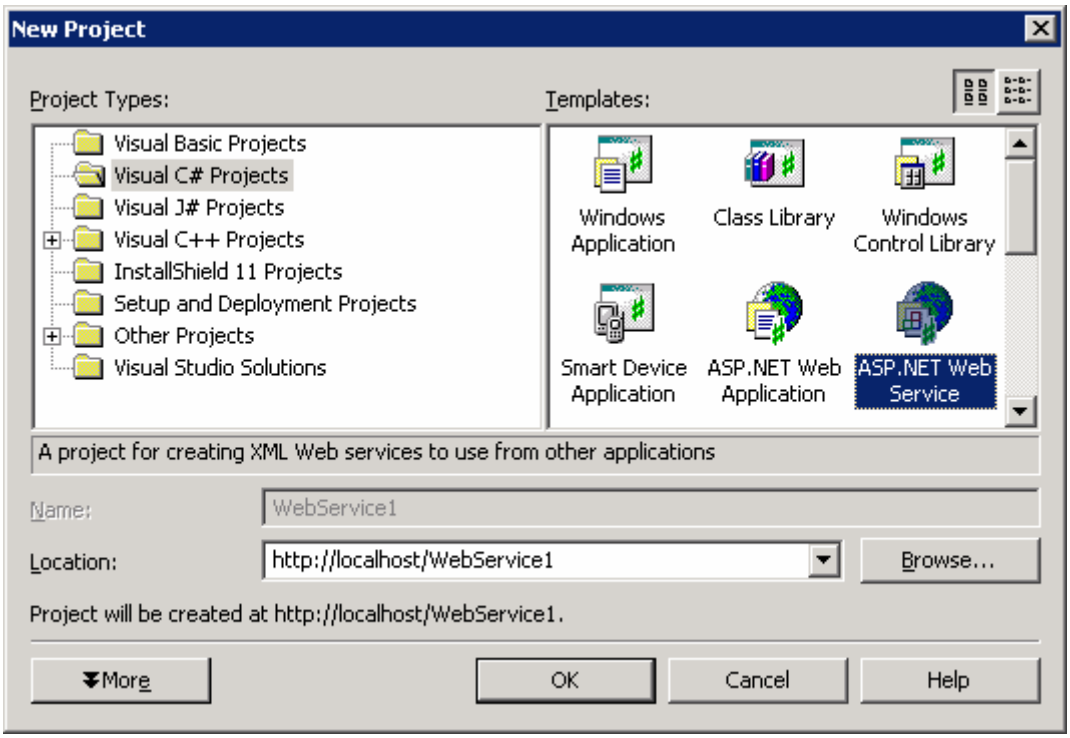
Instructions for writing Web Services using Microsoft .NET:

Pre-requisites:

Operating System: Microsoft Windows XP Professional / Microsoft Windows 2000 Professional / Microsoft Windows 2003 Server
.NET Runtime: Microsoft .NET Framework 1.x or later
Visual Studio: 2003 or 2005
IIS: Required if using Visual Studio .NET 2003

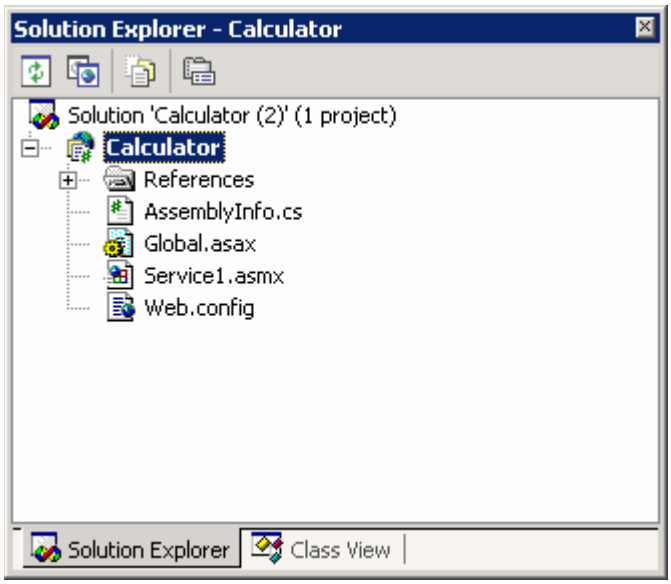
Instruction for Visual Studio .NET 2003
[Note: I am assuming no prior experience in working with Visual Studio .NET 2003]

