Microsoft .NET 2.0 Web Service and Web Service Client Tutorial

Matthew Kubicina
CIS 764 – Database Design
Kansas State University

Requirements

a. Microsoft .NET Framework 2.0
b. Microsoft Visual Studio 2005

Outline

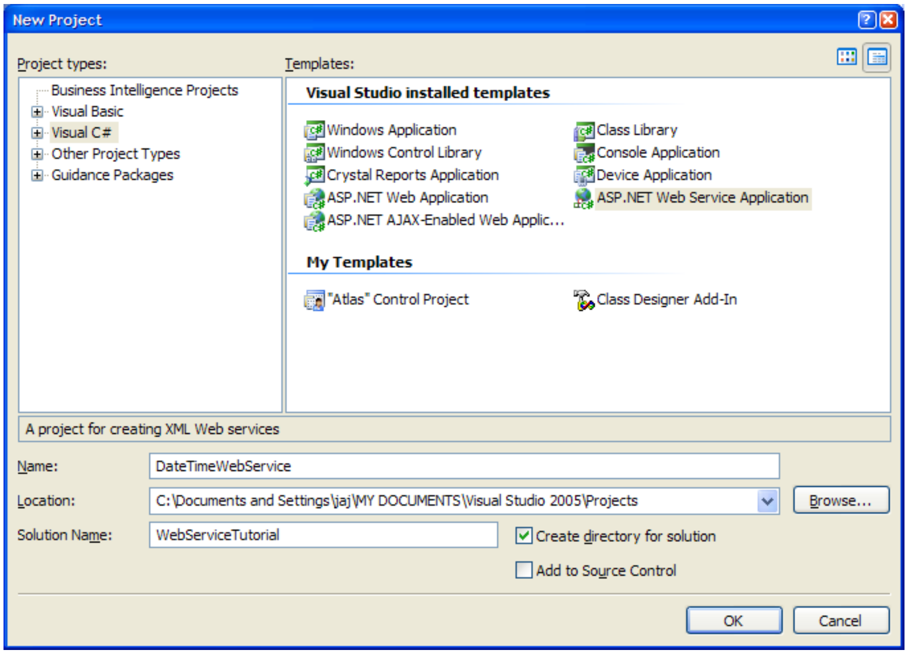
Part 1 – Create the Visual Studio Solution

a. Open Visual Studio 2005.

b. Go to the “File” menu and then select “New Project”.

c. In the project type listing, click on “Visual C#”. In the templates section, select “ASP.NET Web Service Application”. In the project settings, fill in the following properties and click “OK”.

Name: DateTimeWebService
Location: Default
Solution Name: WebServiceTutorial
Create directory for solution: Checked



A project for creating XML Web services

Name: DateTimeWebService

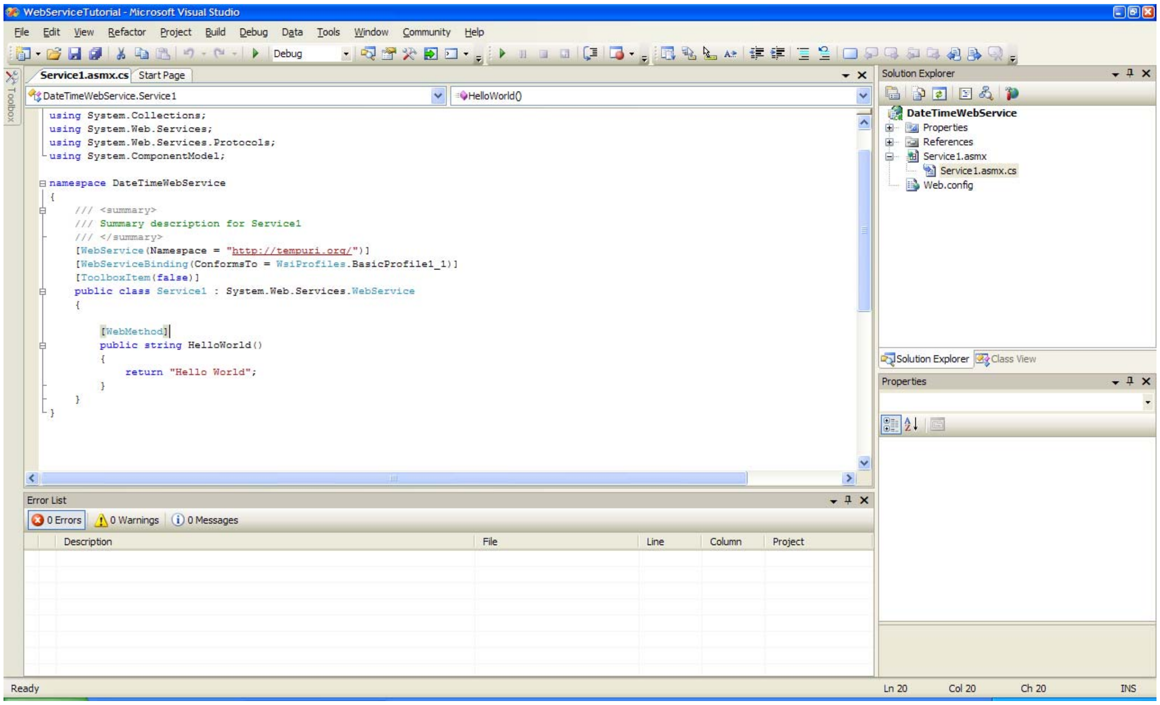
Location: C:\Documents and Settings\jaj\MY DOCUMENTS\Visual Studio 2005\Projects

