

Accessing Natural from Visual Basic

Contents

| | |
|-------------------------------------------------|----|
| 1. Introduction | 2 |
| 2. Prerequisites | 2 |
| 3. Visual Basic..... | 2 |
| 4. Natural | 2 |
| 5. Discovery..... | 2 |
| 6. Accessing Web Service with Visual Basic..... | 7 |
| 6.1. Initial Setup | 7 |
| 6.2. Designing the Form | 10 |
| 5.3 Writing the Code..... | 14 |
| 5.4 Running the code | 15 |
| 7. Conclusion..... | 15 |

1. Introduction

In this tutorial we will show you how to build a Visual Basic application to access Natural via the SOA Gateway.

2. Prerequisites

As a prerequisite you must install Natural and Visual Basic.

It is also assumed you already have a SOA Gateway server and Control Centre installed. See [here](#) for more info about installing the SOA Gateway.

3. Visual Basic

To build and run Visual Basic applications, you will need a Visual Studio IDE. If you do not already have a Visual Basic IDE installed, we recommend using the Microsoft Visual Studio Express range of products. They can be downloaded freely from Microsoft website, packaged for a number of languages, including Visual Basic. See [here](#) for more information about downloading, installing, and configuring Visual Basic Express.

4. Natural

Natural is a programming language designed to simplify the implementation of business solutions. It takes a very pragmatic and non-theoretical approach to common programming tasks such as database access.

Natural is similar to JAVA in that it is a semi-interpreted language; Natural source programs are “compiled” into a platform independent byte code (object) format which is then executed (interpreted) by a platform dependent runtime environment. While originally developed for mainframe computers, Natural has been available on Windows and UNIX platforms for several years now.

This tutorial is based on the Natural Community Edition which can be downloaded from [Natural download site](#).

When Natural is installed successfully load the objects used for this tutorial. Details on this can be found [here](#). The sample objects are suitable for a Windows/Unix/Linux environment.

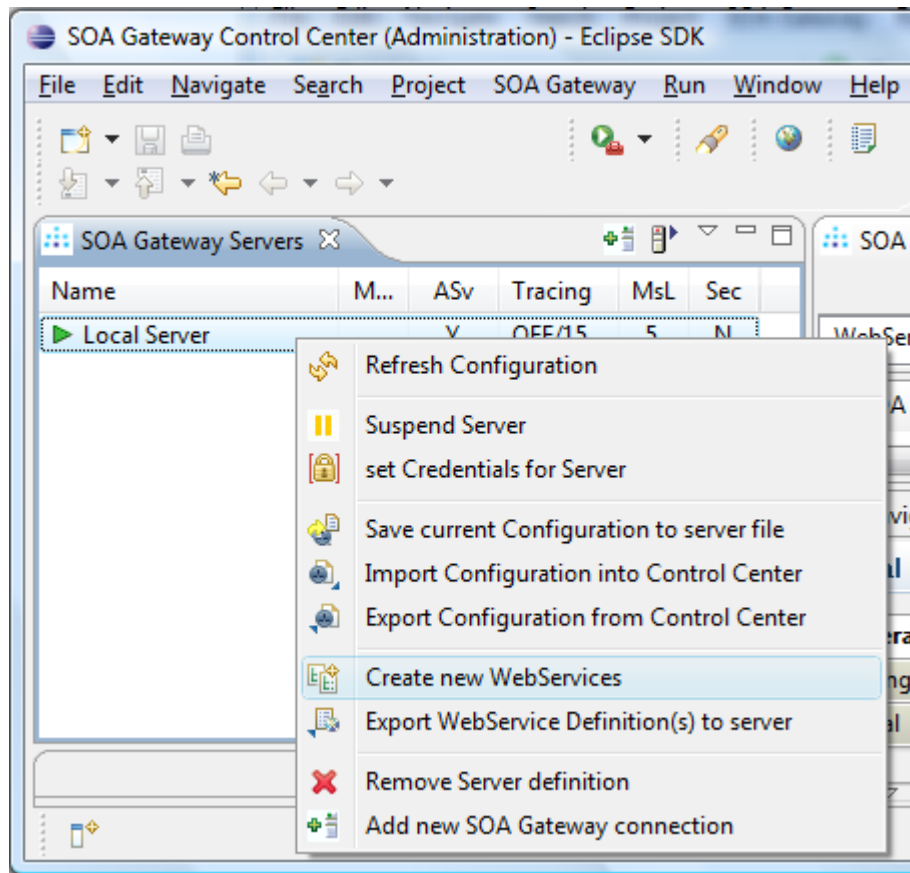
5. Discovery

In this section we’ll show you how to create a web services for one of the programs. This web service can be used by Visual Basic (and many others) to give you direct real-time access to your Natural program.

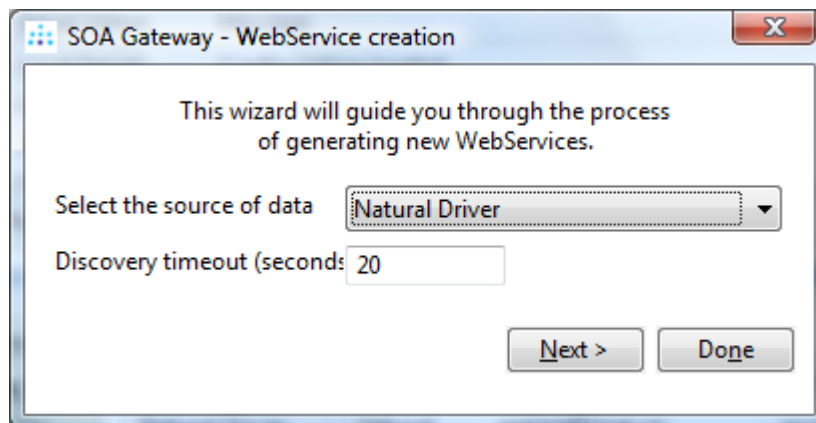
Web Service Creation using SOA Gateway

Start your SOA Gateway Control Centre. See [here](#) for an introduction to the Control Centre.

In your servers view, right click the entry which represents your local SOA Gateway Server. Select “Create new WebServices”.

