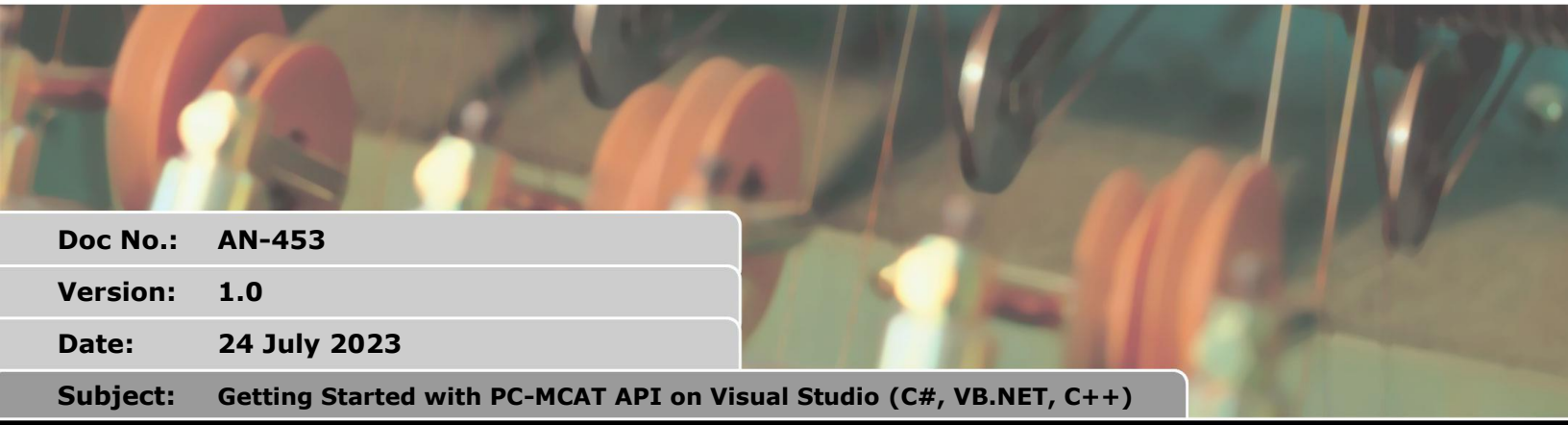
Shannon Way,
Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292 333
Fax: +44 (0)1684 297 929

187 Northpointe Blvd,
Suite 105
Freeport, PA 16229
United States of America
Tel: +1 724-540-5018
Fax: +1 724-540-5098

Tomson Centre
118 Zhang Yang Rd., B1701
Pudong New Area, Shanghai,
Postal code: 200122
CHINA
Tel/Fax: +86 21 587 97659

SCMC House
16/6 Vishal Nagar
Pimpale Nilakh, Wakad, Pune
PIN 411027
INDIA
Tel: +91 206 811 4902



Doc No.:	**AN-453**
Version:	**1.0**
Date:	**24 July 2023**
Subject:	**Getting Started with PC-MCAT API on Visual Studio (C#, VB.NET, C++)**

APPLICATION NOTE

1. Introduction

The PC-MCAT API is part of the PC-MCAT installation.

The PC-MCAT API communicates to the PC-MCAT via a shared memory interface. This means that applications that use the PC-MCAT API must execute on the PC-MCAT itself.

The shared memory interface is used to implement access to the following.

- Parameters: system, axis and process parameters.
- I/O: digital and analogue I/O.
- Moves: the moves are loaded directly into the PC-MCAT motion buffers.
- Programs.

The PC-MCAT API for .Net consists of the following files:

- PC-MCAT API for C++
  - pcmcat_api_x86.lib – This library implements the API commands compiled for 32 bits.
  - pcmcat_api_x64.lib– This library implements the API commands compiled for 64 bits.
- PC-MCAT API for .Net
  - pcmcat_net.dll – This DLL implements the .Net assembly (x86 and x64 compatible).

The PC-MCAT API for C++ libraries and headers are installed into the PC-MCAT program directory, usually C:\Program Files\TrioMotion\PC-MCAT\ApiCPP.

The PC-MCAT API for .Net DLL is installed into the PC-MCAT program directory, usually C:\Program Files\TrioMotion\PC-MCAT.

All the required dynamic libraries (DLLs) required for the C++ applications to run are installed in C:\Program Files\TrioMotion\PC-MCAT.

The source code of the examples presented below can be downloaded from the website.

2. PC-MCAT API Reference

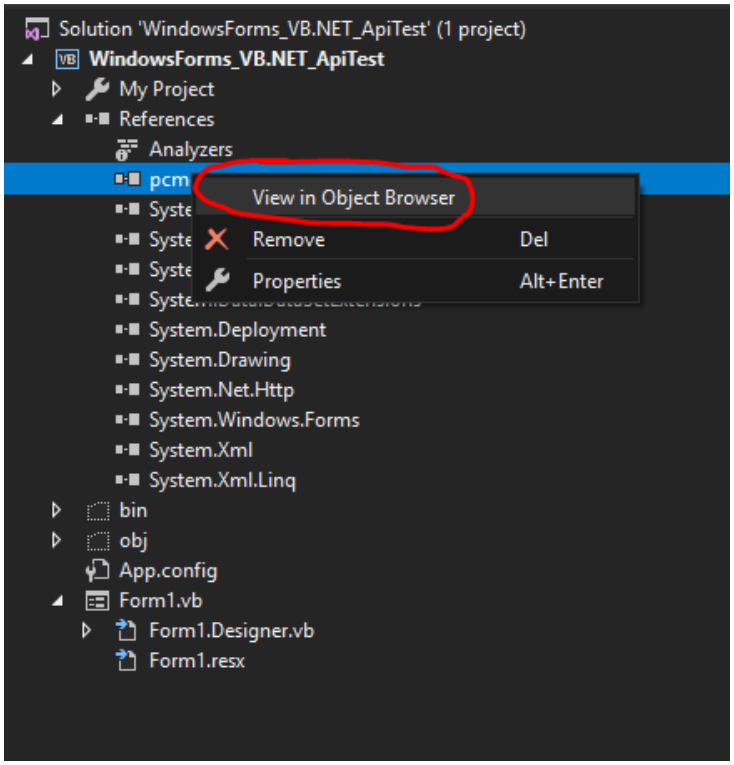
When using the C++ version of the PC-MCAT API, the headers files found in "C:\Program

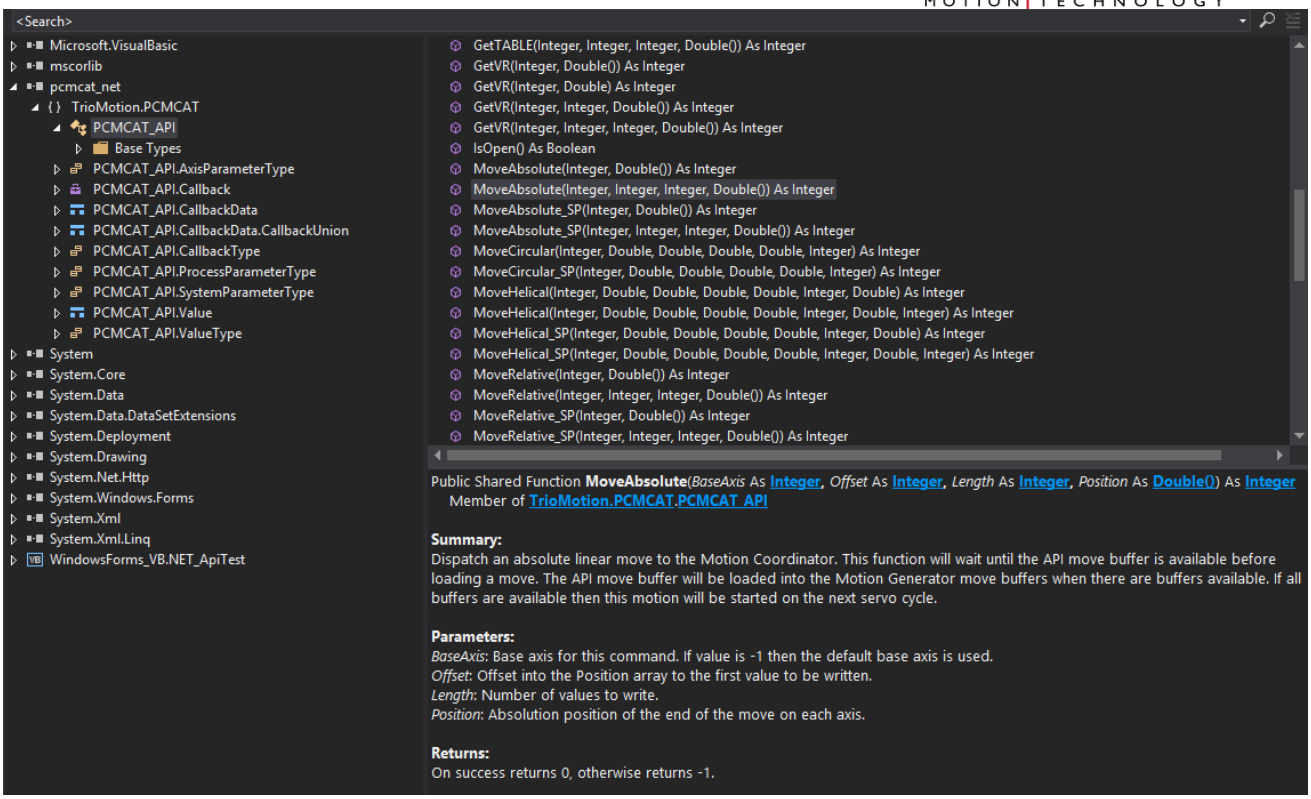
Files\TrioMotion\PC-MCAT\ApiCPP" can be used as a reference, a description of the functionality, data types and return types can be found. Below an example of the "pcmcat_api_moveabs" function, which can be found – as any of the other API functions – in "pcmcat_api.h":

```
/// <summary>
/// Dispatch an absolute linear move to the Motion Coordinator.
/// This function will wait until the API move buffer is available before loading a move.
/// The API move buffer will be loaded into the Motion Generator move buffers when there are buffers available.
/// If all buffers are available then this motion will be started on the next servo cycle.
/// </summary>
/// <param name="axis_count">Number of axes involved in the move.</param>
/// <param name="base_axis">Base axis for this command. If value is -1 then the default base axis is used.</param>
/// <param name="position">Absolution position of the end of the move on each axis.</param>
/// <returns>
/// On success returns 0, otherwise returns -1
/// </returns>
int PCMCATLIBRARY_API pcmcat_api_moveabs(int base_axis, int axis_count, double position[]);
```

The header files could also be used as a reference when using the .NET assembly, however, to avoid any confusion, the "pcmcat_net.xml" metadata file found in "C:\Program Files\TrioMotion\PC-MCAT" could be used instead, where the name of the functions, parameter passing methods and data types would match those of the .NET implementation of the API.

In addition to the "pcmcat_net.xml", the Visual Studio IntelliSense would provide all that information automatically as well. All this metadata can be consulted throughout the Object Browser in Visual Studio directly:



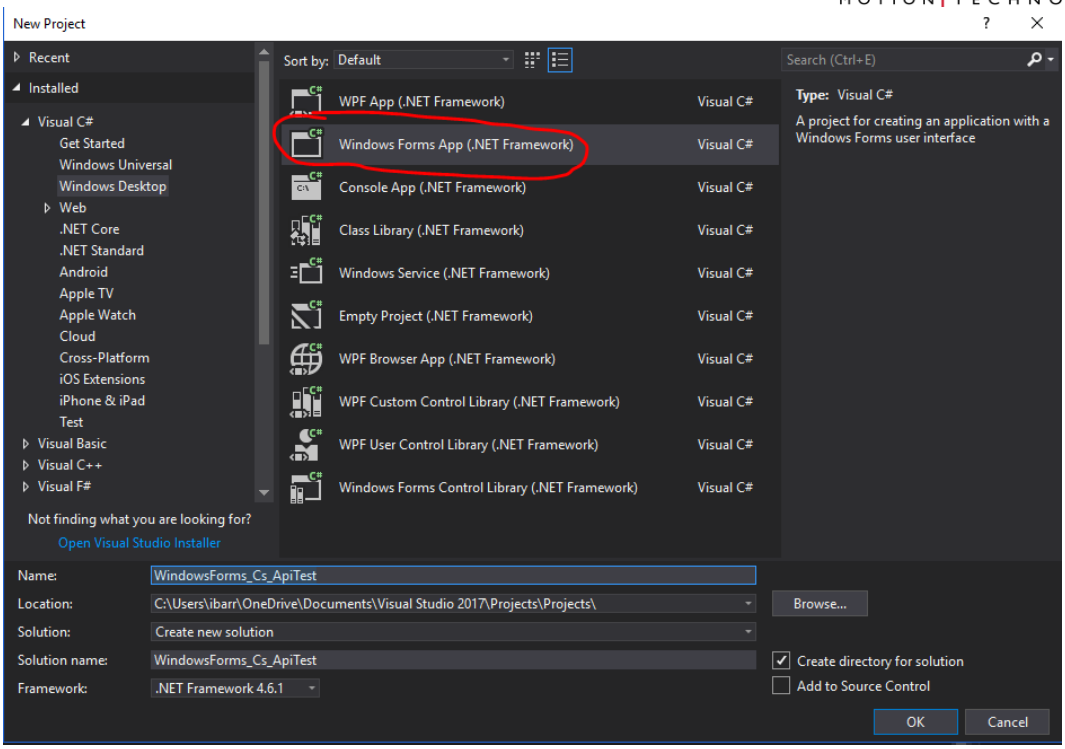


3. Requirements

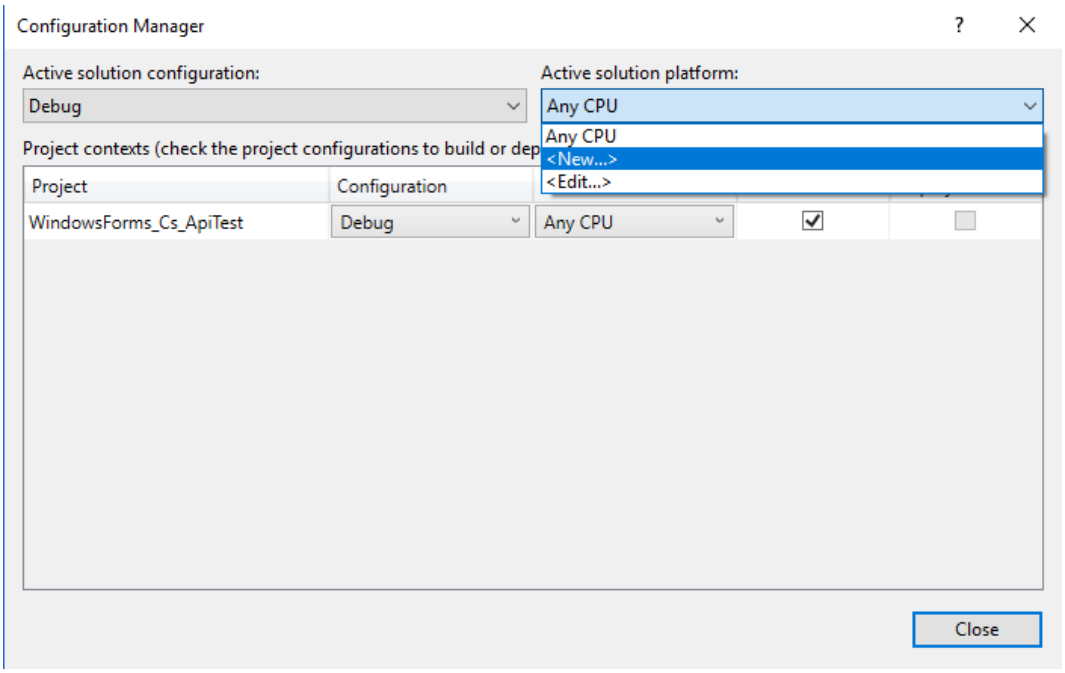
- PC-MCAT or PC-MCAT-2.
- Visual Studio 2015 or later versions.

