Walkthrough: Creating a Measurement Studio Application with Windows Forms Controls and Network Variable

Measurement Studio includes user interface controls, such as a waveform graph control, and network variable functionality to transfer live measurement data between applications over the network. This walkthrough is designed to help you learn how to add network variable functionality to a Windows Forms application by taking you through the following steps:

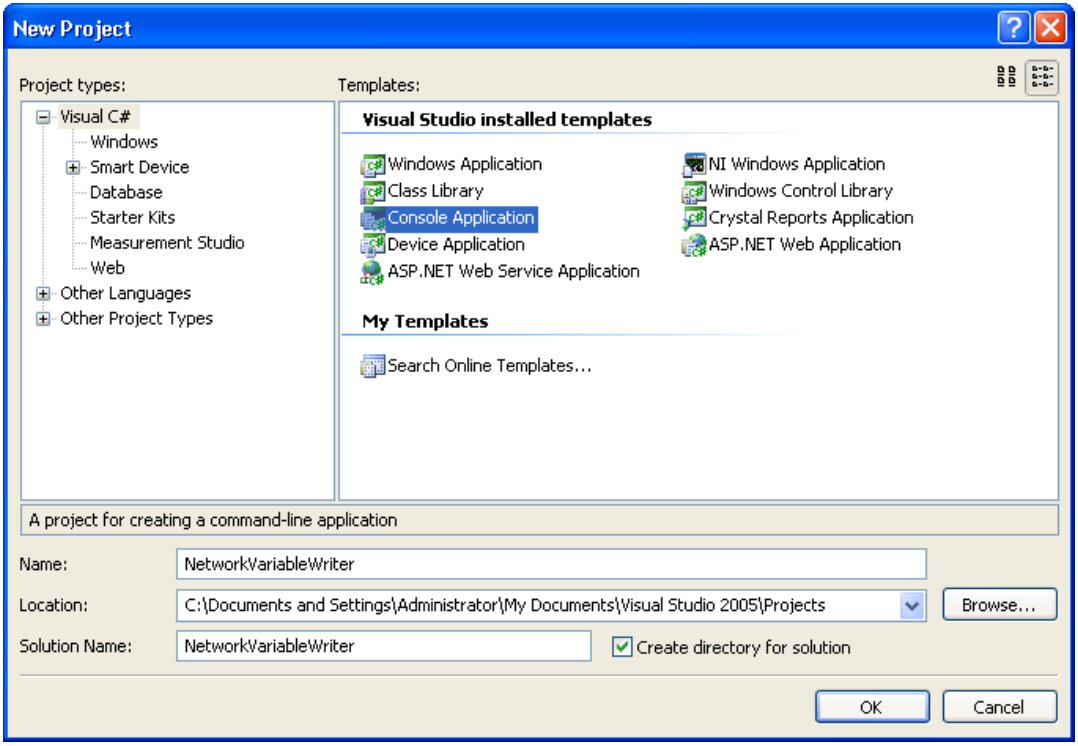
- **Writing an array of data to the server**—Using `NationalInstruments.NetworkVariable.NetworkVariableBufferedWriter<TValue>`, you will create and run a console application that writes an array of values to the server.
- **Setting up a Windows Forms project**—Using the Measurement Studio Application Wizard, you will create a new project that references the Measurement Studio Network Variable class library and Windows Forms controls.
- **Configuring the network variable data source control**—Using the Toolbox and the `NationalInstruments.NetworkVariable.WindowsForms.NetworkVariableDataSource` smart tag, you will add and configure a data source control to your application.
- **Displaying the array of data on a Windows Forms page**—Using the Toolbox, you will add and configure a `NationalInstruments.WaveformGraph` control to display the data.