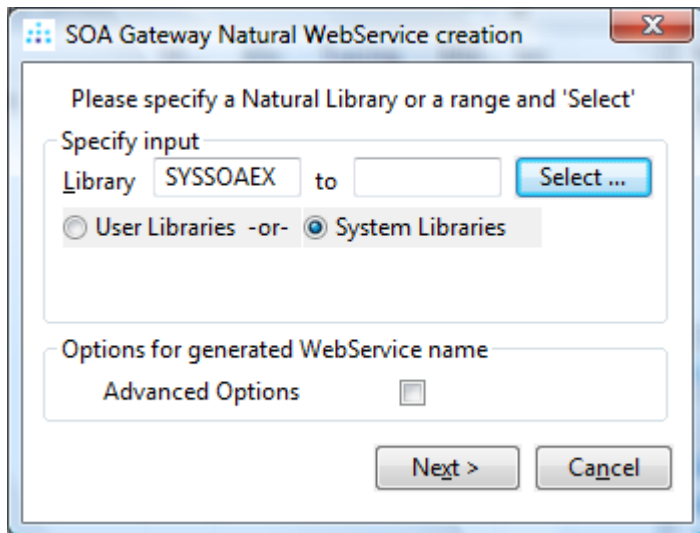


From the next dialog, choose “Natural Driver”. If you do not see have a Natural Driver in the list, see how to create one [here](#).

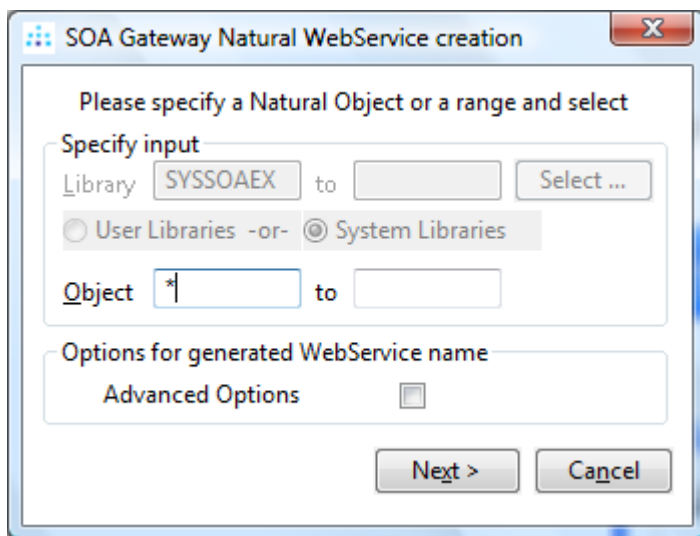


Click Next.

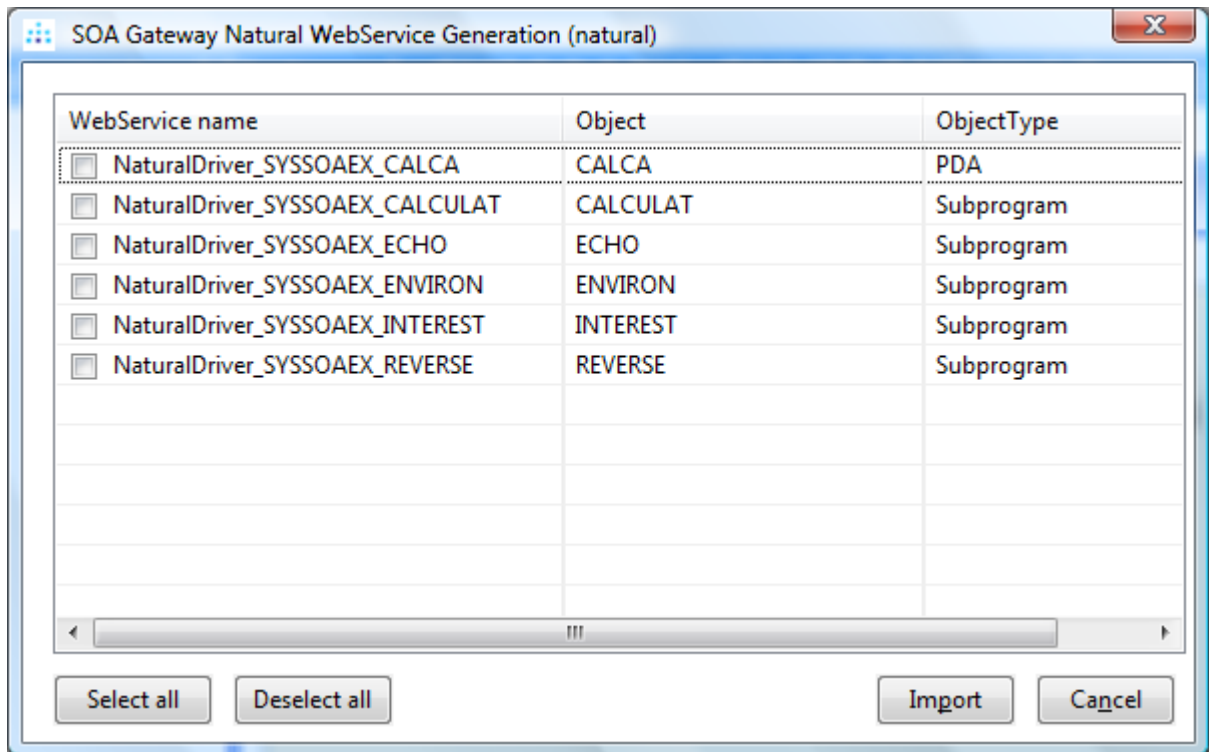
Enter “SYSSOAEX” for the Library name, select “System Libraries”, click “Next >”



