

|  |
| --- |
| **C# Starter Guide** |
| **Controlling USBTMC Test Instruments Remotely** |

# Pre-requisites

The example provide with this document was created using the software detailed below. This is a proven way to achieve communication between the computer and the instrument.

## Download and Install:

1. Microsoft Visual Studio Community:

A fully-featured, extensible, free IDE for creating modern applications… available for free download here:

<https://www.visualstudio.com/vs/community/>

1. NI VISA: <http://www.ni.com/download/ni-visa-16.0/6184/en/> (or latest version, check compatibility with your Windows Operating system).

## Description of required software

1. **Microsoft Visual Studio Community 2015:** A fully-featured, extensible, free IDE for creating modern applications for Android, iOS, Windows, as well as web applications and cloud services. It is a limited version, but provides everything the user needs to get started and start communicating with your test instrumentation.
2. **NI VISA: From LabVIEW’s website:**

“The Virtual Instrument Software Architecture (VISA) is a standard for configuring, programming, and troubleshooting instrumentation systems comprising GPIB, VXI, PXI, Serial, Ethernet, and/or USB interfaces. VISA provides the programming interface between the hardware and development environments such as LabVIEW, LabWindows/CVI, and Measurement Studio for Microsoft Visual Studio. NI-VISA is the National Instruments implementation of the VISA I/O standard. NI-VISA includes software libraries, interactive utilities such as NI I/O Trace and the VISA Interactive Control, and configuration programs through Measurement & Automation Explorer for all your development needs.”

# Creating a New Project

1. Open Visual Studio
2. Click **File: New > Project**
3. Under Templates, select **Visual C#**
4. Click on “Windows Form Application”
5. Name the project and select a location
6. Click **OK** to open the project.

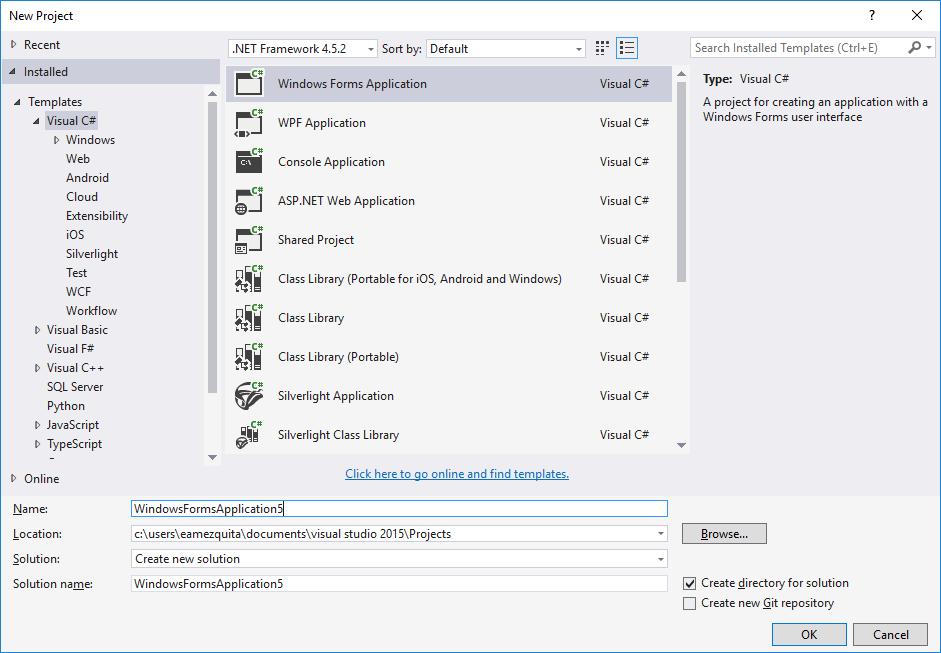


Figure 1 - Creating a Windows Form Application

# Getting ready

Before getting started with the code, we need to make sure our code has all the references that needs to establish communication, create a resource manager, and open a messaged-based session with the instrument. Here is where we take advantage of the advantages of using NI VISA and Visual Studio.

We have to add the following references to the project:

* National instruments.common
* National instruments.visa
* VisaCom lib

1. Right click on **References** on the Solution Explorer
2. Click on the **Add Reference o**n the next pop up window.
3. The window below will be shown.
4. Click on **“Assemblies”.**
5. On the Search field, type “national instruments” and the two options below will be shown.
6. Check the boxes for National Instruments Common and National Instruments VISA.

