

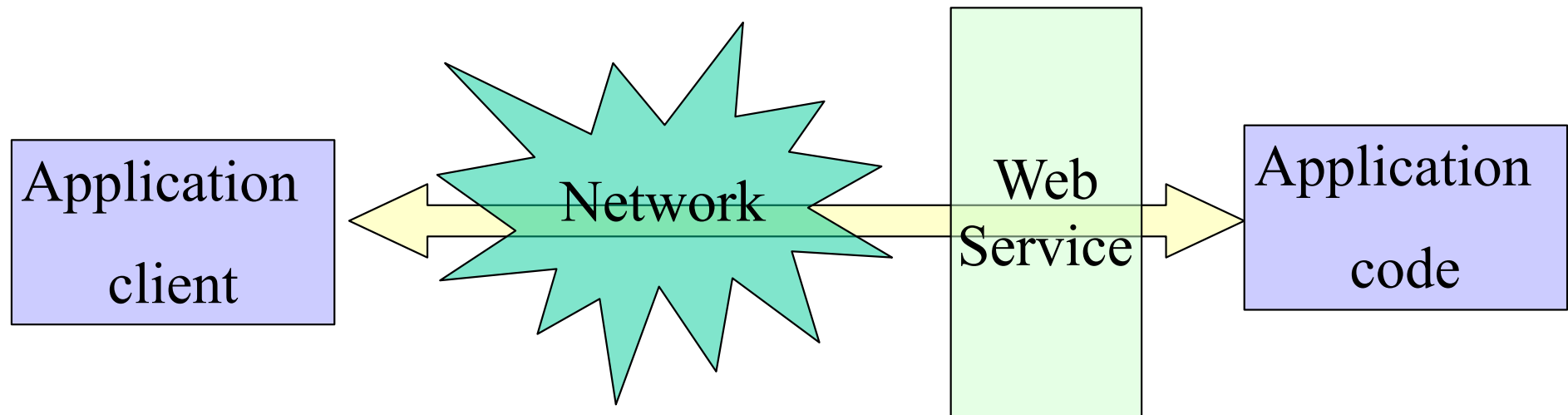
Web Services with MS .Net

Outline

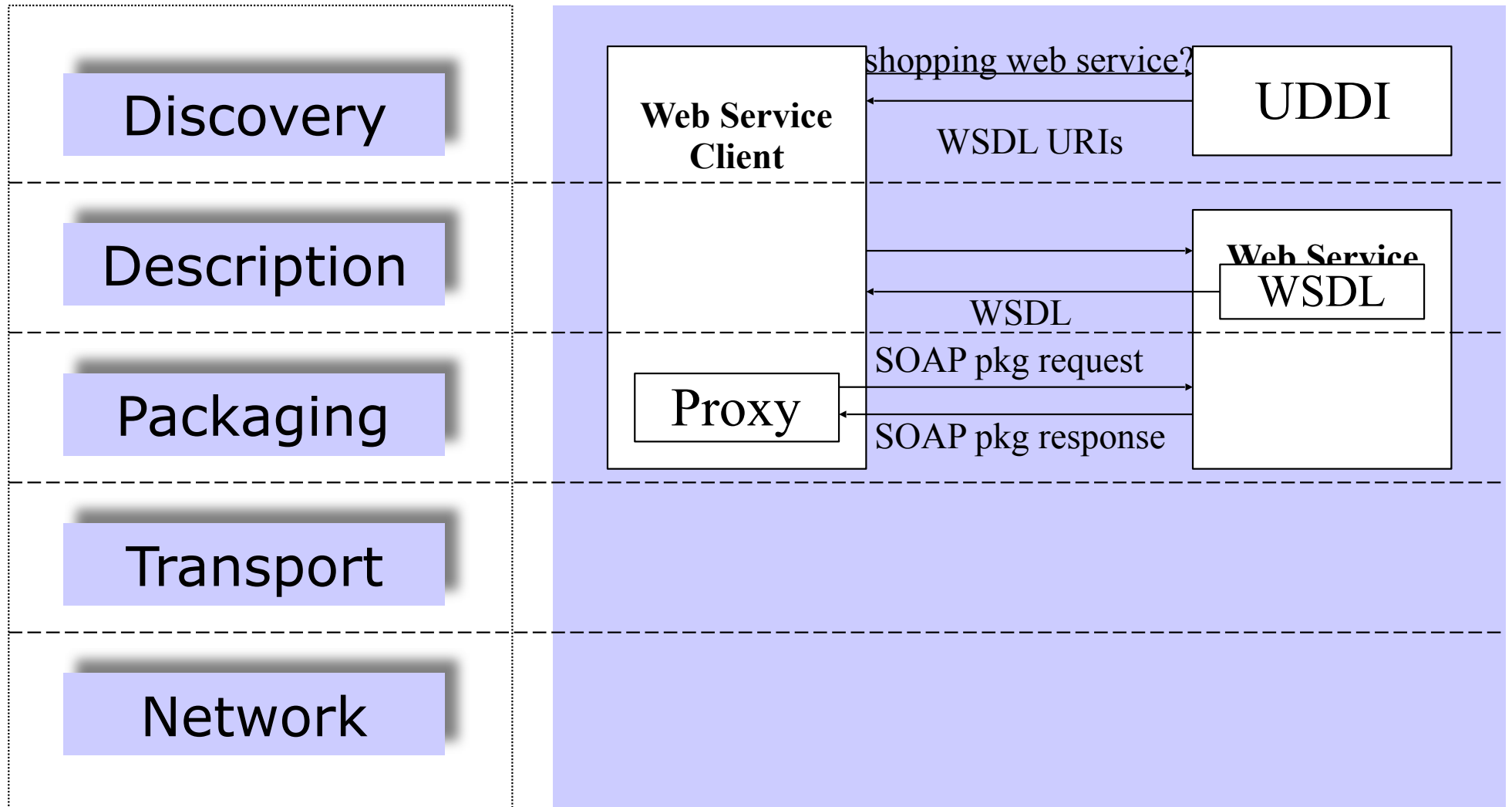
- Overview of web services
- Create a web service with MS .Net

What is a Web Service

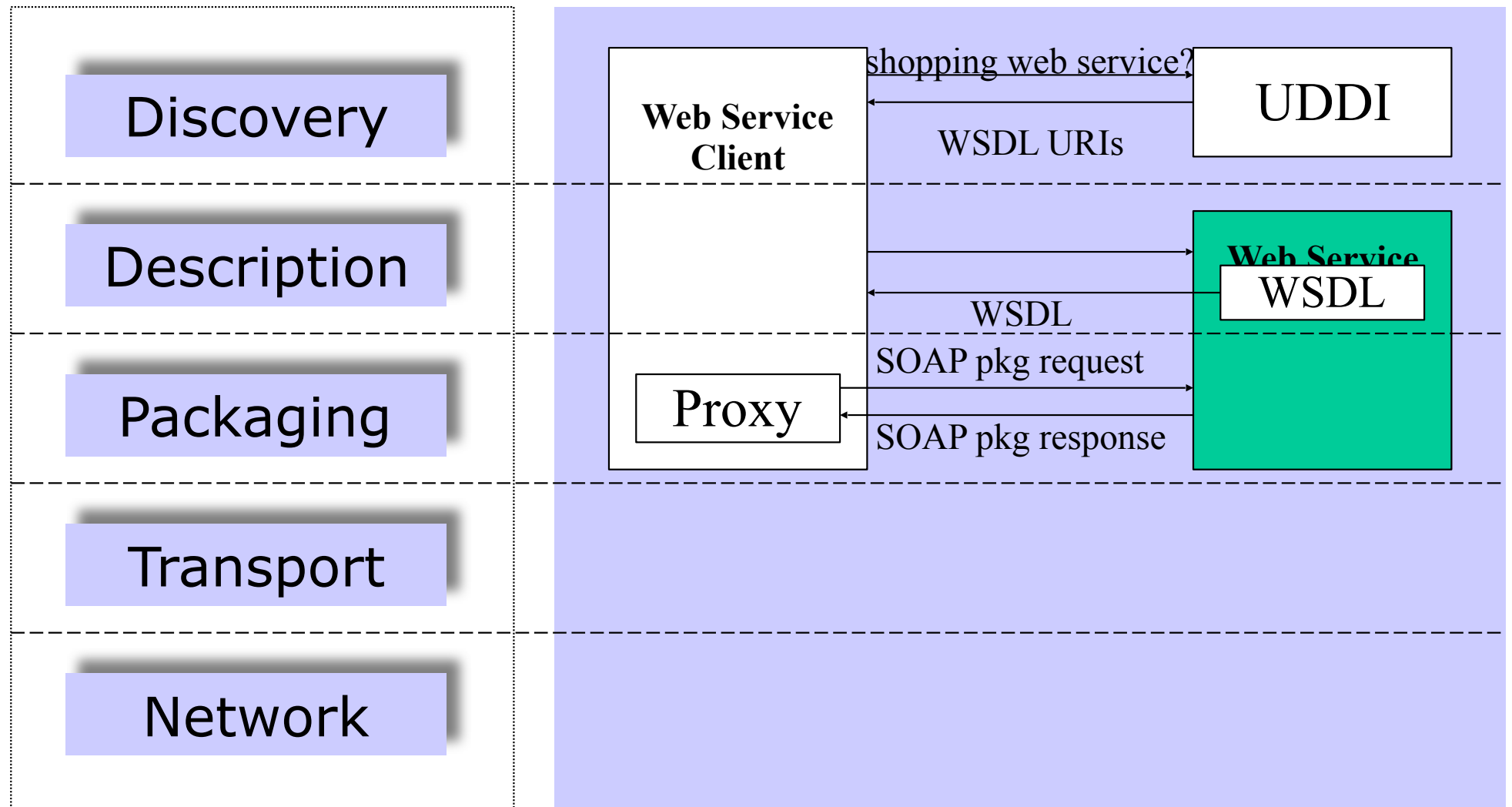
- A web service is a network accessible interface to application functionality, built using standard Internet technologies.
- Clients of web services do **NOT** need to know how it is implemented.