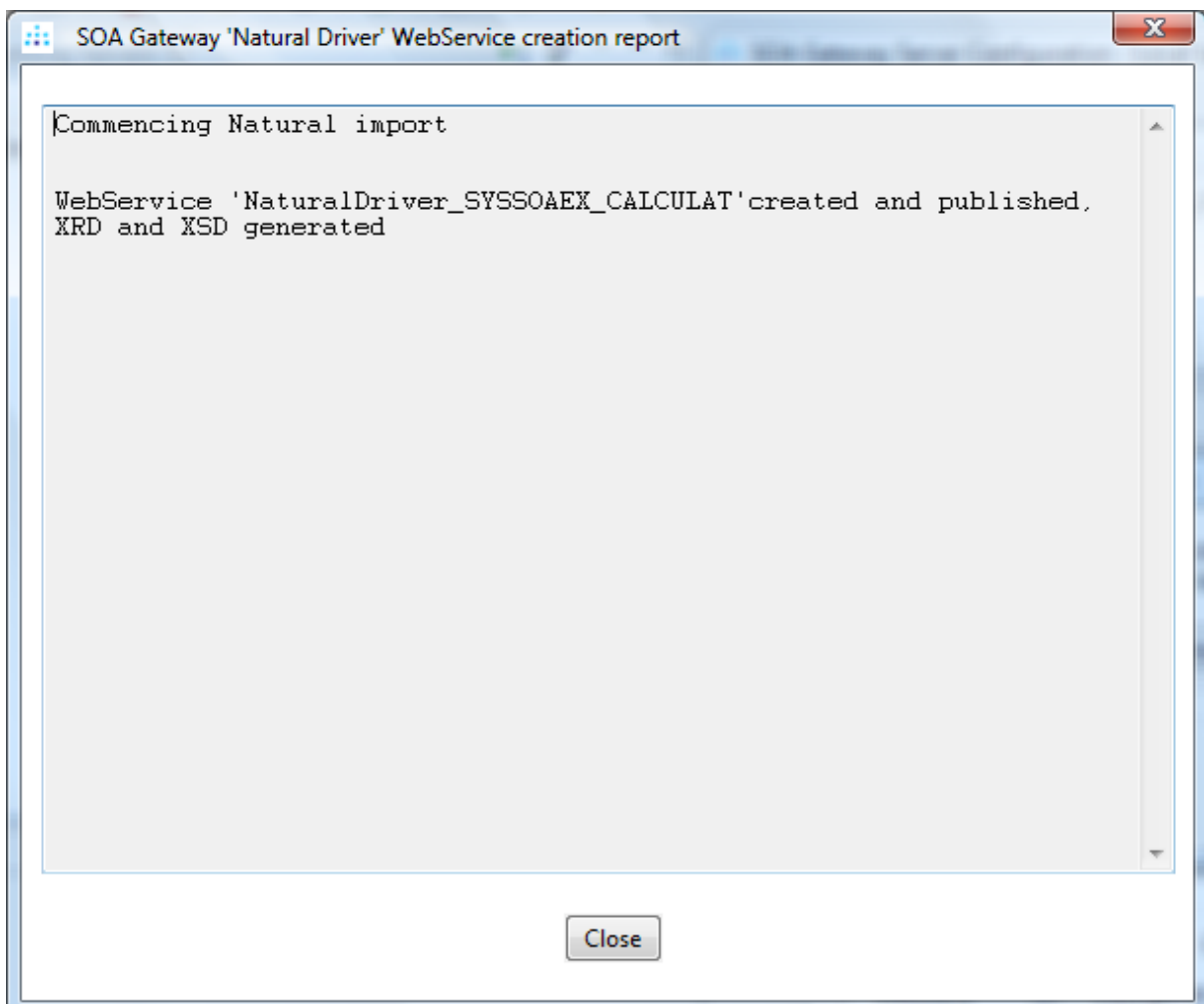
Put an asterisk (*) in the “Object” field and click “Next >”



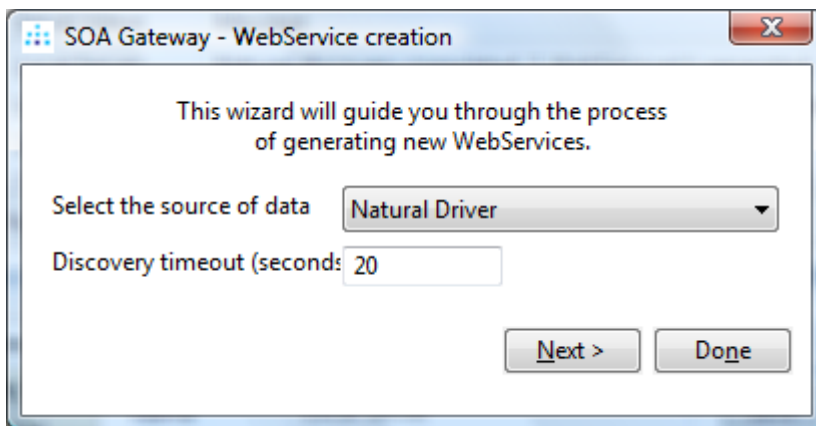
A list of possible WebServices is presented:



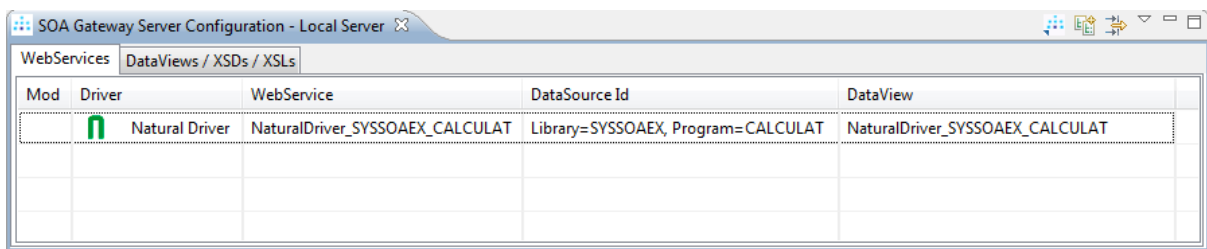
Select the “Natural_SYSSOAEX_CALCULAT” WebService and click “Import”.



