Walkthrough: Creating a Measurement Studio Application with Web 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 Web 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 Web Forms project**—Using the Measurement Studio Application Wizard, you will create a new project that references the Measurement Studio Network Variable class library and Web Forms controls.
- **Configuring the network variable data source control**—Using the Toolbox and the `NationalInstruments.NetworkVariable.WebForms.NetworkVariableDataSource` smart tag, you will add and configure a data source control to your application.
- **Displaying the array of data on a Web page**—Using the Toolbox, you will add and configure an `NationalInstruments.UI.WebForms.AutoRefresh` control and a `NationalInstruments.UI.WebForms.WaveformGraph` control to display the data.

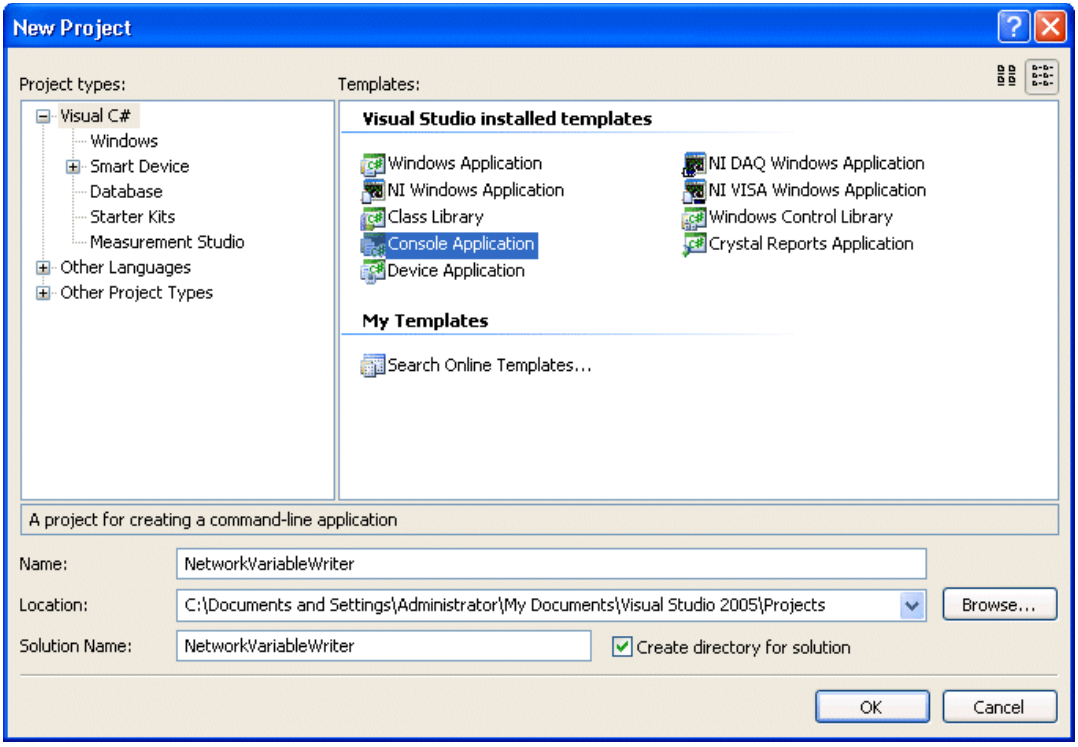
Before You Begin

The following components are required to complete this walkthrough:

