Web Services

Jim Fawcett

CSE681 – SW Modeling & Analysis

Spring 2005

References

- Programming Microsoft .Net,
Jeff Prosise, Microsoft Press, 2002
- Web Services, Mark Sapossnek,
Powerpoint presentation available from
www.gotdotnet.com

Web Service Definition

- A web service is a set of methods exposed through a web interface.
  - Accessible through HTTP
  - Provides internet access to RPC-like calls that define the service
  - Web service messages are encoded in an XML dialect called Simple Object Access Protocol (SOAP)
- Service model assumes services are always available

Benefits of the Web Service Model

- Web services use this special architecture because it:
  - Can be used from any platform.
  - Uses a standard, well-know channel.
  - Is routable and will pass through most firewalls.
  - Uses the same security mechanisms as any web site.

Service Oriented Architecture

- Framework provides a set of fundamental operations via web services
  - May also provide local services using Windows services
- All applications based on that framework share the common services
  - Don’t have to recreate the same functionality for each new application
- Can provide those same services to Partner businesses, suppliers, and customers
- Longhorn’s Indigo model is a service oriented architecture

Comparing MicroSoft Web Service with ASP.Net

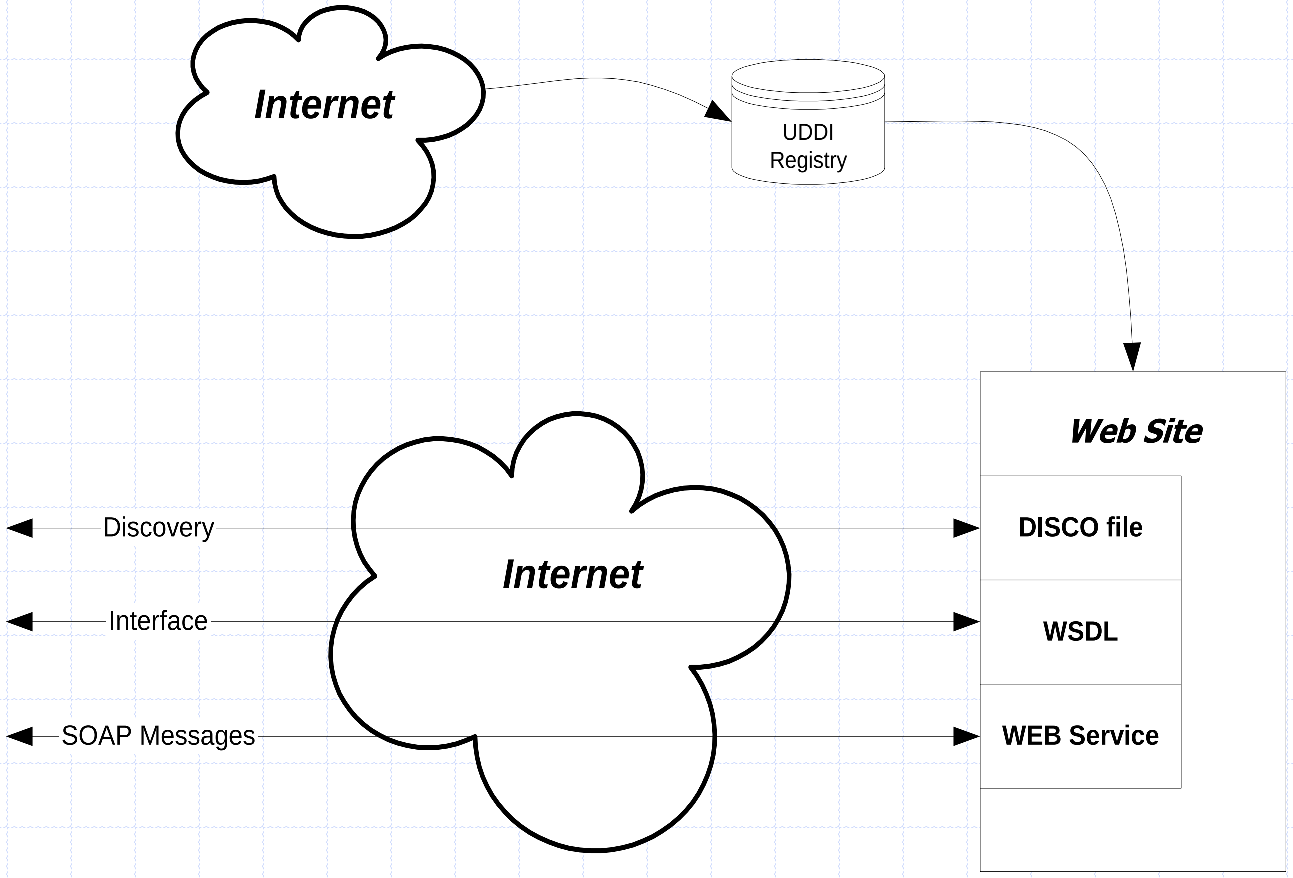
- ASP.NET
  - Uses ASP pipeline
  - Applic.aspx
  - Applic.aspx.cs
  - Uses Session, …
  - Visual Interface invoked from browser