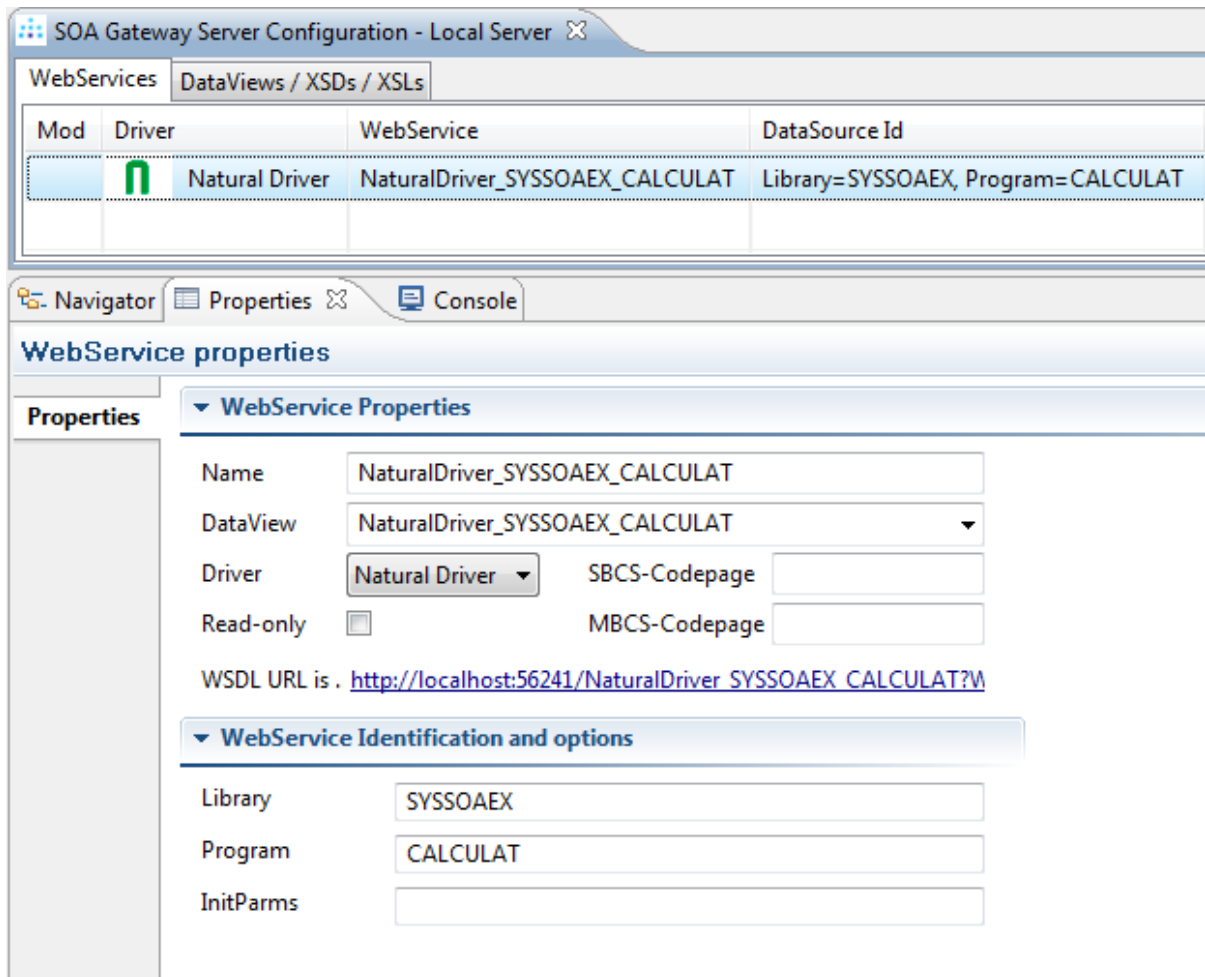
Select Close.



Select Done.



Click the WebService in the Configuration View. The Properties View displays the details.



Web Service Description Language (WSDL) is a standard, XML-based language that is used to describe a Web Service. The WSDL itself is usually interpreted by a web service client, such as Visual Basic.

As WSDL is XML-based, it will open in your browser of choice. To see the WSDL click on the link at 'WSDL URL is' in the WebService Properties window.

This WSDL is the starting point for using Web Services, and can be used time and again by different web service clients.

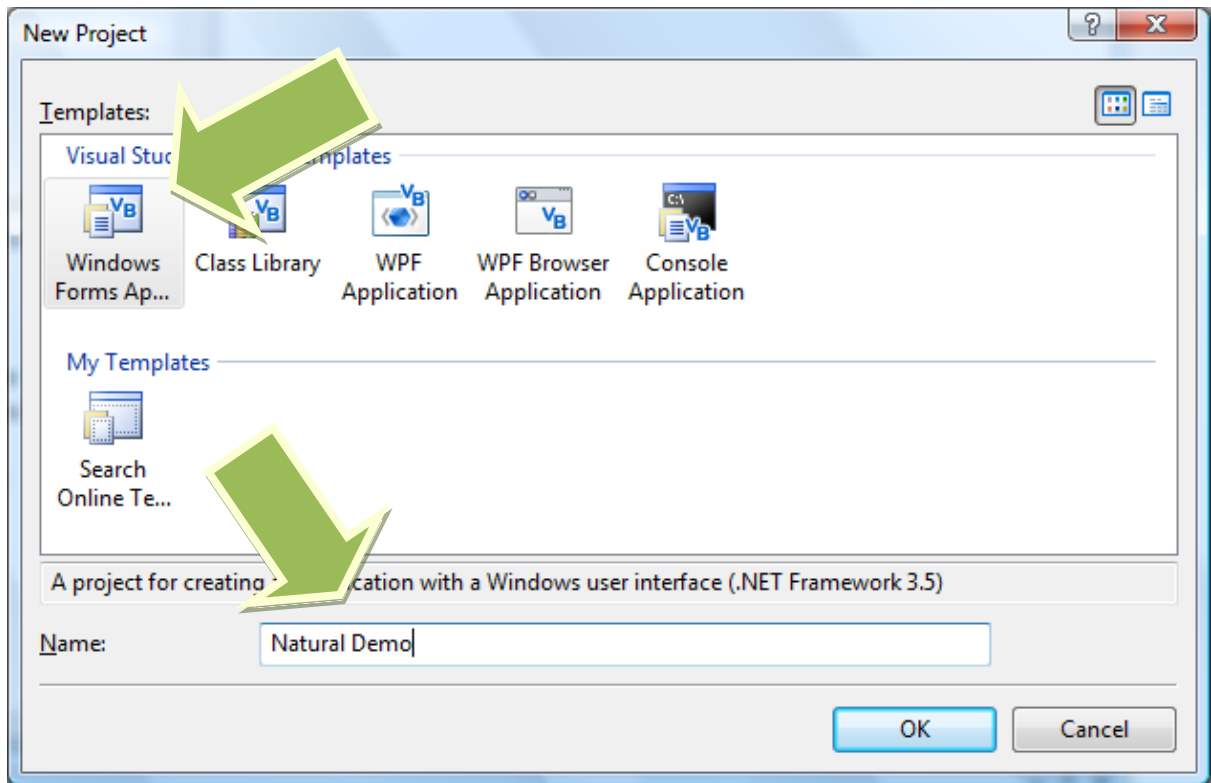
6. Accessing Web Service with Visual Basic

We will use Visual Basic to build an application which accesses our Natural program Web Service via the WSDL.

