TELEDYNE DALSA
Everywhereyoulook™

# *Developing a Visual Basic .NET interface for a Sherlock 7 investigation*

# Overview

Although a Sherlock 7 investigation can be run from the Sherlock IDE, it is often desirable, and sometimes necessary, to hide the Sherlock GUI behind a custom interface.  This tutorial walks you through the steps of creating a Visual Basic .NET front end for a Sherlock 7 investigation.

# Requirements


To follow the steps in this tutorial, you will need

- Sherlock 7
- Microsoft Visual Basic .NET 2005 or newer[1,2]
- Familiarity with Visual Basic .NET

[1]A separate tutorial explains how to create an interface using Visual Basic 6
[2] No claim is made for the applicability of this tutorial to earlier versions of Visual Basic .NET (i.e., 2002 and 2003)

# Five steps

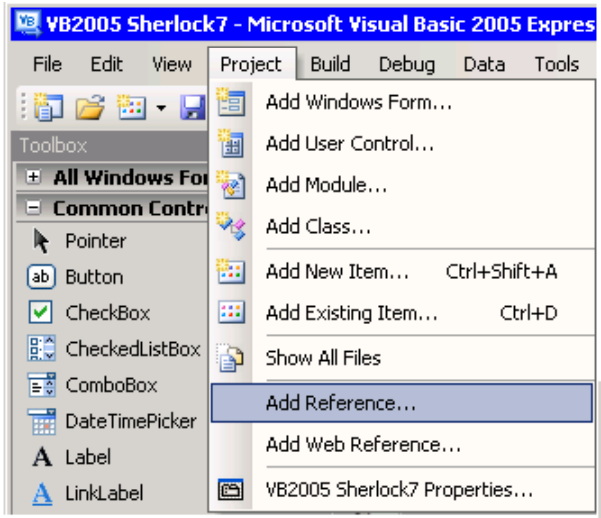
TELEDYNE DALSA
Everywhereyoulook™

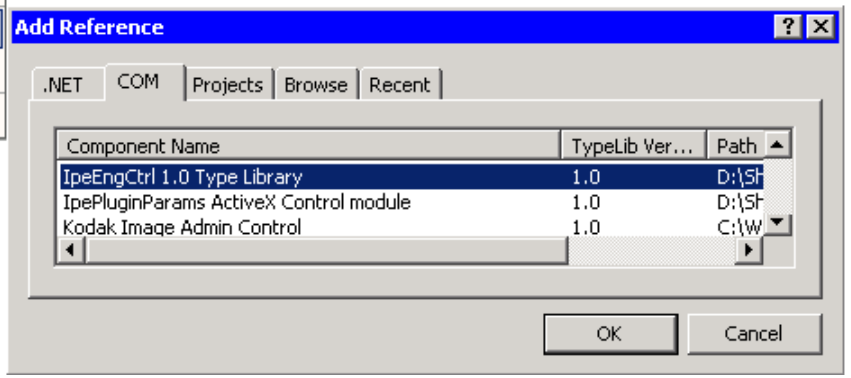
The five steps in creating a Visual Basic .NET (from here on, VB) front end for a Sherlock 7 investigation:

1. Add a reference to the Sherlock runtime engine
2. Declare and create a Sherlock object
3. Set up Sherlock display (optional, but common)
4. Use the Sherlock object to load, execute, control and communicate with an investigation
5. Destroy the Sherlock object

# Add the Sherlock reference

Open a new VB Windows Application project. From the project's main menu, select **Project → Add Reference...** On the **Add Reference** dialog, click the **COM** tab, select **IpeEngCtrl 1.0 Type Library**, and click the **OK** button.



# Create the Sherlock object


The Sherlock object must be created and initialized before you call any of its methods.

```
Public Class Form1
        Private WithEvents hSherlock As IpeEngCtrlLib.Engine
        Dim nErr As IpeEngCtrlLib.I_ENG_ERROR

Public Sub New()
        InitializeComponent()

        hSherlock = New IpeEngCtrlLib.Engine
        nErr = hSherlock.EngInitialize()
End Sub
```

# Load an investigation

The Form_Load subroutine is usually a good place to load the investigation you want to run.

```
Private Sub Form_Load(ByVal...SystemEventArgs) Handles MyBase.Load
    nErr = hSherlock.InvLoad("Widget.ivs")
End Sub
```