Solution Name: WebServiceTutorial

☑ Create directory for solution

□ Add to Source Control

d. Visual Studio creates a solution and a web service project. The web service, contained in the “Service1.asmx” file, contains a default “Hello World” method.



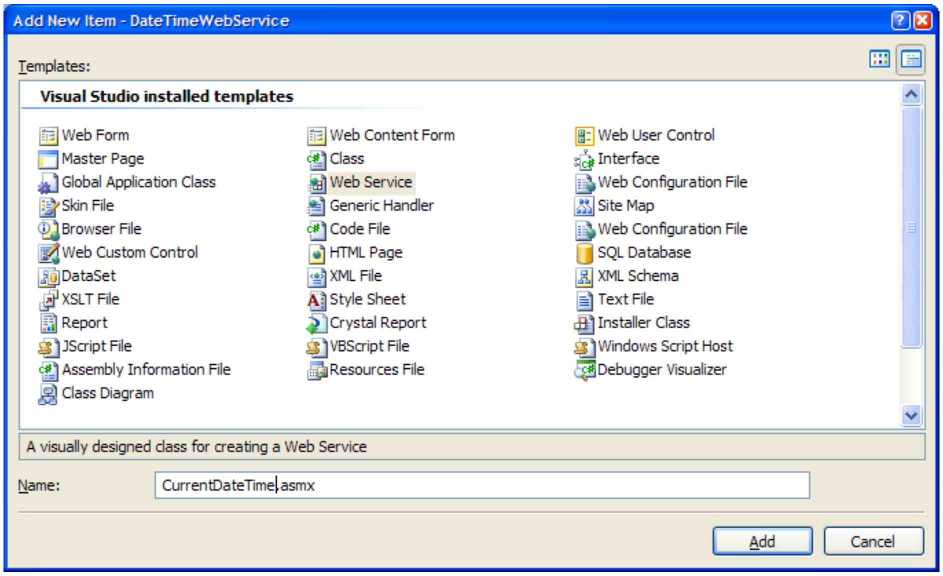
```
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.ComponentModel;

namespace DateTimeWebService
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [ToolboxItem(false)]
    public class Service1 : System.Web.Services.WebService
    {

        [WebMethod]
        public string HelloWorld()
        {
            return "Hello World";
        }
    }
}
```

e. In solution explorer, delete the “Service1.asmx” file to remove the default web service file by right clicking on the file and selecting “Delete”. Click “OK” on the confirmation box that will appear.

Step 2 – Create the Web Service

a. In the solution explorer, right click on the “DateTimeWebService” project node and select “Add”, “New Item”.

b. Select “Web Service” in the list of templates. Name the file “CurrentDateTime.asmx” and click “Add”.



A visually designed class for creating a Web Service

Name: CurrentDateTime.asmx

c. Visual Studio adds the file to the project and opens it. In the file is a default “Hello World” web function. Delete the “HelloWorld” function and the “WebMethod” attribute from the source code in the file.

d. Create a new function in the “CurrentDateTime” class. The function should look like the code snippet below.

