UniLink – Polling a channel directly using Visual Basic

Sometimes you may wish to poll a Unimeter or UniMux directly, instead of from within the normal 'master form' timer loop.

Firstly there are two methods of calling the value from UniLink.

Using Names:

In the 'Modules' section of the database created by the UniLink Wizard, there is the following function in the 'UniLinkComms' module:

GetUnimeterValue( ):
Public Function **GetUnimeterValue**(ByRef App As Object, Name As String, ByRef AppRunning As Boolean) As Double

This is called after opening the Object (UniLink.exe), which is done in the master form's Form_Activate( ) function:

UniLinkAppRunning = ConnectUniLink(App, AppCreated)

This uses the following function in the 'UniLinkComms' module to start up UniLink.exe:

Public Function ConnectUniLink(ByRef App As Object, ByRef AppCreated As Boolean) As Boolean

After running this function, if UniLinkAppRunning = True, then a connection has been made to UniLink.exe, and the App object points to that connection.

So to get a value from a single unit or channel, the following would do the job: (The code in italics is not required if the amendment you are doing is in the master form code, as it is already there.

Defined in the top of the master form:
Public UniLinkAppRunning As Boolean
Public App As Object
Public AppCreated As Boolean

UniLinkAppRunning = ConnectUniLink(App, AppCreated)

```
Dim Stime As Date
Dim UniValue as Double
Dim Name as String
Dim Command as Long

Name = "Pump 1"
'The name string must match the string used in UniLink as
'shown on the front screen of UniLink (saved in
'UniLink.ini) It may include spaces, and if used in the
'Access database is in the UniDetails table 'Name' field.
'(Remember the 'ControlName' field can not contain spaces)
```

```
Command =0
'Generally use Command:=0 (gets local value from UniLink if
'it is polling, else forces poll if it's not polling. Use 1
'if you want to force a poll immediately)

On Error GoTo AppErr
UniValue = App.GetDeviceValue(Name, Command)
'Test for No Error Condition (Less than 1 Million)
If (UniValue < 1000000) Then
     Stime=Now    'Good value – record time
      'Now save UniMeterValue & Time to a Database Table
Else
     GetUnimeterValue = 0 'Or some other error value
End If

AppErr:
    MsgBox "Unable to Access 'UniLink' Server"
```

The functions in UniLinkComms are also provided in UniLinkLIBbas.MDA. This is a collection of functions which can be used in a non UniLink Wizard built Visual Basic application.

Using Comms Port number and ID of Unimeter:

Browse the UniLink automation library and select the most appropriate function eg:

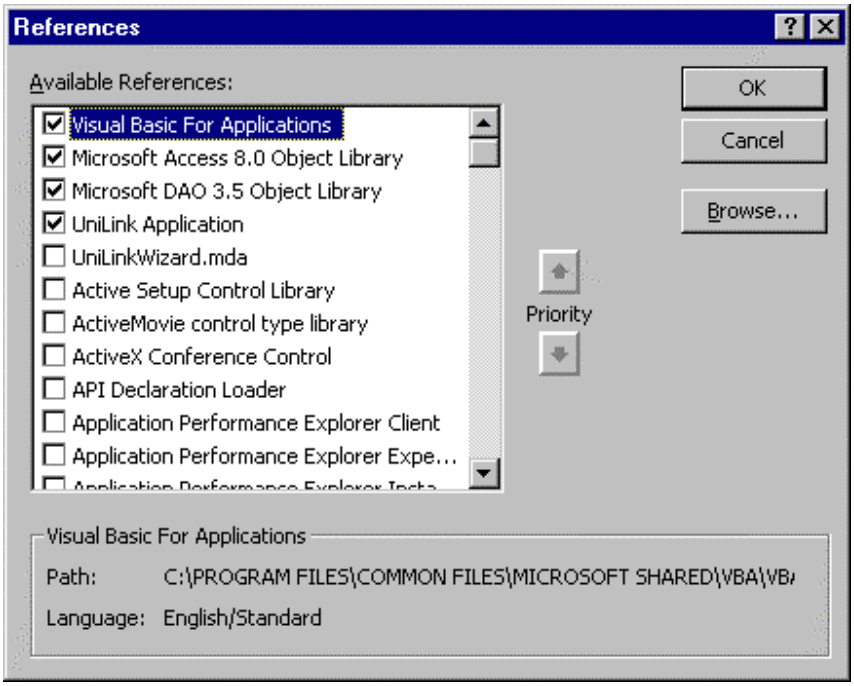
Function GetVal(Port As Long, UniID As Long) As Double:

```
Dim Port as Long
Dim UniID as Long
DimVal as Double
Port=2
UniID=1
      Val= GetVal(Port, UniID)
      If (Val < 1000000) then
            'Val OK
      Endif
```

