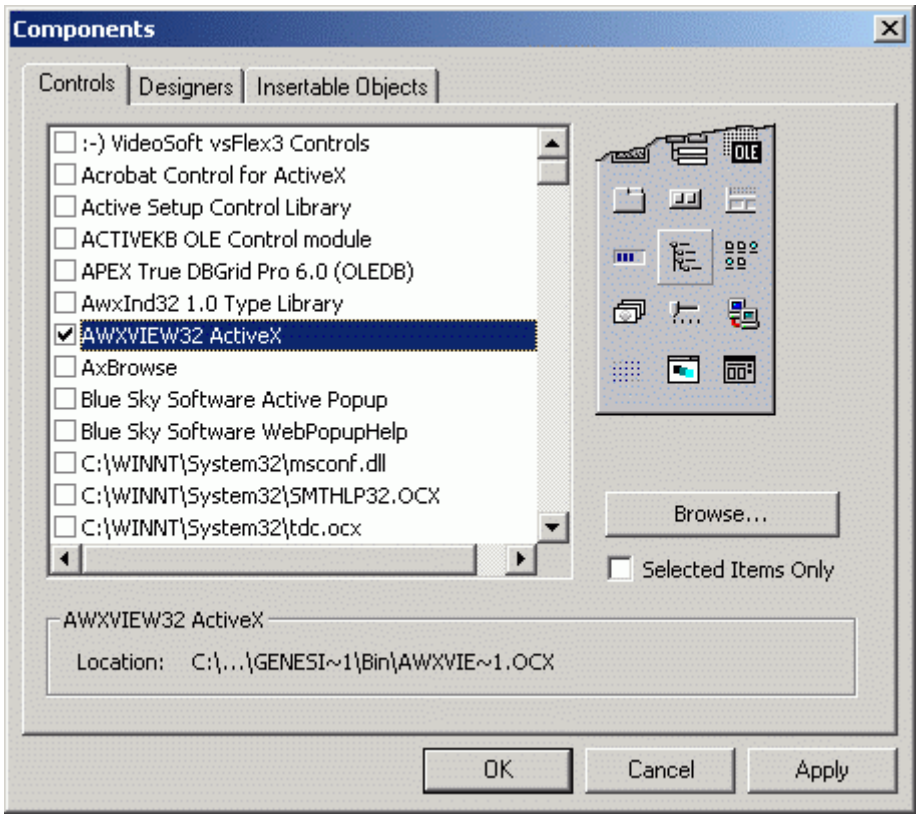
This example describes how to integrate the Alarm Logger when programming in Microsoft Visual Basic (VB). The example shows how to build a form to enable logging and printing.

Getting Started

1. Open a new **Standard EXE** project in Visual Basic. This opens a new blank Visual Basic form.

2. Before adding anything to this form, you must add the **AlarmWorX32 Viewer ActiveX** component to your toolbox. To do this, right-click anywhere in the toolbox and select **Component** from the pop-up menu. This opens the **Components** dialog box, as shown in the figure below. Select **AWXVIEW32 ActiveX** and click **OK.**

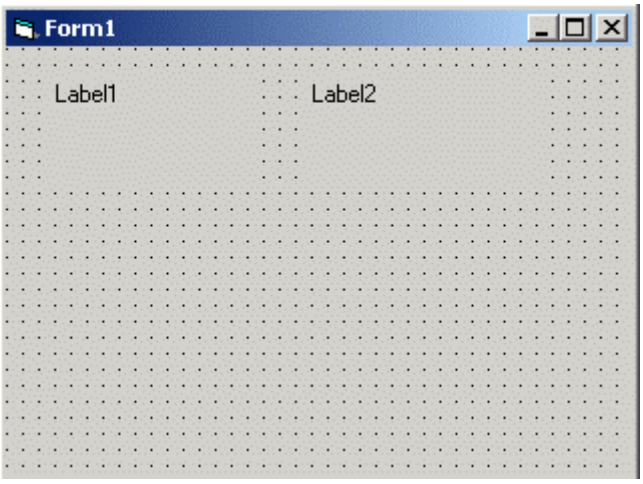
3. You are now ready to begin configuring the interface for the new Visual Basic form.