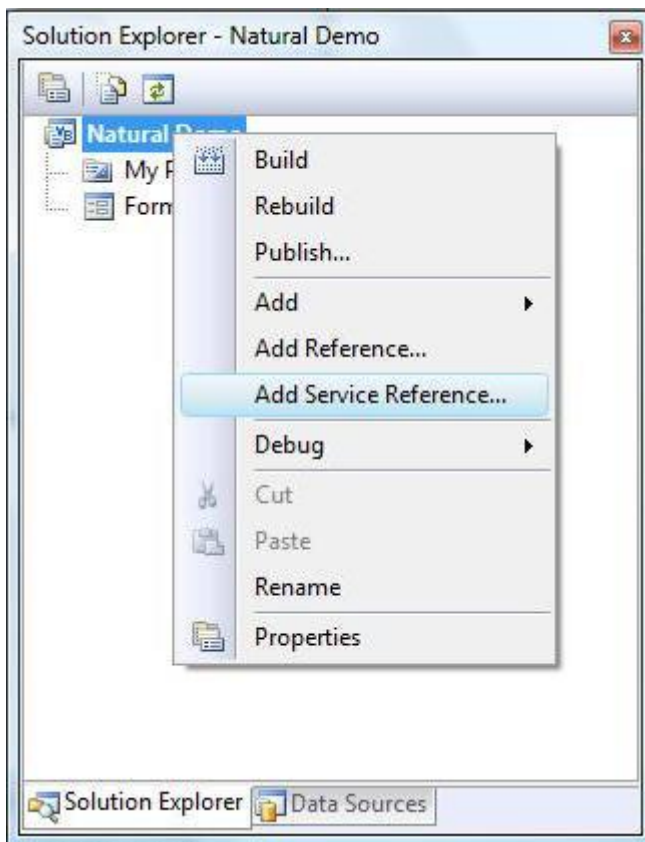
6.1. Initial Setup

Start *Microsoft Visual Basic Express*. The version we are using here is Visual C# 2008. Open the File menu item and select New Project...

In the New Project dialog (see below) select the Windows Forms Application template and name the project **Natural Demo**. Hit OK.

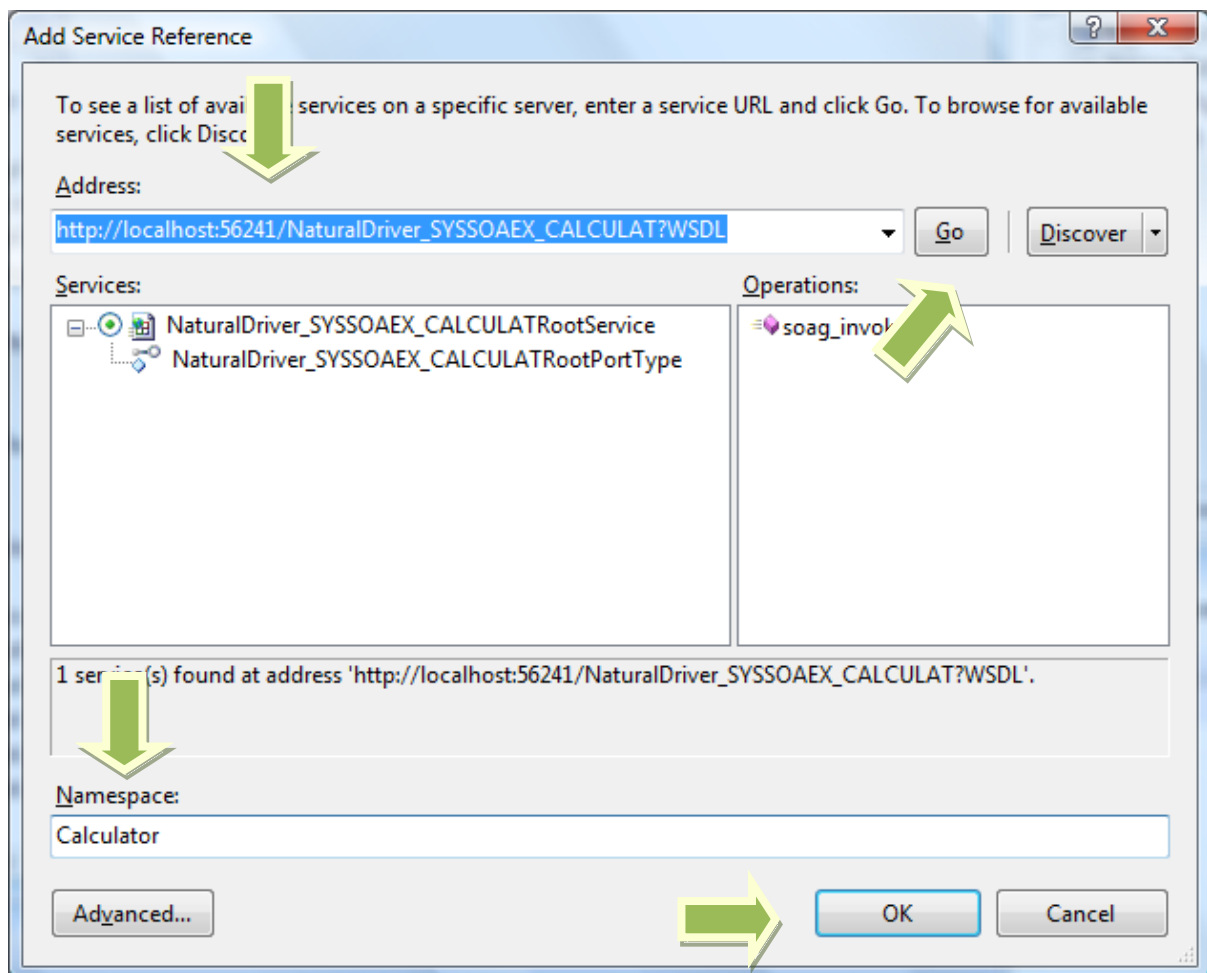


In the Solution Explorer right-click your project name and select “Add Service Reference”