1. Create a new project and select ASP.NET Web Service project template



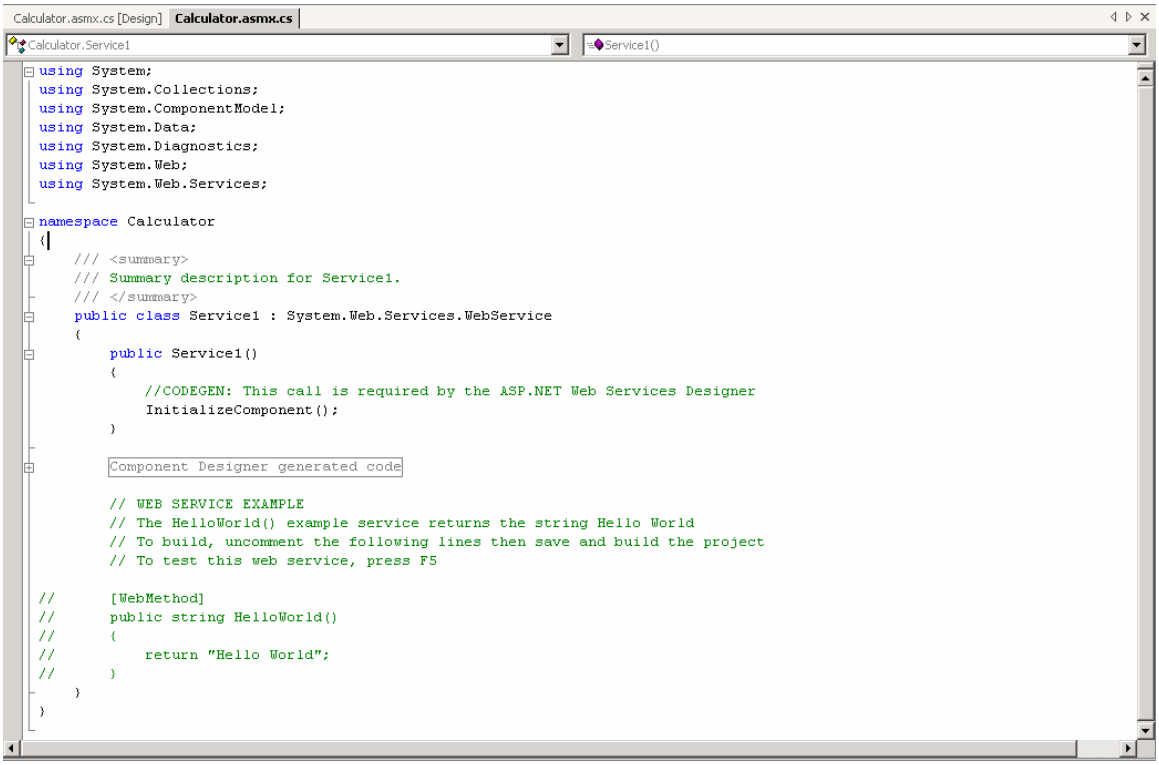
You can specify your own name for the web service. Let us say we are creating a web service called **Calculator** and click ok.

2. You should now see the **Solution Explorer**. If you do not see it, just click on **View** and then click on **Solution Explorer**.



3. Service1.asmx is our web service. The URL for such a service would be http://localhost/Service1.asmx. You can rename your asmx file to the name of your choice. Let us rename it to Calculator.asmx

4. Now right click on the asmx file and select **View Code.**



Service1 is the class name with in the namespace **Calculator.** You can rename it to anything you want to. Let us rename it to **Calculator**.

5. Now build the project. You can now see you web service at http://localhost/Calculator/Calculator.asmx. .NET creates a nice interface which lists all the methods this service provides. At this point we do not have any methods our service provides.