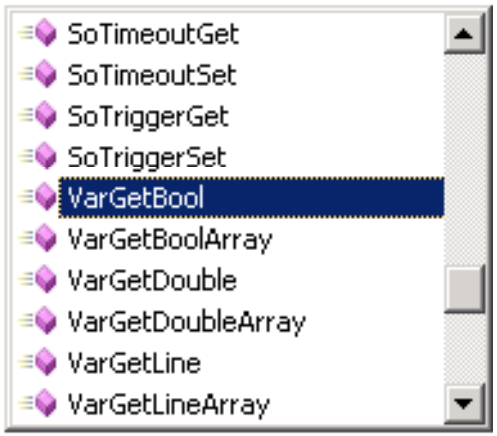
If you have several investigations that can use the same VB front end (for example, different but similar parts to be analyzed), the investigation name can be contained in a variable that is filled based on user input or some other mechanism.

# Autocomplete and pop-up tips

TELEDYNE DALSA
Everywhereyoulook™

As you type Sherlock object code, autocomplete displays a list of the matching methods…

```
nErr = hSherlock.var
```




…and pop-up tips show you the required parameters for the methods.

Name of the Sherlock Boolean variable to read, as a String

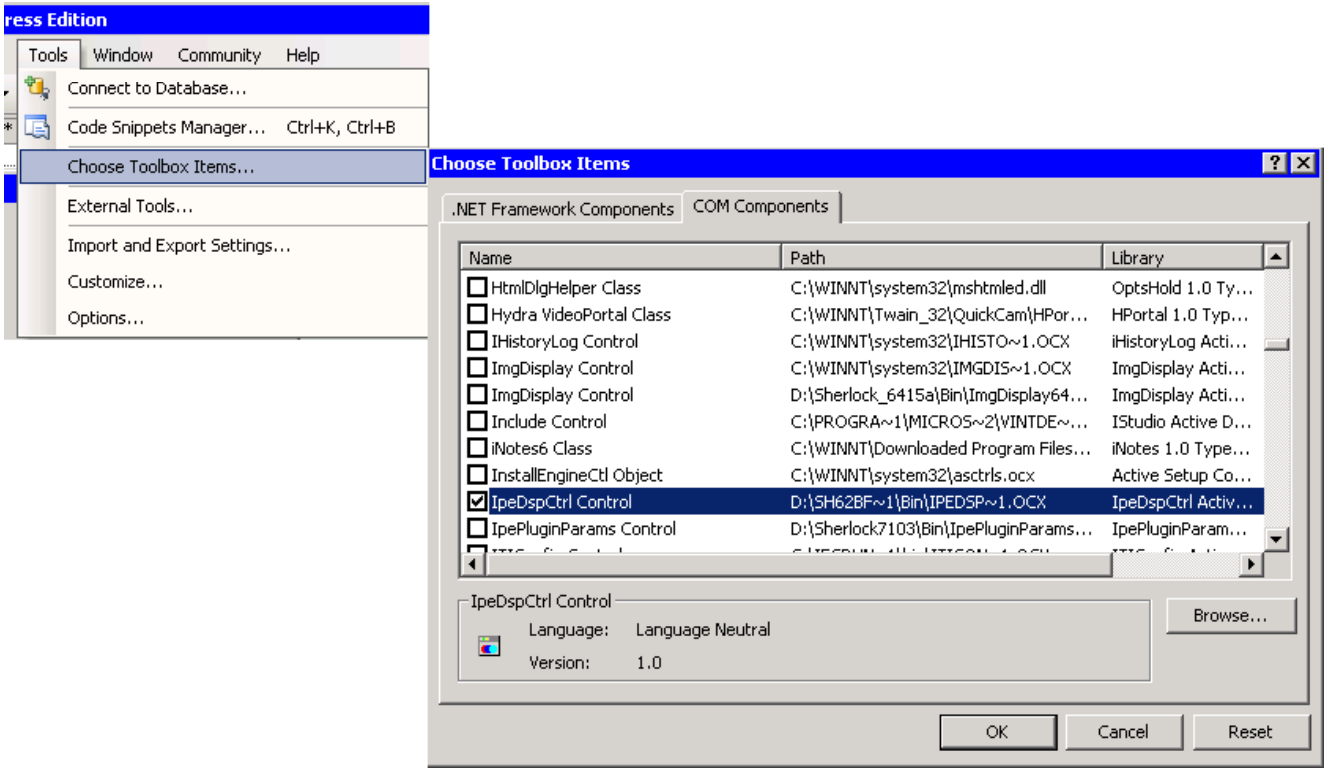
Returned value of the variable, as a Boolean