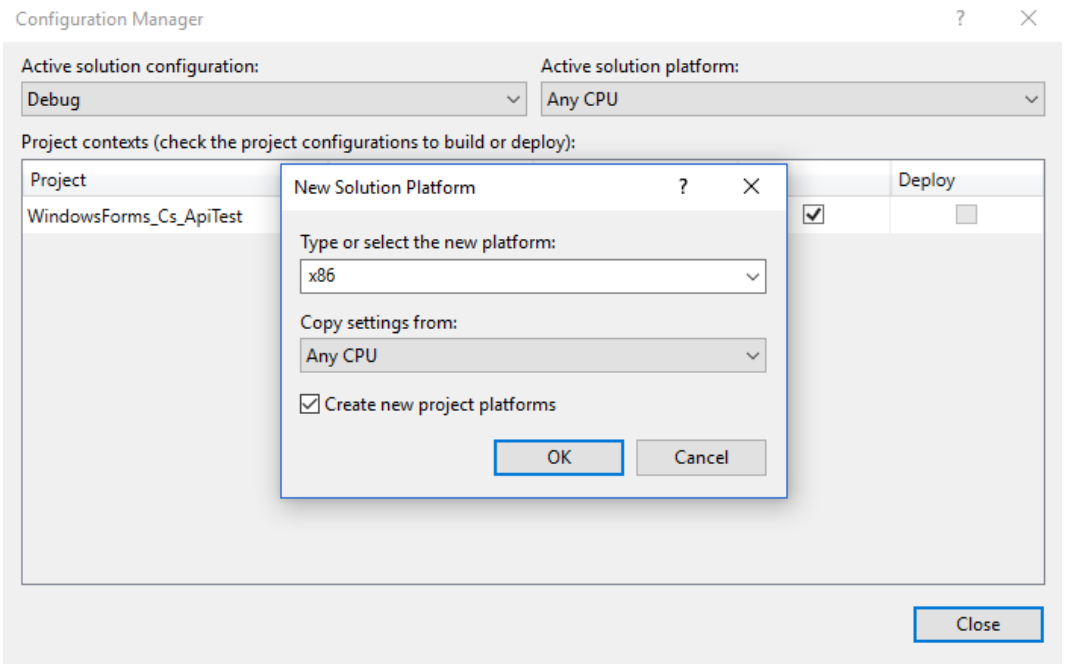
4. Create a 32 bit Visual C# PC-MCAT API application

- Open Visual Studio 2015 and create a new Windows Forms Application solution called WindowsForms_Cs_ApiTest.

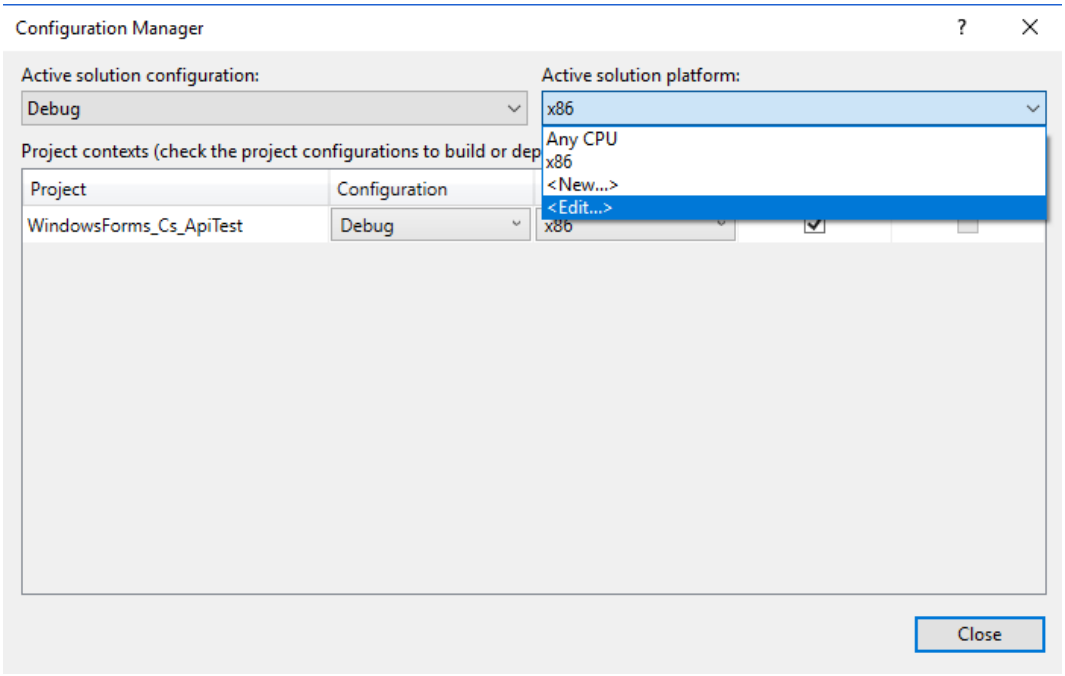


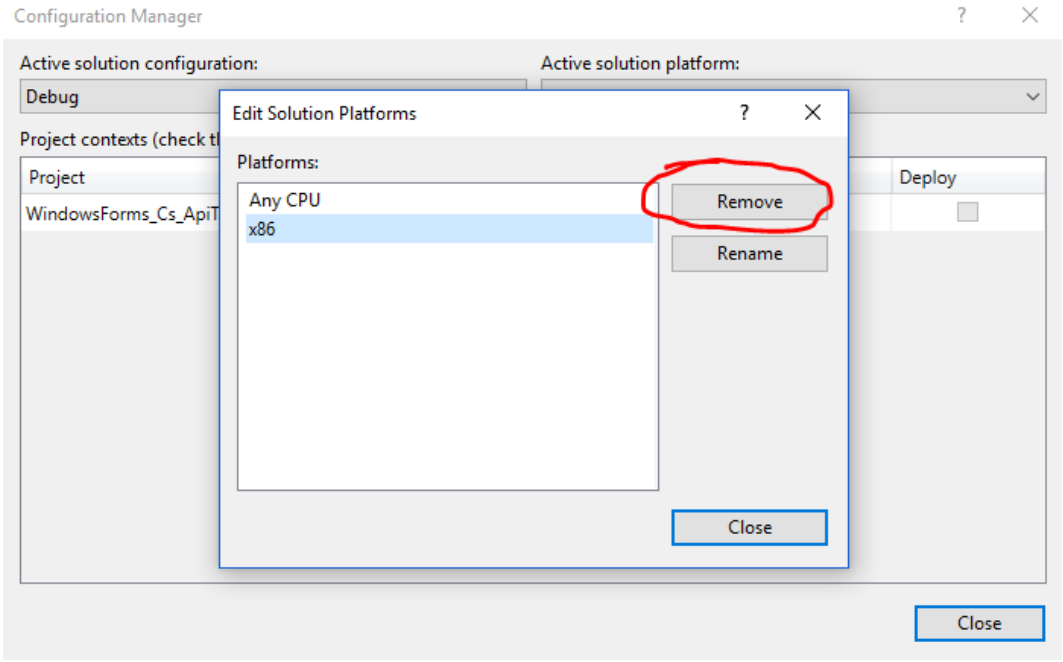
- Use the Build->Configuration Manager menu option to create a new configuration x86.



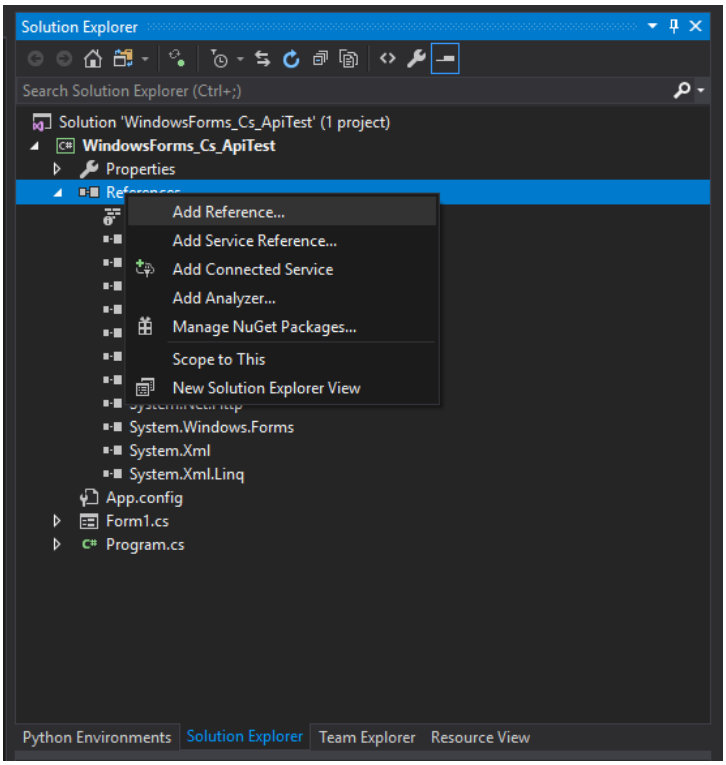


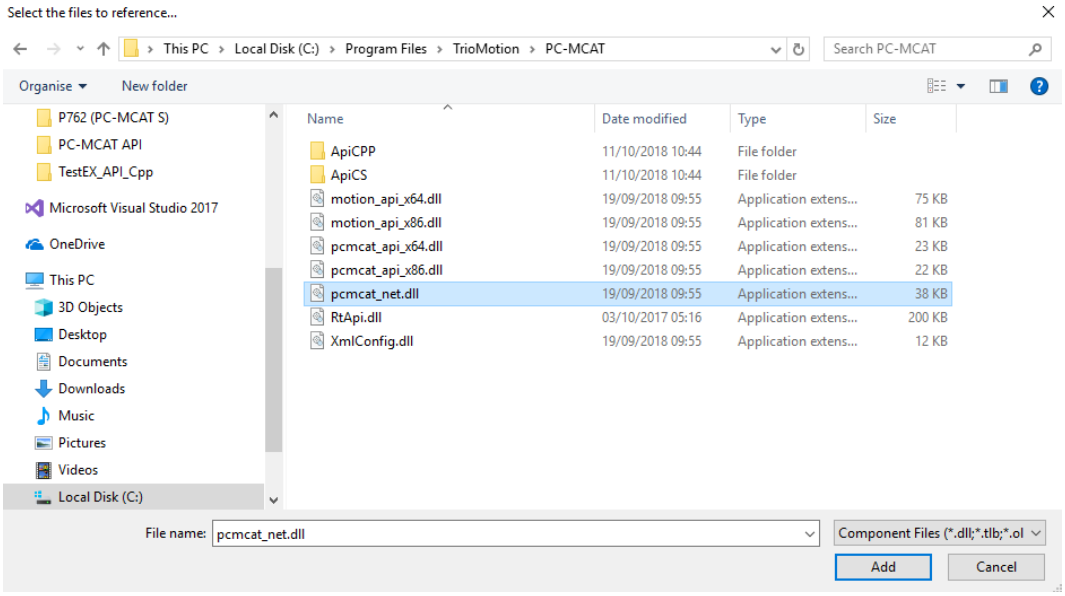
- Delete the AnyCPU configuration.



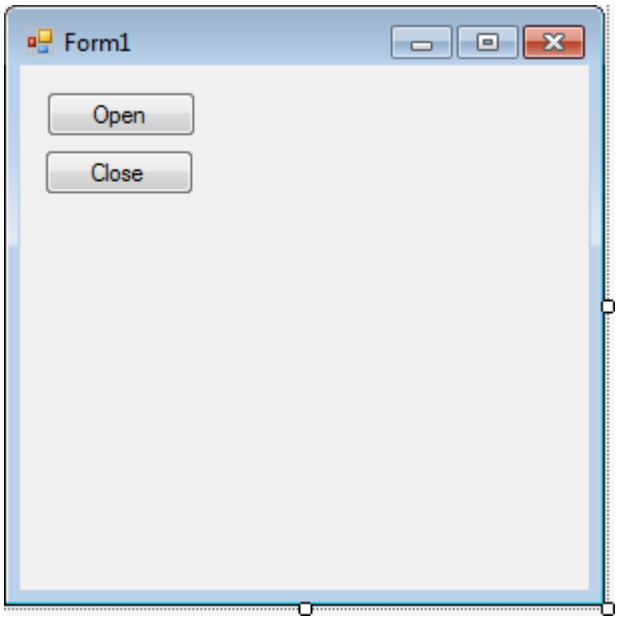


- Add a reference to the PC-MCAT API .Net assembly in the folder C:\Program Files\TrioMotion\PC-MCAT.





- Create two buttons: Open and Close.