Or:
Function GetMuxChannels(Port As Long, MuxID As Long) As String
'(Then decode with Public Function Decode_Mux_String(Muxstr As String) As Integer
' in the UniLinkComms module)

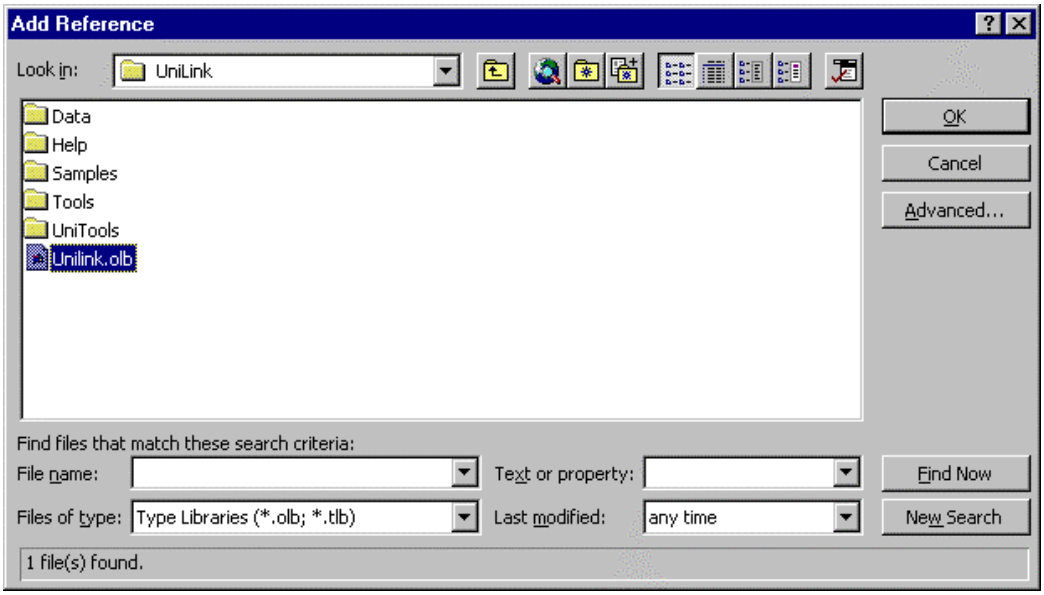
UniLink Automation Library

To browse the UniLink.olb library, open a code module, select ‘Tools | References’



If ‘UniLink Application’ is not shown as a reference:

- Click ‘Browse’
- Select the UniLink folder
- Select UniLink.olb
- Click OK & Click Ok again.