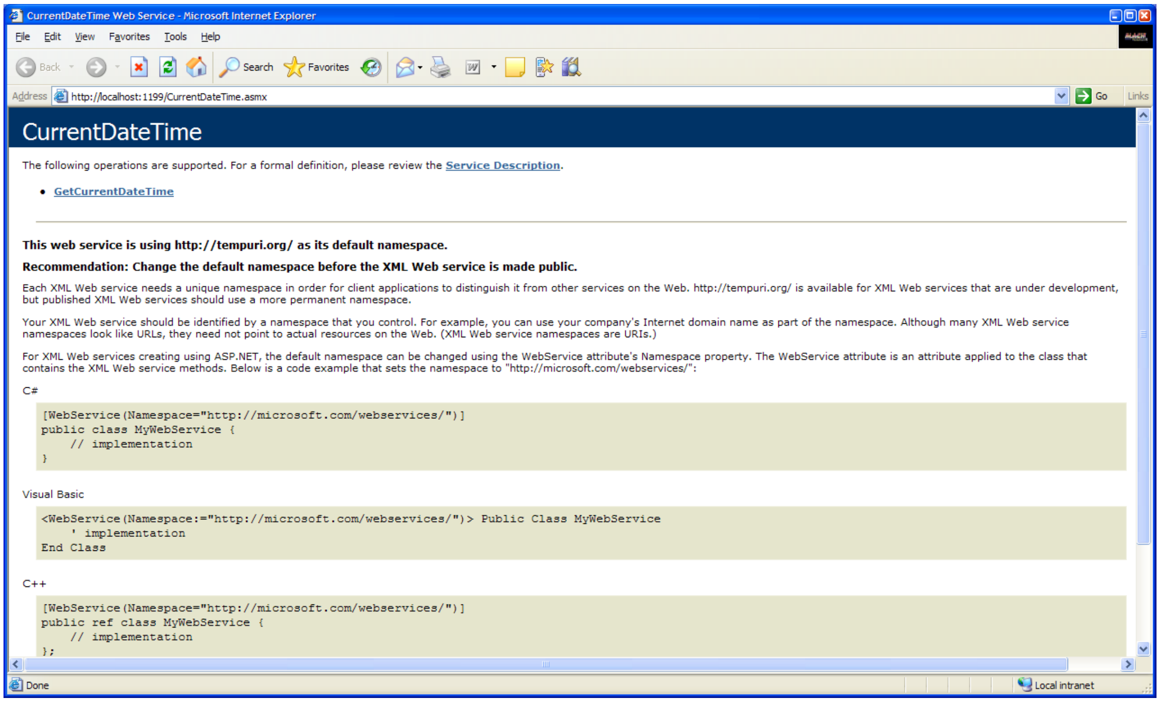
```
public DateTime GetCurrentDateTime()
{
        return DateTime.Now;
}
```

e. To make the newly created function exposed via the web service, add the “WebMethod” attribute above the function definition. The code should look like the following when completed.

```
[WebMethod]
public DateTime GetCurrentDateTime()
{
        return DateTime.Now;
}
```

Part 3 – Testing the Web Service

a. To test the web service function, go to “Debug”, “Start Debugging”. Visual Studio will compile the web service and launch the web service URL automatically.


CurrentDateTime Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:1199/CurrentDateTime.asmx
Go
Links
Done
Local intranet

CurrentDateTime

The following operations are supported. For a formal definition, please review the Service Description.

- GetCurrentDateTime

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

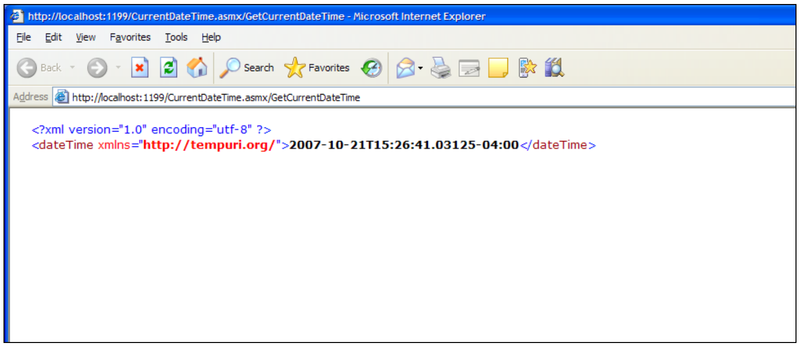
```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

b. Click on the link named “GetCurrentDateTime” to open the test harness for the function we just created. Click the “Invoke” button to execute the function on the web server.

c. A new window will open containing the response from the web server. If the web service is responding correctly, it should return an XML soap message containing the current date/time on the web server.


http://localhost:1199/CurrentDateTime.asmx/GetCurrentDateTime - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:1199/CurrentDateTime.asmx/GetCurrentDateTime