- Web Service
  - Uses ASP pipeline
  - Applic.asmx
  - Applic.asmx.cs
  - Uses Session, …
  - RPC Interface invoked by ASP or Winform app through proxy

Web Service Protocols

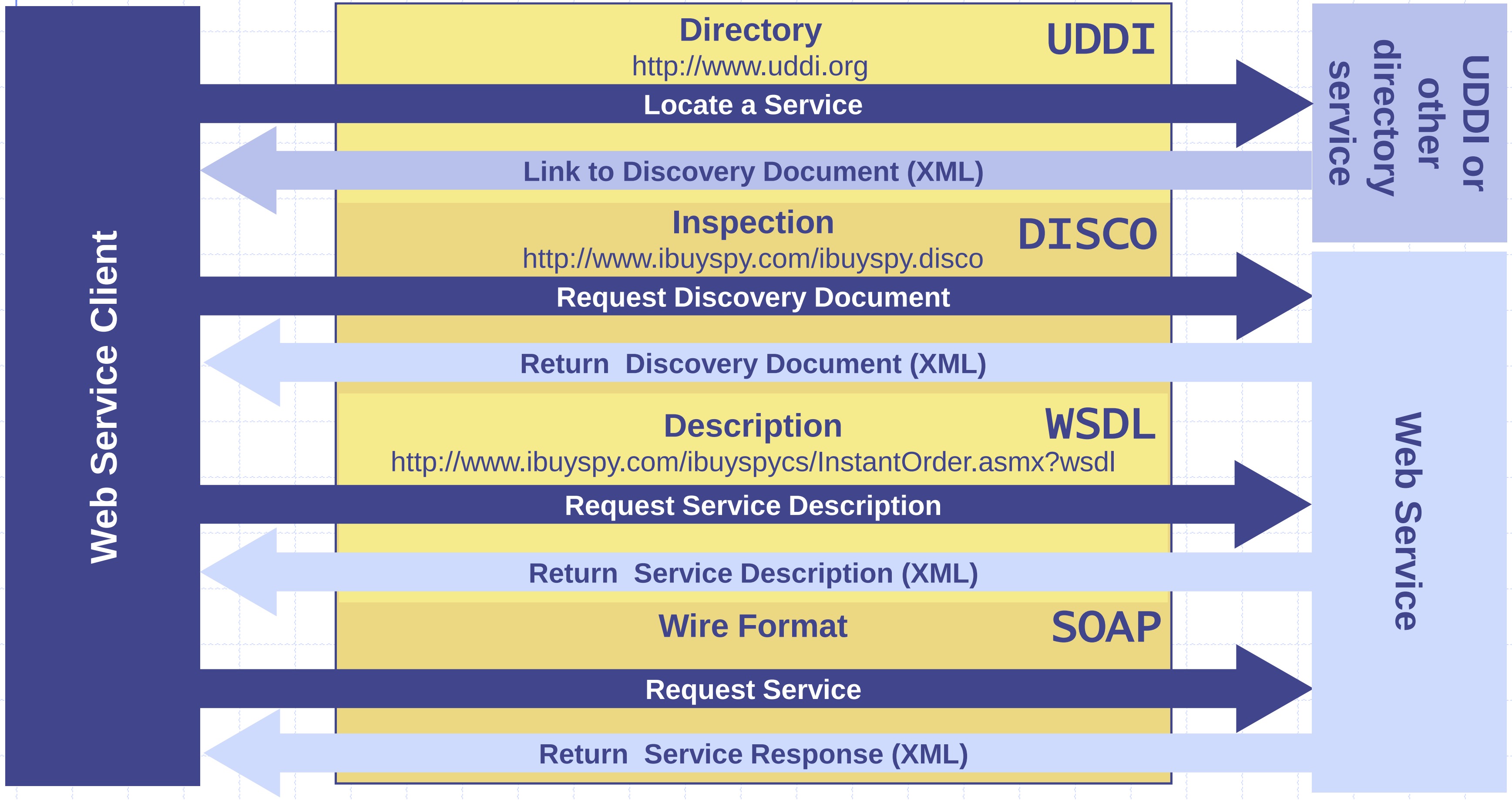
- Web services are based on four protocols:
  - Extensible Markup Language (XML)
    - defines complex data structures
  - Web Service Description Lanaguage (WSDL)
    - Specifies the interface of the web service
  - Discovery Protocol (DISCO)
    - Pointer to all web services on a particular web site
  - Universal Description, Discovery, and Integration (UDDI)
    - Central repository of web service descriptions