Error return of the function call

```
Public Function VarGetBool(bstrName As String, ByRef pval As Boolean) As IpeEngCtrlLib.I_ENG_ERROR
```

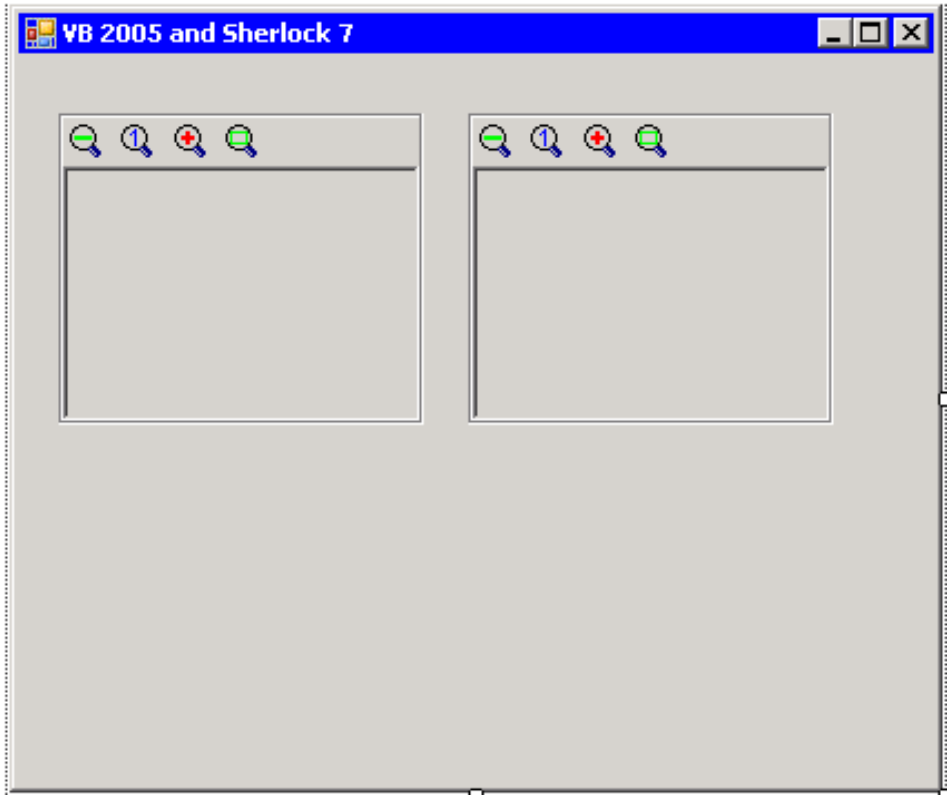
# Add the IpeDspCtrl control

To display an image window on the VB form, you must add an IpeDspCtrl display control. On the VB Toolbox, open the **Components** tab. From the VB project's main menu, select **Tools → Choose Toolbox Items…** On the **Choose Toolbox Items** dialog, click the **COM** Components tab, select **IpeDspCtrl Control**, and click the **OK** button. The IpeDspCtrl tool is added to the Components tab.

# Add display controls

You can add as many display controls to your VB form as you need. Their default names are AxIpeDspCtrl1, AxIpeDspCtrl2, etc., but you can rename them.



# Connect a display to an image window

To display an image window, you must connect it to a display control. You do not have to display any image windows, but it is common to display at least one.

```
Private Sub Form_Load()
    nErr = hSherlock.InvLoad("Widget.ivs")
    ' Connect the display object to Sherlock
    AxIpeDspCtrl1.ConnectEngine(hSherlock.GetEngineObj())
    ' Connect the display object to a Sherlock image window
    AxIpeDspCtrl1.ConnectImgWindow("imgA")
End Sub
```

# Run the investigation

Commands to run the investigation are usually executed in response to command button or menu item events.

```
Private Sub btnRunOnce_Click(ByVal...System.EventArgs)
            Handles btnRunOnce.Click
 ' Run the investigation once
 nErr = hSherlock.InvModeSet(IpeEngCtrlLib.I_MODE.I_EXE_MODE_ONCE)
End Sub

Private Sub btnRunContinuous_Click (ByVal...System.EventArgs)
            Handles btnRunContinuous.Click
 ' Run the investigation continuously
 nErr = hSherlock.InvModeSet(IpeEngCtrlLib.I_MODE.I_EXE_MODE_CONT)
End Sub
```