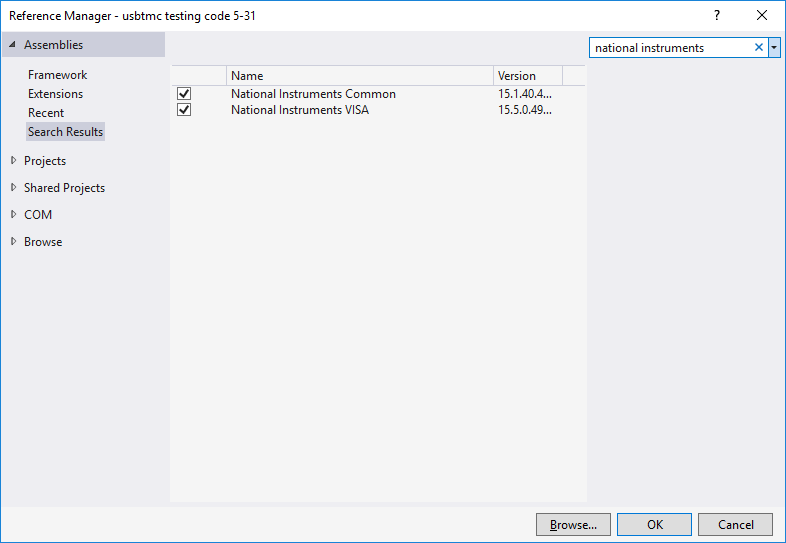
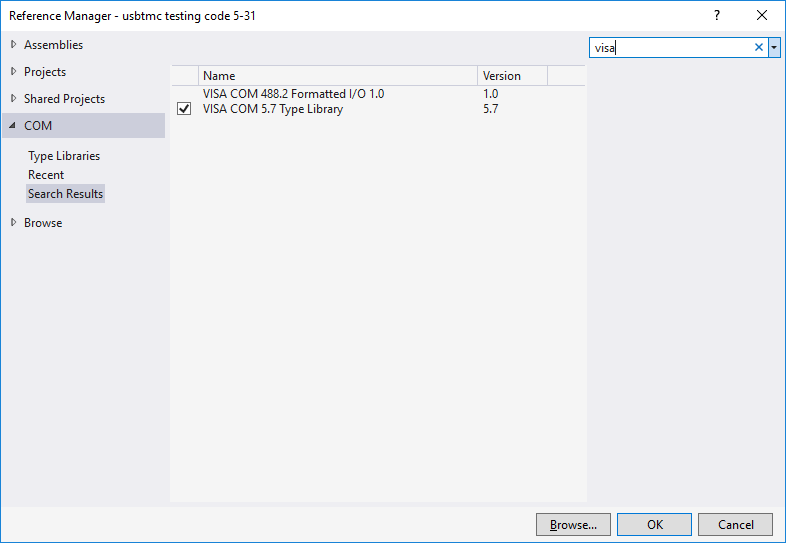