Components Dialog

Visual Basic Form Configuration

Click the **Label** button in the toolbox and draw two rectangles on the blank VB form that you created earlier. Your form should now look like the figure below.



VB Form With Two Labels

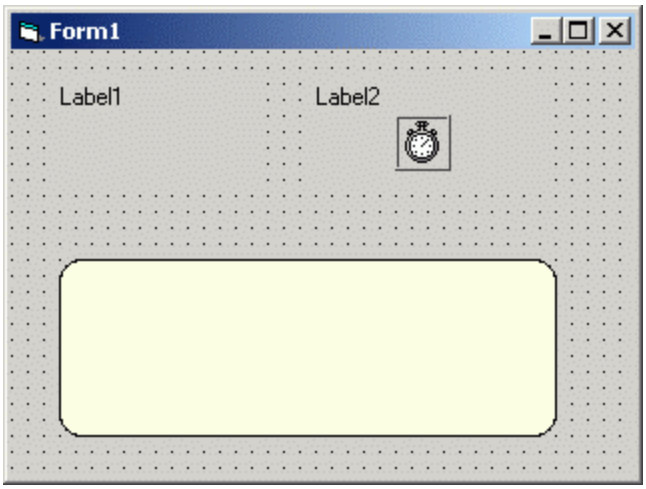
These labels will be referenced in later code so that, while the program is running, the date will be displayed on the field **Label1,** and the time will be displayed on the field **Label2.**

You must now insert a timer object over the **Label2** field.

Click the **Timer** button in the toolbox and draw a square over the **Label2** field. The timer will appear in the form.

Next add a shape where the action buttons will be located.

Click the **Shape** button in the toolbox and insert the shape on the form so that your form looks like the one in the figure below.



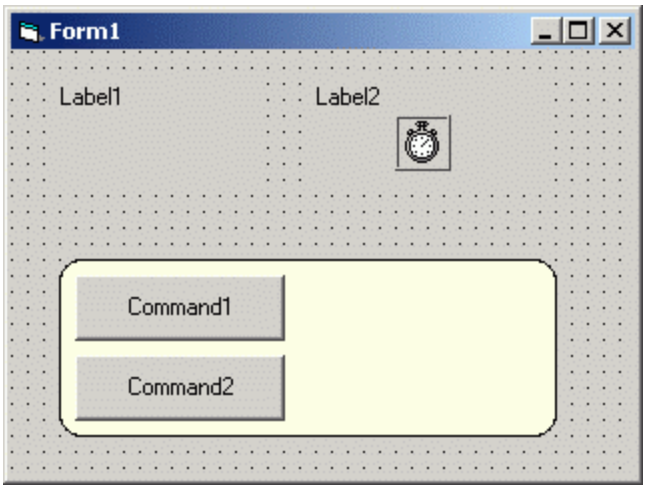
VB Form With Shape and Timer

To configure the shape, select it and then enter the following data in the **Properties** window.

(Name):	Shape1
BackColor:	Clicking the down opens a palette. Select a light color for the back color.
BackStyle:	1 - Opaque
BorderColor:	Choose something dark that will contrast with the Back Color you picked.
BorderStyle:	6 - Inside Solid
BorderWidth:	1
DrawMode:	13 - Copy Pen
FillColor:	Choose the same color as the Back Color
FillStyle:	1 - Transparent
Shape:	4 - Rounded Rectangle

Now you will add the action buttons and text fields on top of the shape.

Click the **CommandButton** button in the toolbox and insert two buttons over the shape that you have just configured. Your form should look like the one in the figure below.



VB Form With Command Buttons

To configure each button, select it and then enter the following data in the **Properties** window.