```
<?xml version="1.0" encoding="utf-8" ?>
<dateTime xmlns="http://tempuri.org/">2007-10-21T15:26:41.03125-04:00</dateTime>
```

d. Close the two web browser windows for the web service. This will stop the debugging process and should return you to the Visual Studio editor.

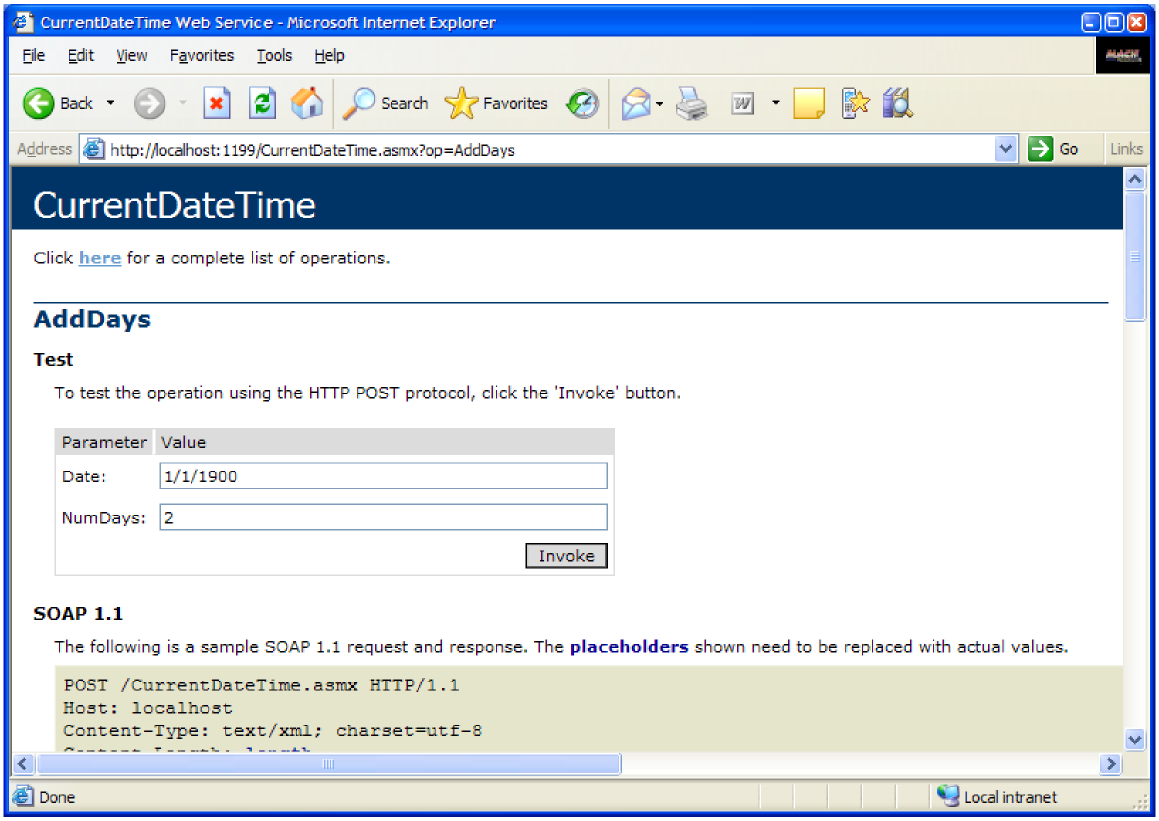
Part 4 – Parameterized Web Methods

a. Add a new web function that accepts two parameters by pasting in the code snippet below. This function accepts a date and an integer representing the number of days to add to the given date instance. The web function will return the result of the calculation.