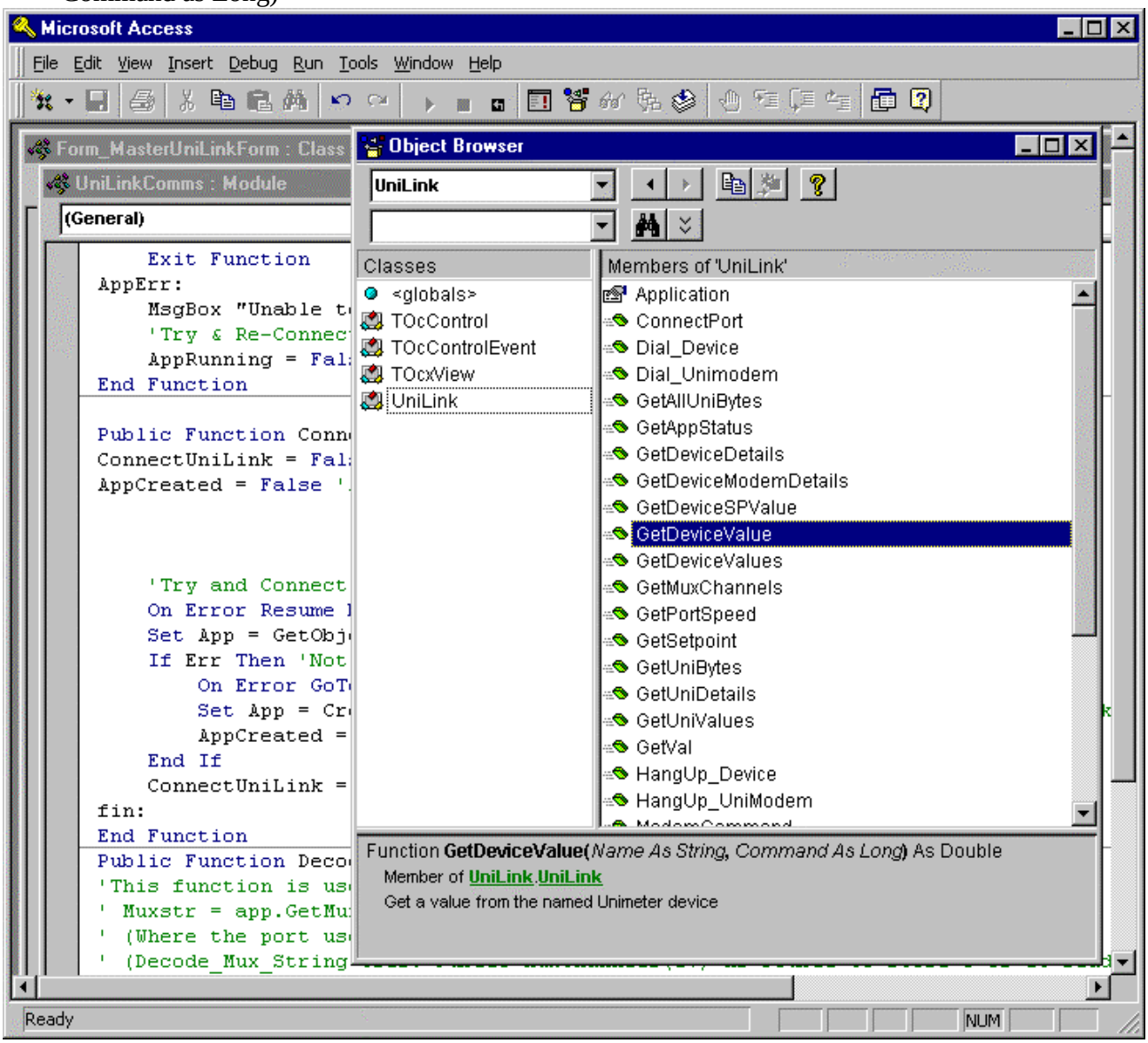
Now to browse the library functions made available by UniLink:

- From the open module, Click on ‘View | Object Browser’
- Select the UniLink library in the top left hand side drop box
- Click on UniLink in the left hand window (Ignore the Toc…. Names)
- Click on the Function you wish to browse – the details are shown in the bottom grey window.

The functions are used by referring to the global ‘App’ object created above:

App.GetDeviceValue(Name, Command)

Make sure the parameters specified are matched by your variable declarations. (Eg: Dim Command as Long)



UniLink Error Return Codes

The codes returned from UniLin.exe to a Visual Basic (Automation) call are based around a value of one million (1000000). If the value expected is say a double, test it for less than a million to make sure it is valid.

```
Dim Val as Double
Val= GetVal(Port, UniID)
If (Val < 1000000) then
      'Val OK
Endif
```

If the returned value is only used for error codes, then the same thing applies before accepting the result.

```
Dim Ret as Long
Ret=SetSetpoint(Port, UniID, SetpointNumber, Value)
If Ret < 1000000 Then
      'Setpoint was set
Else
      'Try again / Message to user
Endif
```