- Double click on the Open button to add an OnClick handler. This will open the file Form1.cs.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace PC_MCAT_API_CSharp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
```

```
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

        }
    }
}
```

- Add the line “using TrioMotion.PCMCAT;” at the top of the file.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using TrioMotion.PCMCAT;

namespace PC_MCAT_API_CSharp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

        }
    }
}
```

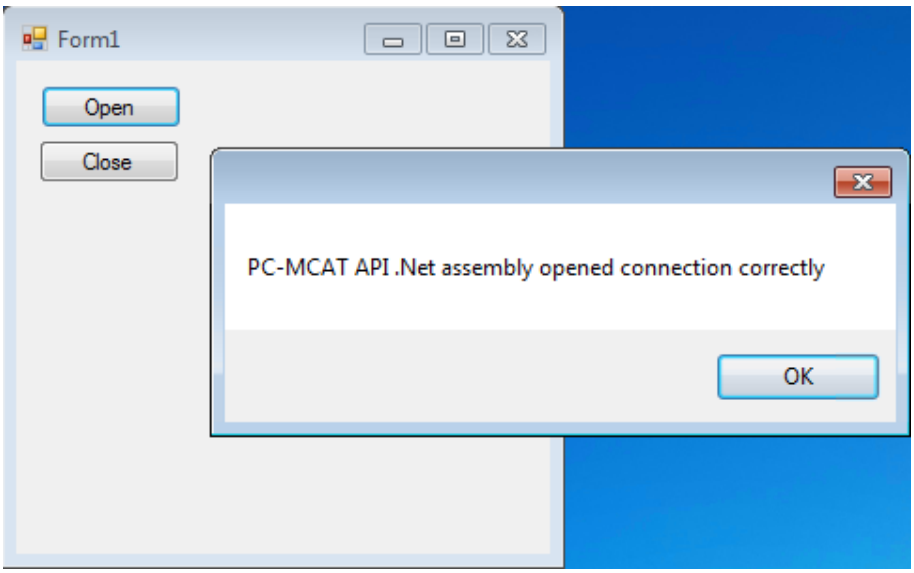
- Add the required code to the handler.

```
private void button1_Click(object sender, EventArgs e)
{
    if (PCMCAT_API.Open(null, IntPtr.Zero) == 0)
    {
        MessageBox.Show("PC-MCAT API .Net assembly opened connection correctly");
    }
    else
    {
        MessageBox.Show("Error opening the PC-MCAT API .Net assembly connection");
    }
}
```

- Repeat for the Close handler.

```
private void button2_Click(object sender, EventArgs e)
{
    PCMCAT_API.Close();
}
```

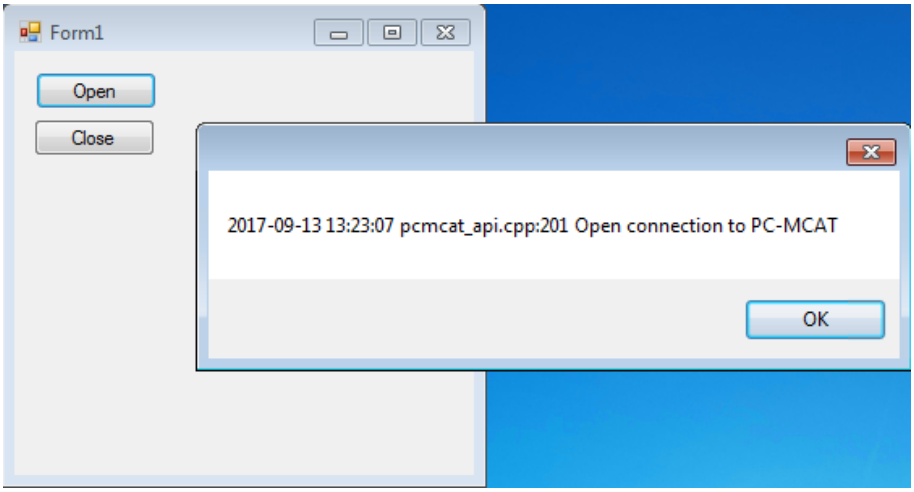
- Test.

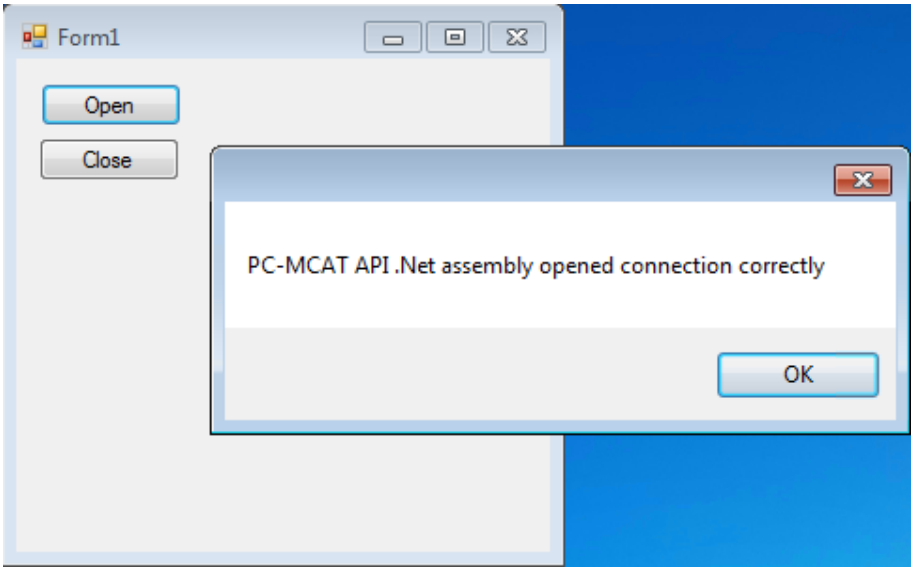


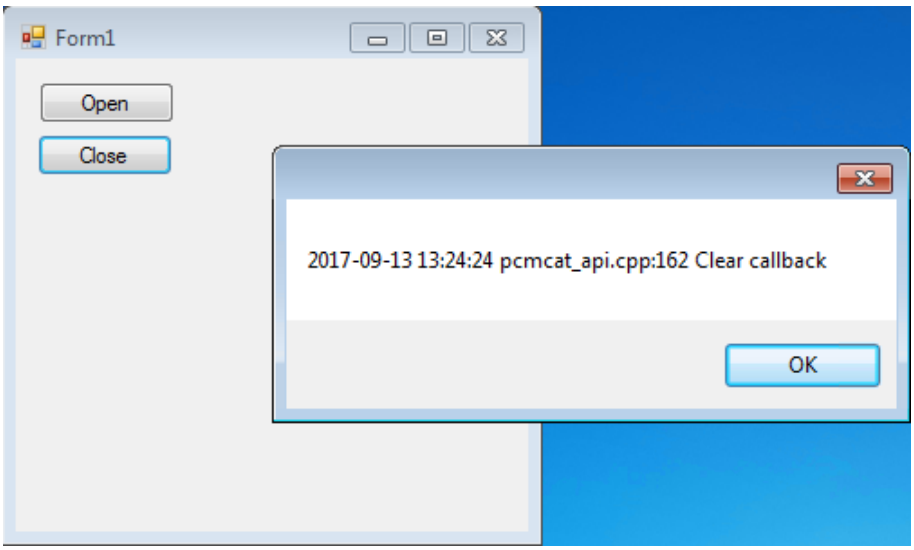
- Add Open callback handler.

```
PCMCAT_API.Callback CallbackDelegate;

void CallbackFunction(IntPtr Context, ref PCMCAT_API.CallbackData Data)
{
    switch (Data.Type)
    {
    case PCMCAT_API.CallbackType.Message:
        MessageBox.Show(Marshal.PtrToStringAnsi(Data.Data.Str));
        break;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    CallbackDelegate = CallbackFunction;
    if (PCMCAT_API.Open(CallbackDelegate, IntPtr.Zero) == 0)
    {
        MessageBox.Show("PC-MCAT API .Net assembly opened connection correctly");
    }
    else
    {
        MessageBox.Show("Error opening the PC-MCAT API .Net assembly connection");
    }
}
```

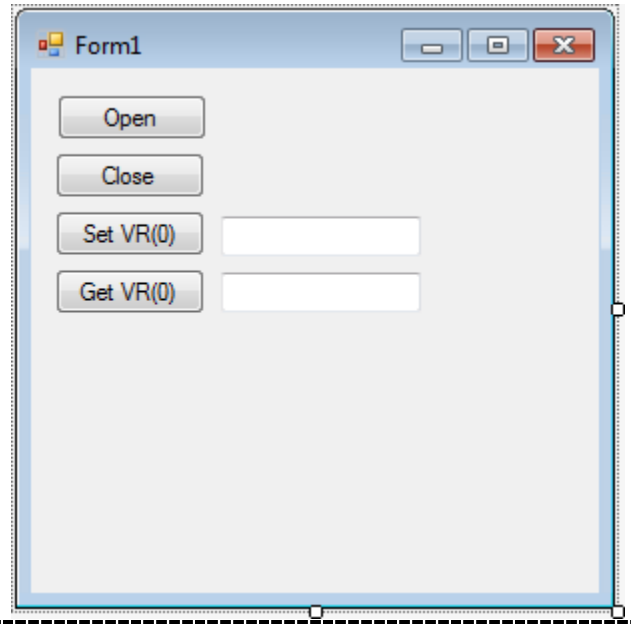
- Test.


Form1
Open
Close
2017-09-13 13:23:07 pcmcat_api.cpp:201 Open connection to PC-MCAT
OK


Form1
Open
Close
PC-MCAT API .Net assembly opened connection correctly
OK

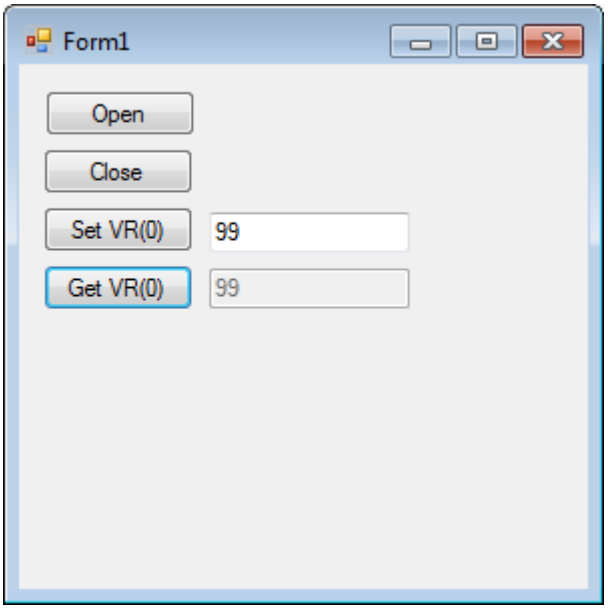

Form1
Open
Close
2017-09-13 13:24:24 pcmcat_api.cpp:162 Clear callback
OK

- Get/Set VR(0).



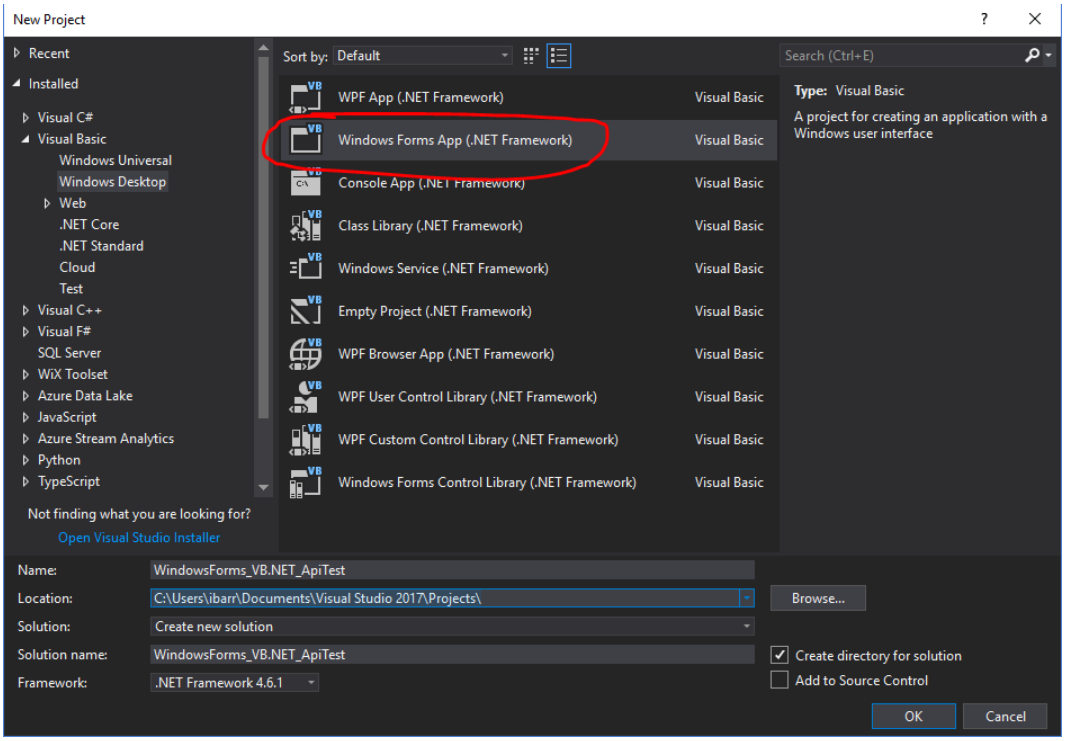

```
private void button3_Click(object sender, EventArgs e)
{
    if (PCMCAT_API.SetVR(0, Double.Parse(textBox1.Text)) != 0)
    {
        MessageBox.Show("Error setting VR(0) to " + textBox1.Text);
    }
}

private void button4_Click(object sender, EventArgs e)
{
    double vr0;
    if (PCMCAT_API.GetVR(0, out vr0) != 0)
    {
        MessageBox.Show("Error getting VR(0)");
    }
    else
    {
        textBox2.Text = vr0.ToString();
    }
}
```

- Test.