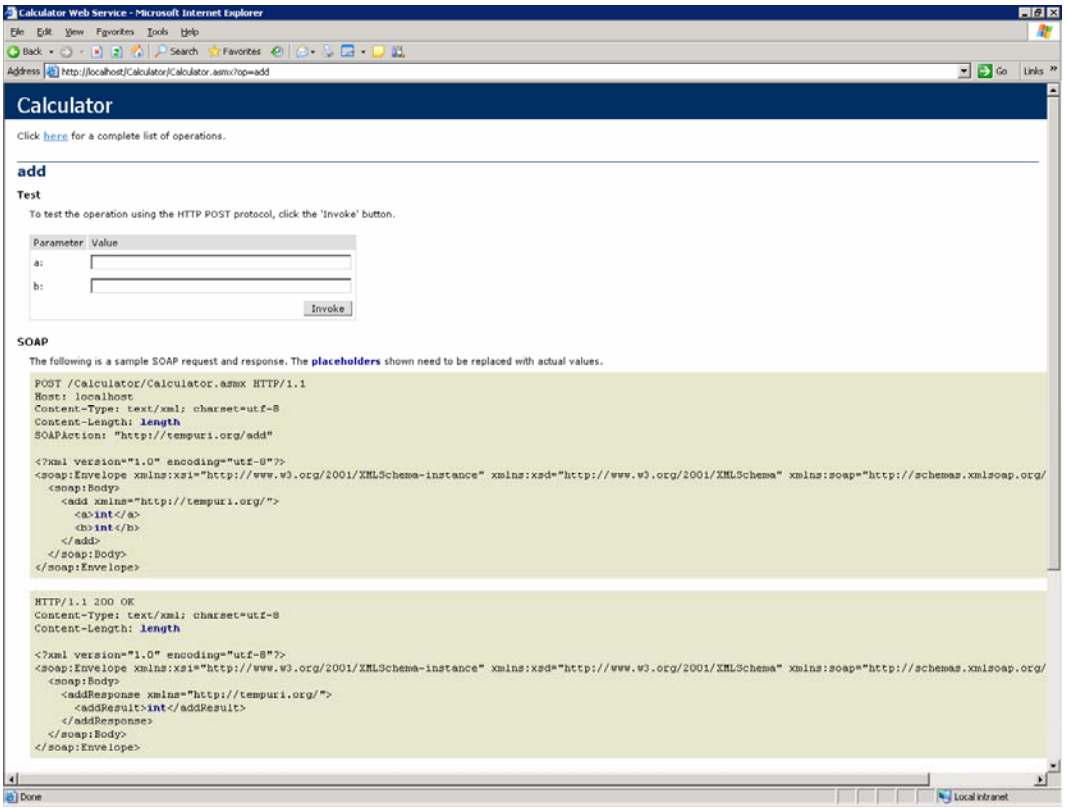
6. Writing the add service. To do that need to create a new web method in the class Calculator. Web methods generally are prefixed by **[WebMethod]** directive. Also make sure that the method is a **public method.**

Add the following lines to the service.

```
[WebMethod]
public int add(int a, int b)
{
      return a + b;
}
```

Build the application. To see what methods your web service supports go back to the URL http://localhost/Calculator/Calculator.asmx. You will see add service listed.

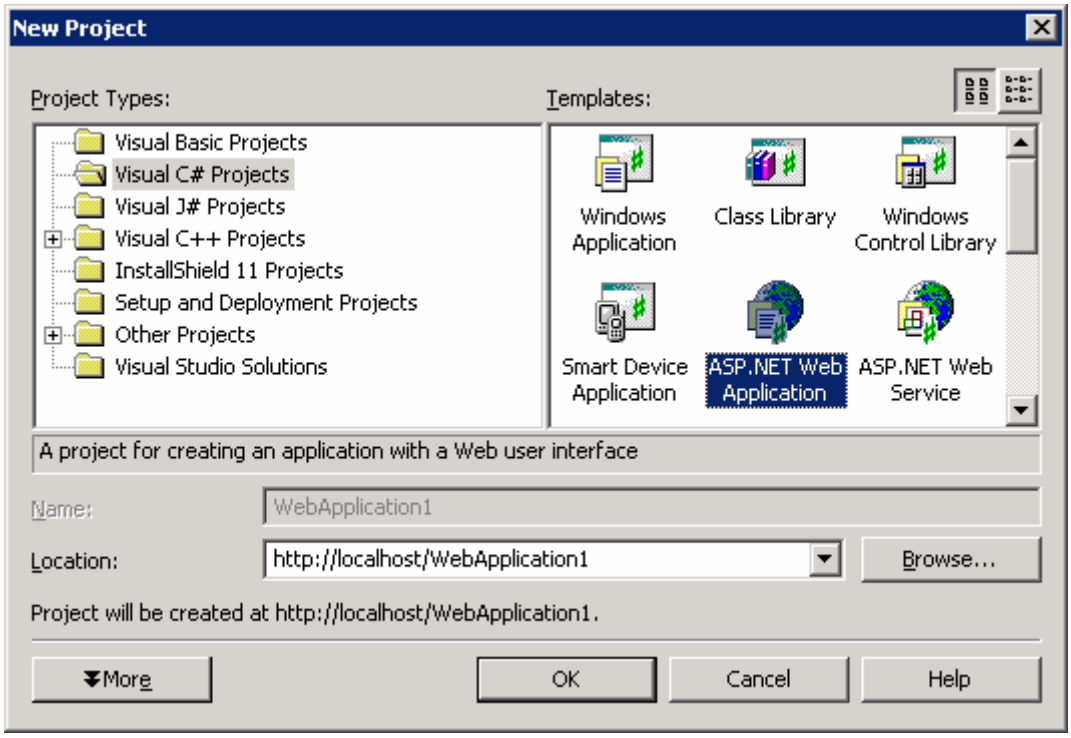
7. Click on add you will see this page give some values for a and b and click on invoke. This will return the result in another page