Web Service Structure



C# Web Services, Banerjee, et. al., WROX, 2001

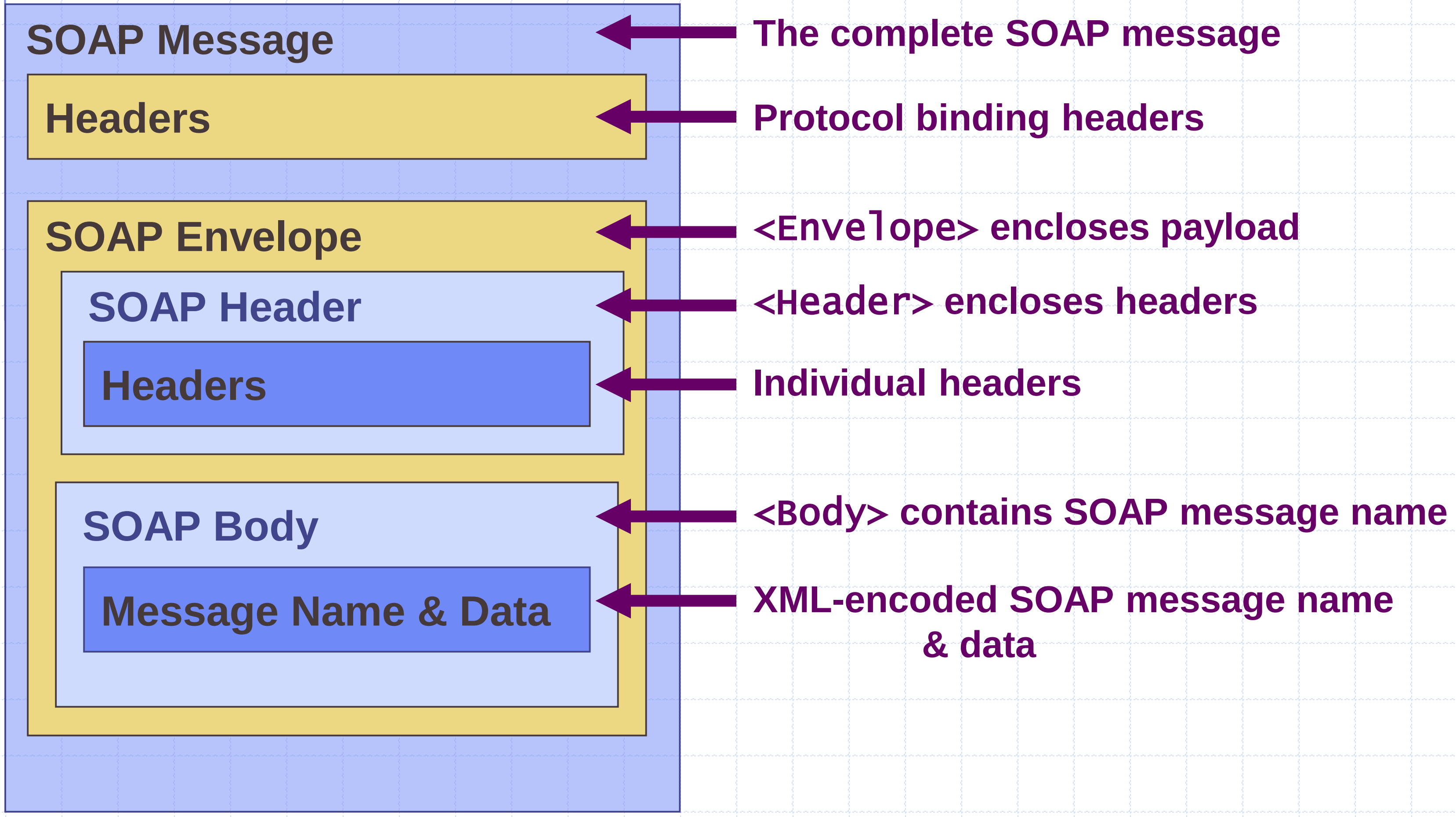
Underlying Technologies
Web Services Stack