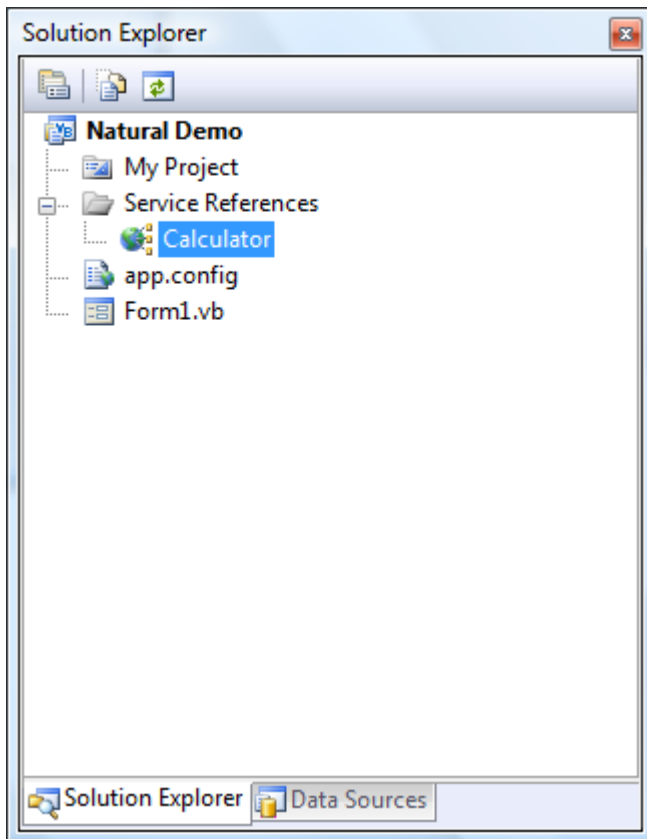


- Copy the URL of the web service created earlier into the Address text box

- Click Go.
- Once the WSDL has been loaded, change the Namespace to **Calculator**
- Click **OK**



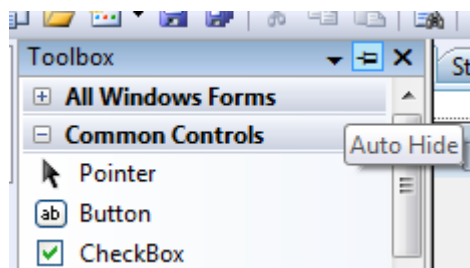
You should now have the Calculator Service Reference loaded into your Solution Explorer.



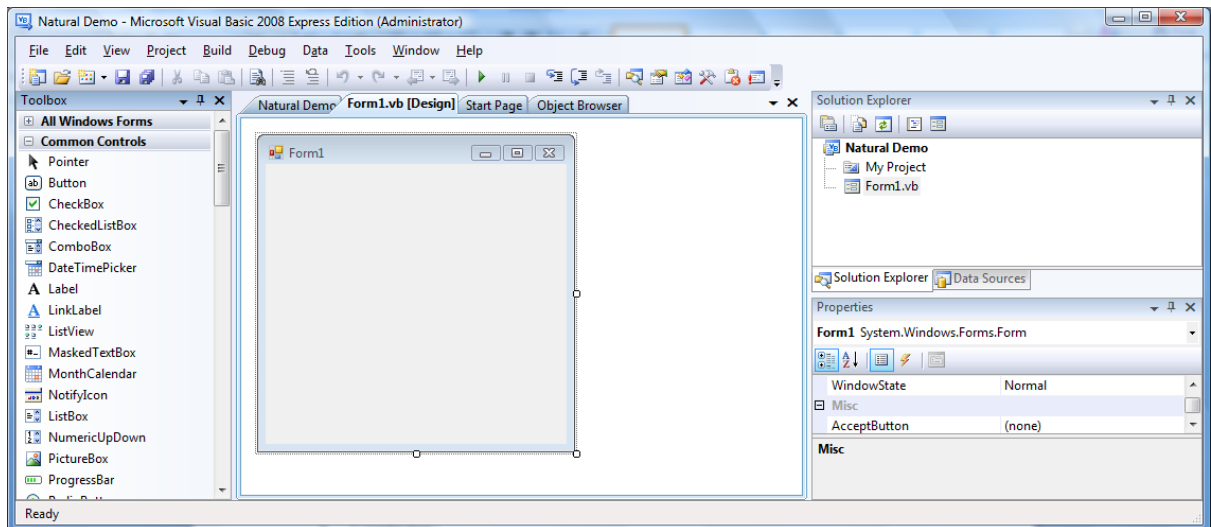
6.2.Designing the Form

In this section we'll add the necessary controls to our Form.

If your Form is not visible double-click on Form1.vb in the Solution Explorer. You may be looking at the Code View so ensure that you are in the Designer View (View -> Designer), and that you have the control Toolbox available (View->Toolbox). If you wish to dock the Toolbox while you design the Form click on the Auto Hide icon in the Toolbox menu bar.

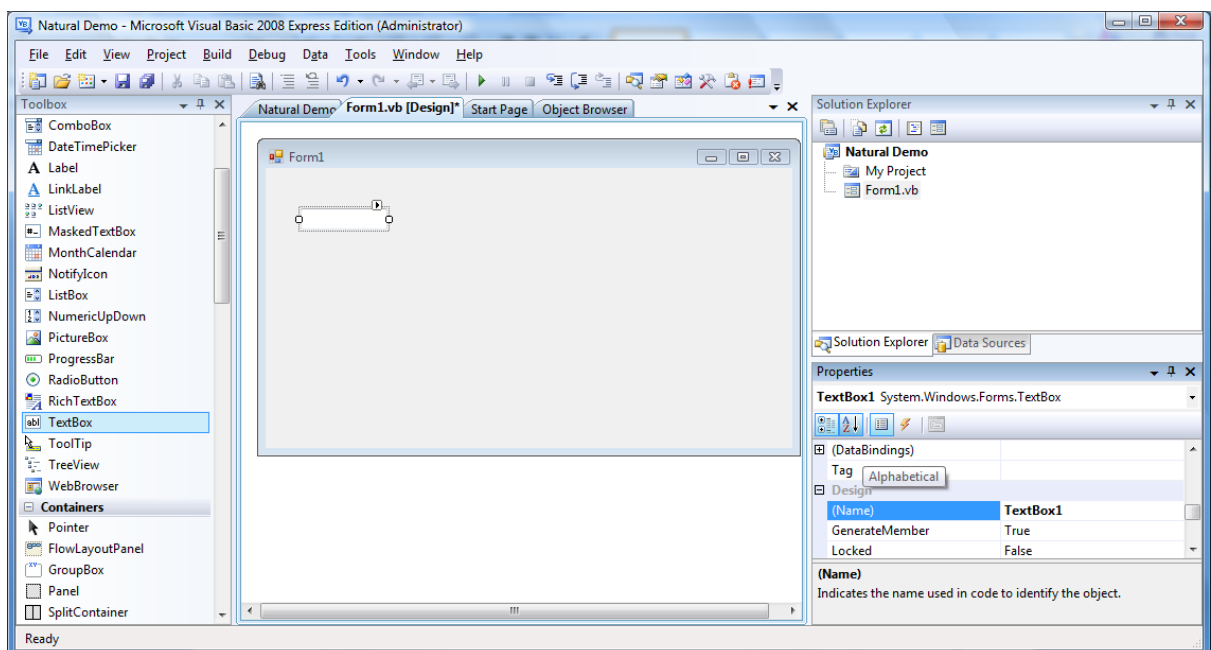


At this point you should be looking at a Form as shown below. Resize it by dragging on the small squares at the right and bottom of the Form. Go to the Properties Window for the Form (F4) and set the Text property to Natural Calculator Demo.