```
[WebMethod]
public DateTime AddDays(DateTime Date, int NumDays)
{
        return Date.AddDays(NumDays);
}
```

b. To test the new web service function, go to “Debug”, “Start Debugging”. Visual Studio will compile the web service and launch the web service automatically.

c. Click on the link named “AddDays” to open the test harness for the function we just created.


CurrentDateTime Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Invoke
Done
Local intranet

http://localhost:1199/CurrentDateTime.asmx?op=AddDays

CurrentDateTime

Click here for a complete list of operations.

AddDays

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
Date:	1/1/1900
NumDays:	2

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /CurrentDateTime.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
```

d. In the test harness, enter some values for the two available parameters. After clicking the “Invoke” button, the web service should return the value of the calculation.

e. Close the two web browser windows for the web service. This will stop the debugging process and should return you to the Visual Studio editor.

Part 5 – Create the Client Application

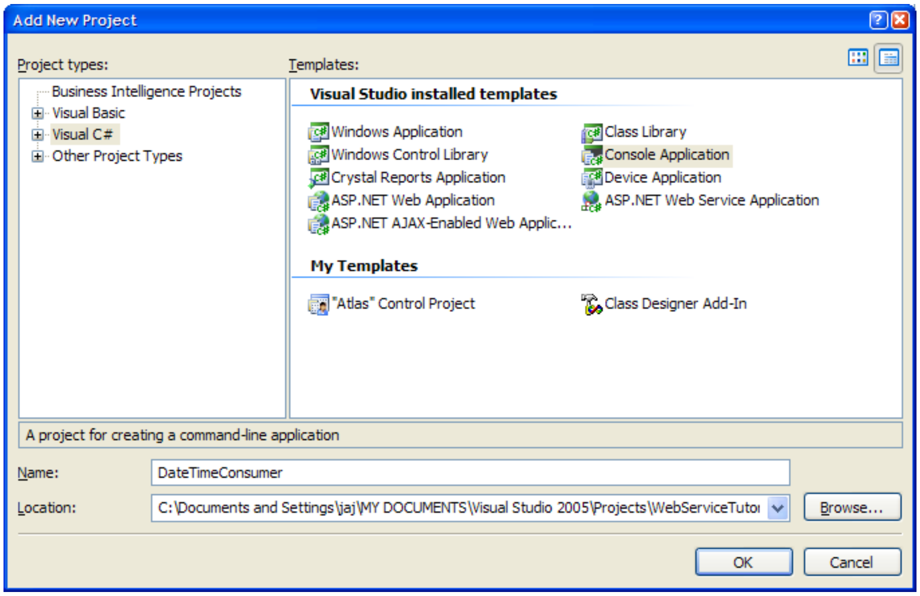
a. To create the client application project, go to "File", "Add", "New Project".

b. In the project type list, click on "Visual C#".

c. In the templates section, select "Console Application".

d. In the project settings, fill in the following properties and click "OK".