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

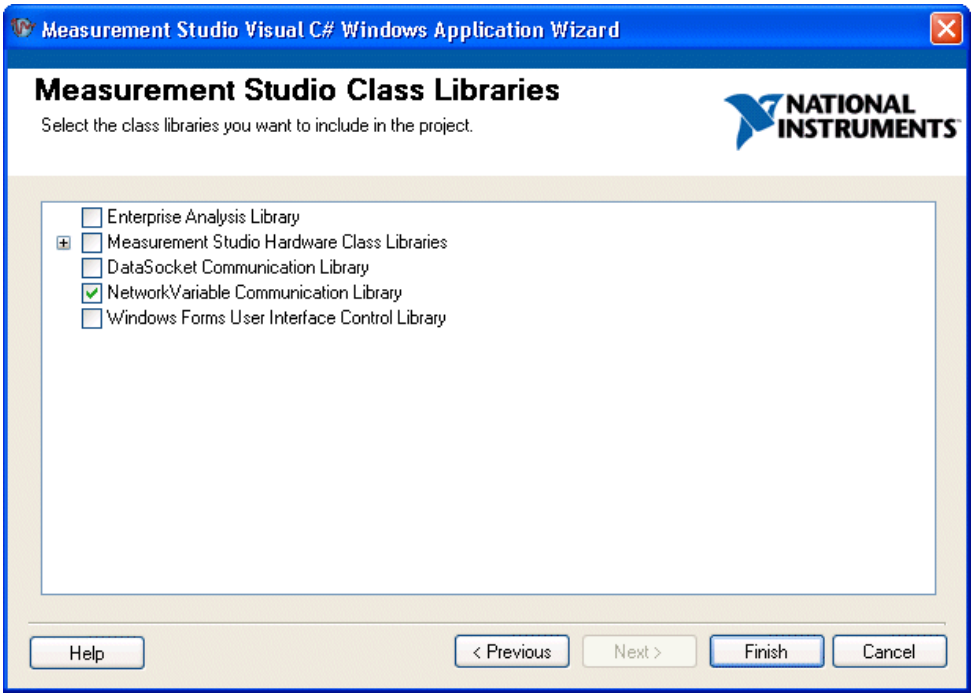
Writing an Array of Data to the Server

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

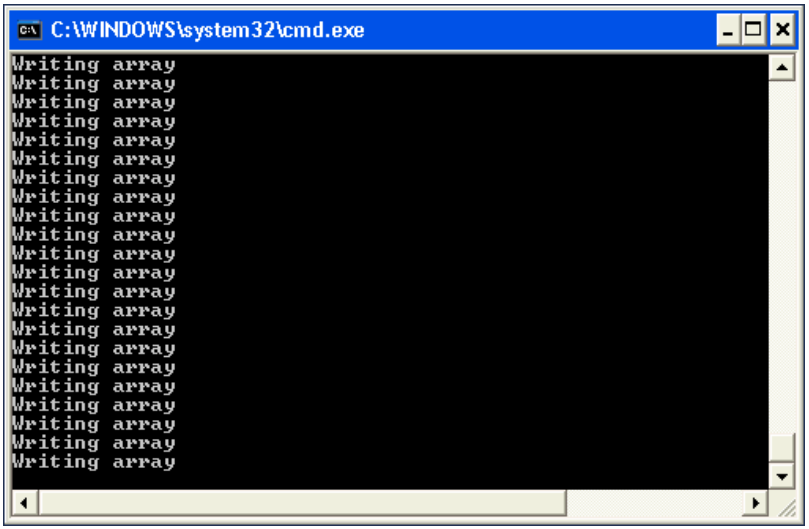
```
Imports NationalInstruments.NetworkVariable
Imports System.Threading
Module Module1
    Private Function GenerateDoubleArray(ByVal phase As Double) As Double()
        Dim values(999) As Double
        Dim x As Integer
        For x = 0 To 999
            values(x) = Math.Sin(((2 * Math.PI * x) / 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.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[]> bufferedWriter = new
            NetworkVariableBufferedWriter<double[]>(Location);
            bufferedWriter.Connect();
            int phase = 0;
            while (true)
            {
                double[] value = GenerateDoubleArray(phase);
                Console.WriteLine("Writing array");
                bufferedWriter.WriteValue(value);
                Thread.Sleep(500);
                phase++;
            }
        }
    }
}
```