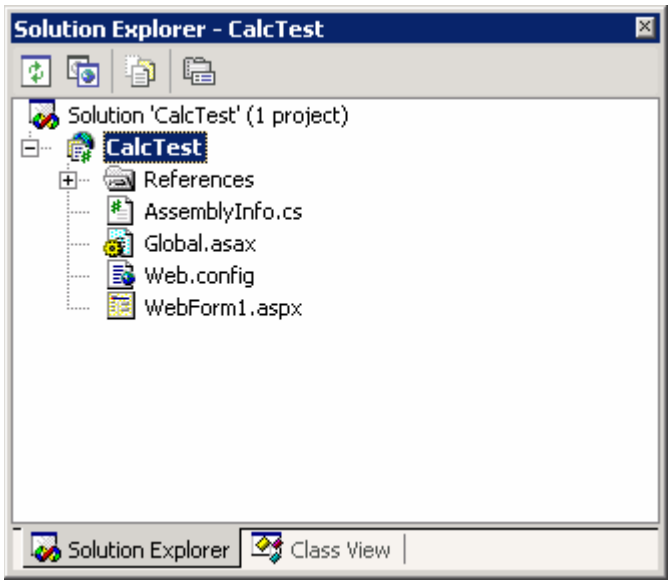
8. WSDL. In order to see the wsdl for the current service, enter the following URL: http://localhost/Calculator/Calculator.asmx?wsdl. Now we have completed writing the add web service.

Writing the client application in .NET.

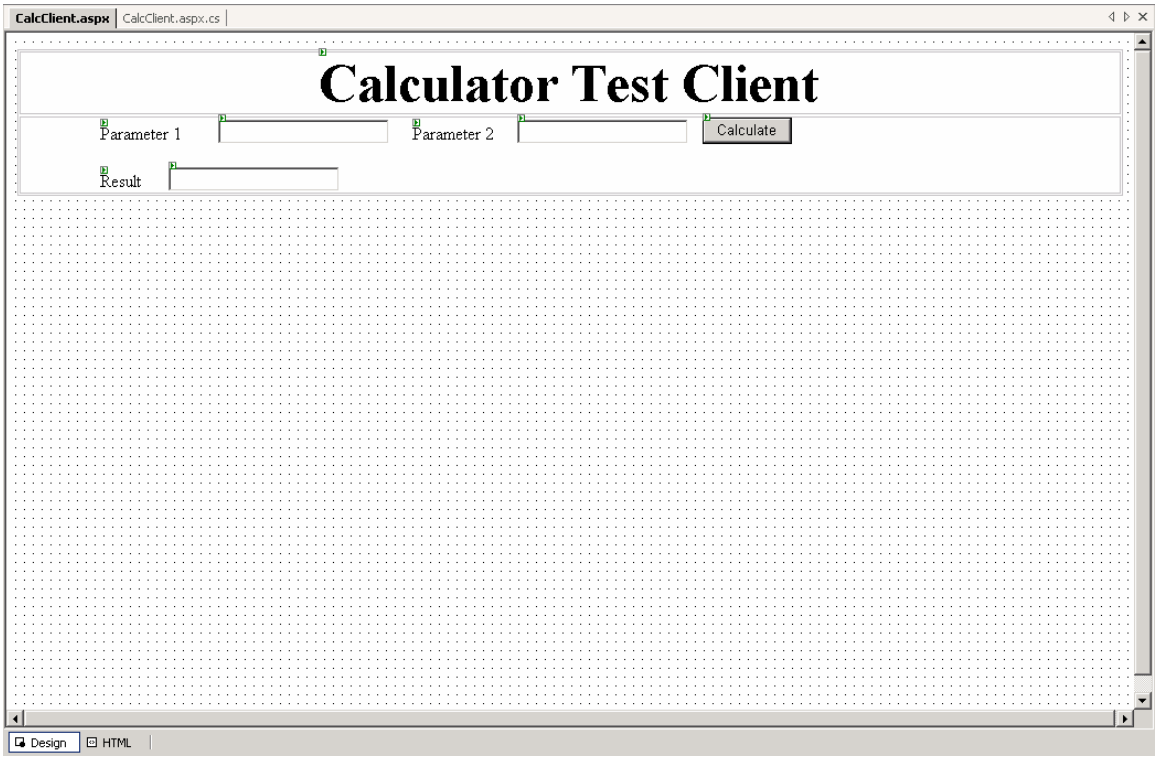
1. Create a new project and select ASP.NET web application give it some name and click OK