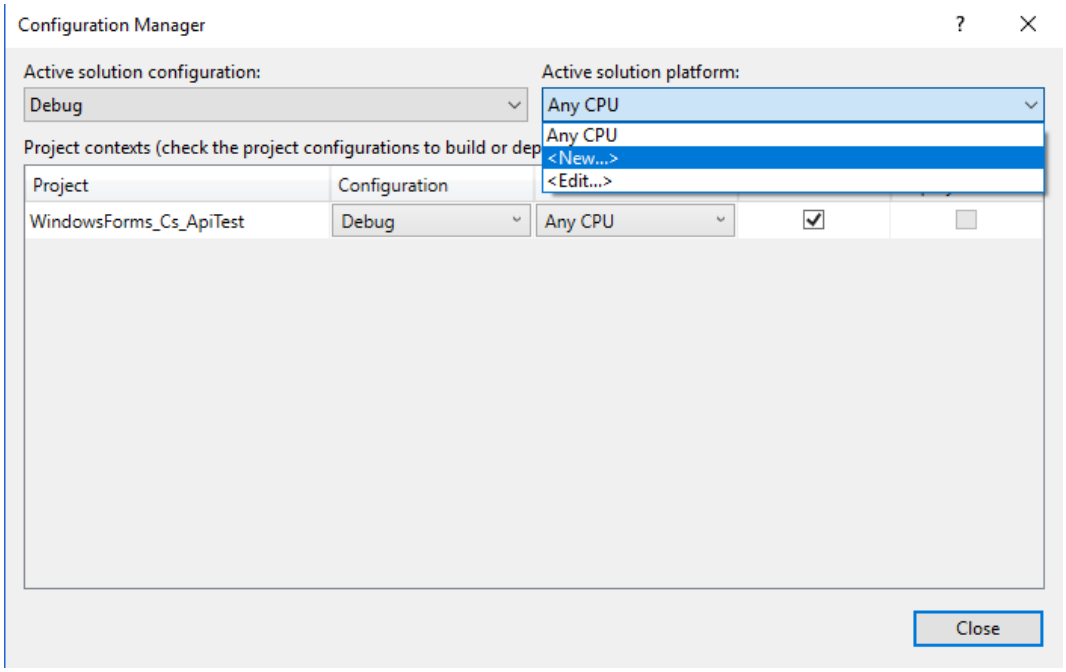


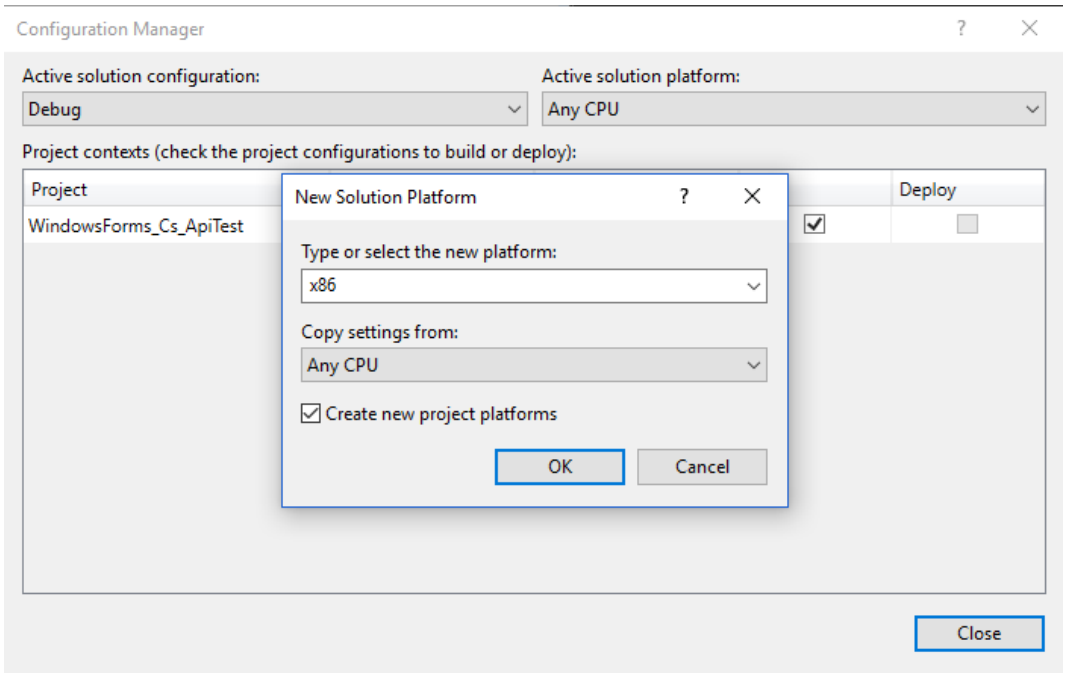
5. Create a 32 bit VB.NET PC-MCAT API application

- Open Visual Studio 2015 and create a new Windows Forms Application solution called WindowsForms_VB.NET_ApiTest.

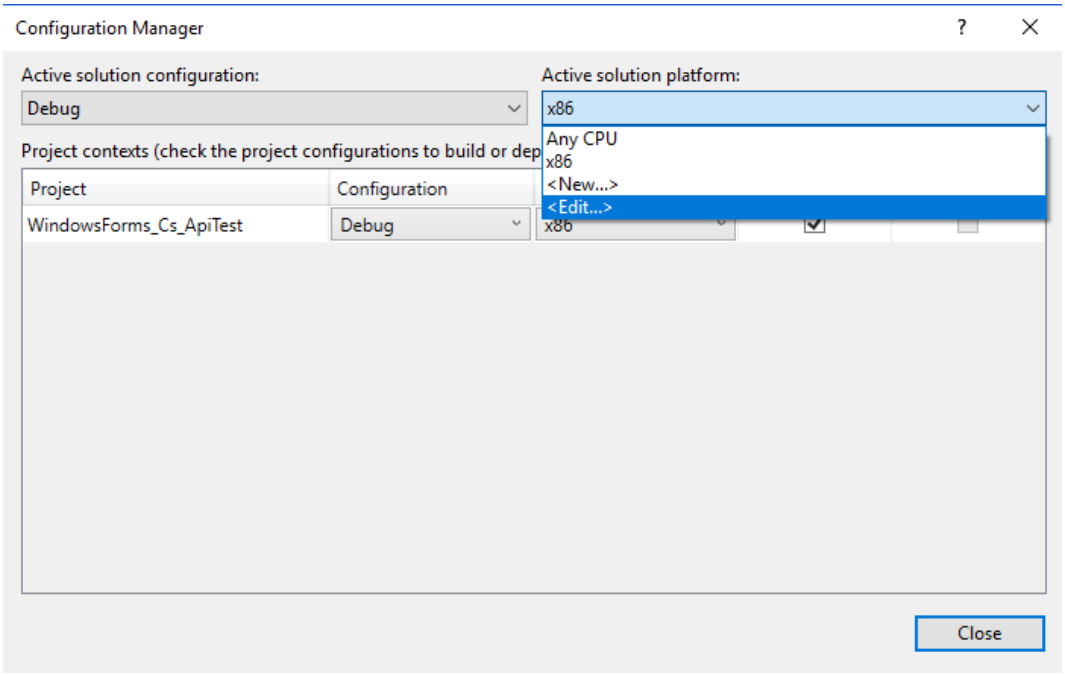


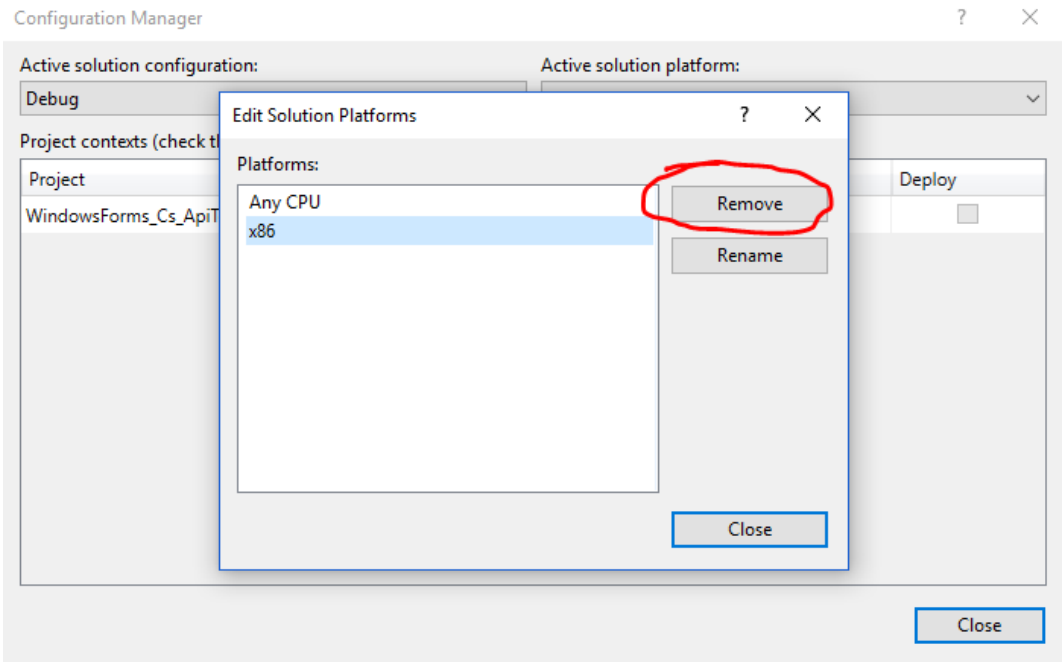
- Use the Build->Configuration Manager menu option to create a new configuration x86.



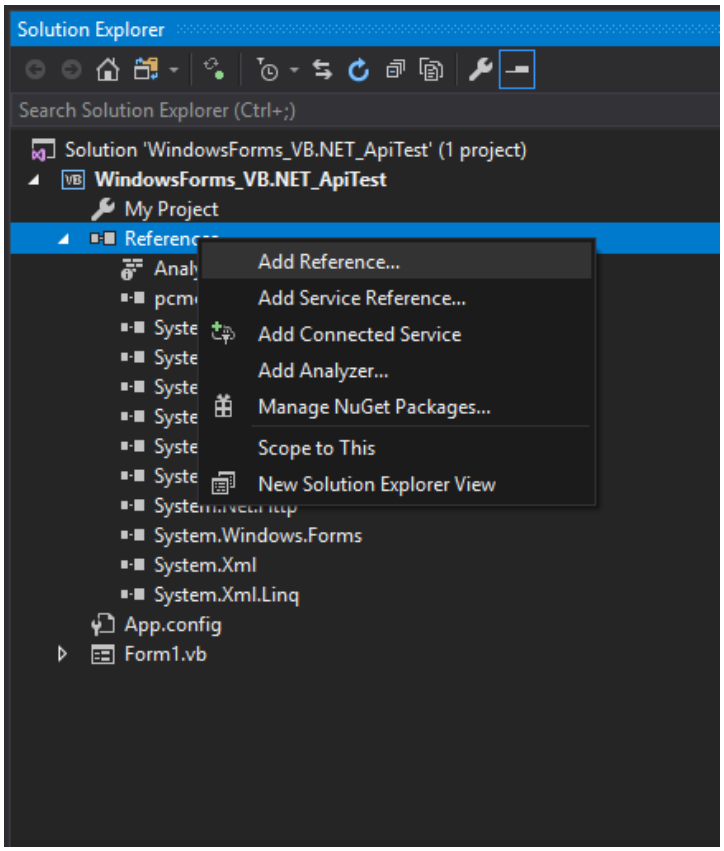


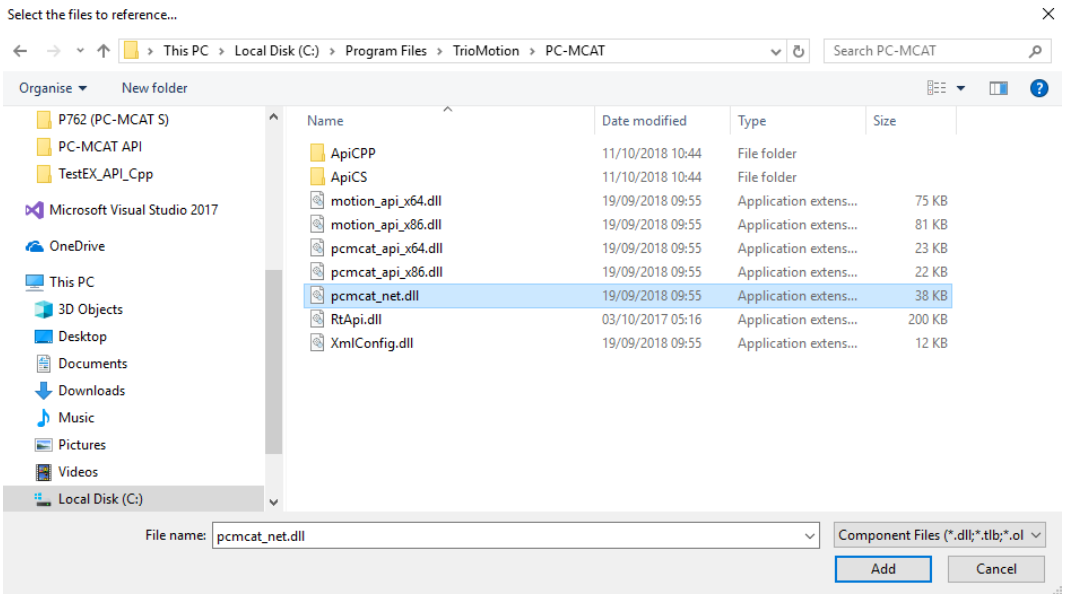
- Delete the AnyCPU configuration.



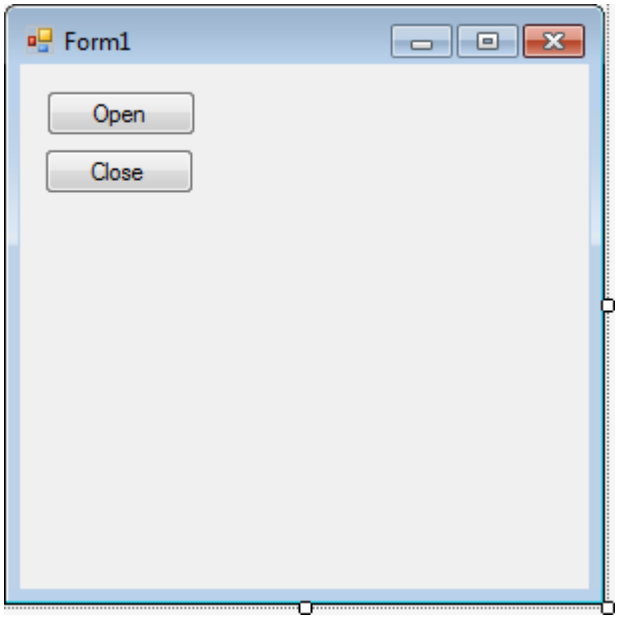


- Add a reference to the PC-MCAT API .Net assembly in the folder C:\Program Files\TrioMotion\PC-MCAT.





- Create two buttons: Open and Close.



- Double click on the Open button to add an OnClick handler. This will open the file Form1.vb.

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

    End Sub
End Class
```

- Import “TrioMotion.PCMCAT” and “IteropServices” adding this code at the top of the Form.vb file.