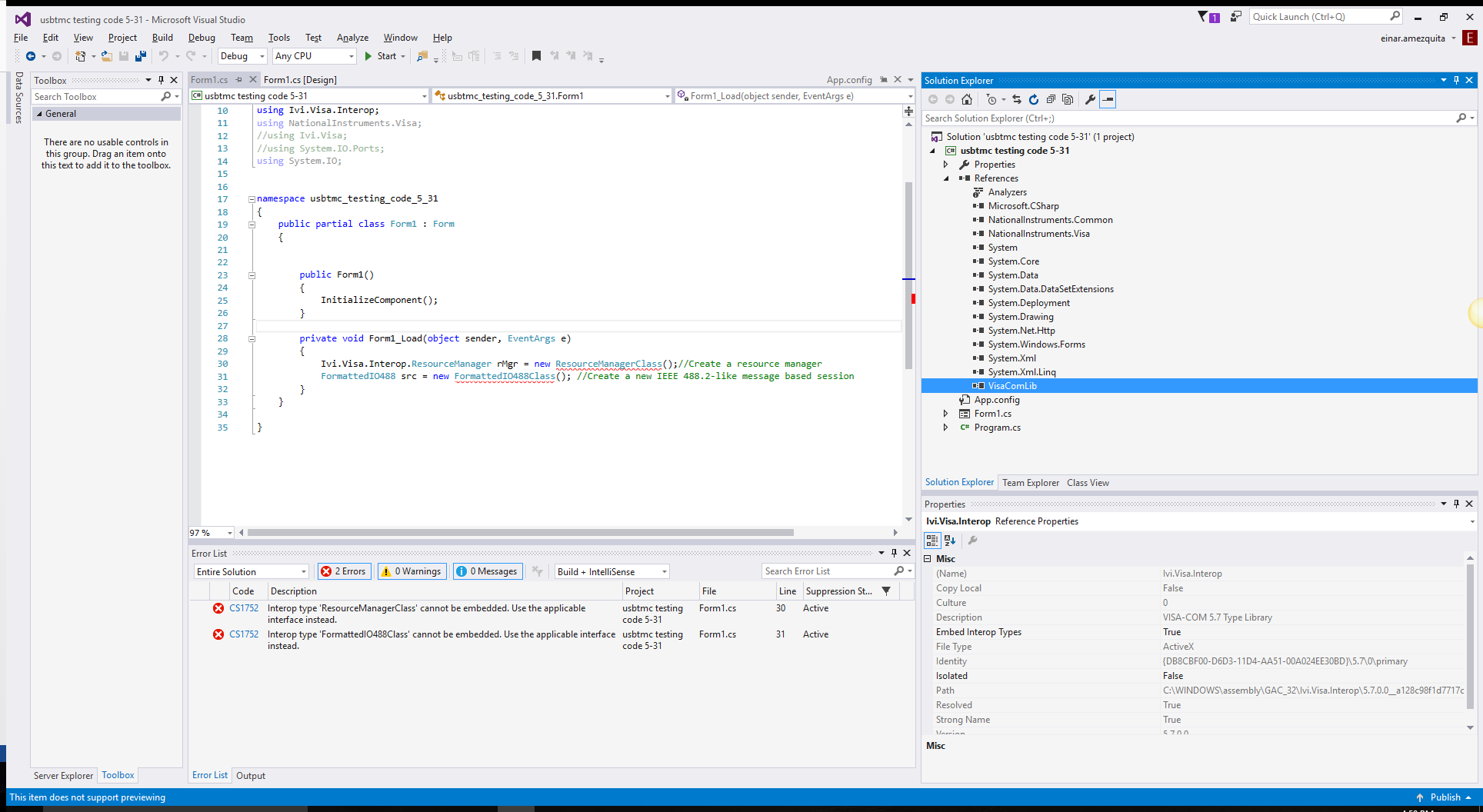
Figure - Adding National Instruments Assemblies

1. In the same window, click COM.

Figure 3 - Adding VISA COM Library



1. Click OK to add the references to your code.
2. Back on the Solution Explorer, click on the option **VisaComLib**
3. In the properties of **VisaComLib,** turn the **Embed Interop Types** to **False.**



1. Next step is to add “using” statements. The “using” keyword is used to include the object namespace in the program. A namespace is a collection of classes.

We have to add the following statements (see attached Form1.cs file) to the header of the code. They should be added in addition to the “using” statements prefilled by Visual Studio:

using Ivi.Visa.Interop;

using NationalInstruments.Visa;

using System.IO;

1. Inside the **Public Partial Class Form1:Form** create a Resource Manager and a Message Based Session.
2. Your code should like something like this:
3. using System;
4. using System.Collections.Generic;
5. using System.ComponentModel;
6. using System.Data;
7. using System.Drawing;
8. using System.Linq;
9. using System.Text;
10. using System.Threading.Tasks;
11. using System.Windows.Forms;
12. using Ivi.Visa.Interop;
13. using NationalInstruments.Visa;
14. //using Ivi.Visa;
15. //using System.IO.Ports;
16. using System.IO;
17. namespace USBTMC\_project
18. {
19. public partial class Form1 : Form
20. {
21. Ivi.Visa.Interop.ResourceManager rMgr =

new ResourceManagerClass();//Create a resource manager

* 1. FormattedIO488 src = new FormattedIO488Class(); //Create a new IEEE 488.2-like message based session

//your code

To run the executable created as an example:

Unzip the USBTMC project

Open the USTMC project folder

Bin>Debug> Double click on USBTMC project (type Application, 17 KB)

This executable can be modified to meet the user requirements.