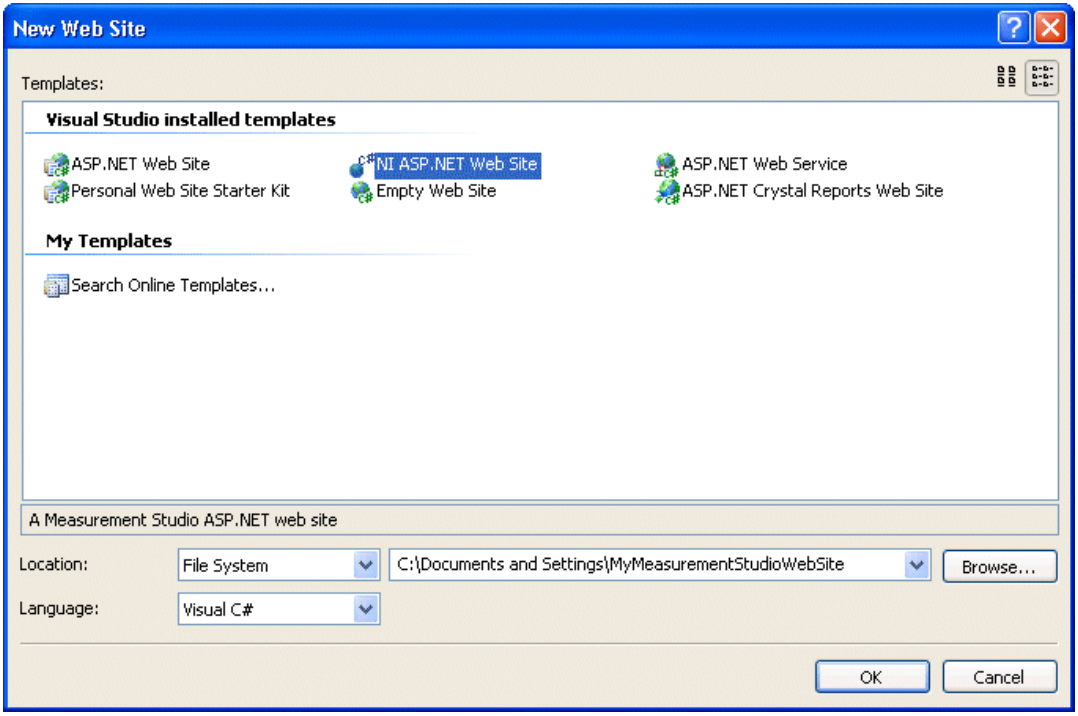
9. Select **Debug»Start Without Debugging** to run the application.



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

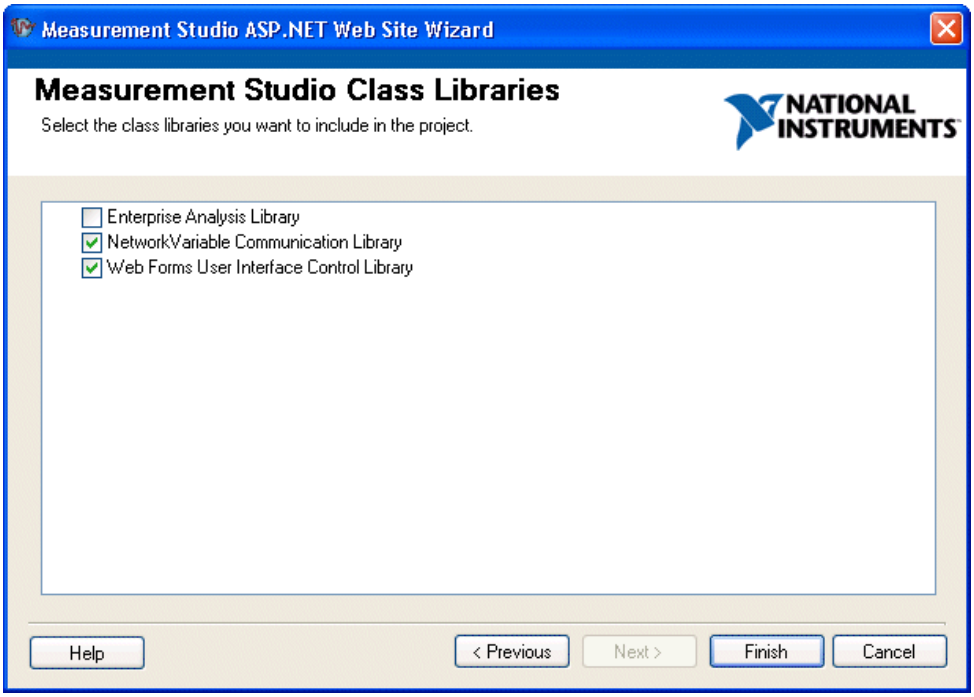
Setting Up a Web Forms Project

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



3. In the Templates pane, select **NI ASP.NET Web Site**. Select `File System` for Location and specify a file path of your choice.
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. Click **OK**. The Measurement Studio ASP.NET Web Site Wizard launches.