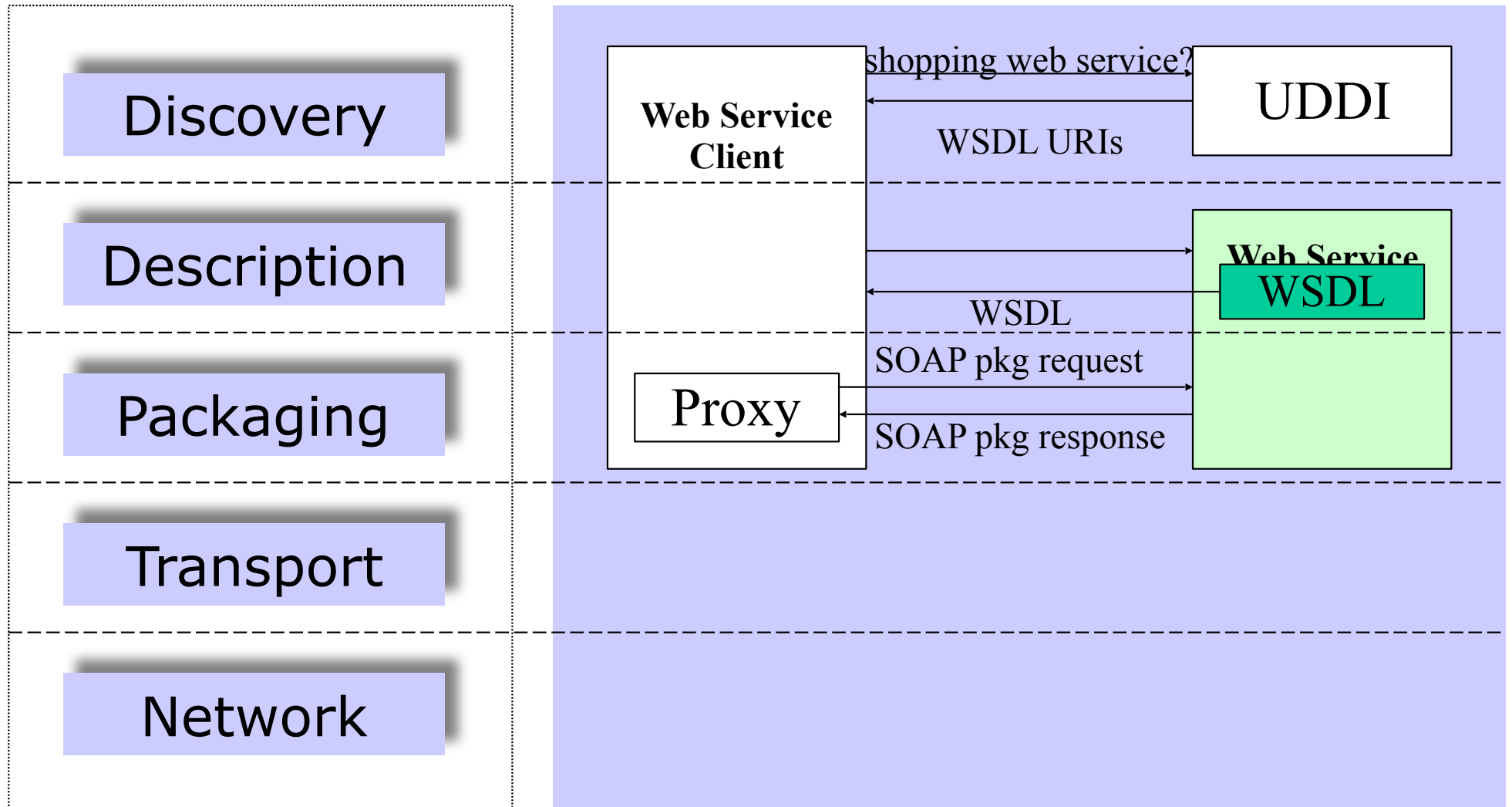
Web Service Technology Stack



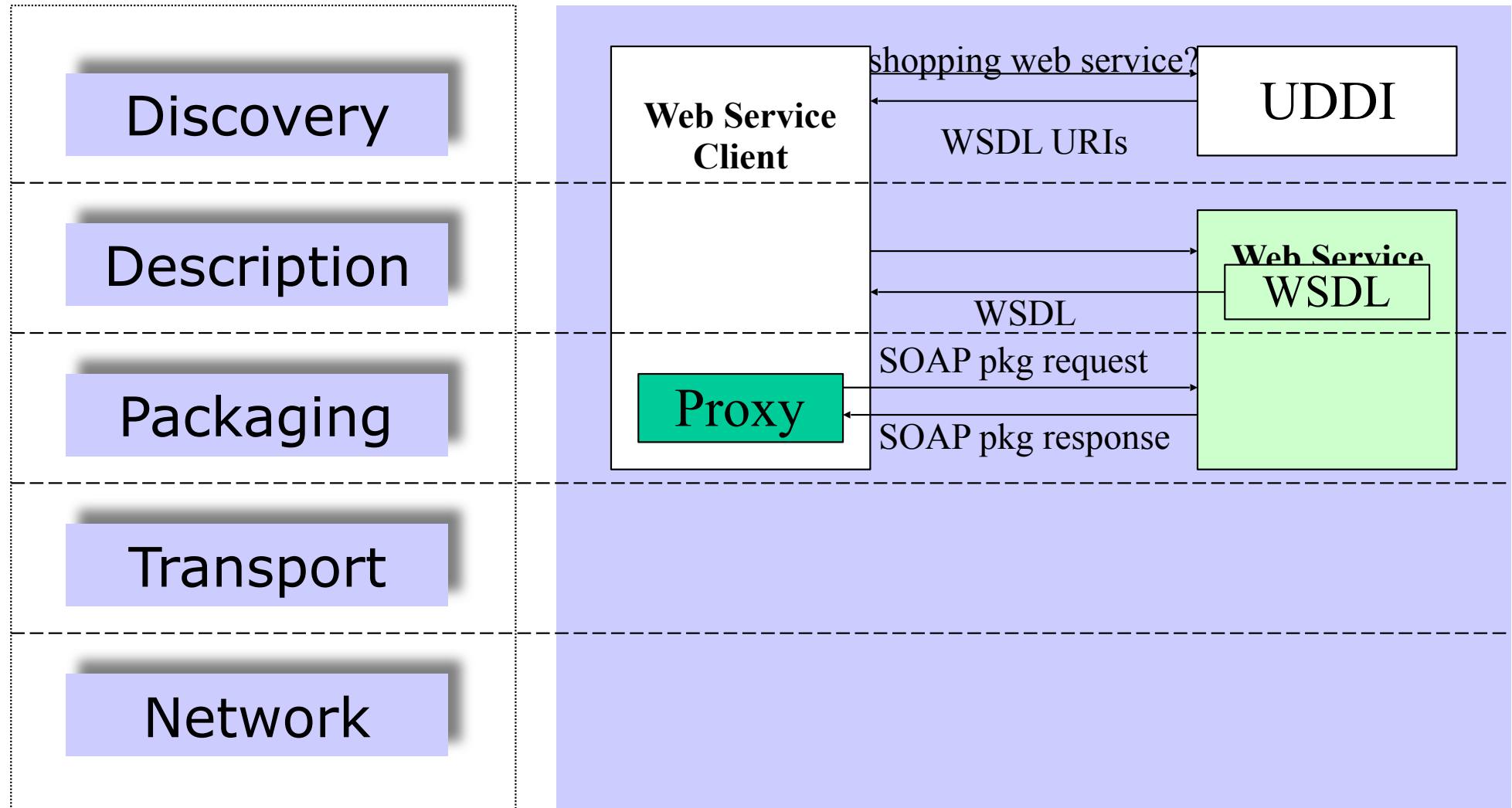
Step 1. Write Web Service Method



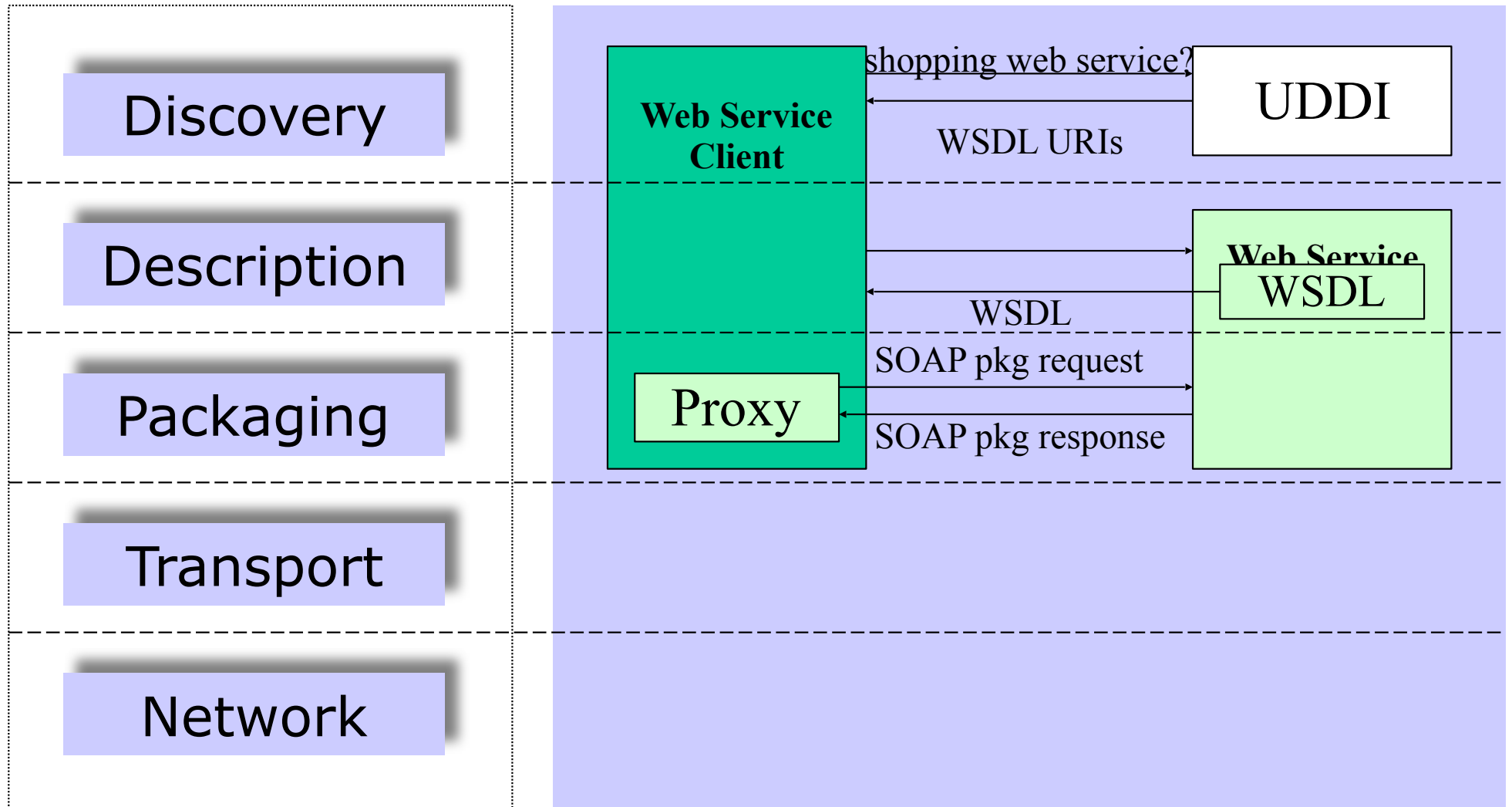
Step2. Describe Web Service using WSDL



Step3. Write Proxy to Access Web Service



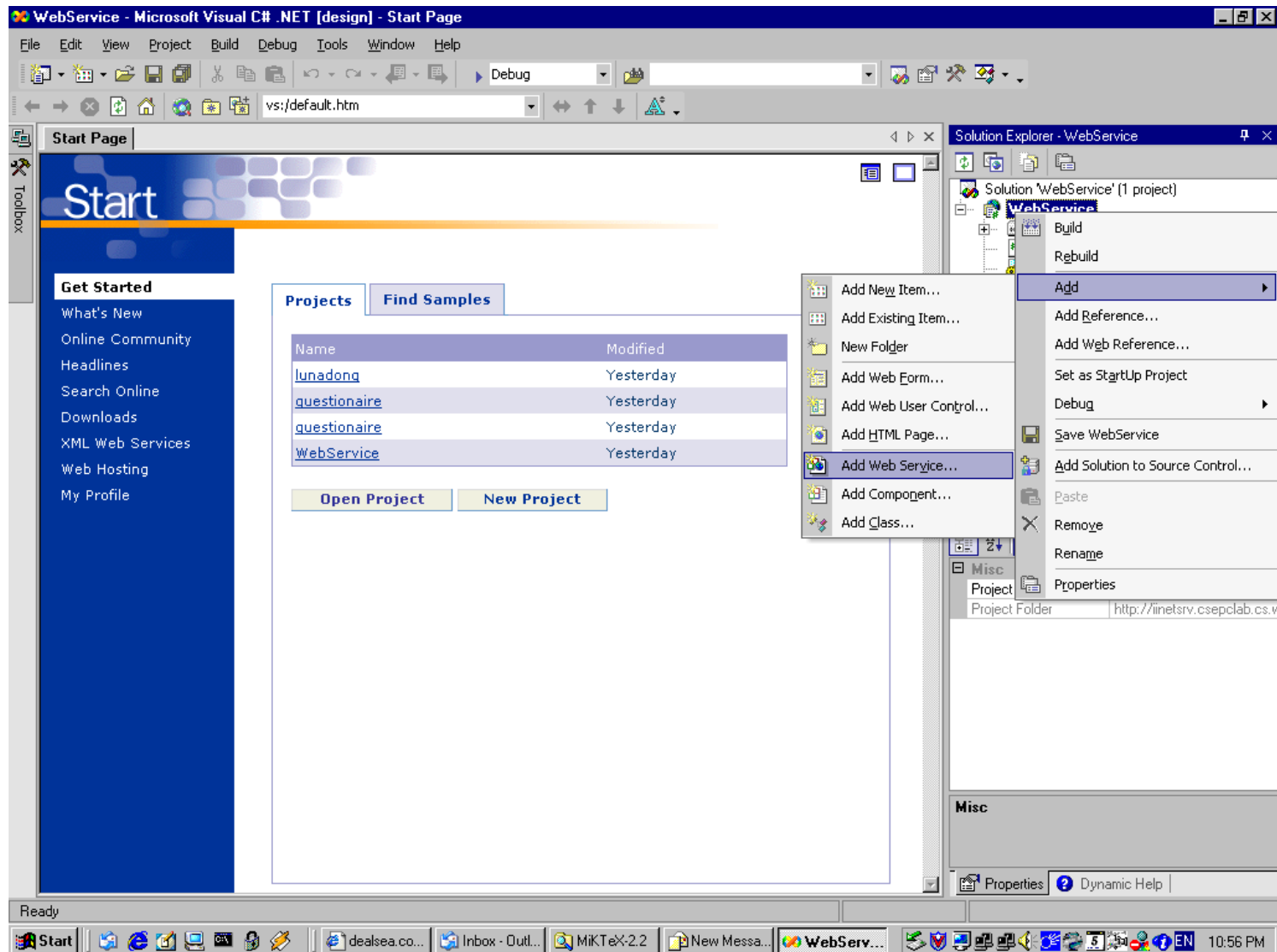
Step4. Write Client to Invoke Proxy



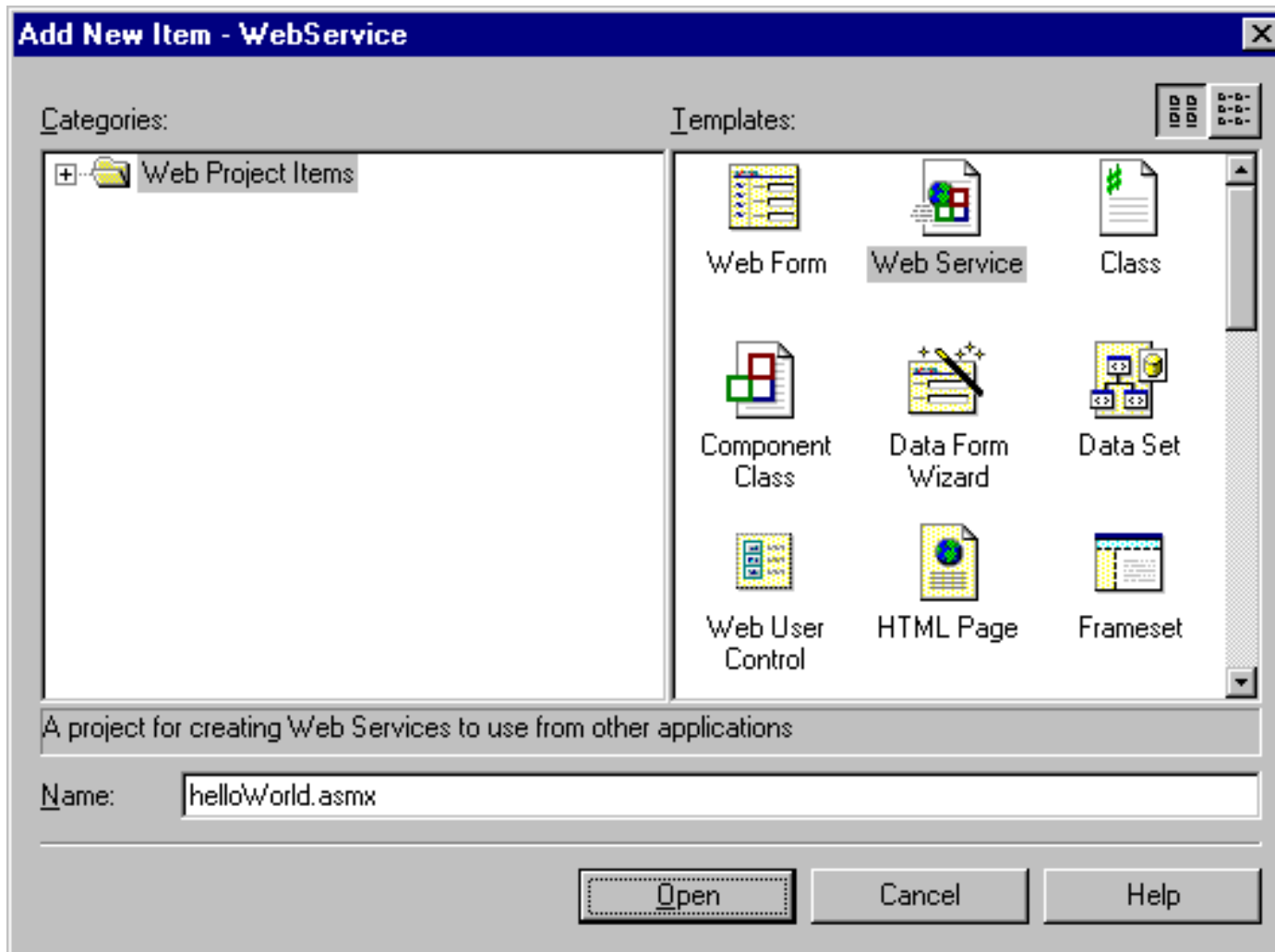
Step 1. Create a Web Service

- Functionality is implemented in .asmx.vb or .asmx.cs files.
 - `<%@WebService Language="C#" Class="helloWorld" %>`
- Use System.Web.Services Namespace
 - `Using System.Web.Services`
- Inherit your own class from WebService Base Class
 - `public class helloWorld : System.Web.Services.WebService`
- Declare the WebMethod Attribute
 - `[WebMethod]`
`public string HelloWorld(string name)`
`{...}`