```
Imports System.Runtime.InteropServices
Imports TrioMotion.PCMCAT
```

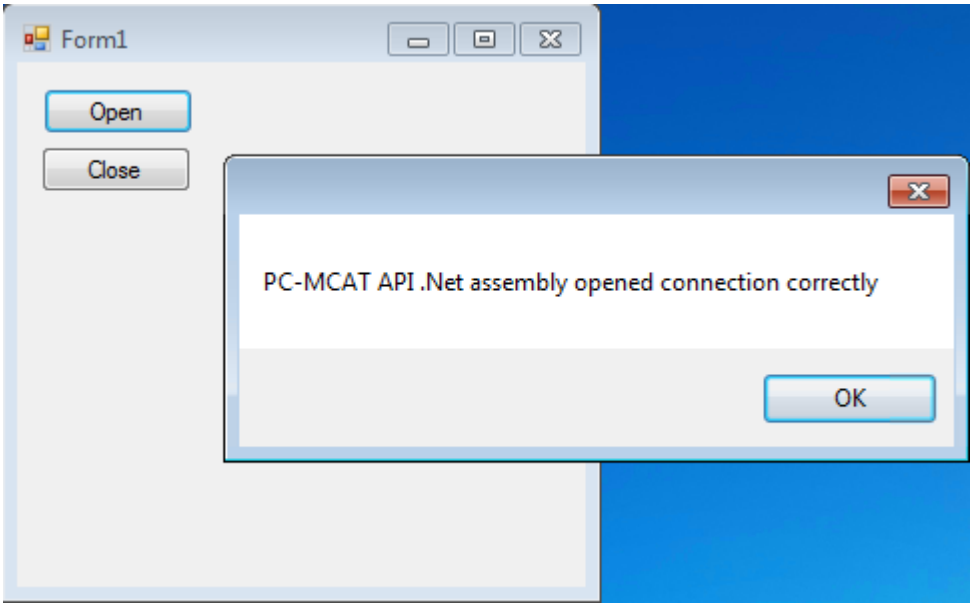
- Add the required code to the handler.

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If PCMCAT_API.Open(Nothing, IntPtr.Zero) = 0 Then
            MessageBox.Show("PC-MCAT API .Net assembly opened connection correctly")
        Else
            MessageBox.Show("Error opening the PC-MCAT API .Net assembly connection")
        End If
    End Sub
End Class
```

- Repeat for the Close button handler.

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
   PCMCAT_API.Close()
End Sub
```

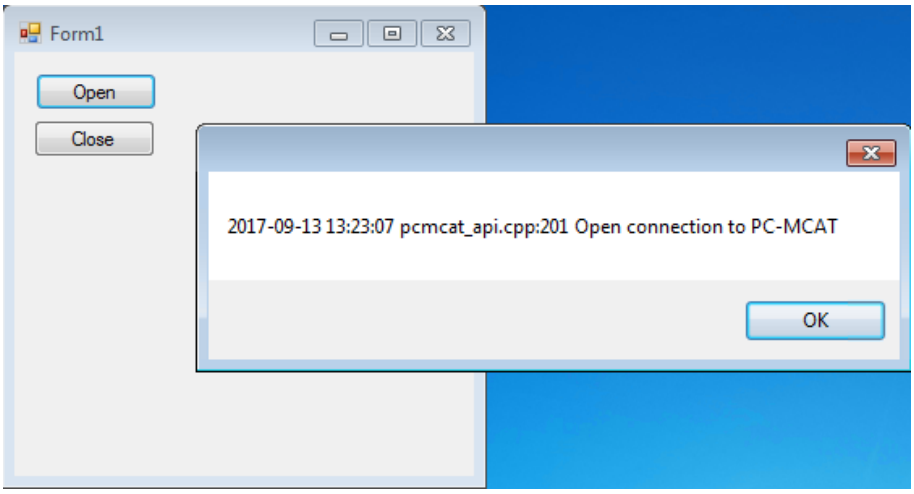
- Test.

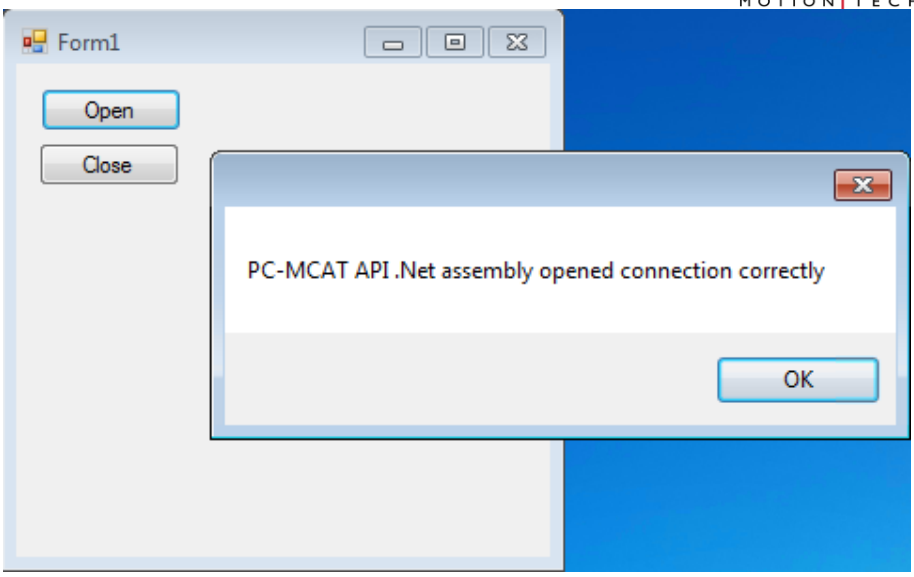


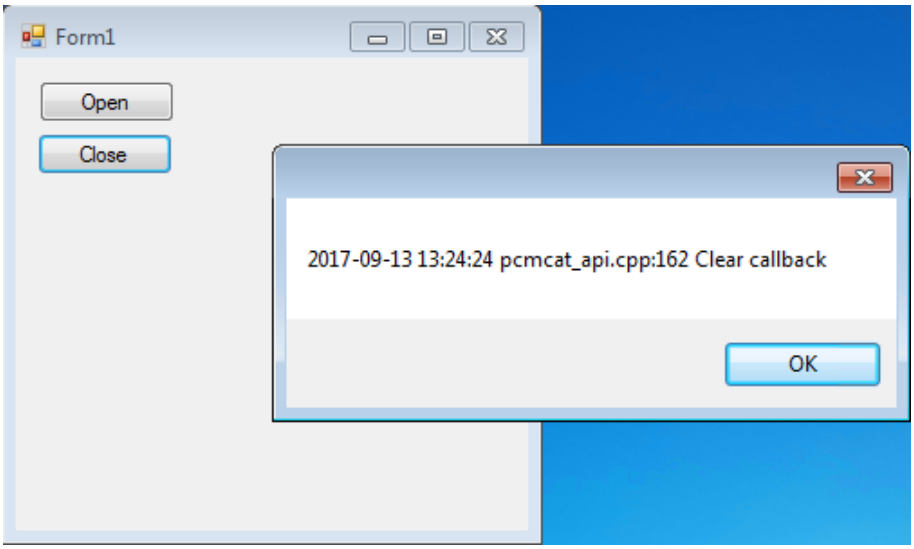
- Add Open Callback handler.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
   If PCMCAT_API.Open(AddressOf CallbackFunction, IntPtr.Zero) = 0 Then
      MessageBox.Show("PC-MCAT API .Net assembly opened connection correctly")
    Else
      MessageBox.Show("Error opening the PC-MCAT API .Net assembly connection")
    End If
End Sub

Private Sub CallbackFunction(Context As IntPtr, ByRef Data As PCMCAT_API.CallbackData)
   Select Case Data.Type
      Case PCMCAT_API.CallbackType.Message
         MessageBox.Show(Marshal.PtrToStringAnsi(Data.Data.Str))
      End Select
End Sub
```

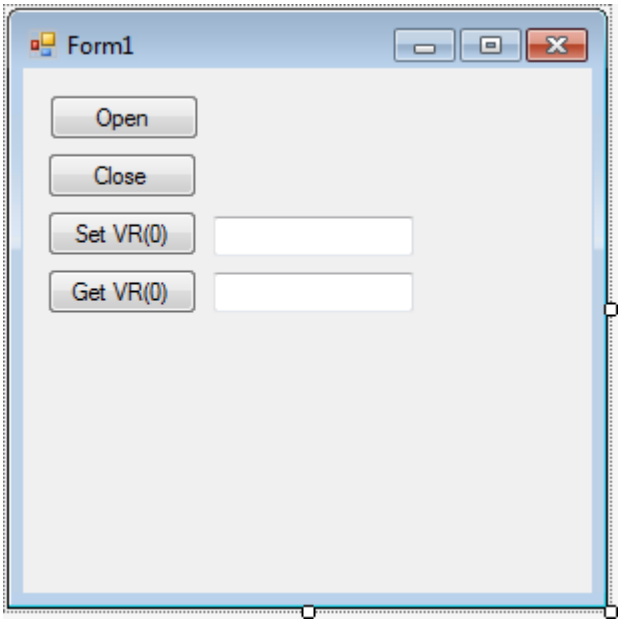
- Test.




Form1
Open
Close
PC-MCAT API .Net assembly opened connection correctly
OK

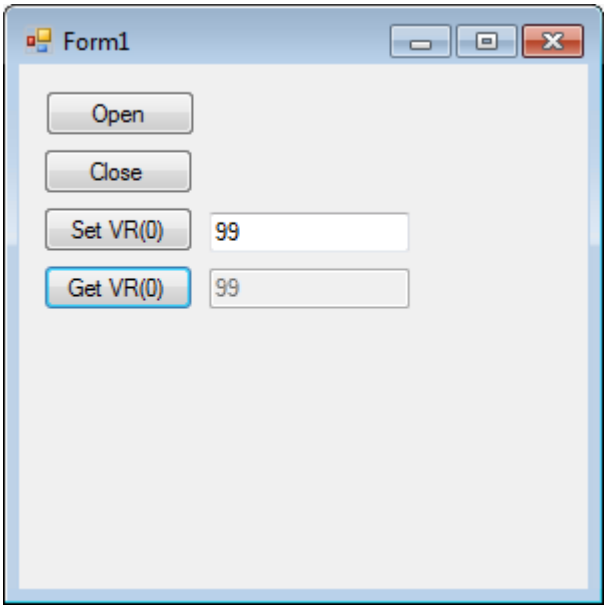

Form1
Open
Close
2017-09-13 13:24:24 pcmcat_api.cpp:162 Clear callback
OK

- Get/Set VR(0).


Form1
Open
Close
Set VR(0)
Get VR(0)

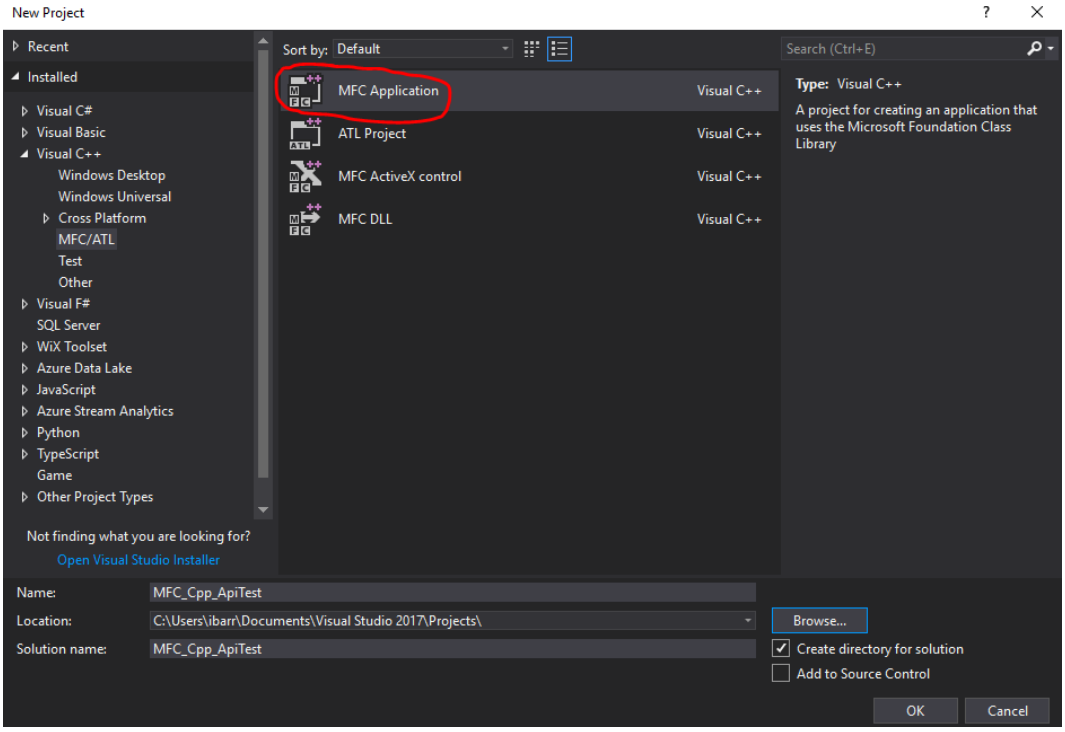
```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
   If PCMCAT_API.SetVR(0, Double.Parse(TextBox1.Text)) <> 0 Then
      MessageBox.Show("Error setting VR(0) to " + TextBox1.Text)
   End If
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
   Dim vr0 As Double
      If PCMCAT_API.GetVR(0, vr0) <> 0 Then
         MessageBox.Show("Error getting VR(0)")
      Else
         TextBox2.Text = vr0.ToString()
      End If
End Sub
```

- Test.