6. Select **Network Variable Communication Library** and **Web 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**.

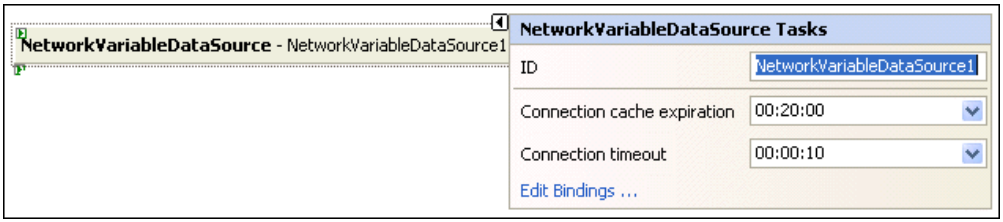
7. Click **Finish** to display `Default.aspx` in the Web Forms Designer.
8. You can rename the title of your Web page. Click inside the `<title>` tag and rename the title to **Measurement Studio Network Variable and Web Forms Controls Walkthrough**.

Configuring the Network Variable Data Source Control

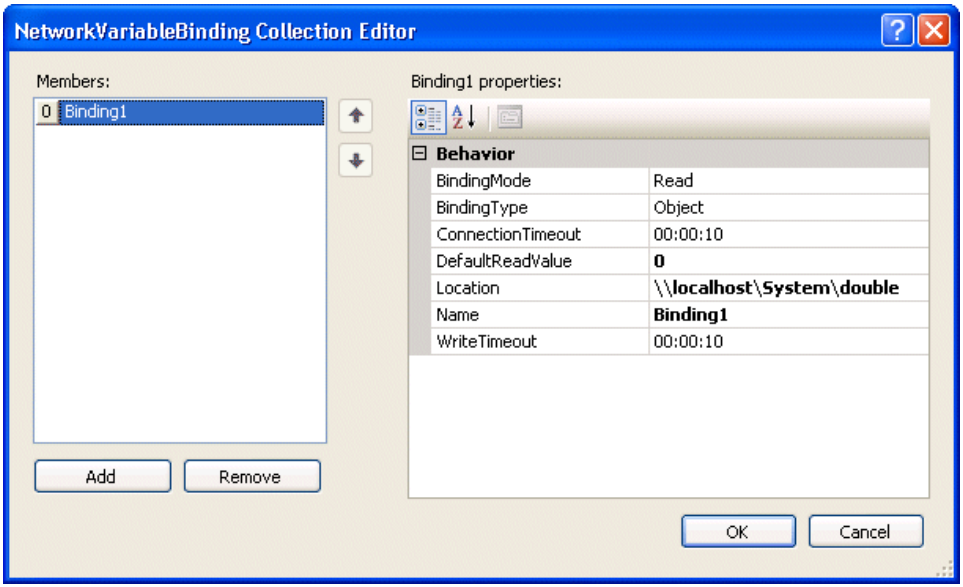
1. Click **Design** in the lower left corner to switch from Source View to Design View.
2. Select **View»Toolbox** to display the Toolbox. The toolbox contains components and controls that you can add to your project.
3. Expand the **Measurement Studio** group on the Toolbox.
4. Select the NetworkVariableDataSource control in the toolbox and drag and drop it on the form. The `NationalInstruments.`

`NetworkVariable.WebForms.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.WebForms.NetworkVariableDataSource` control encapsulates `NationalInstruments.NetworkVariable` functionality.

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



6. Select **Add**. You add a binding to create a connection with the underlying network variable, and you use the NetworkVariableBinding Collection Editor to configure the binding properties. Select **Object** for the BindingType. You select **Object** because this walkthrough binds to `NationalInstruments.UI.WebForms.WaveformGraph.BindingData`. Enter **0** as the **DefaultReadValue**.