Before You Begin

The following components are required to complete this walkthrough:

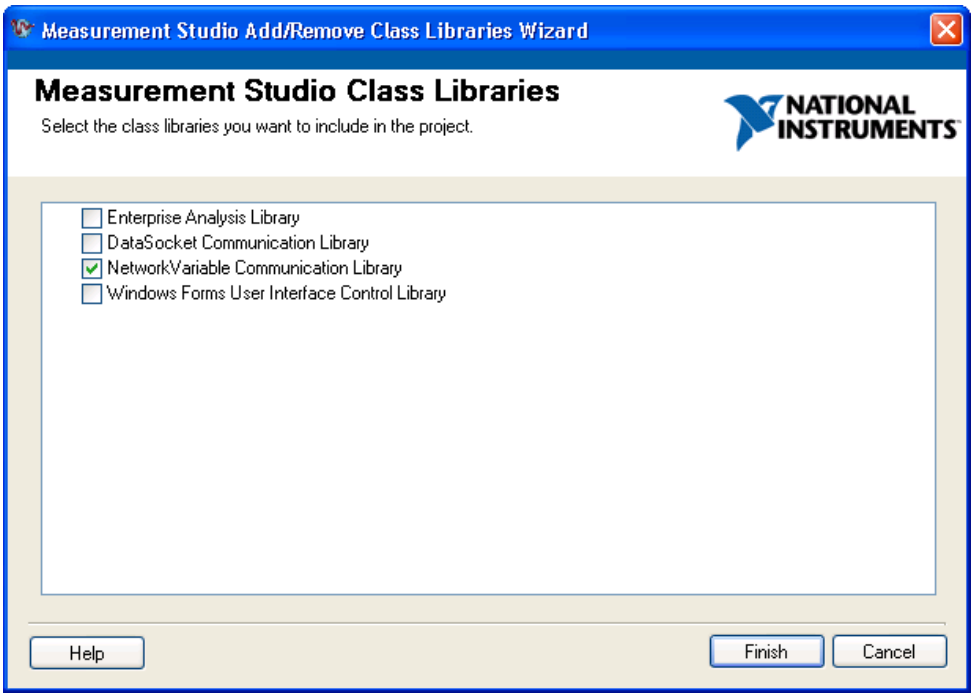
- Microsoft Visual Studio .NET 2005
- Measurement Studio 8.1 or later (Professional or Enterprise package)

1. Select **Start»All Programs»Microsoft Visual Studio 2005»Microsoft Visual Studio 2005**.
2. Select **File»New»Project**. The New Project dialog box launches.



3. In the Project Types pane, select **Visual C#** or **Visual Basic**, depending on which language you want to create the project in.
4. In the Templates pane, select **Console Application**. Specify `NetworkVariableWriter` for **Name** and specify a **Location** of your choice.
5. Click **OK**.
6. Select **Measurement Studio»Add/Remove .NET Class Libraries**. The Measurement Studio Add/Remove Class Libraries Wizard launches. You use this wizard to add Measurement Studio components to your project.

7. Select **NetworkVariable Communication Library**. Click **Finish**.



8. In `Program.cs`, add the following code to write an array of data to the server:

[VB.NET]

```
Imports NationalInstruments.NetworkVariable
Imports System.Threading
Module Module1
Private Function GenerateDoubleArray(ByVal phase As Double) As Double()
    Dim values(999) As Double
    Dim x As Integer
    For x = 0 To 999
        values(x) = Math.Sin(((2 * Math.PI * x) / 1000) + phase) * 2
    Next x
    Return values
End Function
Sub Main()
    Const location As String = "\\localhost\system\double"
    Dim bufferedWriter As NetworkVariableBufferedWriter(Of Double()) = _
    New NetworkVariableBufferedWriter(Of Double())(location)
    bufferedWriter.Connect()
    Dim phase As Integer = 0
    While (True)
```