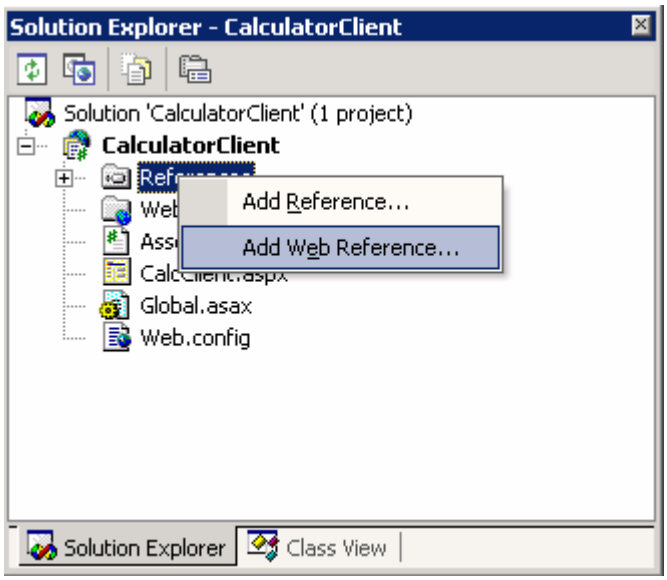
2. You will see a solution explorer that a WebForm1.aspx file. You can rename it to any name you want to. Let us name it as CalcTest.aspx



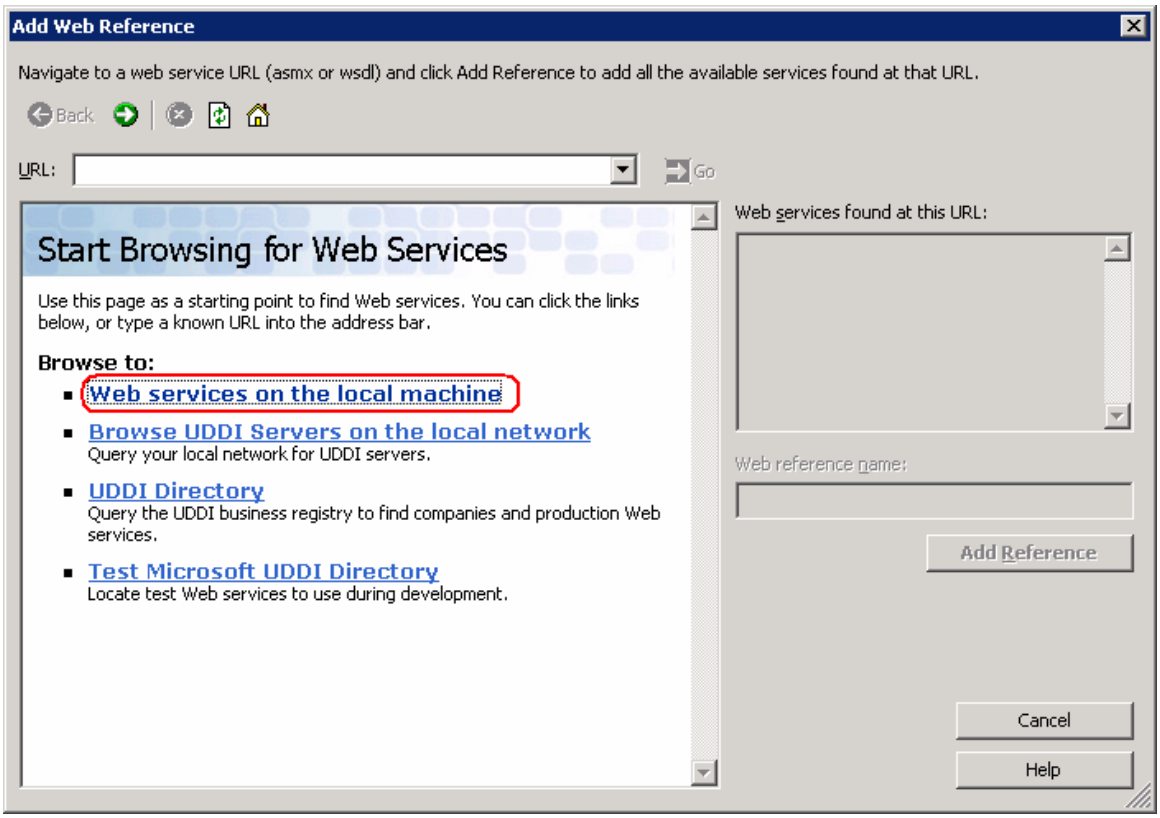
3. Now we will design the front end for the page we created. We shall have three text boxes and labels. Two for input and one for the out put. Add a button also.
[Note: you will find these controls in the toolbox.]