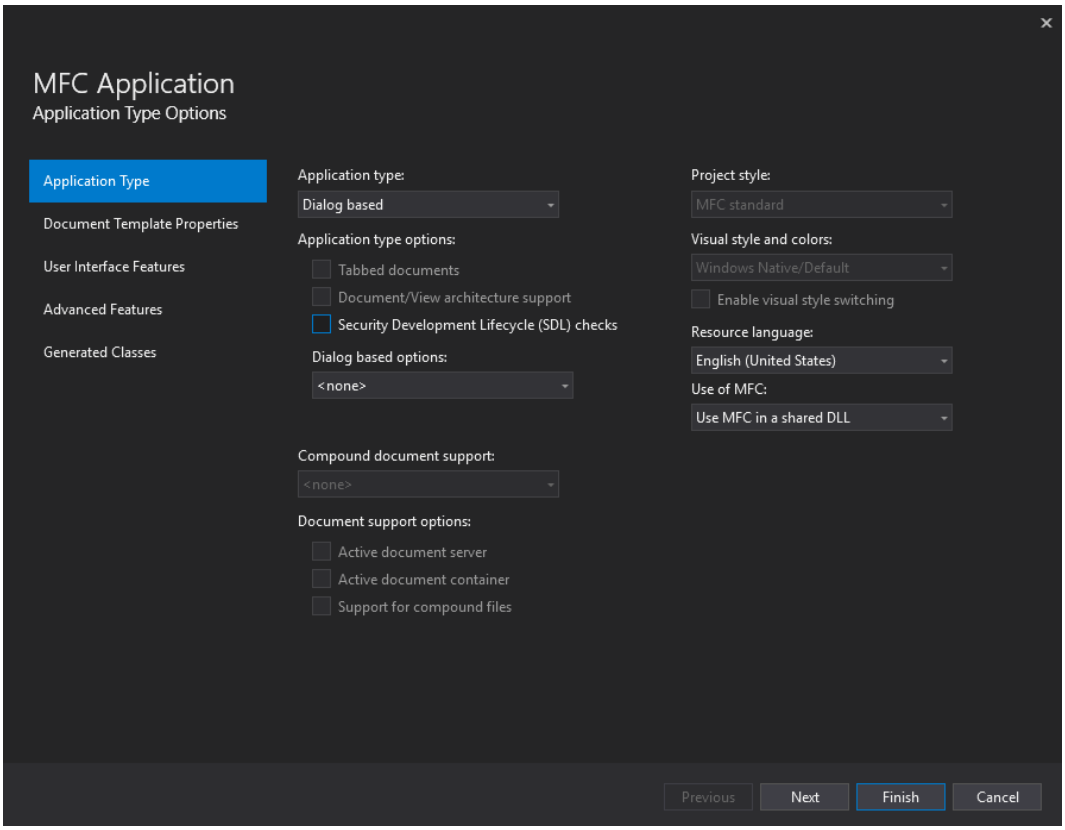


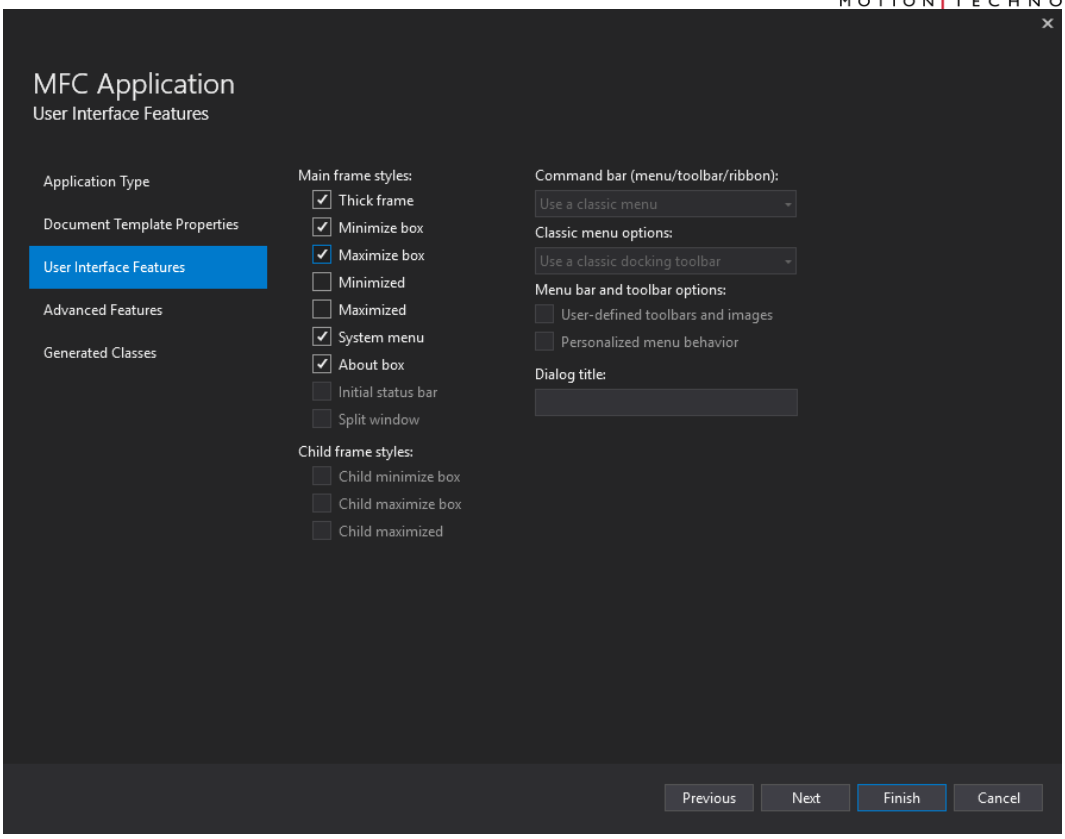
6. Create a 32/64 bit MFC C++ PC-MCAT API application

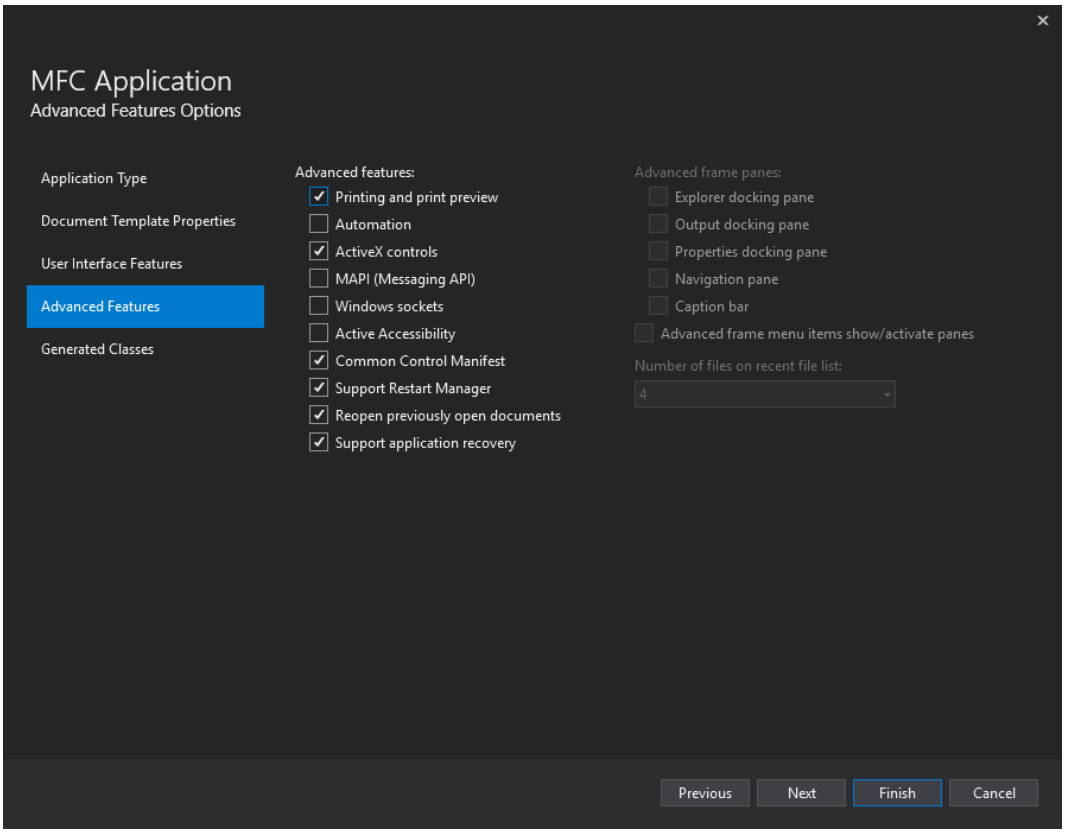
- Open Visual Studio 2015 and create a new MFC Application solution called MFC_Cpp_ApiTest.

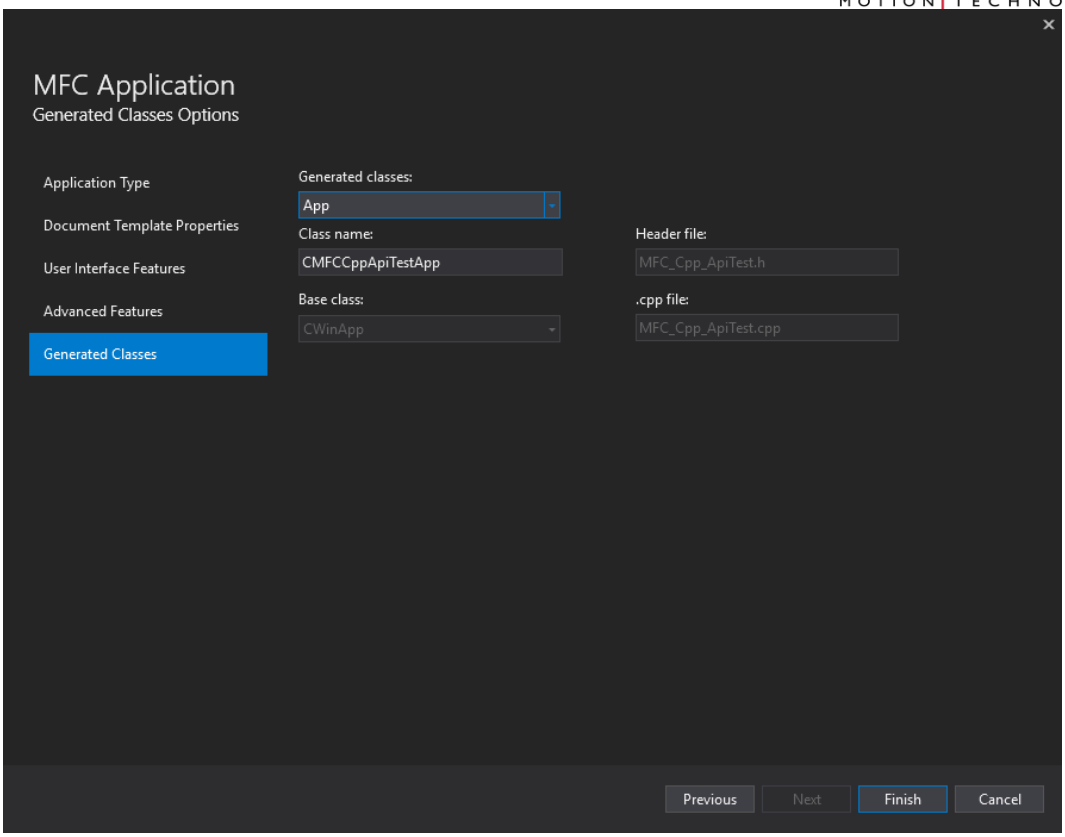


- Create the MFC application with the following settings.

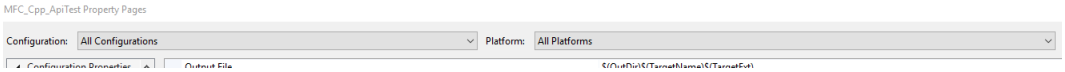



MFC Application
User Interface Features
Application Type
Document Template Properties
User Interface Features
Advanced Features
Generated Classes
Main frame styles:
Thick frame
Minimize box
Maximize box
Minimized
Maximized
System menu
About box
Initial status bar
Split window
Child frame styles:
Child minimize box
Child maximize box
Child maximized
Command bar (menu/toolbar/ribbon):
Use a classic menu
Classic menu options:
Use a classic docking toolbar
Menu bar and toolbar options:
User-defined toolbars and images
Personalized menu behavior
Dialog title:
Previous
Next
Finish
Cancel

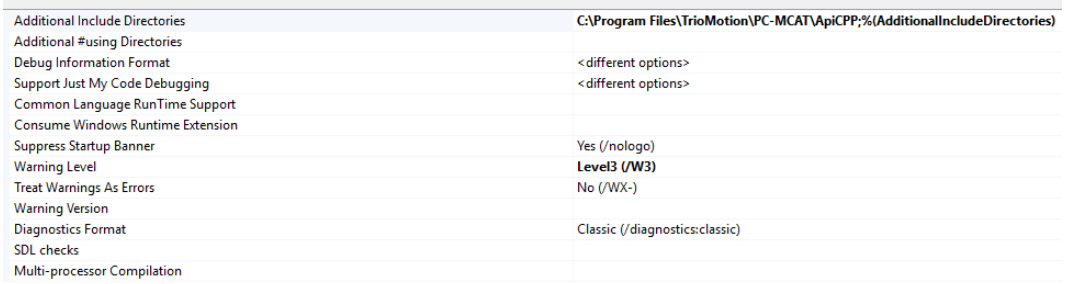

MFC Application
Advanced Features Options
Application Type
Document Template Properties
User Interface Features
Advanced Features
Generated Classes
Advanced features:
Printing and print preview
Automation
ActiveX controls
MAPI (Messaging API)
Windows sockets
Active Accessibility
Common Control Manifest
Support Restart Manager
Reopen previously open documents
Support application recovery
Advanced frame panes:
Explorer docking pane
Output docking pane
Properties docking pane
Navigation pane
Caption bar
Advanced frame menu items show/activate panes
Number of files on recent file list:
4
Previous
Next
Finish
Cancel



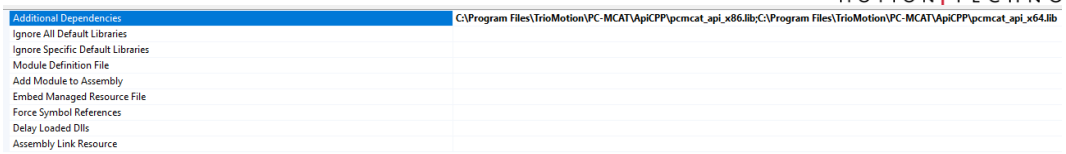
- When adding the C++ headers and libraries to the project as explained below, make sure the project properties are modified for All Configurations and All Platforms.