Name: DateTimeConsumer
Location: Default

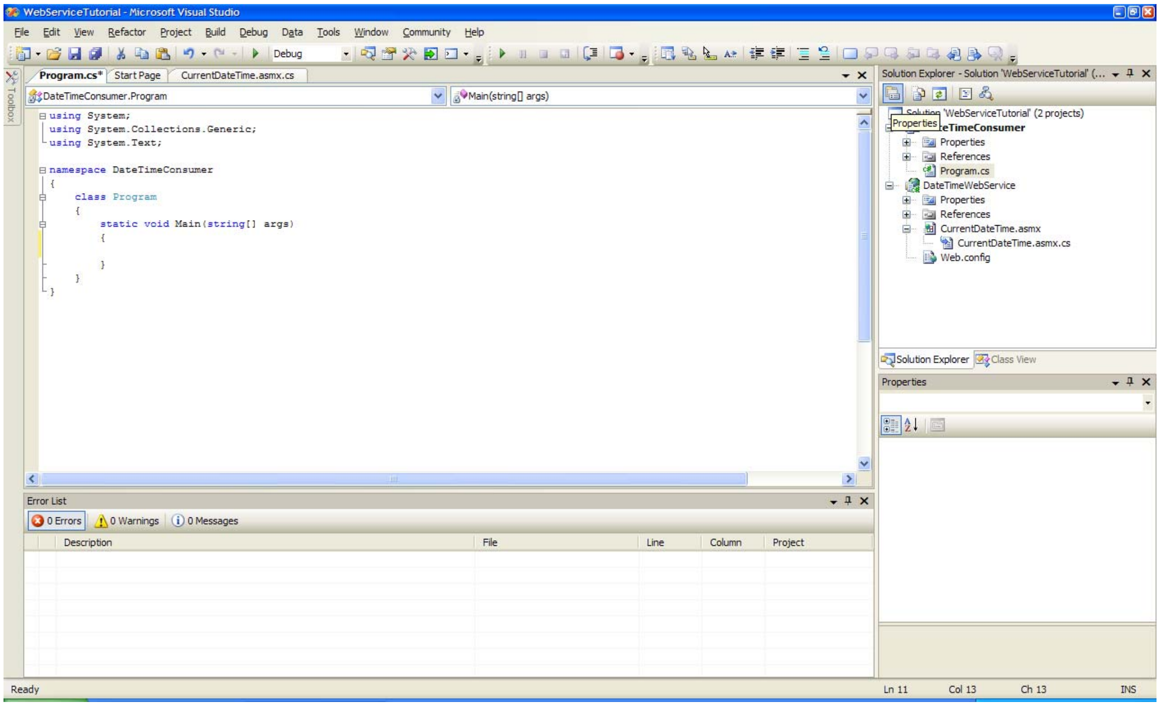


A project for creating a command-line application

Name: DateTimeConsumer

Location: C:\Documents and Settings\jaj\MY DOCUMENTS\Visual Studio 2005\Projects\WebServiceTutoı

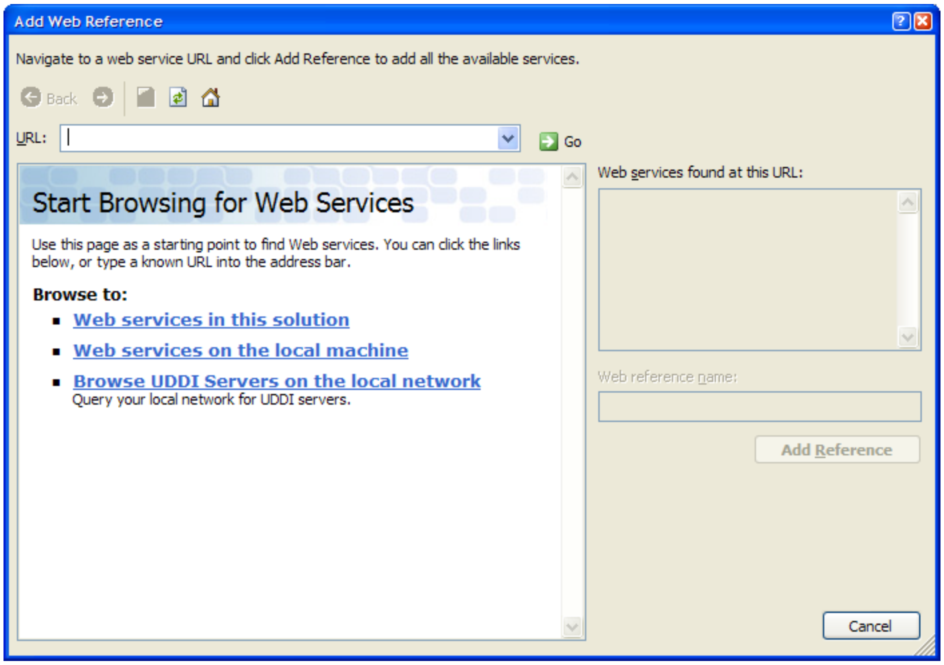
e. Visual Studio adds the new project to the existing solution and opens up the "Main" static class.



```
using System;
using System.Collections.Generic;
using System.Text;

namespace DateTimeConsumer
{
    class Program
    {
        static void Main(string[] args)
        {

        }
    }
}
```

Part 6 – Add the Web Reference

a. Add a reference to the existing web service by right clicking on the "DateTimeConsumer" project and selecting "Add Web Reference".



Navigate to a web service URL and click Add Reference to add all the available services.