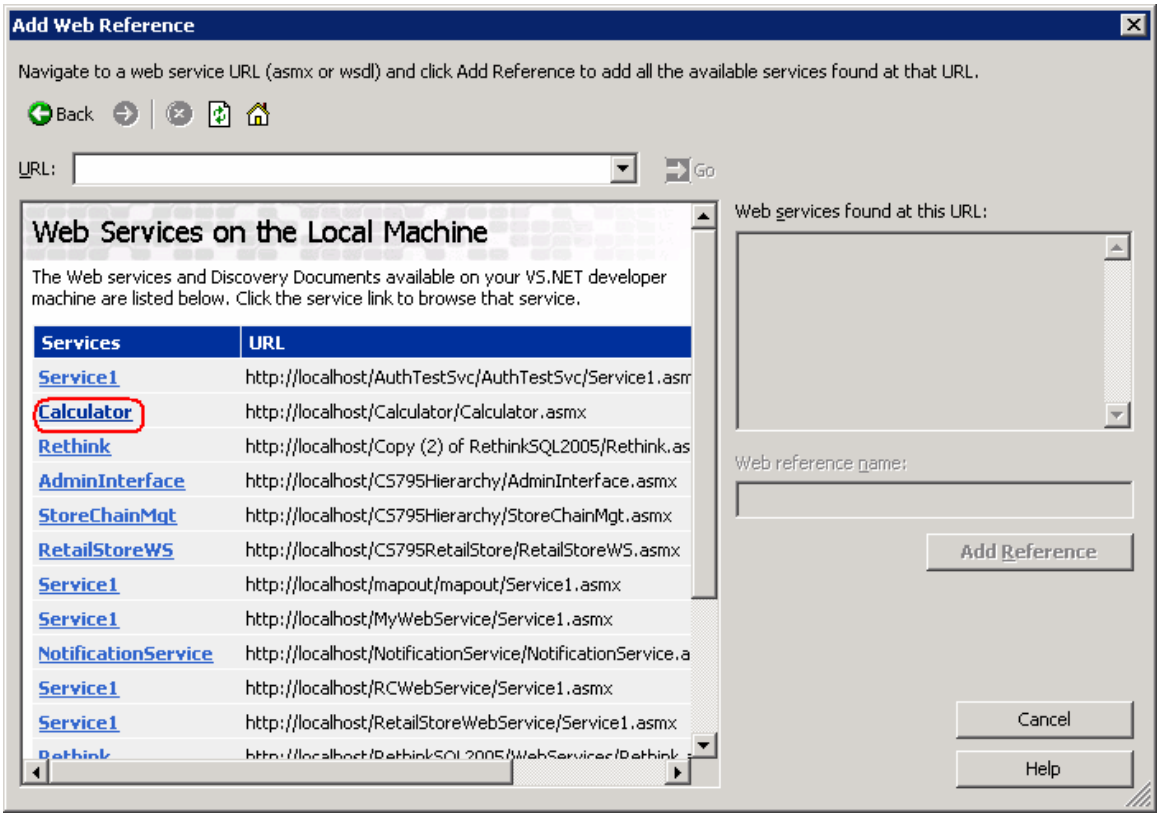
4. Adding the web service as a web service to the client: Right click on references and select Add Web Reference as shown