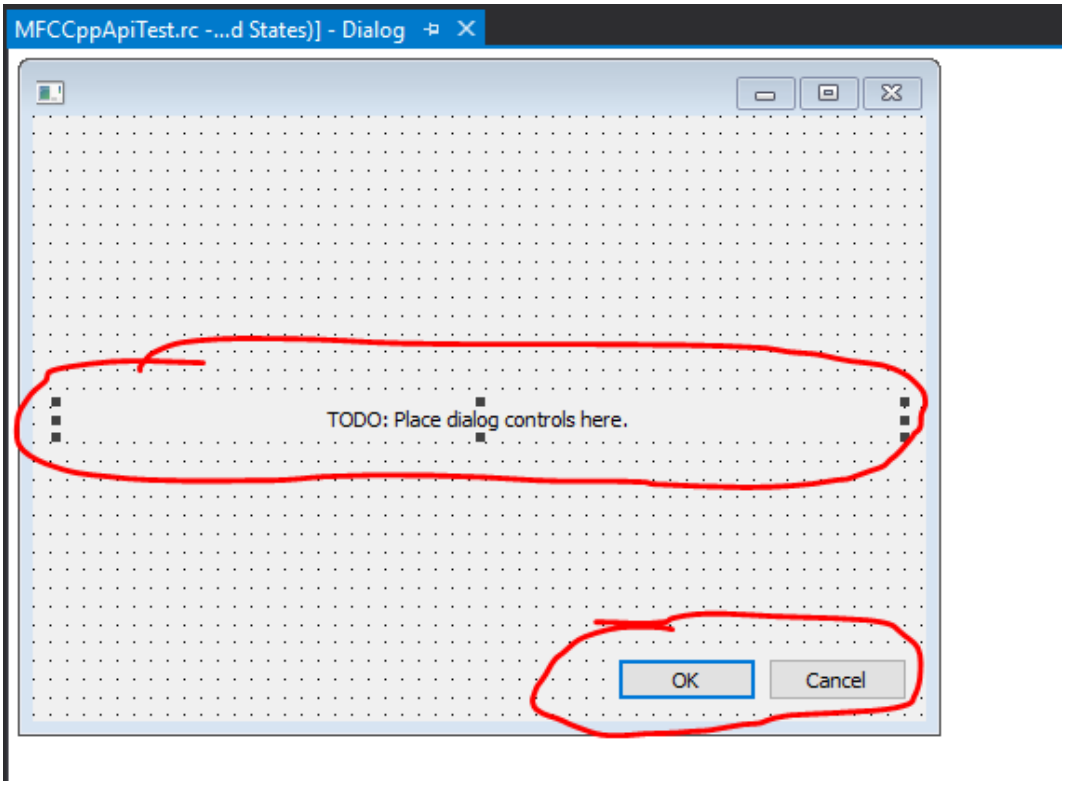
- Once the project is generated, right click on the project in the Solution Explorer, then go to Configuration Properties->C/C++->General->Additional Include Directories and add the PC-MCAT\ApiCPP directory:
  - C:\Program files\TrioMotion\PC-MCAT\ApiCPP



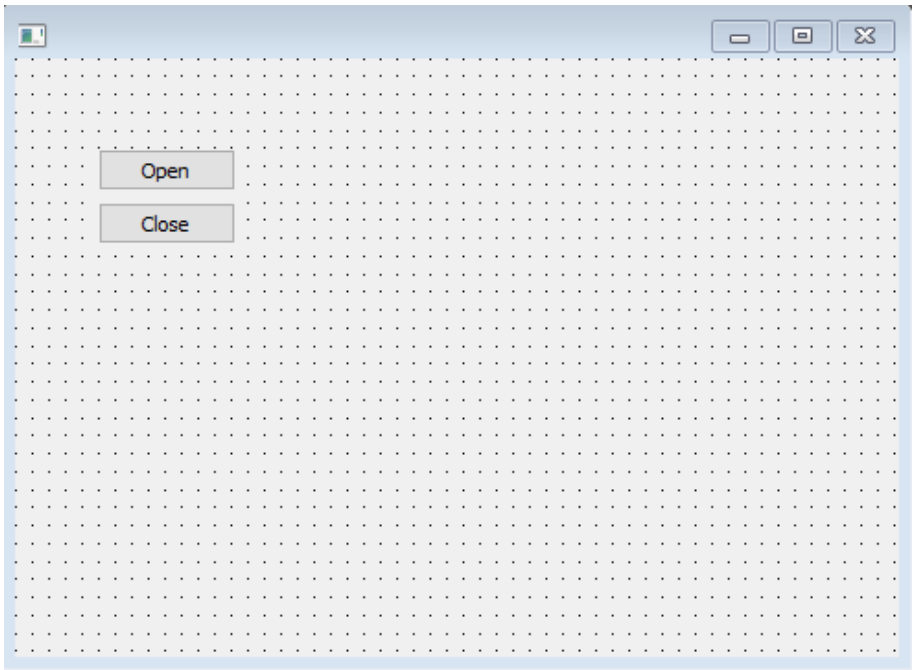
- Within the same project properties dialogue, go to Configuration Properties->Linker->Input and add the path to the .lib files found in that directory:
  - C:\Program Files\TrioMotion\PC-MCAT\ApiCPP\pcmcat_api_x86.lib
  - C:\Program Files\TrioMotion\PC-MCAT\ApiCpp\pcmcat_api_x64.lib



- Open the Resource View to manipulate the dialogue and remove the default static text and buttons controls.



- Create two buttons: Open and Close.



- Double click on the Open button to add a BN_CLICKED handler. This will open the file MFC_Cpp_ApiTestDlg.cpp.

```
void CMFCCppApiTestDlg::OnBnClickedButton1()
{
   // TODO: Add your control notification handler code here
}
```

- Add the “pcmcat_api.h” header file to the “MFC_Cpp_ApiTestDlg.h” file using the #include directive.

```
#include "pcmcat_api.h"
```

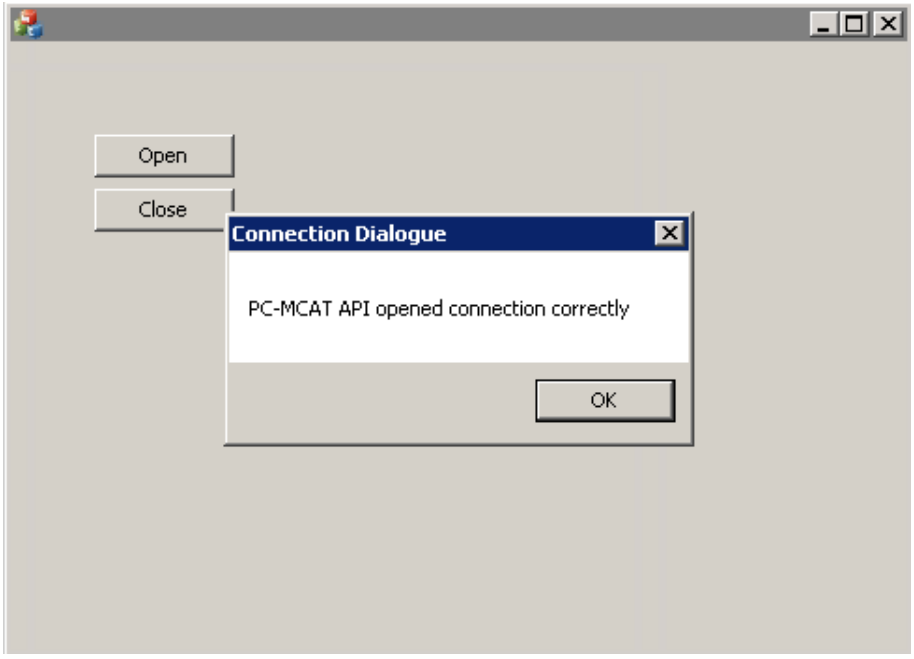
- Add the required code to the handler.

```
void CMFCCppApiTestDlg::OnBnClickedButton1()
{
   // TODO: Add your control notification handler code here
   if (pcmcat_api_open(NULL, NULL) == 0)
      MessageBoxA(this->GetSafeHwnd(), "PC-MCAT API opened connection correctly",
"Connection Dialogue", MB_OK);
   else
      MessageBoxA(this->GetSafeHwnd(), "Error opening the PC-MCAT API connection", "Error",
MB_OK | MB_ICONERROR);
}
```

- Repeat for the Close handler.

```
void CMFCCppApiTestDlg::OnBnClickedButton2()
{
   // TODO: Add your control notification handler code here
   pcmcat_api_close();
}
```

- Test running the application (F5).



- Add Open callback handler.
  - Function declaration in “MFC_Cpp_ApiTestDlg.h”.

```
static void __stdcall static_pcmcat_api_callback(void *context, pcmcat_api_callback_data_t
*pcmcat_api_callback_data);
```

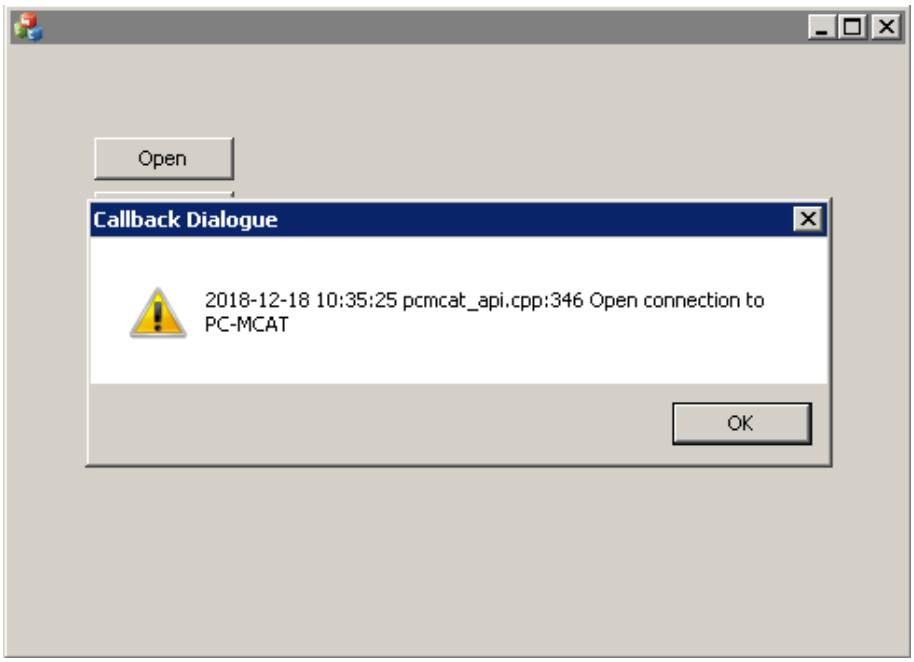
  - Function definition in “MFC_Cpp_ApiTestDlg.cpp”.

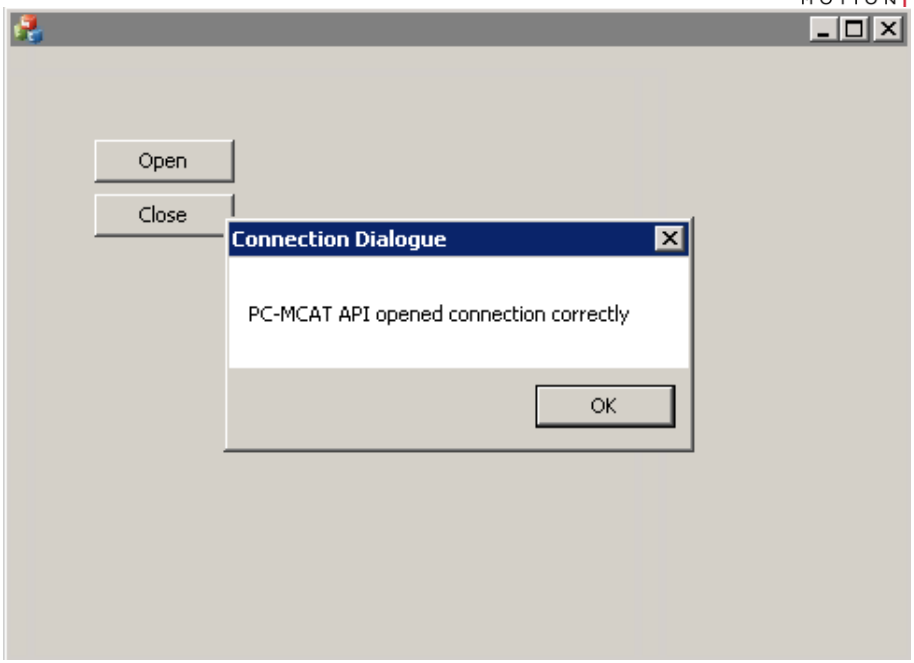
```
void __stdcall CMFCCppApiTestDlg::static_pcmcat_api_callback(void * context,
pcmcat_api_callback_data_t * pcmcat_api_callback_data)
{
   if (!pcmcat_api_callback_data)
      return;

   switch (pcmcat_api_callback_data->type)
   {
   case pcmcat_api_callback_type_message:
      if (pcmcat_api_callback_data->data.str)
         MessageBoxA(((CMFCCppApiTestDlg *)context)->GetSafeHwnd(),
pcmcat_api_callback_data->data.str, "Callback Dialogue", MB_OK | MB_ICONWARNING);
   }
}
```

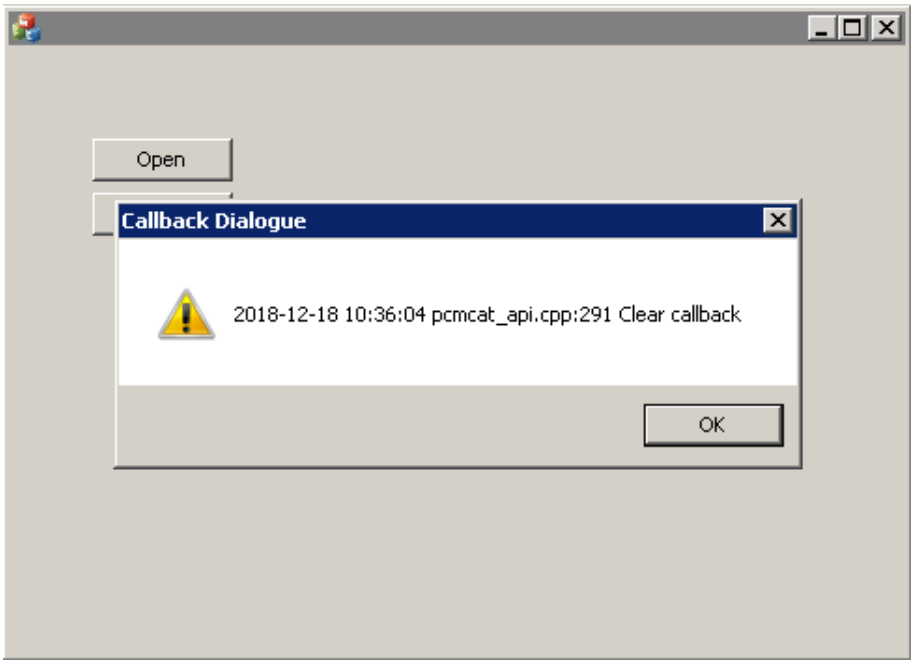
- Modify Open button handler.

```
void CMFCCppApiTestDlg::OnBnClickedButton1()
{
   // TODO: Add your control notification handler code here
   if (pcmcat_api_open(static_pcmcat_api_callback, this) == 0)
      MessageBoxA(this->GetSafeHwnd(), "PC-MCAT API opened connection correctly",
"Connection Dialogue", MB_OK);
   else
      MessageBoxA(this->GetSafeHwnd(), "Error opening the PC-MCAT API connection", "Error",
MB_OK | MB_ICONERROR);
}
```