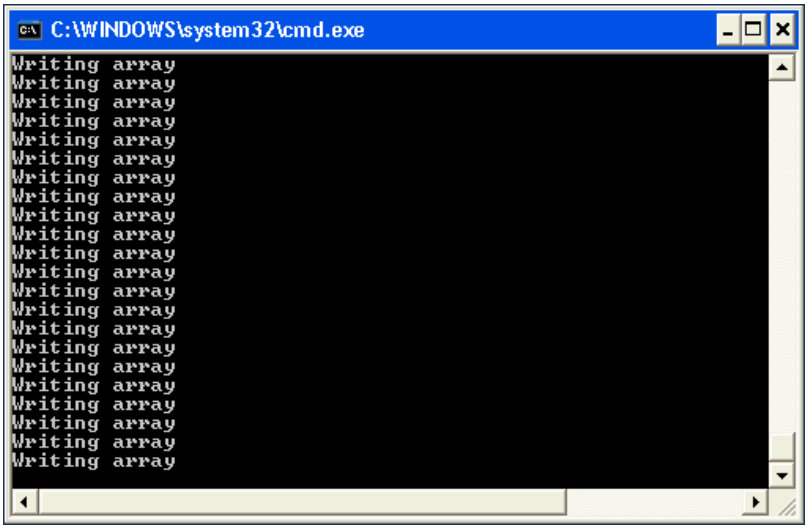
```
        Dim values As Double() = GenerateDoubleArray(phase)
        Console.WriteLine("Writing Array")
        bufferedWriter.WriteValue(values)
        Thread.Sleep(500)
        phase = phase + 1
    End While
End Sub
End Module
```

[C#]