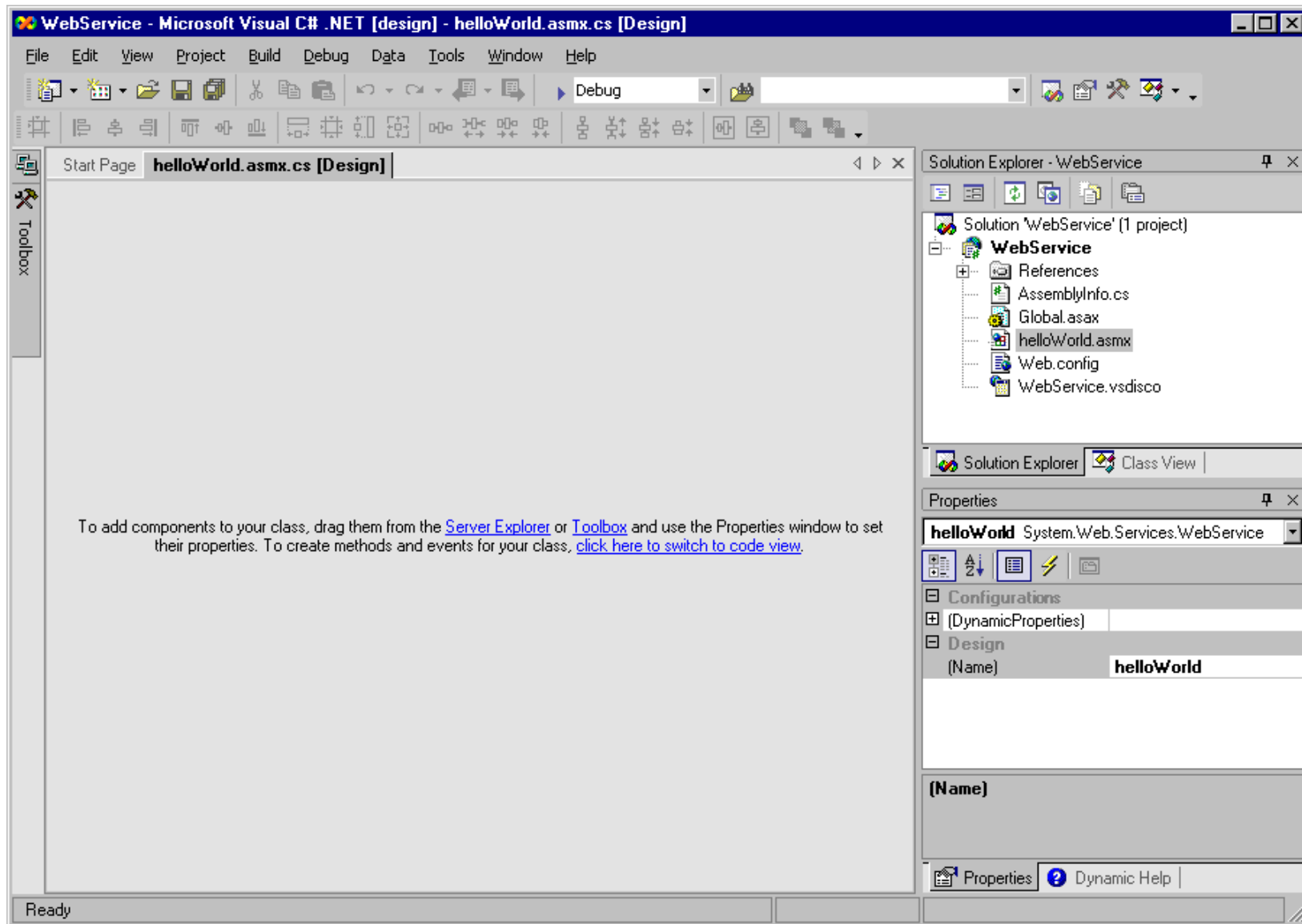
Creating a Web Service with .Net



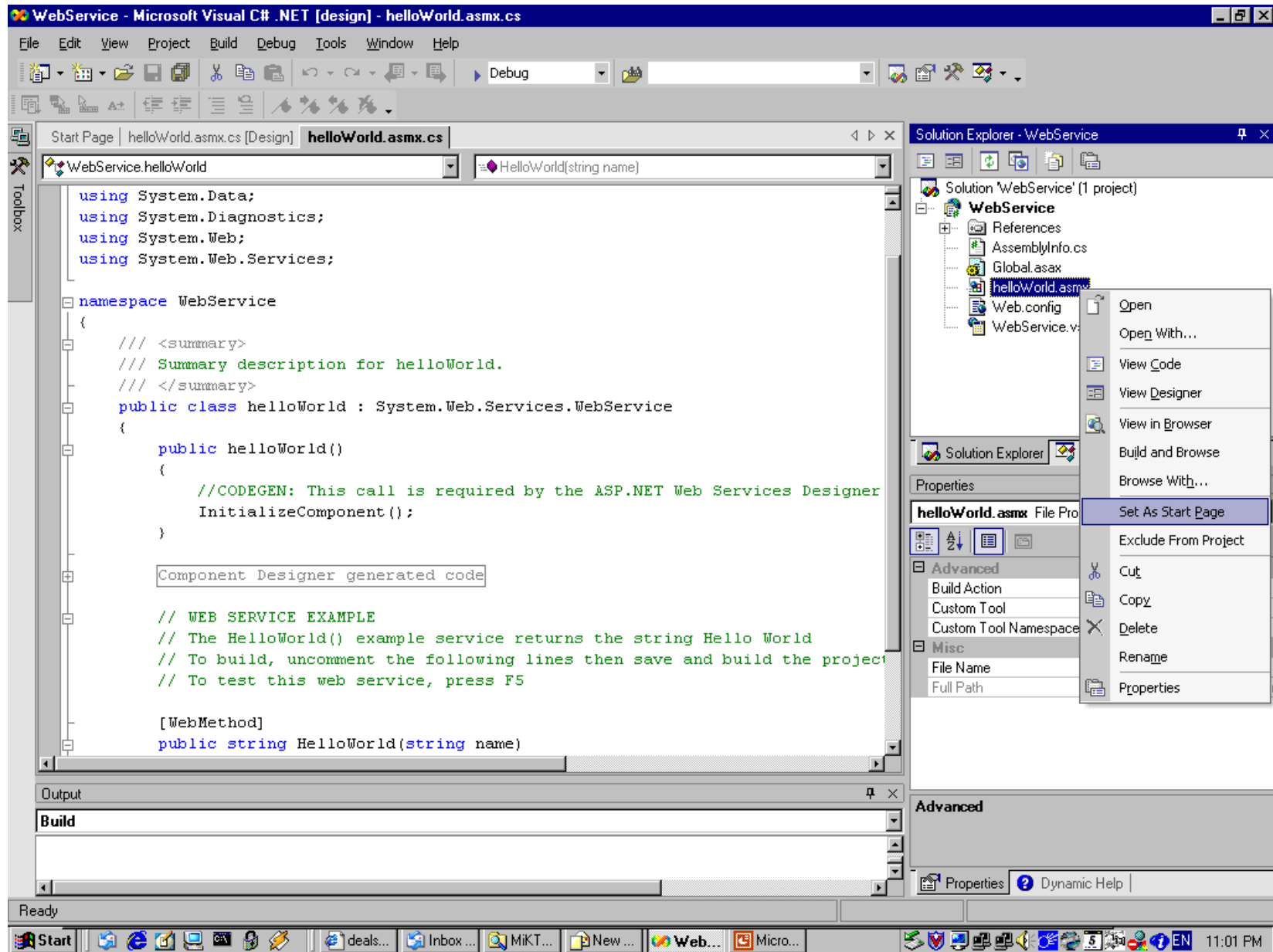
Creating a Web Service with .Net



Creating a Web Service with .Net



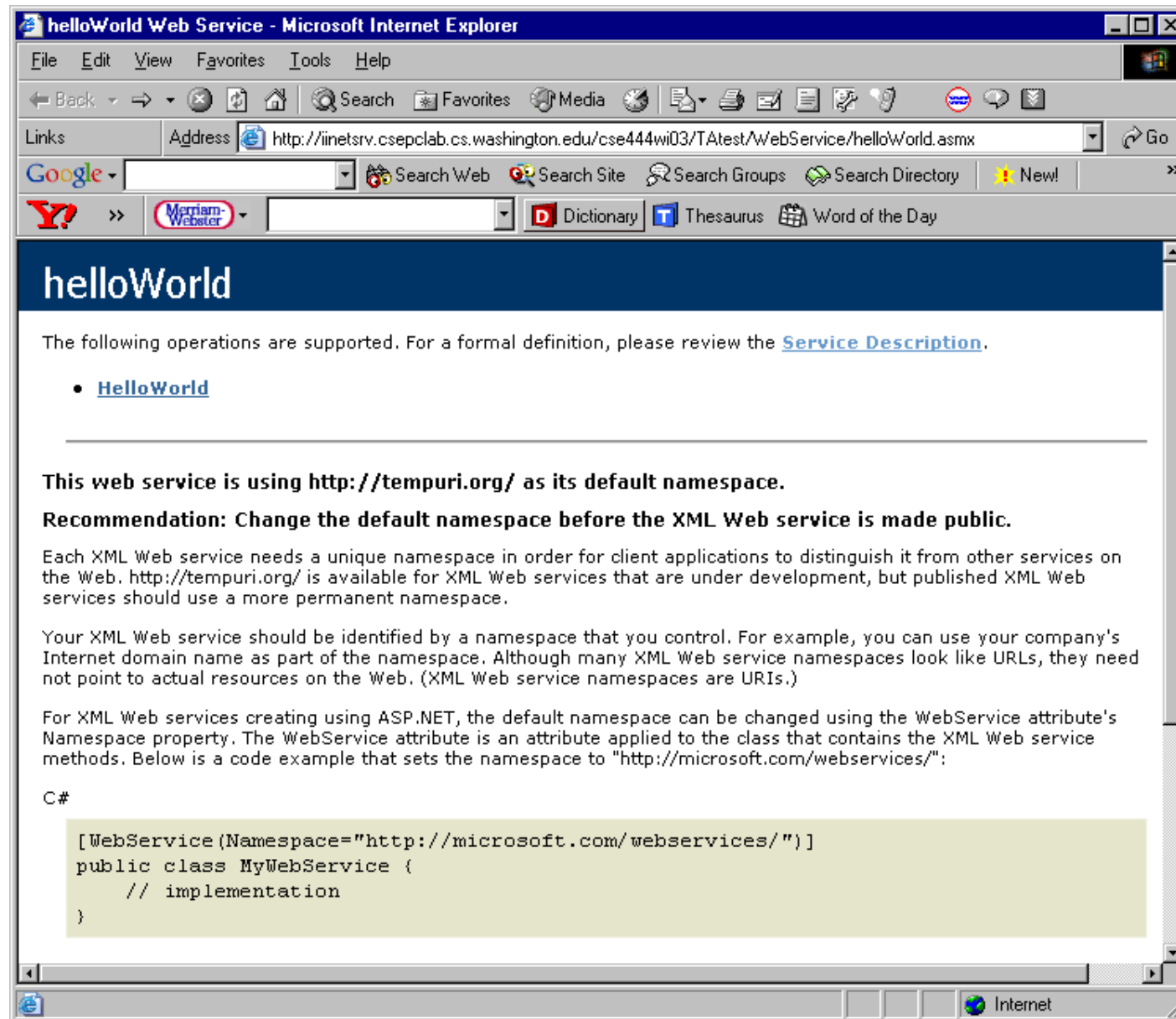
Creating a Web Service with .Net



Creating a Web Service with .Net

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

Compile and View Your Web Service



Compile and View Your Web Service

helloWorld Web Service - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News RSS Feeds

Address [tp://iinetsrv.csepclab.cs.washington.edu/cse444wi03/TAtest/WebService/helloWorld.asmx?op=HelloWorld](http://iinetsrv.csepclab.cs.washington.edu/cse444wi03/TAtest/WebService/helloWorld.asmx?op=HelloWorld) Go

Google Search Web Search Site Search Groups Search Directory New!

Y! Merriam-Webster Dictionary Thesaurus Word of the Day

helloWorld

Click [here](#) for a complete list of operations.