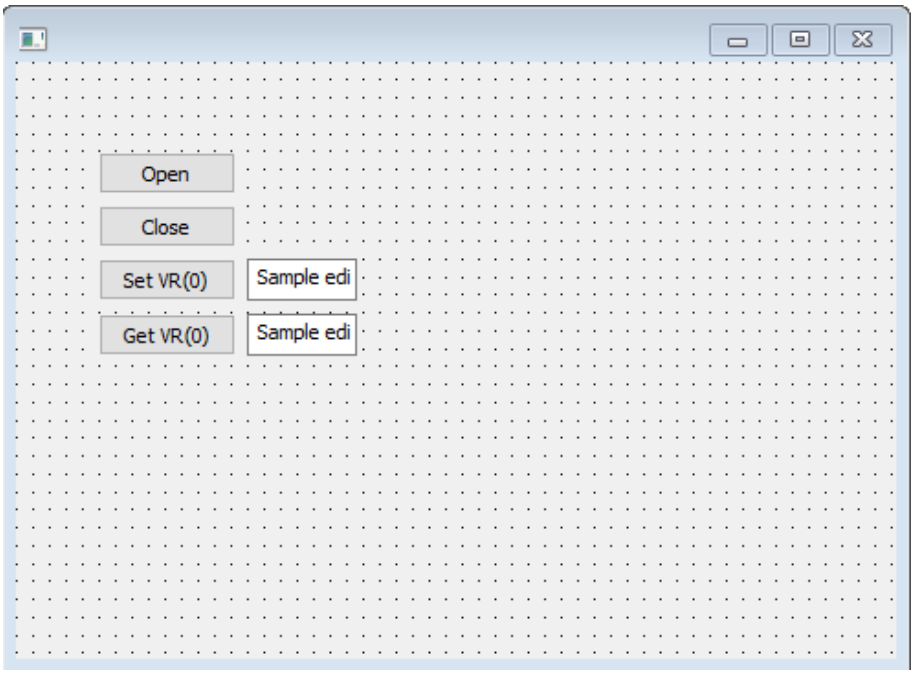
- Test running the application (F5).




Open
Close
Connection Dialogue
PC-MCAT API opened connection correctly
OK


Open
Callback Dialogue
2018-12-18 10:36:04 pcmcat_api.cpp:291 Clear callback
OK

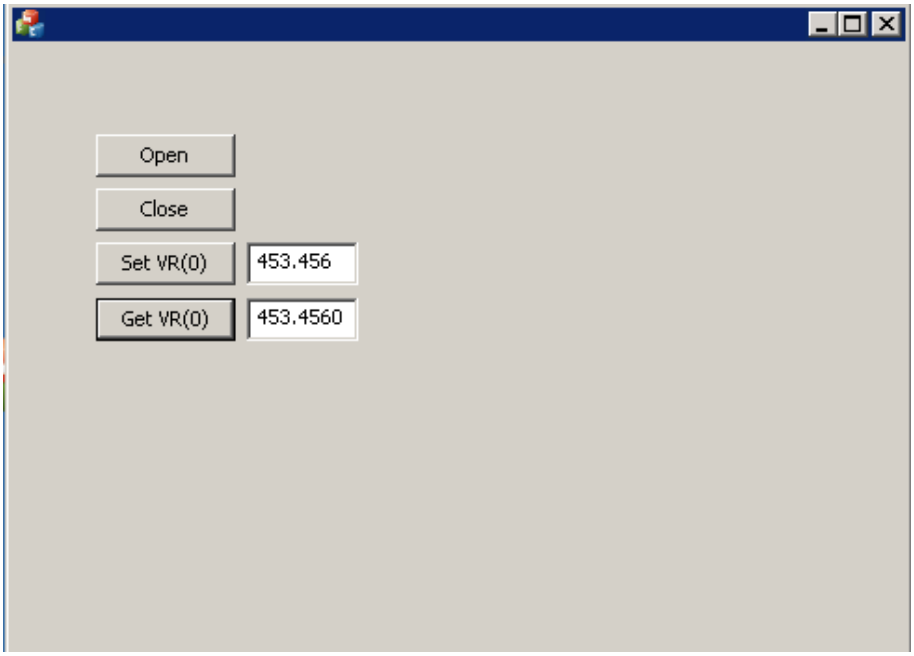
- Get/Set VR(0). Add two more buttons as below, one of them to set the VR(0) value and the other one to read it. Two Edit Controls will be added to show these values.



- Add the buttons event handlers that would Set and Get VR(0) respectively.

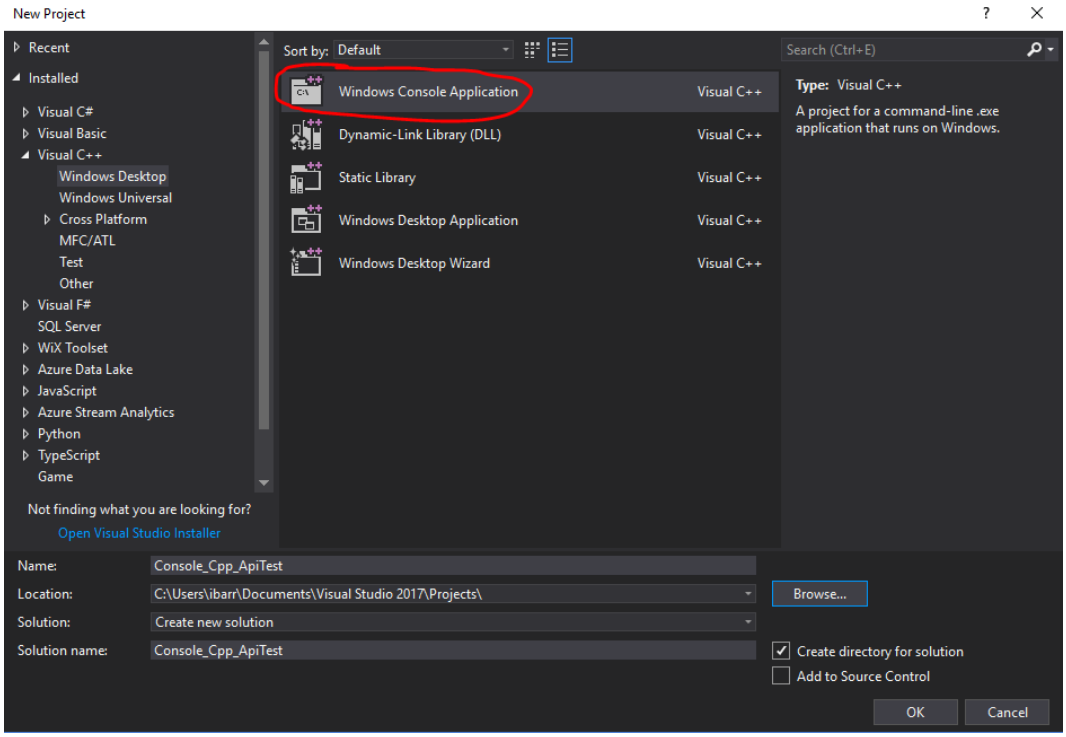
```
void CMFCCppApiTestDlg::OnBnClickedButton3()
{
   // TODO: Add your control notification handler code here
   CString str_value;
   double value;

   ((CEdit *)GetDlgItem(IDC_EDIT1))->GetWindowTextW(str_value);
   value = _wtof(str_value);

   if (pcmcat_api_set_vr(0, 1, &value) != 0)
      MessageBoxW(L"Error setting VR(0) to " + str_value, L"Error", MB_OK | MB_ICONERROR);
}


void CMFCCppApiTestDlg::OnBnClickedButton4()
{
   // TODO: Add your control notification handler code here
   CString str_value;
   double value;

   if (pcmcat_api_get_vr(0, 1, &value) != 0)
      MessageBoxW(L"Error Getting value of VR(0)", L"Error", MB_OK | MB_ICONERROR);
   else
   {
      str_value.Format(L"%.4f", value);
      ((CEdit *)GetDlgItem(IDC_EDIT2))->SetWindowTextW(str_value);
   }
}
```

- Test running the application (F5).

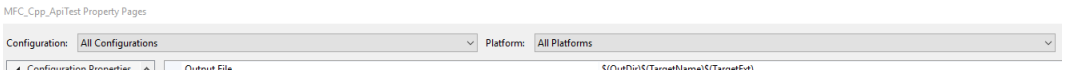


7. Create a plain 32/64 bit C++ PC-MCAT API application

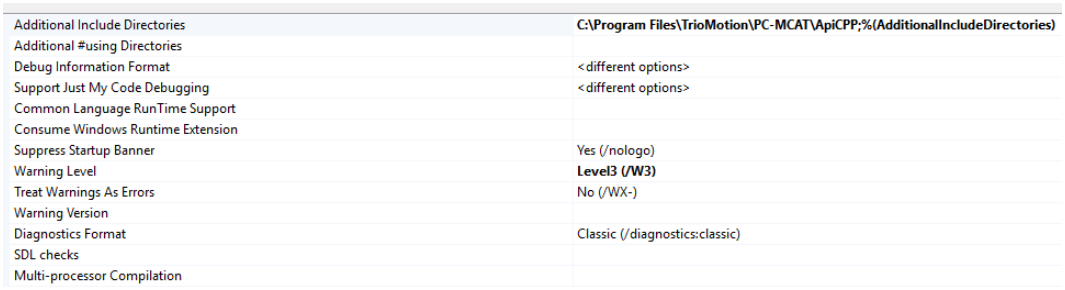
- Open Visual Studio 2015 and create a new Windows Console Application solution called Console_Cpp_ApiTest.



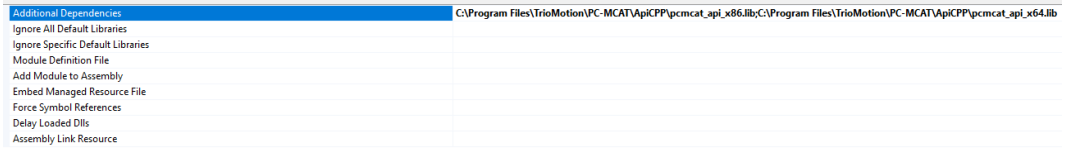
- When adding the C++ headers and libraries to the project as explained below, make sure the project properties are modified for All Configurations and All Platforms.



- Once the project is generated, right click on the project in the Solution Explorer, then go to Configuration Properties->C/C++->General->Additional Include Directories and add the PC-MCAT\ApiCPP directory:
    - C:\Program files\TrioMotion\PC-MCAT\ApiCPP



- Within the same project properties dialogue, go to Configuration Properties->Linker->Input and add the path to the .lib files found in that directory:
    - C:\Program Files\TrioMotion\PC-MCAT\ApiCPP\pcmcat_api_x86.lib
    - C:\Program Files\TrioMotion\PC-MCAT\ApiCpp\pcmcat_api_x64.lib



- Open the automatically generated "Console_Cpp_ApiTest.cpp" file and include the following headers using the #include directive.

```
#include "pch.h"
#include <iostream>
#include <string>
#include <basetsd.h>
#include "pcmcat_api.h"
```

- Within the main function, add code to open a connection, Set VR(0) value according to the user input and Read the value back to check whether it has been properly set. After that close the connection.

```
int main()
{
   std::string input_str;
   double set_value;
   double get_value;

   if (0 == pcmcat_api_open(NULL, NULL))
   {
      std::cout << "Connection Open\n";
      std::cout << "Introduce the number you want VR(0) to be set to-> ";
      std::cin >> input_str;

      set_value = (double)atof(input_str.c_str());

      if (0 == pcmcat_api_set_vr(0,1,&set_value))
      {
         std::cout << "VR(0) Written -> " << set_value << "\n";
```

```
      if (0 == pcmcat_api_get_vr(0,1,&get_value))
      {
         std::cout << "VR(0) Read-> "<< get_value <<"\n";
         if ((get_value - set_value) < DBL_EPSILON)
            std::cout << "VR(0) has been written correctly\n";
         else
            std::cout << "VR(0) written and read values don't match\n";
      }
      else
      {
         std::cout << "Error reading VR(0)\n";
      }
   }
   else
   {
      std::cout << "Error writting VR(0)\n";
   }
}
else
{
   std::cout << "Connection Not Open\n";
}
pcmcat_api_close();
std::cout << "----- Press 'x' plus Enter to exit -----\n";
while (input_str != "x")
   std::cin >> input_str;
}
```

- Add the Callback function to the code and modify the pcmcat_api_close() function call accordingly.

```
void __stdcall static_pcmcat_api_callback(void *context, pcmcat_api_callback_data_t
*pcmcat_api_callback_data);

int main()
{
   std::string input_str;
   double set_value;
   double get_value;

   if (0 == pcmcat_api_open(static_pcmcat_api_callback, NULL))
   {
      std::cout << "Connection Open\n";
      std::cout << "Introduce the number you want VR(0) to be set to-> ";
      std::cin >> input_str;

      set_value = (double)atof(input_str.c_str());

      if (0 == pcmcat_api_set_vr(0,1,&set_value))
      {
         std::cout << "VR(0) Written -> " << set_value << "\n";

         if (0 == pcmcat_api_get_vr(0,1,&get_value))
         {
            std::cout << "VR(0) Read-> "<< get_value <<"\n";
            if ((get_value - set_value) < DBL_EPSILON)
               std::cout << "VR(0) has been written correctly\n";
            else
               std::cout << "VR(0) written and read values don't match\n";
         }
         else
         {
            std::cout << "Error reading VR(0)\n";
         }
      }
      else
```

```
      {
         std::cout << "Error writting VR(0)\n";
      }
   }
   else
   {
      std::cout << "Connection Not Open\n";
   }
   pcmcat_api_close();
   std::cout << "----- Press 'x' plus Enter to exit -----\n";
   while (input_str != "x")
      std::cin >> input_str;
}

void __stdcall static_pcmcat_api_callback(void *context, pcmcat_api_callback_data_t
*pcmcat_api_callback_data)
{
   if (!pcmcat_api_callback_data)
      return;

   switch (pcmcat_api_callback_data->type)
   {
   case pcmcat_api_callback_type_message:
      if (pcmcat_api_callback_data->data.str)
         std::cout << pcmcat_api_callback_data->data.str << "\n";
   }
}
```