Start Browsing for Web Services

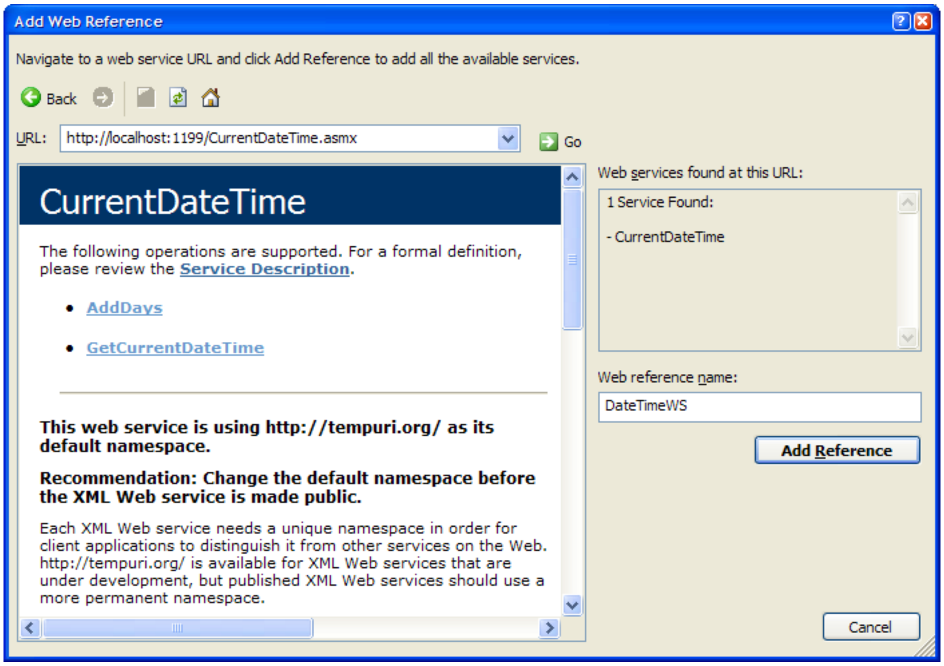
Use this page as a starting point to find Web services. You can click the links below, or type a known URL into the address bar.

Browse to:

- Web services in this solution
- Web services on the local machine
- Browse UDDI Servers on the local network
  Query your local network for UDDI servers.

b. Select "Web Services in this solution". In the list of web services, select "CurrentDateTime"

c. In the "Web Reference Name" property, enter "DateTimeWS" and hit "Add Reference". Visual Studio automatically adds the client classes for the web service to the project.



Navigate to a web service URL and click Add Reference to add all the available services.

URL: http://localhost:1199/CurrentDateTime.asmx

CurrentDateTime

The following operations are supported. For a formal definition, please review the Service Description.