HelloWorld

Test

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

Parameter	Value
name:	<input type="text" value="Luna"/>

Invoke

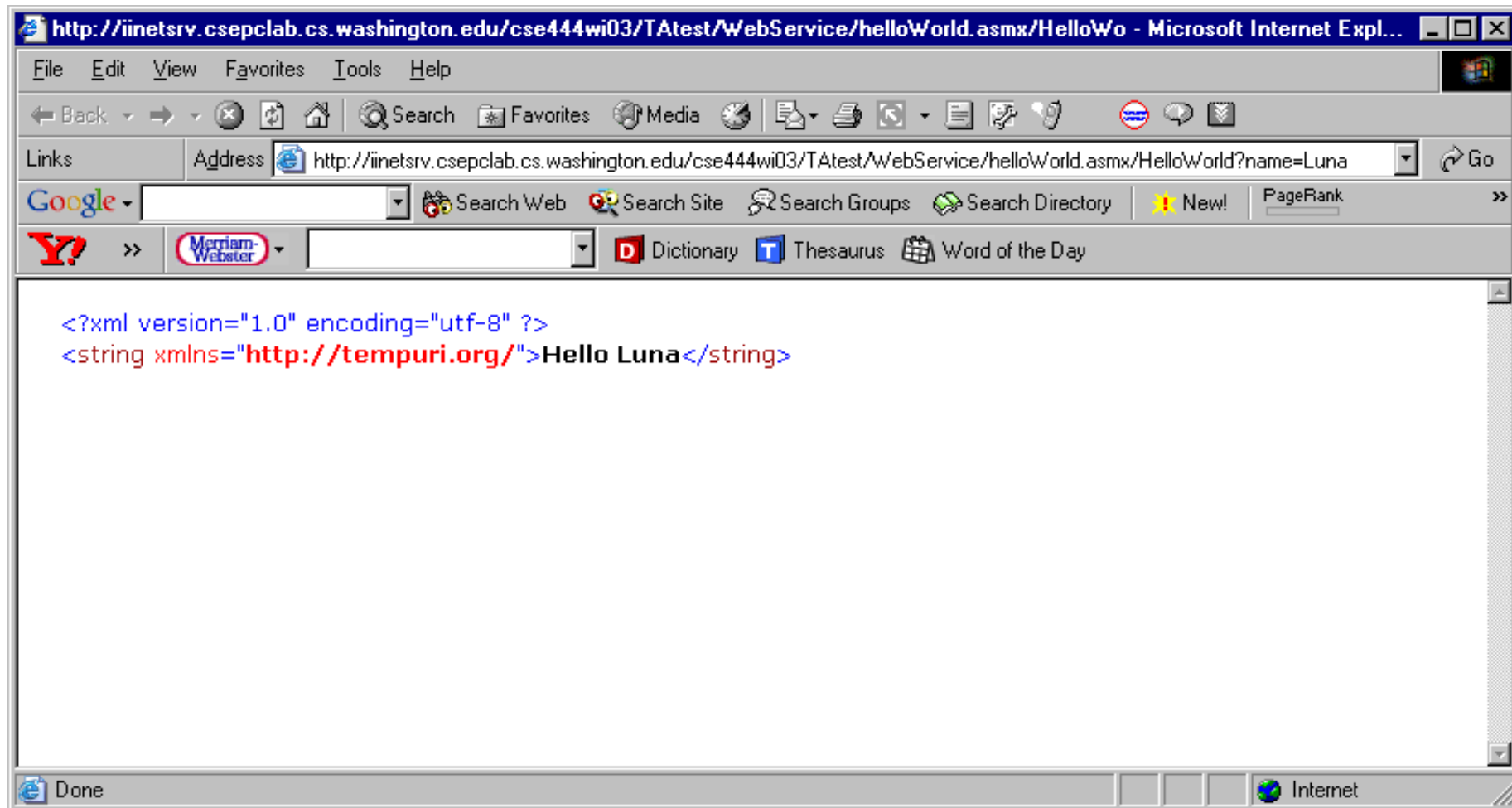
SOAP

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

```
POST /cse444wi03/TAtest/WebService/helloWorld.asmx HTTP/1.1
Host: iinetsrv.csepclab.cs.washington.edu
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/HelloWorld"

<?xml version="1.0" encoding="utf-8"?>
```


Compile and View Your Web Service



Step2. Describe Web Service using WSDL

- WSDL (Web Services Description Language)
- Describes 3 ways to access web service: GET, POST, SOAP
- Element:
 - `<types>`: XML schema for input/output
 - `<message>`:
 - **HelloWorldSoapIn, ...SoapOut, ...HttpGetIn, ...HttpGetOut, ...HttpPostIn, ...HttpPostOut**
 - `<porttype>`
 - **helloWorldSoap, ...HttpGet, ...HttpPost**
 - `<binding>`
 - **s0:helloWorldSoap, s0:...HttpGet, s0:...HttpPost**
 - `<service name = "helloWorld">`
 - `<port name = "...Soap" binding = "s0:...Soap">`
 - `<soap:address location = "http://...">`
 - `</port></service>`

WSDL Generated by .NET

helloWorld Web Service - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News RSS Feeds

Links Address <http://inetsrv.csepclab.cs.washington.edu/cse444wi03/TAtest/WebService/helloWorld.asmx> Go

Google Search Web Search Site Search Groups Search Directory New!

Y! Merriam-Webster Dictionary Thesaurus Word of the Day

helloWorld

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [HelloWorld](#)

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.

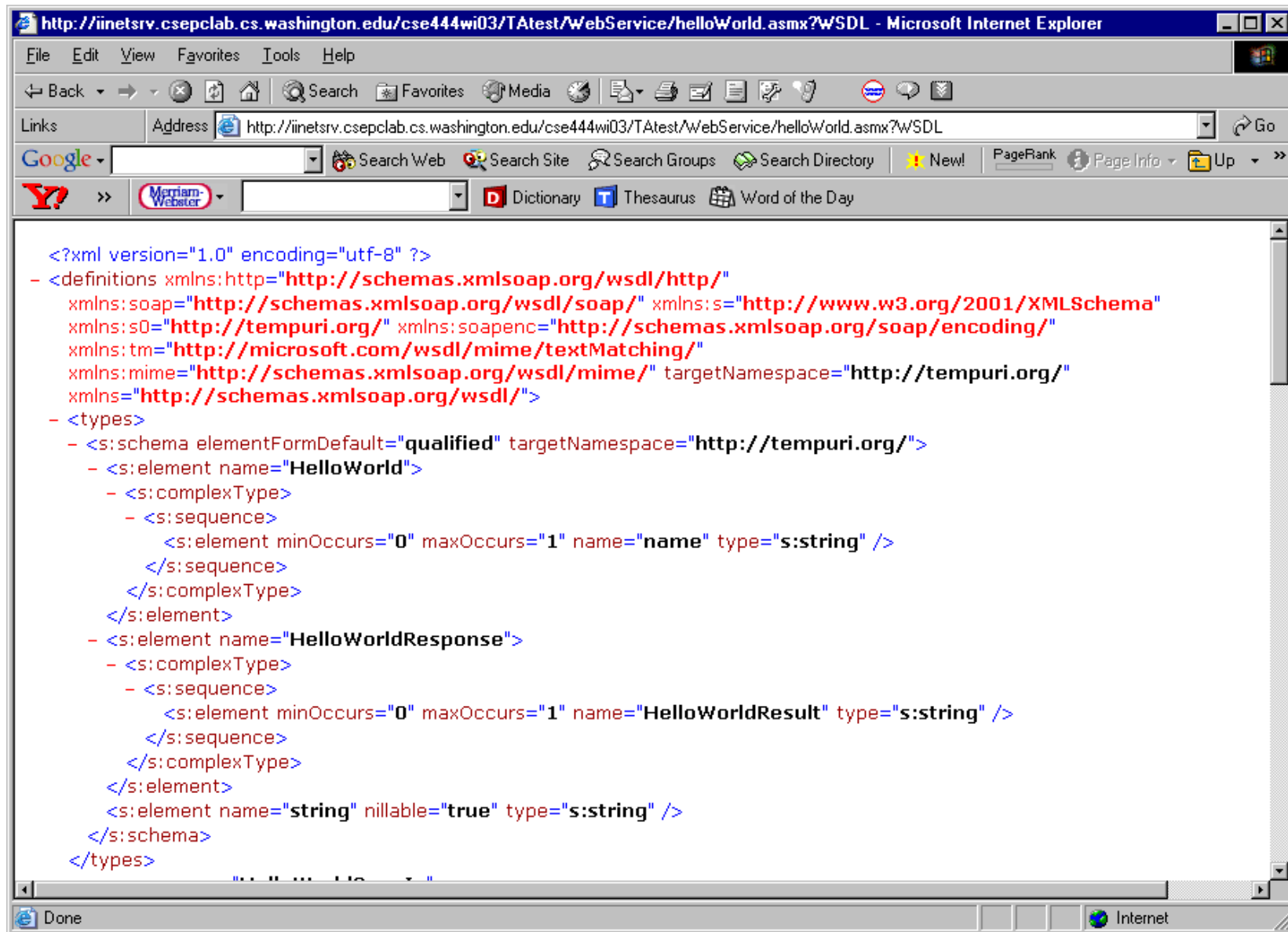
Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are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

Internet

WSDL Generated by .NET



```
<?xml version="1.0" encoding="utf-8" ?>
- <definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:s0="http://tempuri.org/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="http://tempuri.org/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
- <types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
- <s:element name="HelloWorld">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string" />
  </s:sequence>
  </s:complexType>
</s:element>
- <s:element name="HelloWorldResponse">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="HelloWorldResult" type="s:string" />
  </s:sequence>
  </s:complexType>
</s:element>
  <s:element name="string" nillable="true" type="s:string" />
</s:schema>
</types>
```

SOAP (Simple Object Access Protocol)