(Name)	**CmdEnableAlarms**	**CmdEnablePrinting**
Appearance :	1 - 3D	1 - 3D
Caption:	&Enable Logging	E&nable Printing

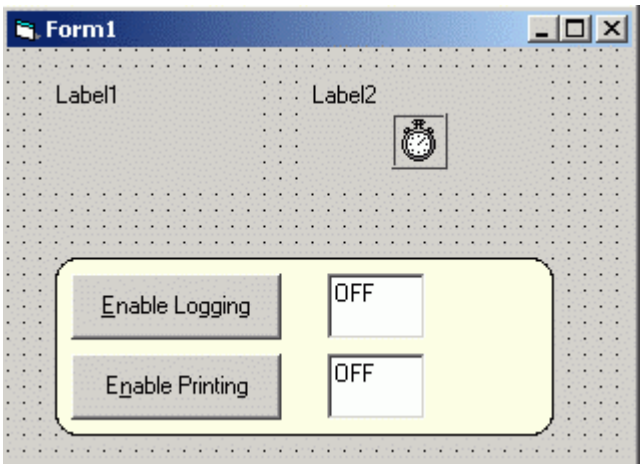
Note: To successfully enable printing, it is necessary to configure a printer in the AlarmWorX32 Logger.

Click the **TextBox** button in the toolbox and insert two textboxes (one after each button). These text boxes will indicate whether logging and printing are enabled.

To configure each textbox, insert the following data in the properties window:

(Name):	**TxtEnableLog**	**TxtPrinting**
Alignment:	0 - Left Justify	0 - Left Justify
Appearance:	1 - 3D	1 - 3D
BorderStyle:	1 - Fixed Solid	1 - Fixed Solid
Font:	MS Sans Serif - 12pt	MS Sans Serif - 12pt
Text:	OFF	OFF

Your form should look like the one in the figure below.



VB Form With Text Boxes

The final button you will add is the **Exit** button, which will effectively exit the user out of this form.

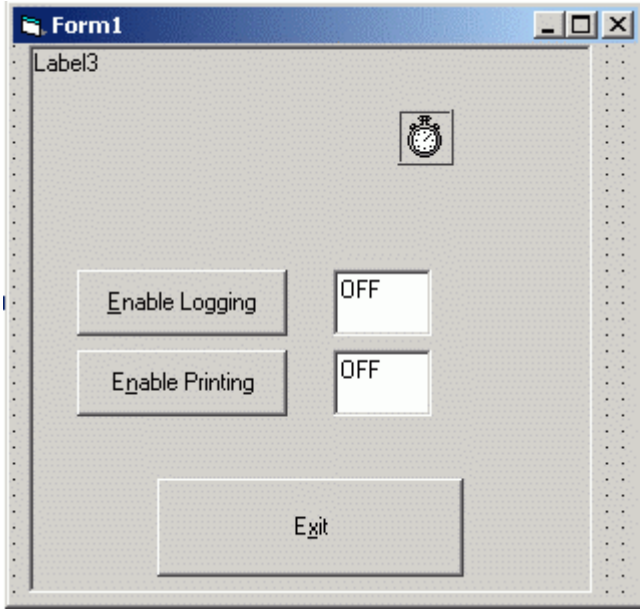
Click the **CommandButton** button in the toolbox and insert a button underneath the previously inserted shape.

To configure the button, enter the following data in the properties window:

(Name):	**ComExit**
Appearance:	1 - 3D
Caption:	E&xit

For the purposes of appearance and grouping, we will add a label component that will enclose all of the components that were just added.

Click the **Label** button in the toolbox and insert the label so your form looks like the one in the figure below.



VB Form With Exit Button

To configure the label, select it and then enter the following data in the **Properties** window.

	Label3
Alignment	0 - Left Justify
Appearance	1 - 3D
Back Style	1 - Opaque
BorderStyle	1 - Fixed Single

Now it is time to enter the code behind the components!

Entering the VB Code

To enter the code behind a component, double-click on the component. This opens the **Project - Form (Code)** window. Before you enter code for any of the components, enter the following definitions by double-clicking on the form itself:

```
Dim b_LoggingEnabled As Boolean

Dim b_PrintingEnabled As Boolean

Dim Logger As New AWXLog32Auto
```

It is important to enter these definitions because they are referenced in later code.