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;
using NationalInstruments.NetworkVariable;
namespace NetworkVariableWriter
{
    class Program
    {
        private static double[] GenerateDoubleArray(double phase)
        {
            double[] values = new double[1000];
            for (int x = 0; x < 1000; x++)
                values[x] = Math.Sin(((2 * Math.PI * x) / 1000) + phase) * 2;
            return values;
        }
        static void Main(string[] args)
        {
            const string Location = @"\\localhost\system\double";
            NetworkVariableBufferedWriter<double[]> bufferedWrite = new
            NetworkVariableBufferedWriter<double[]>(Location);
            bufferedWrite.Connect();
            int phase = 0;
            while (true)
            {
                double[] value = GenerateDoubleArray(phase);
                Console.WriteLine("Writing array");
                bufferedWrite.WriteValue(value);
                Thread.Sleep(500);
                phase++;
            }
        }
    }
}
```

Note You should choose the appropriate code depending on whether you created a VB or C# project.

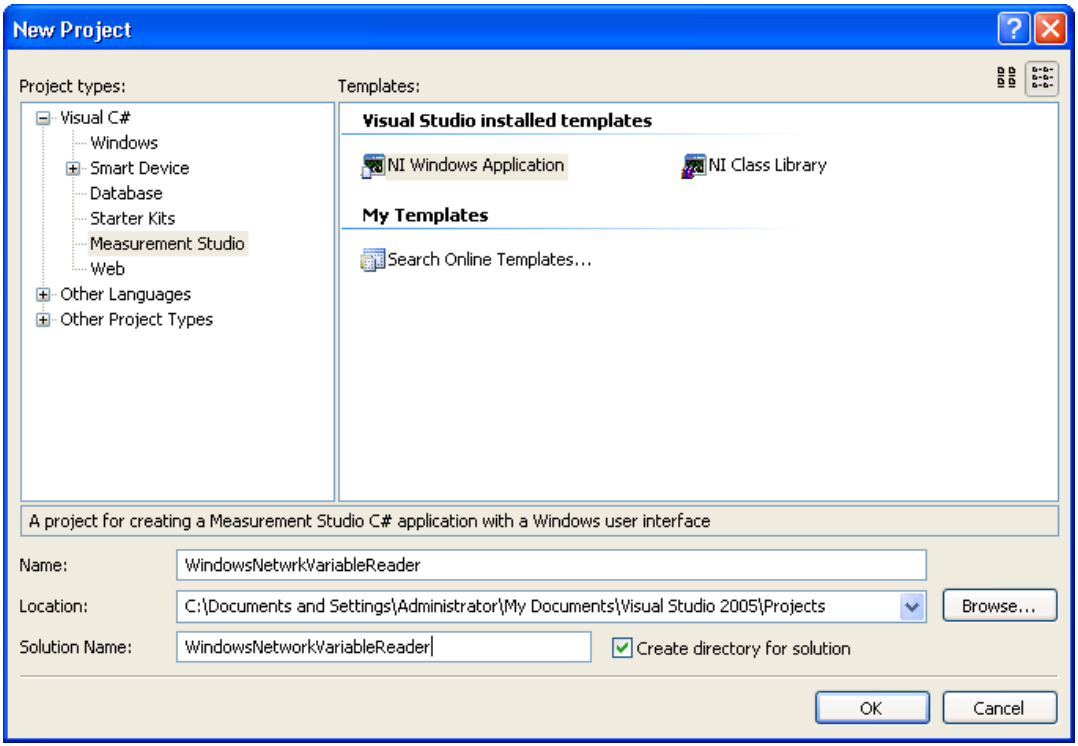
9. Select **Debug»Start Without Debugging** to run the application.



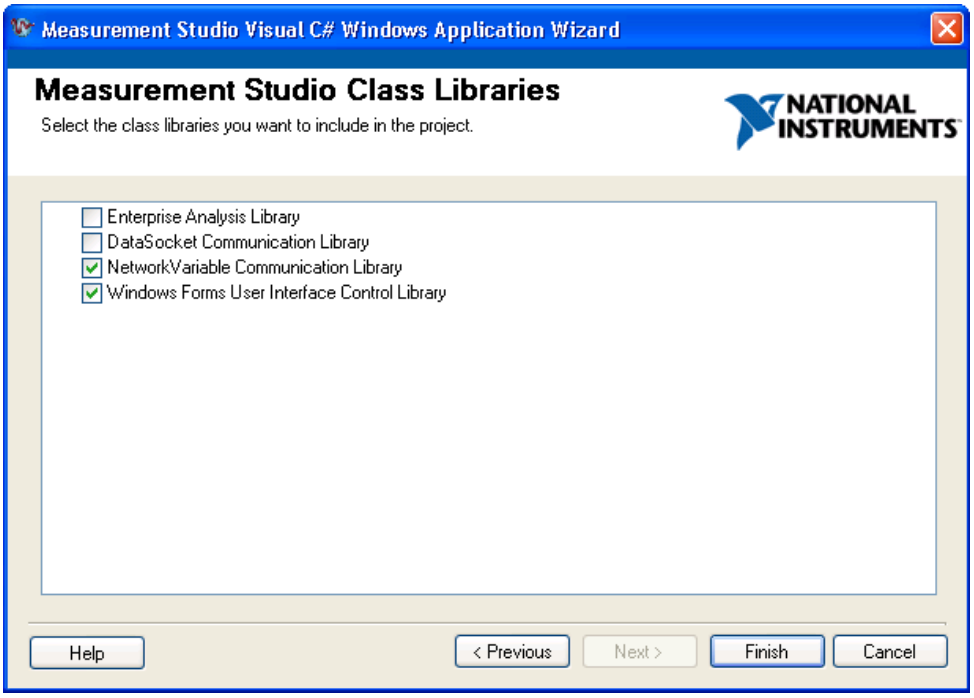
10. Minimize the console, but keep the application running.

Setting Up a Windows Forms Project

1. Select **Start»All Programs»Microsoft Visual Studio 2005»Microsoft Visual Studio 2005**.
2. Select **File»New»Project**. The New Project dialog box launches.



3. In the Project types pane, select **Measurement Studio**.
4. Use the drop-down box to select **Visual C#** or **Visual Basic**, depending on which language you want to create the project in.
5. In the Templates pane, select **NI Windows Application**. Name the project `WindowsNetworkVariableReader` and specify a Location you wish to save to project by clicking **Browse** and navigating to a directory of your choice.
6. Click **OK**. The Measurement Studio Application Wizard launches.
7. Select **Network Variable Communication Library** and **Windows Forms User Interface Control Library**.



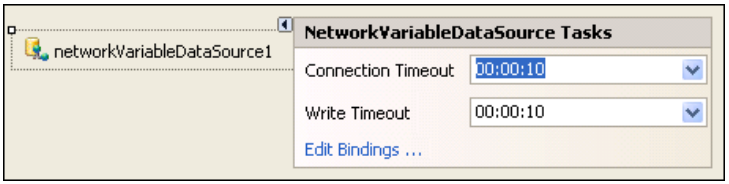
Tip If you are working with an existing project, you can access the Add/Remove Class Libraries dialog box by selecting **Measurement Studio»Add/Remove Class Libraries Wizard**.