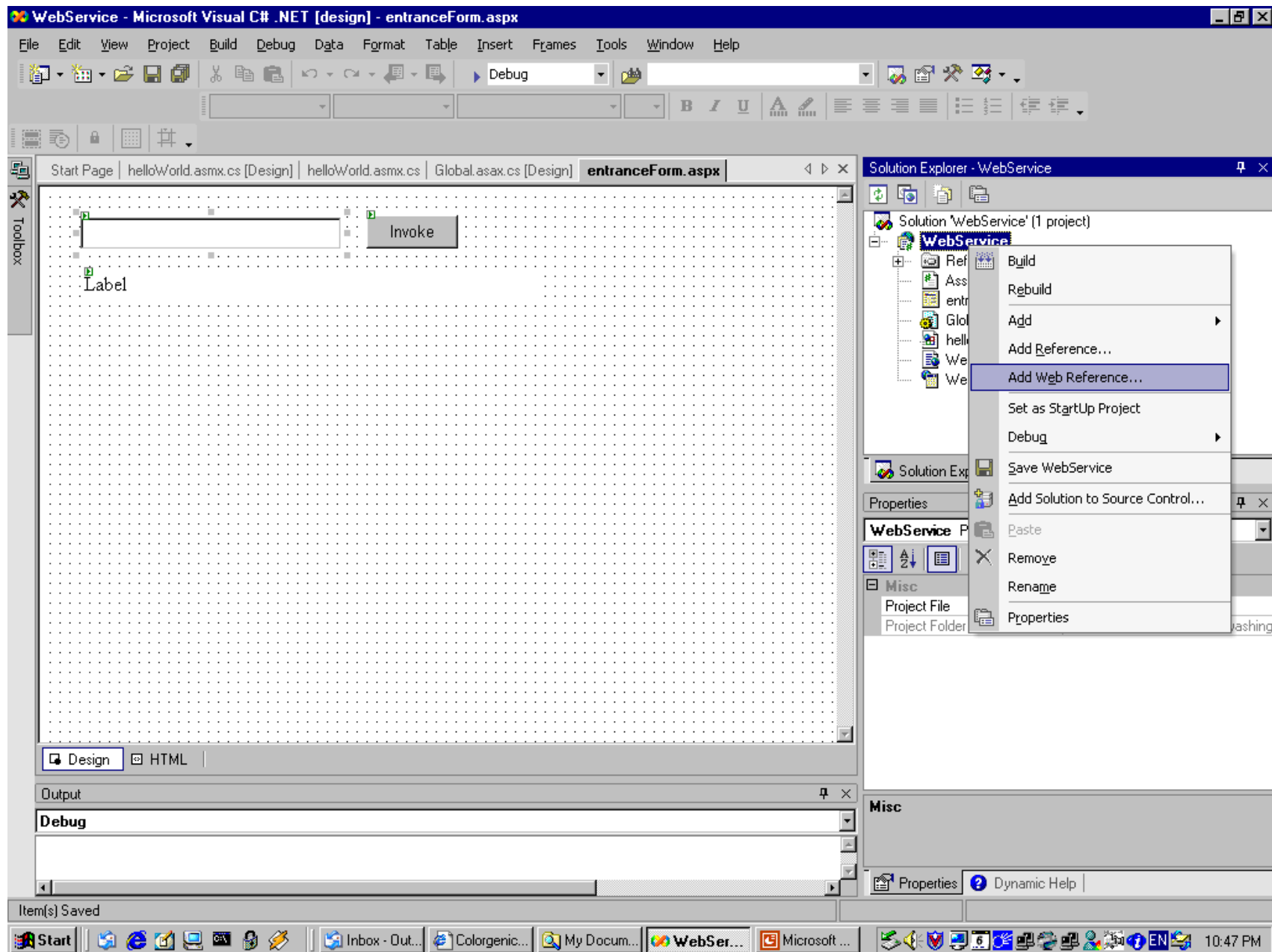
- SOAP Messages
- Using SOAP as RPC (Remote Procedure Call) Messages



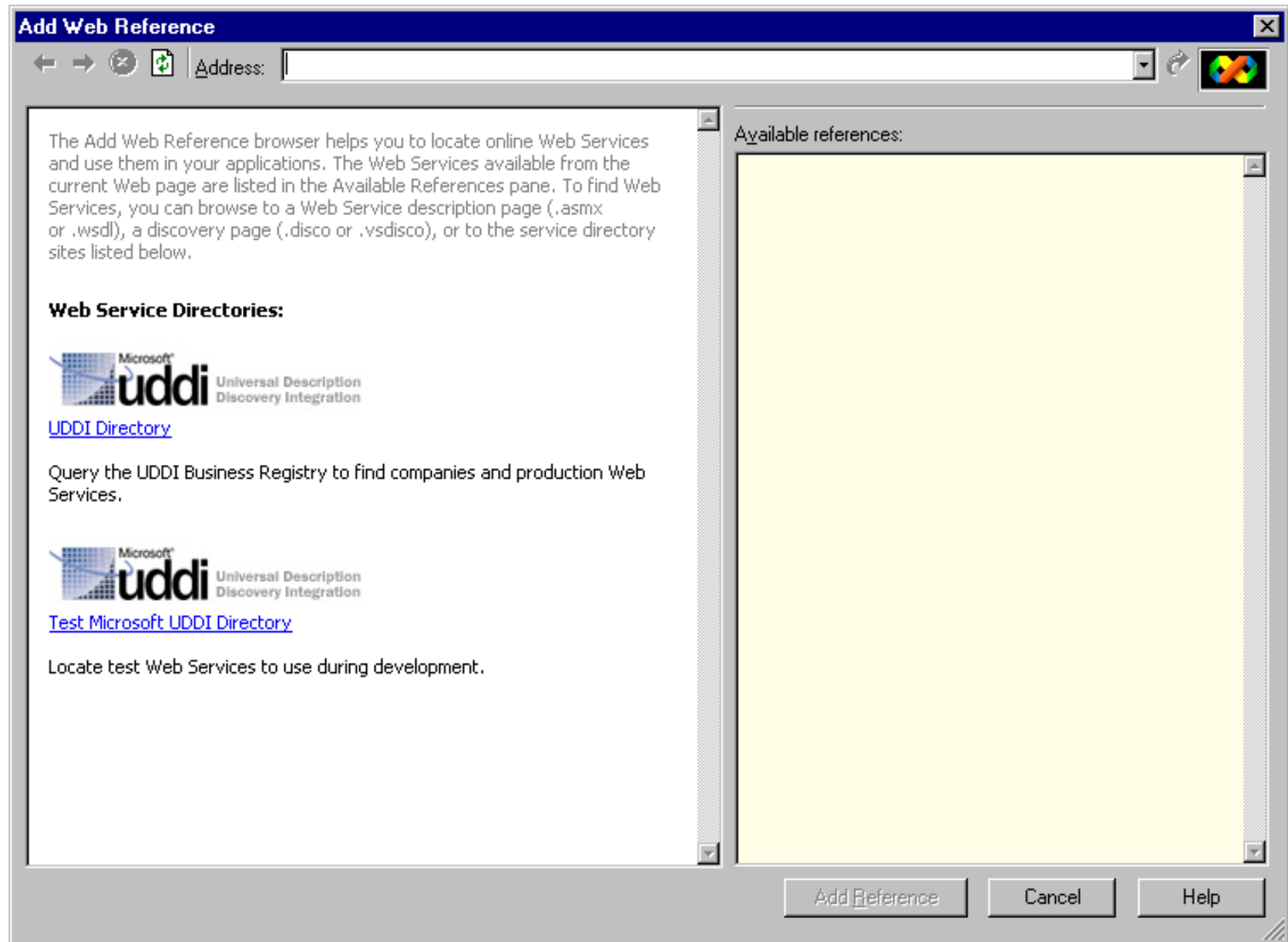
Step3. Write Proxy to Access Web Service

- In software, a proxy is the code that does work on behalf of other code.
 - For web service users, the proxy represents the web service the users wish to call
 - For web service servers, the proxy makes request on the behalf of the user

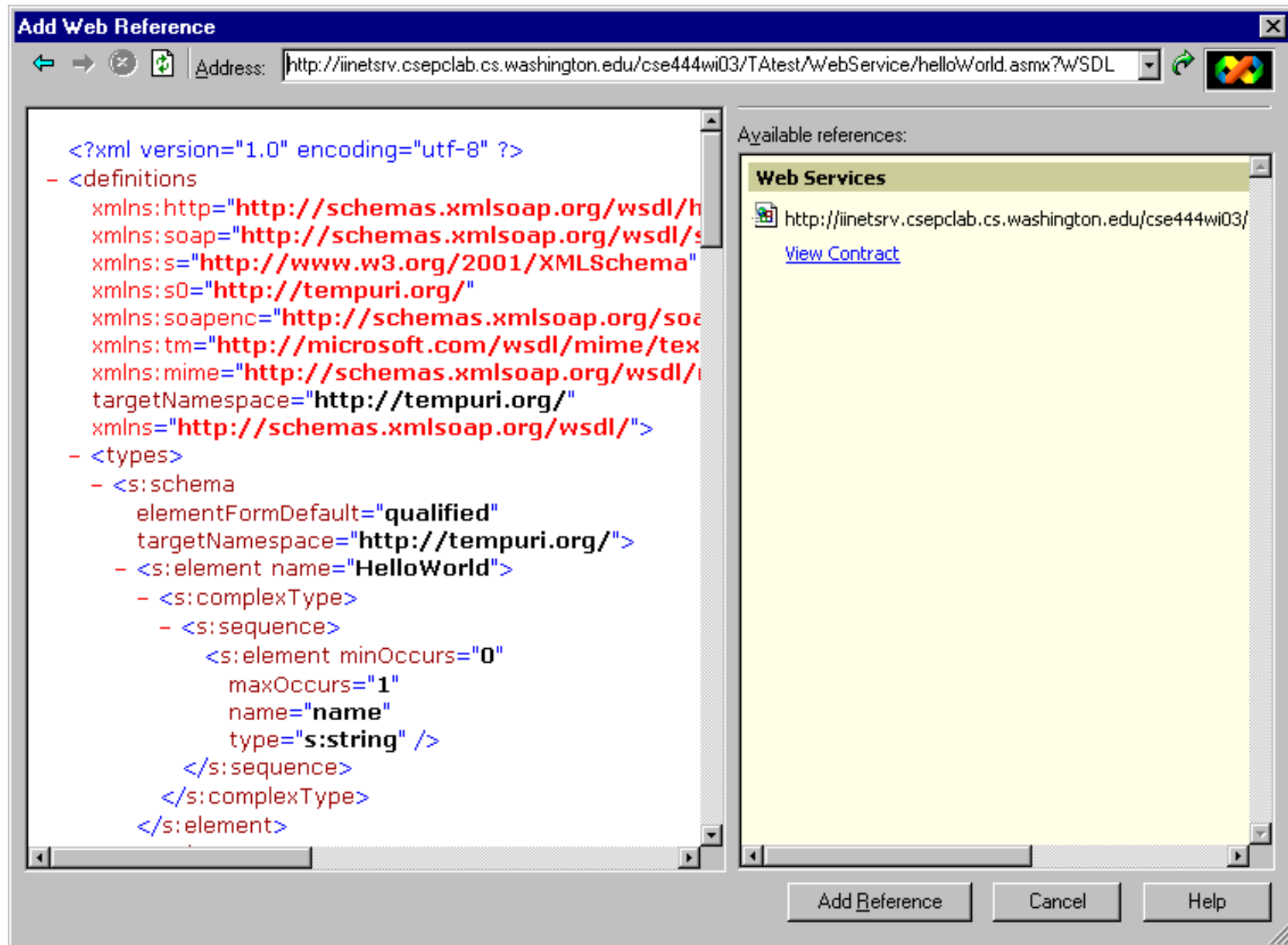
Create a Proxy (Web Reference) with .Net