The Calculator takes 2 input parameters, operand1 and operand2. The code as given expects operand1 to be entered into TextBox1 and operand2 into TextBox2. The result is placed in TextBox3.

To add TextBox1 select the TextBox control from the Toolbox list of controls. Move your mouse over onto Form1 and combining a click and drag action size it to you needs. If you select the new TextBox as shown and hit F4 (View -> Properties Window) it displays its properties in the bottom right window. Note that it has automatically named it TextBox1. You may have to scroll down to the Name property to see it.



Repeat for TextBox2 and TextBox3. For TextBox3 I have also set some additional properties. Click on TextBox3 and hit F4 (or View -> Properties Window). Set the TabStop property to False and the ReadOnly property to True.

Now add 3 Label controls and place each over the text boxes. For each go to the Properties Window (F4) and set the Text property of Label1 to Operand 1, the Text property of Label2 to

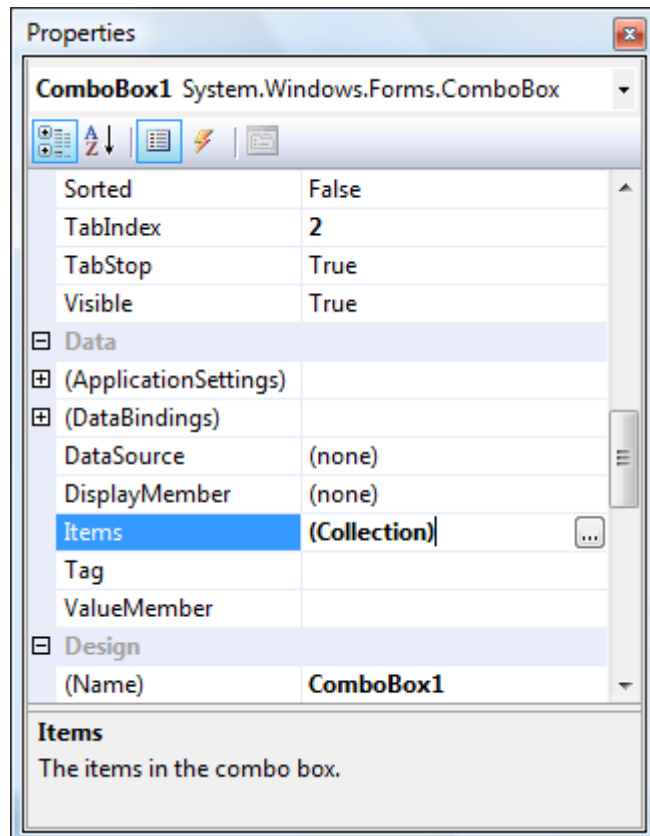
Operand 2 and the Text property of Label3 to Result.

Now add a ComboBox control, placing it between TextBox1 and TextBox2.

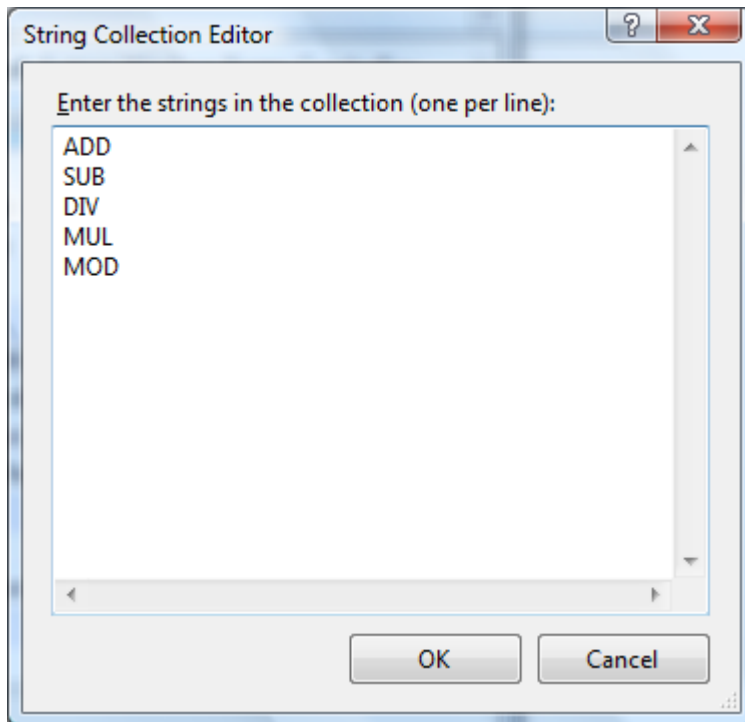
See Figure 1 for example layout.

The combo box is the only control that needs additional setup.

When you add this control, select it on the Form and go to the Properties Window (F 4). Scroll down to the Items property as shown below.



Click on Collection and add the methods for the Natural Program e.g. ADD, MUL, DIV, SUB and MOD. Hit OK.



Finally add a Button control. Go to the Properties Window (F4) and set the Text property to Calculate.

In summary I've used the following controls in this Form and the result is shown in Figure 1.