1. The first component for which you will enter the code is the **Enable Logging** button. When selected, this button will turn on or off alarm logging. Double-click the **Enable Logging** button and enter the following code:

```
Private Sub CmdEnabelAlarms_Click()
 'will Enablel/Disable AlarmLogger.
  On Error GoTo INIT_ERR

  If b_LoggingEnabled = False Then 'if its off then swich it on
      b_LoggingEnabled = True
      TxtEnableLog.Text = "ON"
      Logger.LoggingEnabled = b_LoggingEnabled
  Else                          'or, its on then swich it off
      TxtEnableLog.Text = "OFF"
      b_LoggingEnabled = False
      Logger.LoggingEnabled = b_LoggingEnabled
 End If
Exit Sub
'
INIT_ERR:
If Err = -2147467259 Then   ' AlarmLogger is not Loaded yet then just
waite
   WaitToLoadAlarmLogger
   CmdEnabelAlarms_Click
End If
End Sub

Private Sub WaitToLoadAlarmLogger()
'this function will pause the time for The PauseTime in seconds.
Dim Start ', count
 PauseTime = 5
 'count = 0
 Start = Timer   ' Set start time.

 Do While Timer < Start + PauseTime
   DoEvents    ' Yield to other processes.
   'count = Timer - Start
 Loop
End Sub
```

2. The next component for which you will enter code is the **Enable Printing** button. When selected, this button will enable or disable printing. Double-click the **Enable Printing** button and enter the following code:

```
Private Sub CmdEnablePrinting_Click()
'will Enable/Disable printing the Alarms.
  On Error GoTo INIT_ERR

  If b_PrintingEnabled = False Then  'if its off then swich it on
      b_PrintingEnabled = True
       TxtPrinting.Text = "ON"
       Logger.PrintingEnabled = b_PrintingEnabled
  Else
       TxtPrinting.Text = "OFF"    'or, its on then swich it off
       b_PrintingEnabled = False
       Logger.PrintingEnabled = b_PrintingEnabled
  End If
Exit Sub
 '
INIT_ERR:
If Err = -2147467259 Then   ' AlarmLogger is not Loaded yet then just
wait
   WaitToLoadAlarmLogger
   CmdEnablePrinting_Click
End If
End Sub
```

3. Next you will enter the code for the **Exit** button. When selected, this button will exit the user out of the form. Double-click the **Exit** button and enter the following code:

```
Private Sub ComExit_Click()
   Unload Me
End Sub
```

4. Next you will enter the code for the entire form. Double-click the form and enter the following code:

```
Private Sub Form_Initialize()
       b_PrintingEnabled = False 'set the Alarm Printing to OFF.
End Sub

Private Sub Form_Load()
Dim MyDate, MyTime
```

```
Dim ob As Object
  Timer1.Interval = 1000  ' Set Timer interval for every second.
  MyDate = Format(Date, "dddd, mmm d yyyy") 'set the DATE format.
  Label1.Caption = MyDate  ' load the current date.

  Screen.MousePointer = vbHourglass
  On Error GoTo INIT_ERR
  If Logger.LoggingEnabled = False Then 'if its off then show it OFF in
the TxtEnableLog
      TxtEnableLog.Text = "OFF"
  Else                          'or, its ON then show it OFF in the
TxtEnableLog
      TxtEnableLog.Text = "ON"

  End If
  Screen.MousePointer = VBDEFULT
 Exit Sub

 '
INIT_ERR:
If Err = -2147467259 Then   ' AlarmLogger is not Loaded yet then just
waite
   WaitToLoadAlarmLogger
   Form_Load
End If

End Sub
Private Sub Form_Unload(Cancel As Integer)
   Set Logger = Nothing  ' kill the Object Logger.
End Sub
```

5. Now you will enter the code for the **Timer** function. Double-click the **Timer** button and enter the following code:

```
Private Sub Timer1_Timer()
 'just to update the Clock at every second.
  Label2.Caption = Time 'Load the current time.
End Sub
```

6. Once you have entered all of the above code, you must save and compile your project. Select **Save As** from the **File** menu and save the project as **VBAlarmTest.exe.** To compile the project, select **Make VBAlarmTest.exe** from the **File** menu.