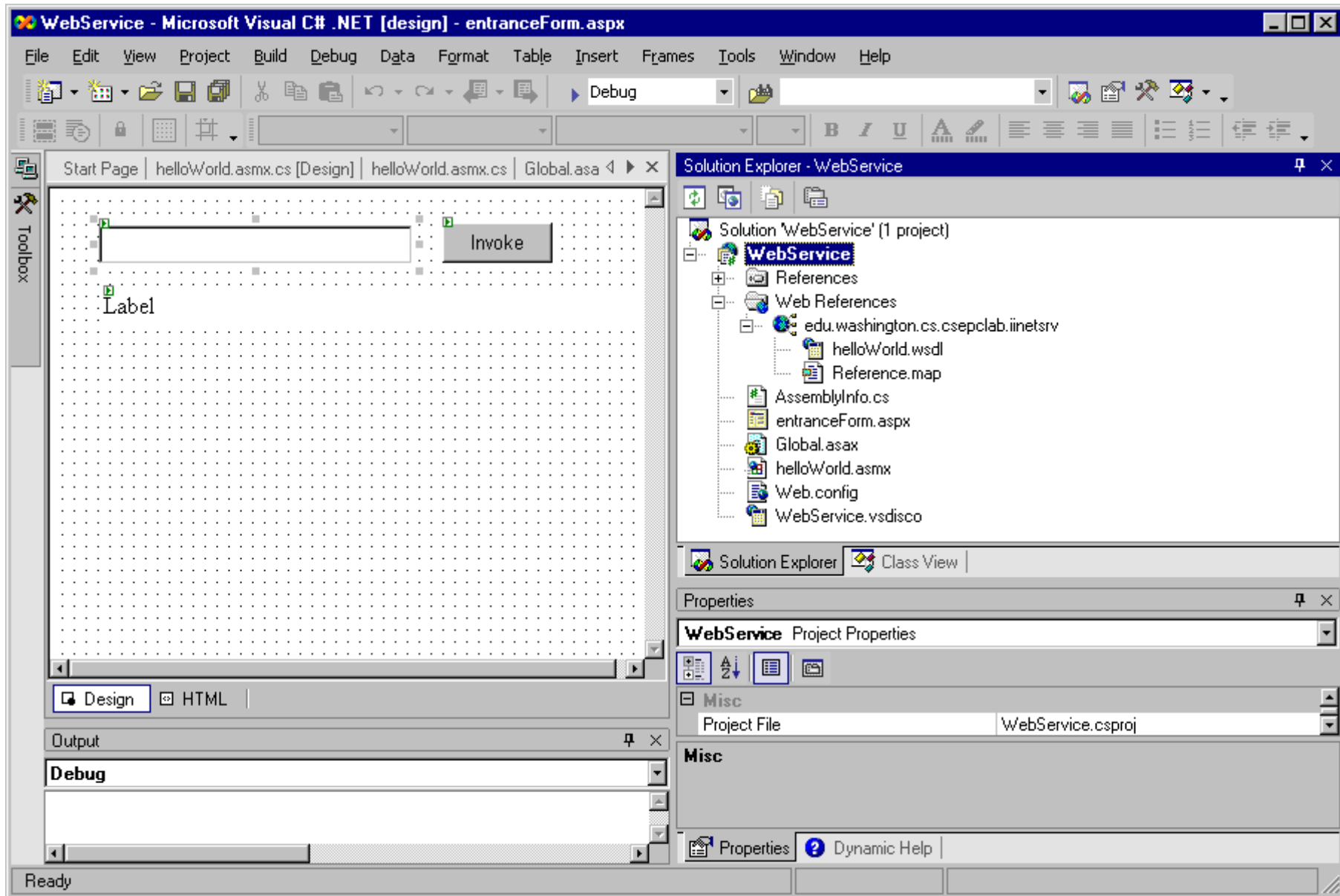
Create a Proxy (Web Reference) with .Net



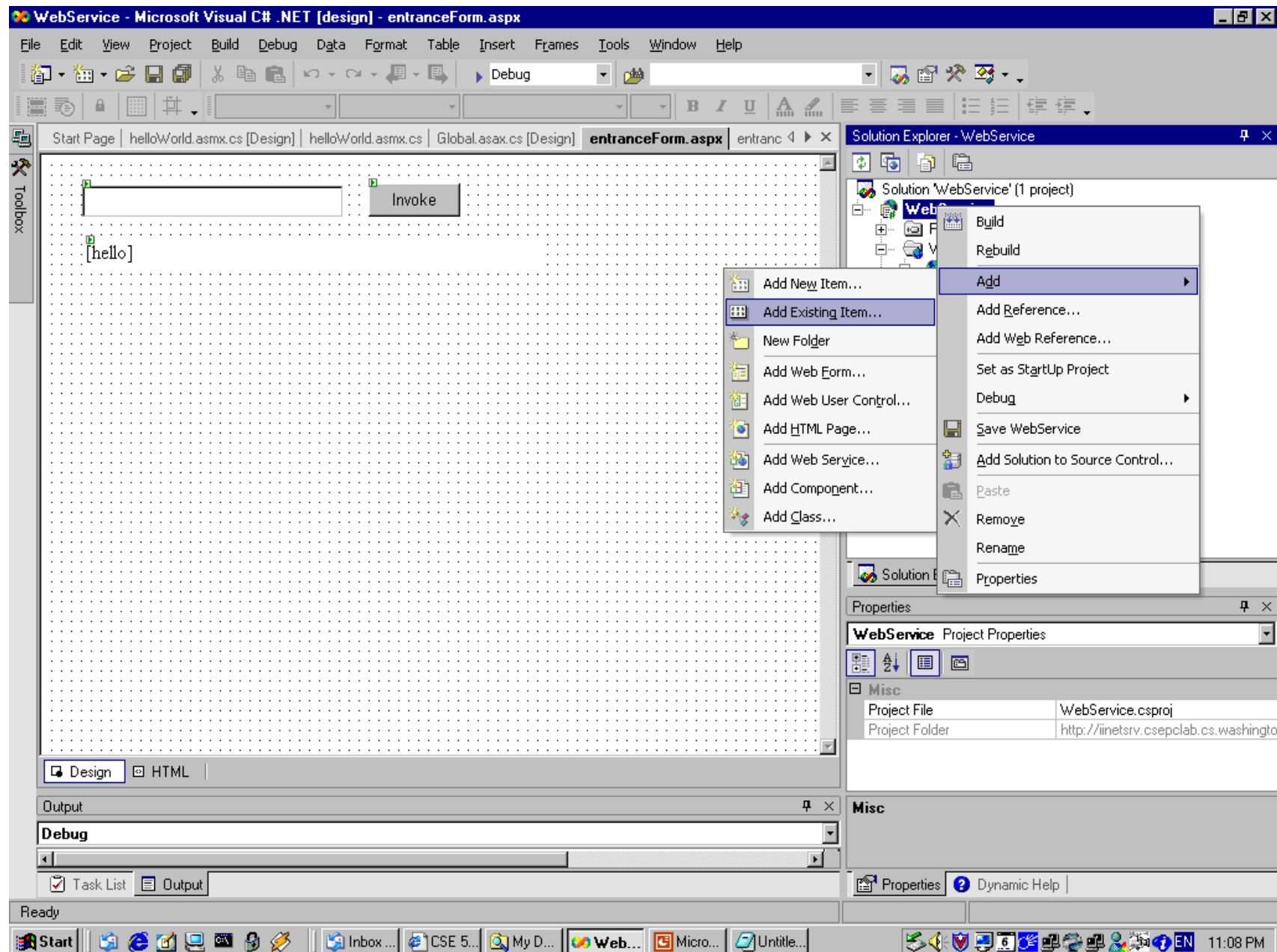
Create a Proxy (Web Reference) with .Net



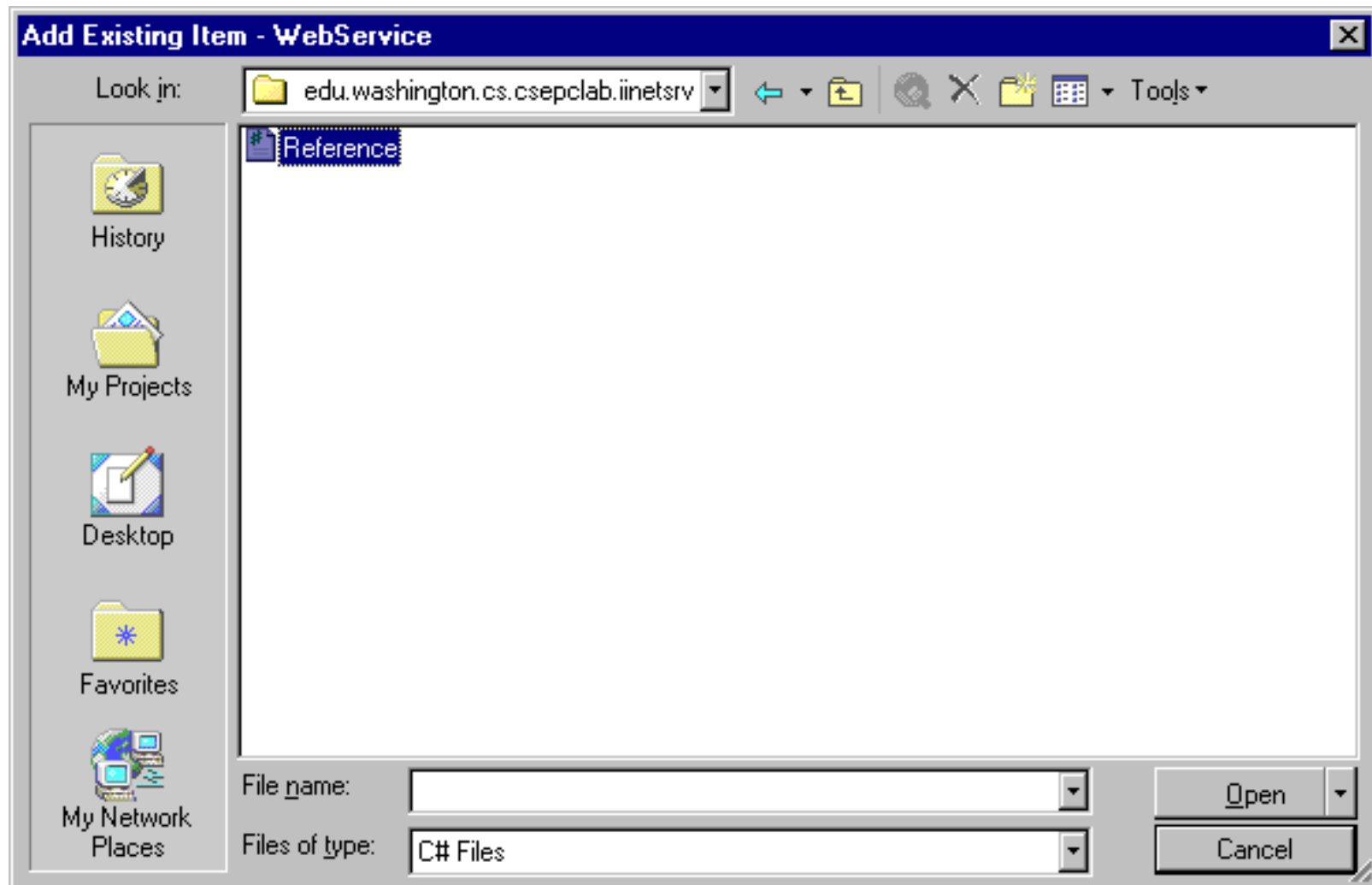
Create a Proxy (Web Reference) with .Net