- Form1 (Text property set to Natural Calculator Demo)
- Label1 (Text property set to Operand 1)
- Label2 (Text property set to Operand 2)
- Label3 (Text property set to Result)
- TextBox1
- TextBox2
- TextBox3 (ReadOnly property = True and TabStop property = False)
- ComboBox1
- Button1 (Text property set to Calculate)

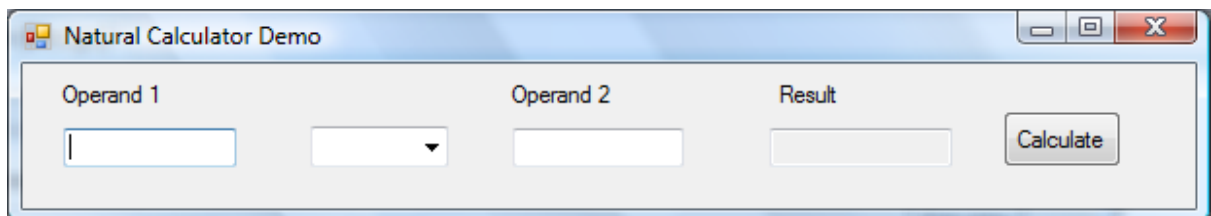


Figure 1

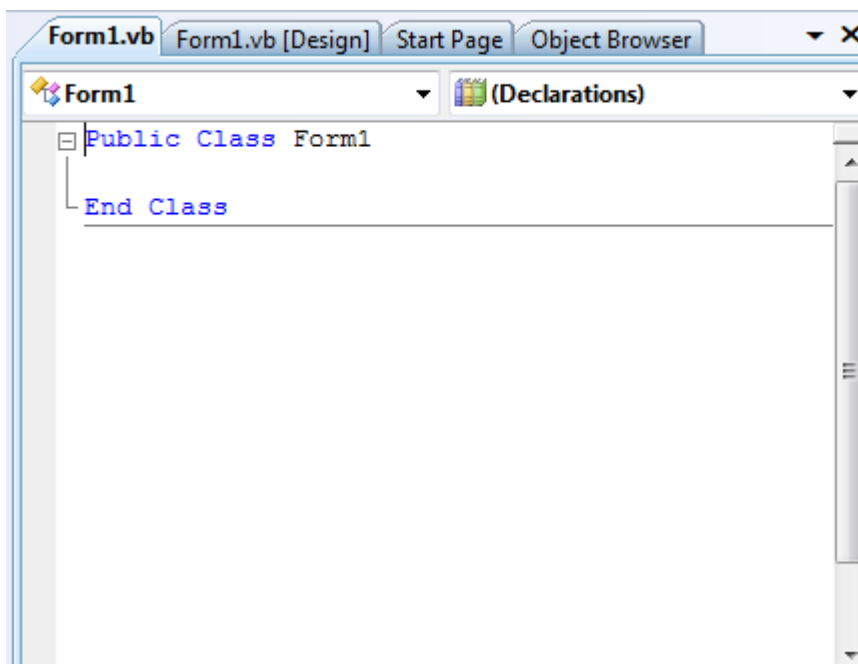
Note that I haven't changed any of the default design names that the Visual Basic designer has given me.

You may change these to whatever you wish, but be aware your Visual Basic code in the next section will have to be cognisant of this!

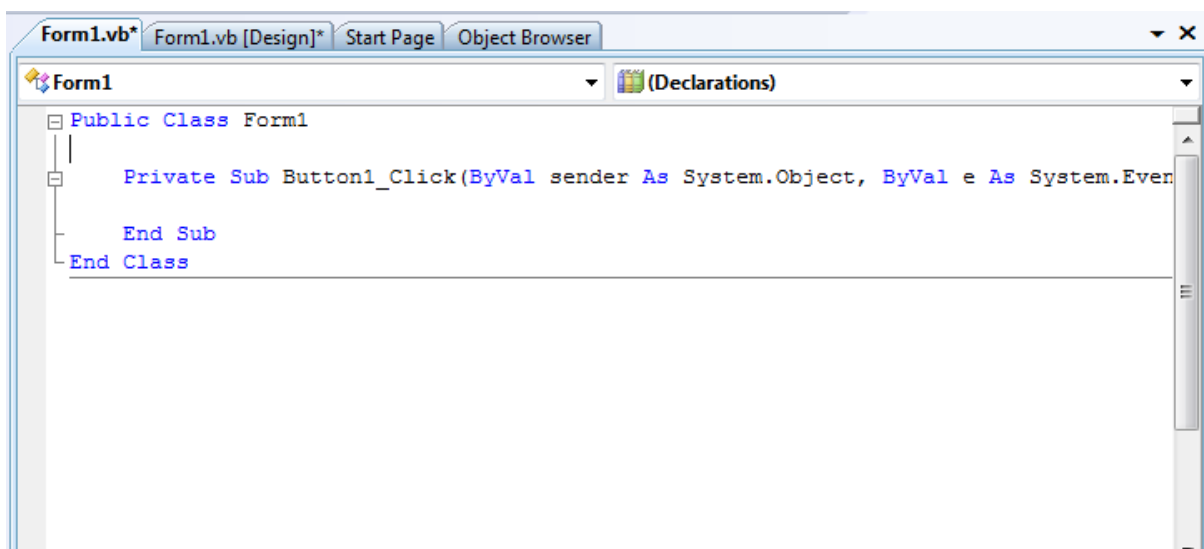
5.3 Writing the Code

Now that the Form controls have been added, we need to write the code to call our Web Service when the Calculate button is clicked.

Switch to your Code view by right-clicking on Form1.vb in the Solution Explorer window and select View->Code. This is what you should see:



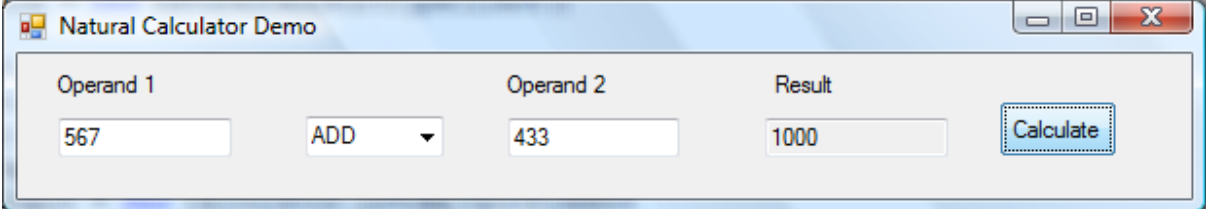
We need to add a handler for when the Calculate button (Button1) is clicked. Switch back to Designer Mode by right-clicking on Form1.vb and selecting View Designer. Select the Calculate button on the form and double-click it. You should now see the following:



At this point you may replace ALL of the code with the code [available here](#).

5.4 Running the code

By hitting F5 or Debug -> Start Debugging, you can run your code and the form is displayed as shown. Note that we have just rudimentary error checking so leaving a textbox empty may cause problems.



| Operand 1 | | Operand 2 | Result | |
|-----------|-----|-----------|--------|-----------|
| 567 | ADD | 433 | 1000 | Calculate |

If you hit problems, you may wish to debug your code by adding breakpoints in your code. See the IDE documentation for further information.

7. Conclusion

This tutorial shows how to access a Natural Program from Visual Basic using the SOA Gateway. As you can see, you have built a powerful application that uses Web Services to retrieve information in real-time.