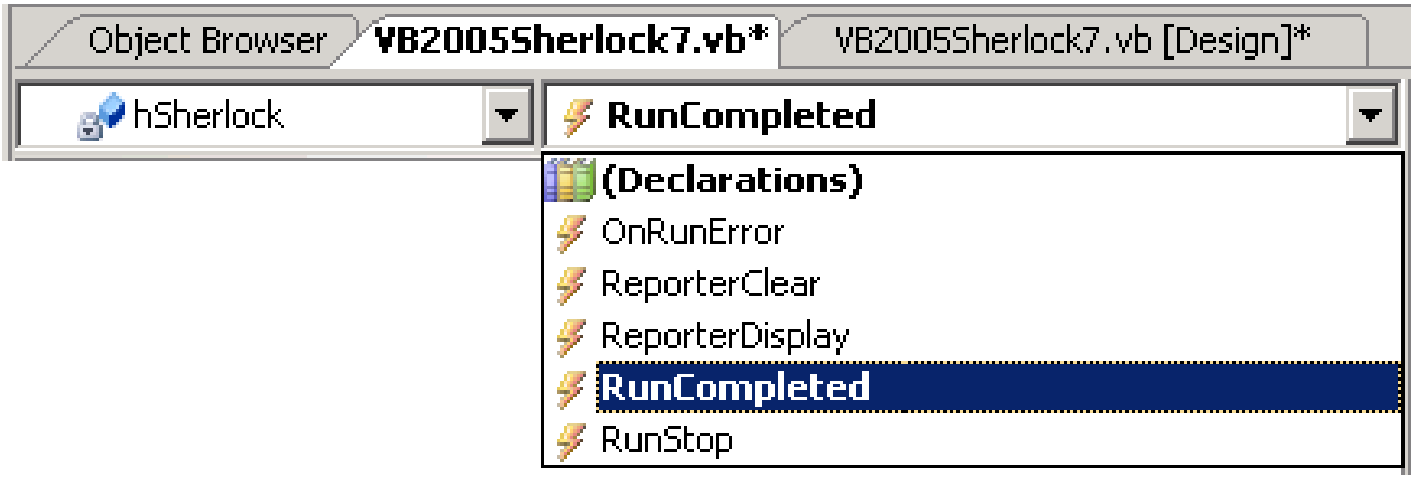
# Run the investigation


```
Private Sub btnHalt_Click(ByVal...System.EventArgs) Handles btnHalt.Click
    ' Halt a running investigation after the current iteration
    nErr = hSherlock.InvModeSet
            (IpeEngCtrlLib.I_MODE.I_EXE_MODE_HALT_AFTER_ITERATION)
End Sub

Private Sub btnAbort_Click(ByVal...System.EventArgs) Handles btnAbort.Click
    ' Abort a running investigation
    nErr = hSherlock.InvModeSet(IpeEngCtrlLib.I_MODE.I_EXE_MODE_HALT)
End Sub
```

# RunCompleted

At the end of every iteration of the investigation, the RunCompleted subroutine is automatically called.



# RunCompleted

Add code to this subroutine to read Sherlock algorithm readings, read and write Sherlock variables, update the VB form, etc.

```
Private Sub hSherlock_RunCompleted() Handles Sherlock.RunCompleted

    Dim dblCount as Double, dblAreas() as Double

    nErr = hSherlock.VarGetDouble("varCount", dblCount)
    labelBlobCount.Text = dblCount

    nErr = hSherlock.VarGetDoubleArray("varAreas", dblAreas)
    …
End Sub
```

# Destroy the Sherlock object


It is "best practice" to destroy the Sherlock object before the VB application terminates.

```
Protected Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
            If Not (hSherlock Is Nothing) Then
                    hSherlock.EngTerminate()
                    hSherlock = Nothing
            End If
            If Not (components Is Nothing) Then
                    components.Dispose()
            End If
            MyBase.Dispose(disposing)
   End If
End Sub
```

# Things to Note

TELEDYNE DALSA
Everywhereyoulook™

1. Every call to the Sherlock object function generates a return value. Best practice is to check this value:

```
nErr = hSherlock. VarGetDouble("varCount", dblCount)
If Not (nErr = IpeEngCtrlLib.I_ENG_ERROR.I_OK) Then
        ' Error-handling code
End If
```

2. To resize the contents of an image window to fill a display control that is either larger or smaller than the image window, call the method AxIpeDspCtrl.SetZoom(0).