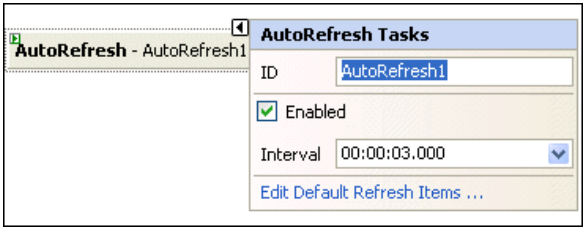


7. Browse to the `\\localhost\System\double` location in the Select Network Item dialog box.

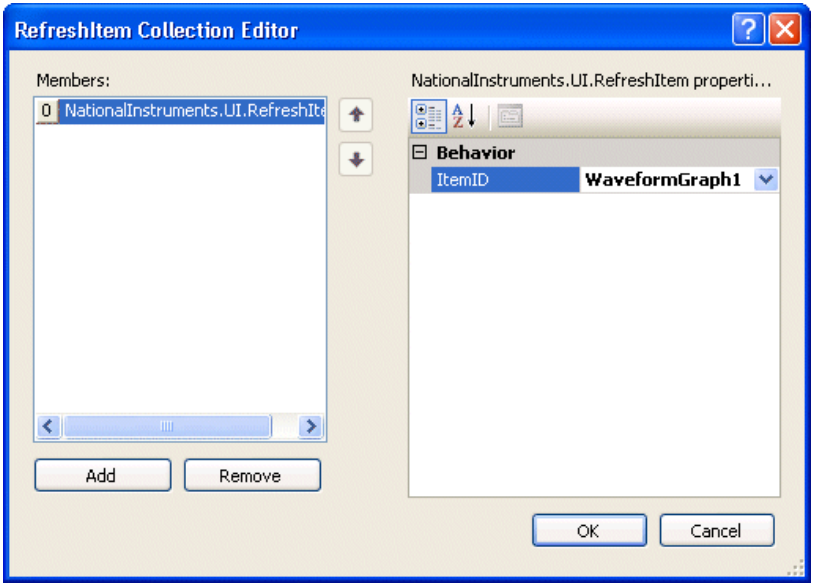
8. Click **OK** to return to the NetworkVariableBinding Collection Editor dialog box.
9. After you configure the binding properties, click **OK** to return to the ASP.NET Designer.

Displaying the Array of Data on a Web Page

1. Select **WaveformGraph** in the Toolbox and drag and drop it on the form.
2. Select **AutoRefresh** in the Toolbox and drag and drop it on the form.
3. In the AutoRefresh smart tag, check **Enabled**. Select **Edit Default Refresh Items** to launch the RefreshItem Collection Editor dialog box.



4. Select **Add**. Select **WaveformGraph1** for the ItemID and click **OK**.



5. Double-click the AutoRefresh control. Add the following code to the AutoRefresh event handler to bind the waveform graph control to the network variable data source control:

[VB.NET]

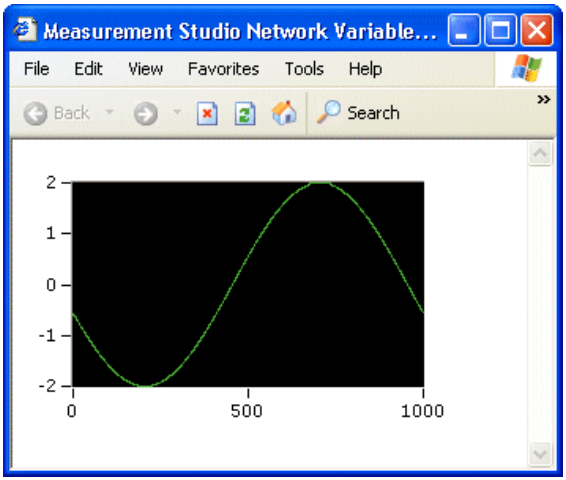
```
WaveformGraph1.BindingData =
NetworkVariableDataSource1.Bindings(0).GetValue()
```

[C#]

```
WaveformGraph1.BindingData =
NetworkVariableDataSource1.Bindings[0].GetValue();
```

6. Select **File»Save Default.aspx** to save your application.

7. Select **Debug»Start Without Debugging** to run the application. The waveform graph displays the array of data.



Note You can also use the `System.Web.UI.WebControls.FormView` control to bind to `NationalInstruments.NetworkVariable.WebForms.NetworkVariableDataSource`. Refer to *Using the Measurement Studio Network Variable Data Source in Web Forms* for more information.