8. Click **Finish** to display `Form1` in the Windows Forms Designer.

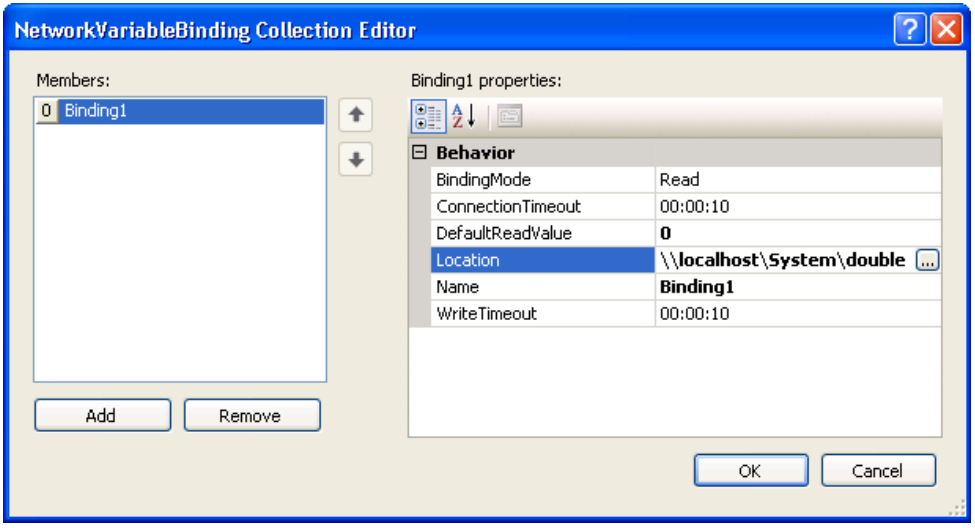
Configuring the Network Variable Data Source Control

1. Select **View»Toolbox** to display the Toolbox. The toolbox contains components and controls that you can add to your project.
2. Expand the **Measurement Studio** group on the Toolbox.
3. Select the NetworkVariableDataSource control in the toolbox and drag and drop it on the form. The `NationalInstruments.NetworkVariable.WindowsForms.NetworkVariableDataSource` control is a data source control with functionality similar to `System.Web.UI.WebControls.ObjectDataSource` and `System.Web.UI.WebControls.SqlDataSource` in the .NET Framework. The `NationalInstruments.NetworkVariable.WindowsForms.NetworkVariableDataSource` control encapsulates `NationalInstruments.NetworkVariable` functionality.

4. In the NetworkVariableDataSource smart tag, select **Edit Bindings** to launch the NetworkVariableBinding Collection Editor dialog box.



5. Select **Add** to create a connection with the underlying network variable, You can use the NetworkVariableBinding Collection Editor to configure the binding properties. Enter **0** as the **DefaultReadValue**.



6. For the **Location**, browse to the `\\localhost\System\double` location in the Select Network Item dialog box.

7. Click **OK** to return to the NetworkVariableBinding Collection Editor dialog box.
8. After you configure the binding properties, click **OK** to return to the Windows Forms Designer.

Displaying the Array of Data on a Windows Form

1. Select **WaveformGraph** in the Toolbox and drag and drop it on the form.
2. Right-click the waveform graph and select **Properties** to display the Properties window for the graph. You can configure the properties of the control in the Properties window.
3. Expand the **Data Bindings** group in the Properties window. Select **Other Data Sources»Form 1 List Instances» networkVariableDataSource1»Binding1** from the **Binding Data** drop-down list. This will bind the waveform graph to the network variable that you are writing to in the console application. The waveform graph will then read and display the data being written to the network variable.
4. Select **File»Save Form1** to save your application.
5. Select **Debug»Start Without Debugging** to run the application. The waveform graph displays the array of data.