- AddDays
- GetCurrentDateTime

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

1 Service Found:

- CurrentDateTime

Web reference name: DateTimeWS

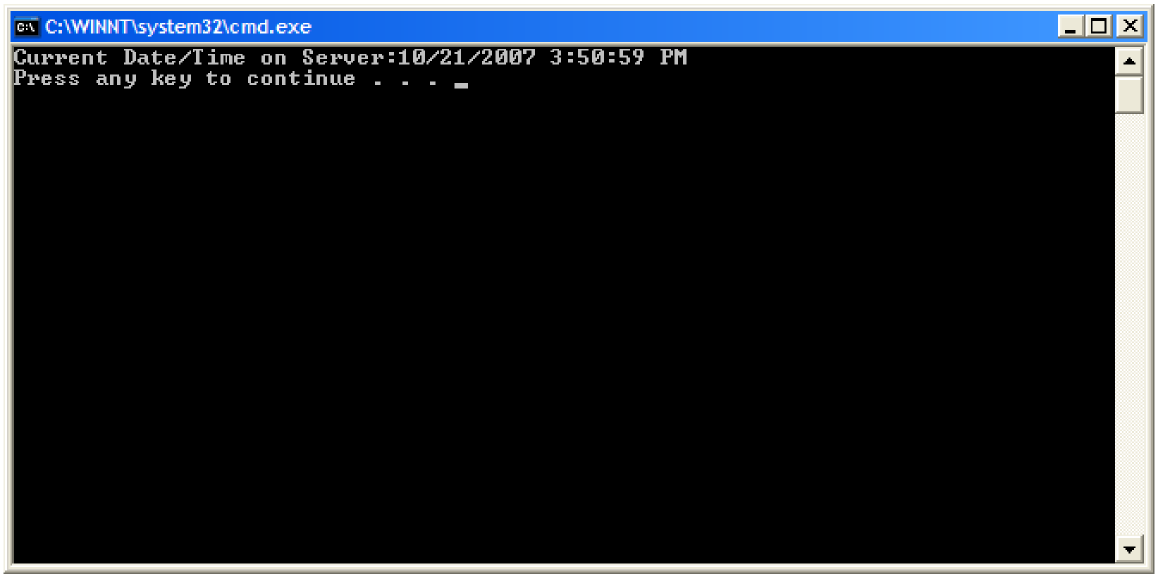
Part 7 – Consuming the Web Service

a. In the “Main” function of the program class, add the following code snippet to represent the web service client. This section of code retrieves the current date time from the web service and displays it to the console for the user.

```
static void Main(string[] args)
{
        DateTimeWS.CurrentDateTime Client = new
                DateTimeConsumer.DateTimeWS.CurrentDateTime();
        DateTime ServerDateTime = Client.GetCurrentDateTime();

        Console.WriteLine("Current Date/Time on Server:" +
               ServerDateTime.ToString());
}
```

Part 8 – Testing the Client Application

a. To test the client program, go to “Debug”, “Start Debugging”. This will automatically start the web service on the web server and begin executing the client application. The program should display the current date/time from the server. The window will automatically close when debugging ends. In order to verify output, utilize the “Start Without Debugging” option and the command prompt will remain open upon termination.



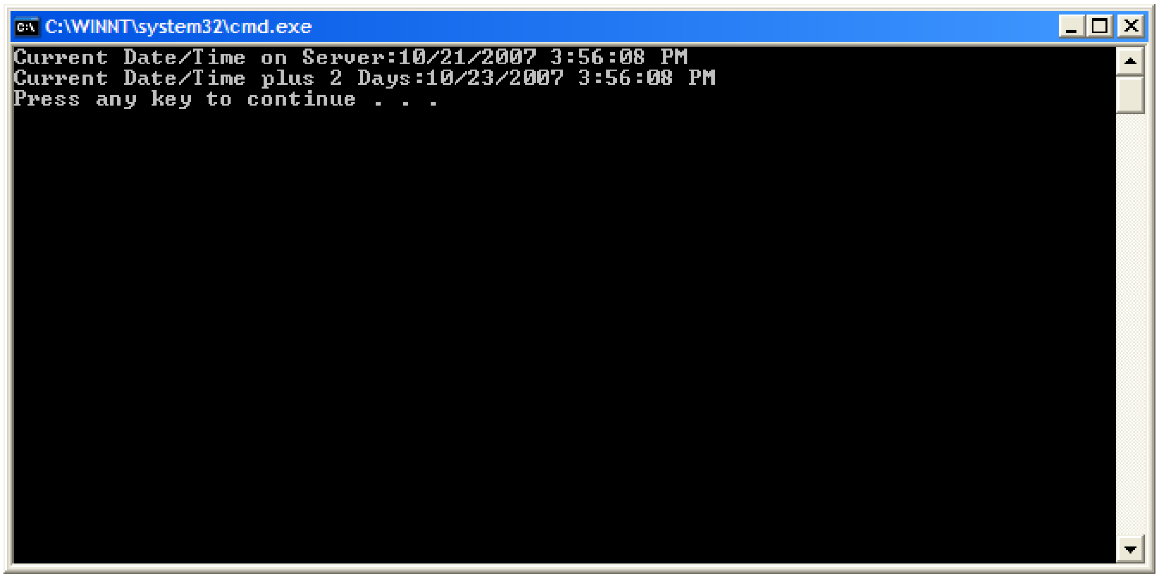
```
Current Date/Time on Server:10/21/2007 3:50:59 PM
Press any key to continue . . . _
```

Part 9 – Passing Parameters to the Web Service

a. To consume the “AddDays” function, paste the following code snippet into the “Main” function below the existing code. This code snippet sends the current date/time on the client machine to the server and the web services adds the number of days specified to the provided date. In this instance, it adds two days to the provided time.

```
DateTime AddedDays = Client.AddDays(DateTime.Now, 2);

Console.WriteLine("Current Date/Time plus 2 Days:" + AddedDays.ToString());
```

b. Running the application should display the results of both web functions to the user in the console.



```
Current Date/Time on Server:10/21/2007 3:56:08 PM
Current Date/Time plus 2 Days:10/23/2007 3:56:08 PM
Press any key to continue . . .
```