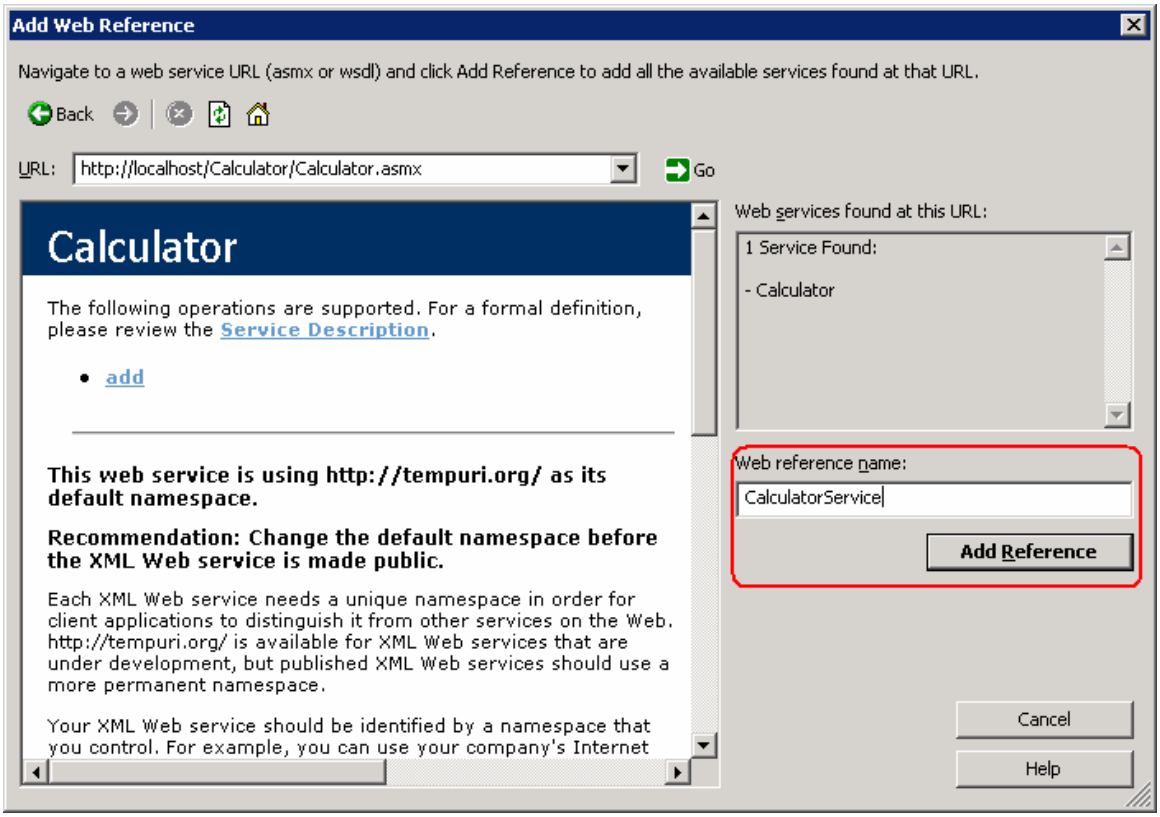
Select **Web services on the local machine** option.



Select Calculator service that we just created as shown



On the next screen, rename the web reference name to CalculatorService and click on Add Reference Button as shown



Now you can see the calculator service with in CalculatorService in your application. Also you can see in WebReference with in your Solution Explorer.

You can see the client side stub for the web service by Right clicking on the service and selecting View in Object Browser option. This is called as a proxy class.

5. Using the Web service.

Now we will add event handler for the button click event for calculate button. To do that just double-click on the button. The action would take you directly to the even handler function.

Just add the following code to the event handler.

```
private void Button1_Click(object sender, System.EventArgs e)
{
      CalculatorService.Calculator calcService = new
CalculatorService.Calculator();
      int param1 = Convert.ToInt32(TextBox1.Text);
      int param2 = Convert.ToInt32(TextBox2.Text);
      int result = calcService.add(param1, param2);
      TextBox3.Text = Convert.ToString(result);
}
```

6. Now run the application and you can test the service we just wrote.