Read Code for Proxies



Read Code for Proxies



Read Code for Proxies

The screenshot displays the Microsoft Visual Studio IDE. The main window shows the code for `Reference.cs` in a web service project. The code is auto-generated and includes the following content:

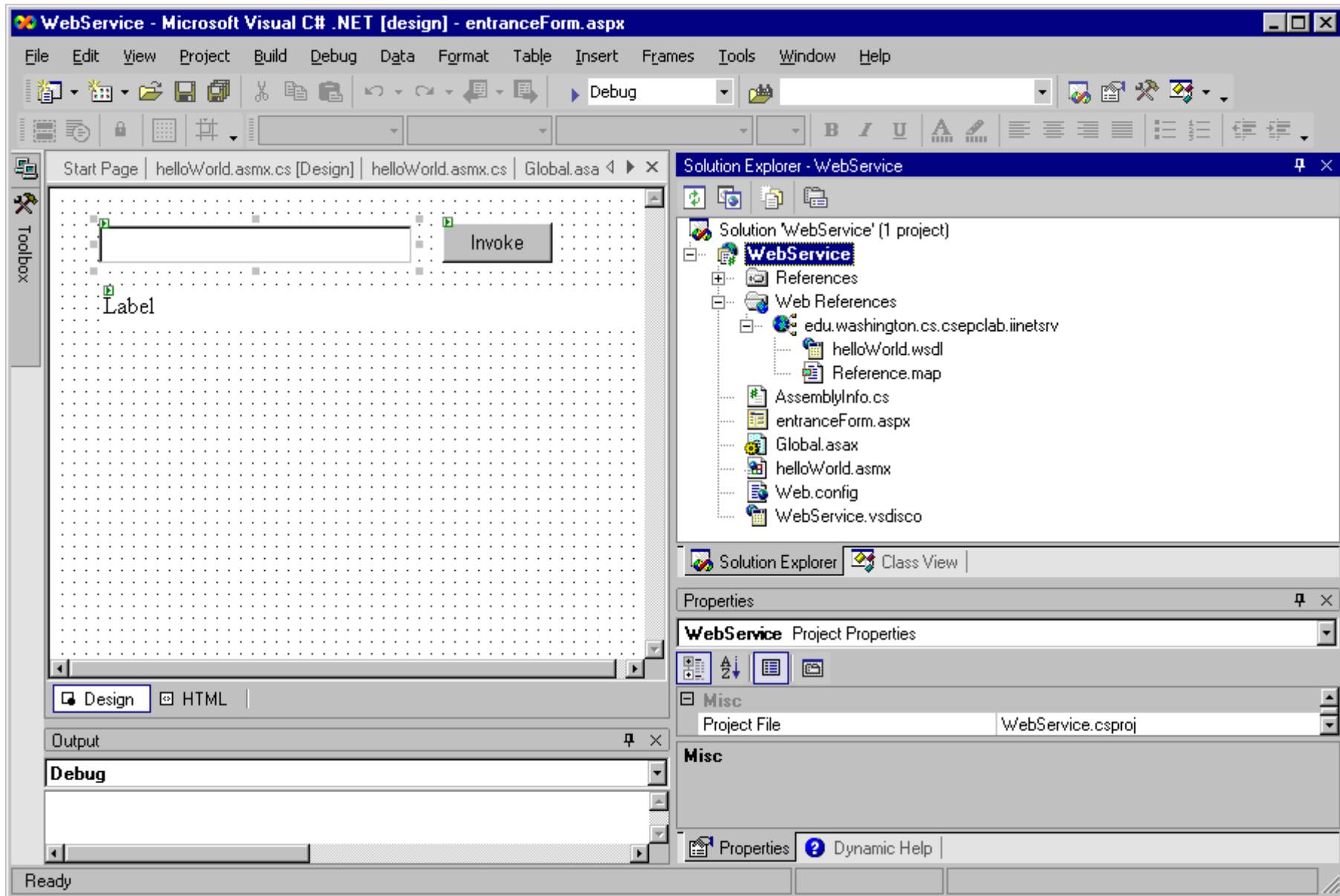
```
// -----  
// <autogenerated>  
//   This code was generated by a tool.  
//   Runtime Version: 1.0.3705.0  
//  
//   Changes to this file may cause incorrect behavior and will be lost if  
//   the code is regenerated.  
// </autogenerated>  
// -----  
  
//  
// This source code was auto-generated by Microsoft.VSDesigner, Version 1.0.3705.0  
//  
namespace WebService.edu.washington.cs.cseplab.iinetsrv {  
    using System.Diagnostics;  
    using System.Xml.Serialization;  
    using System;  
    using System.Web.Services.Protocols;  
    using System.ComponentModel;  
    using System.Web.Services;  
  
    /// <remarks/>  
    [System.Diagnostics.DebuggerStepThroughAttribute()]  
    [System.ComponentModel.DesignerCategoryAttribute("code")]  
    [System.Web.Services.WebServiceBindingAttribute(Name="helloWorldSoap",  
    public class helloWorld : System.Web.Services.Protocols.SoapHttpClientProtocol
```

The Solution Explorer on the right shows the project structure for 'WebService' (1 project), including files like `Reference.cs`, `WebReferences`, `AssemblyInfo.cs`, `entranceForm.aspx`, `Global.asax`, `helloWorld.aspx`, `Web.config`, and `WebService.vsdisco`. The Properties window at the bottom right shows the file properties for `Reference.cs`, including its full path: `C:\Documents and Settings\unad...`. The Output window at the bottom left shows the 'Debug' output.

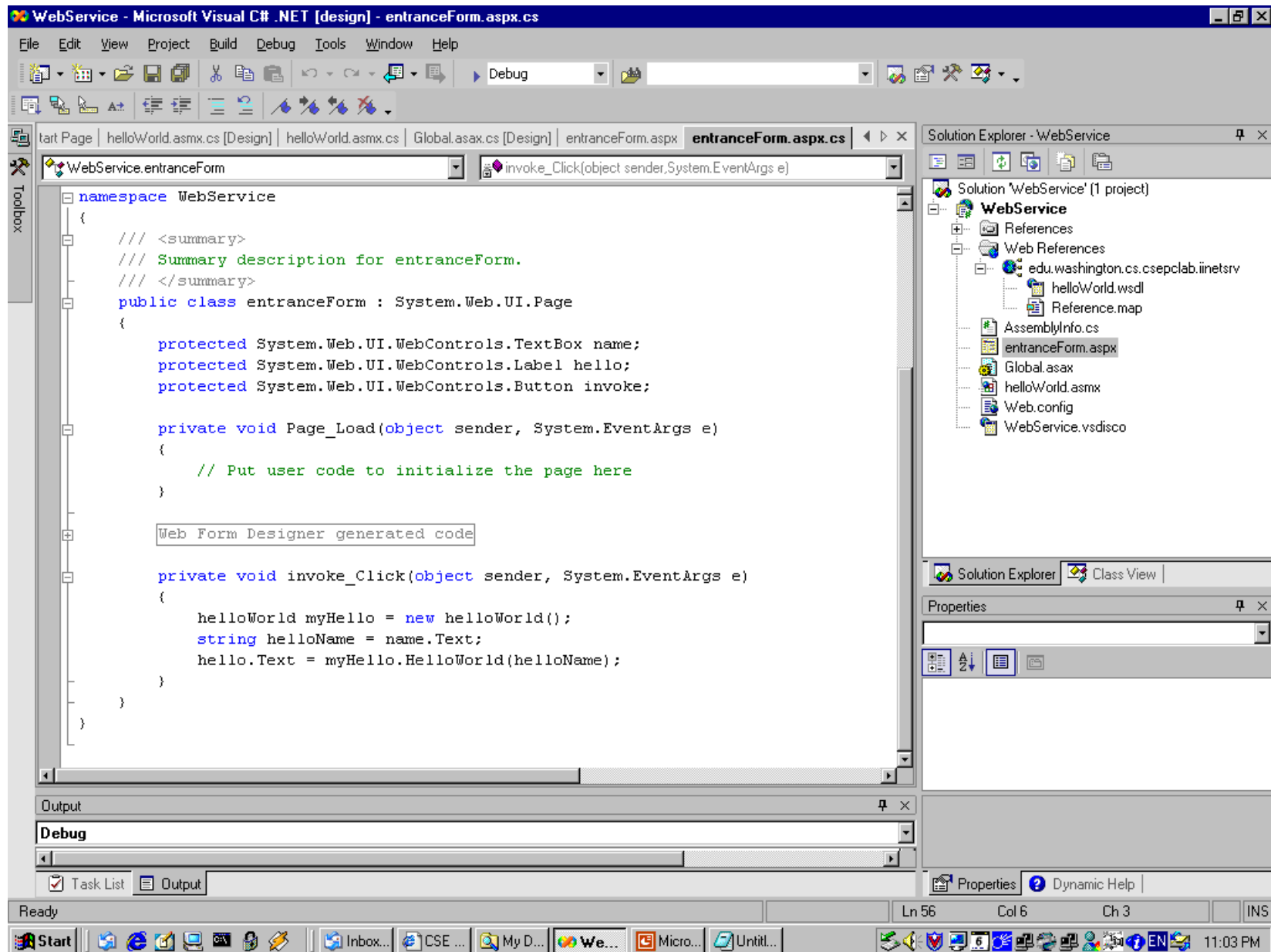
Read Code for Proxies

```
public class helloWorld : System.Web.Services.  
    Protocols.SoapHttpClientProtocol  
{  
    public helloWorld() {  
        this.Url = "http://iinetsrv.csepclab.cs.washington  
        .edu/cse444wi03/TAtest/WebService/helloWorld.  
        asmx";  
    }  
    ...  
}
```

Step4. Write a Client to Invoke Proxy



Step4. Write a Client to Invoke Proxy



Step4. Write a Client to Invoke Proxy

```
private void invoke_Click(object sender, System.EventArgs e)
{
    HelloWorld myHello = new HelloWorld();
    string helloName = name.Text;
    hello.Text = myHello.HelloWorld(helloName);
}
```

Step4. Write a Client to Invoke Proxy