Web Service Client
Directory
http://www.uddi.org
UDDI
Locate a Service
Link to Discovery Document (XML)
Inspection
http://www.ibuyspy.com/ibuyspy.disco
DISCO
Request Discovery Document
Return Discovery Document (XML)
Description
WSDL
http://www.ibuyspy.com/ibuyspycs/InstantOrder.asmx?wsdl
Request Service Description
Return Service Description (XML)
Wire Format
SOAP
Request Service
Return Service Response (XML)
UDDI or other directory service
Web Service

SOAP Messages

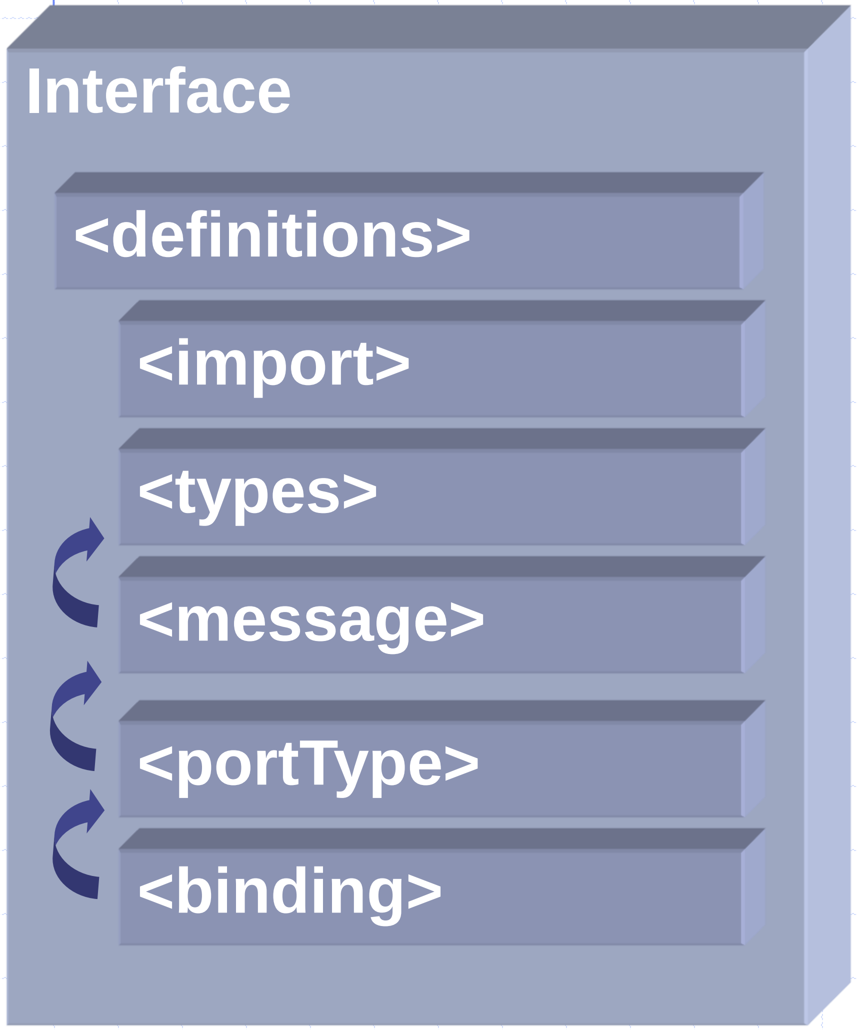
- A SOAP Message can be one of three types:
  - Method call
    - Contains name of method and parameters
  - Method Response
    - Return values
  - Fault Message
    - SOAP fault message if service throws an exception
    - Will get standard HTTP message if transport fails.

SOAP
Message Structure


SOAP Message
Headers
SOAP Envelope
SOAP Header
Headers
SOAP Body
Message Name & Data
The complete SOAP message
Protocol binding headers
<Envelope> encloses payload
<Header> encloses headers
Individual headers
<Body> contains SOAP message name
XML-encoded SOAP message name & data

WSDL

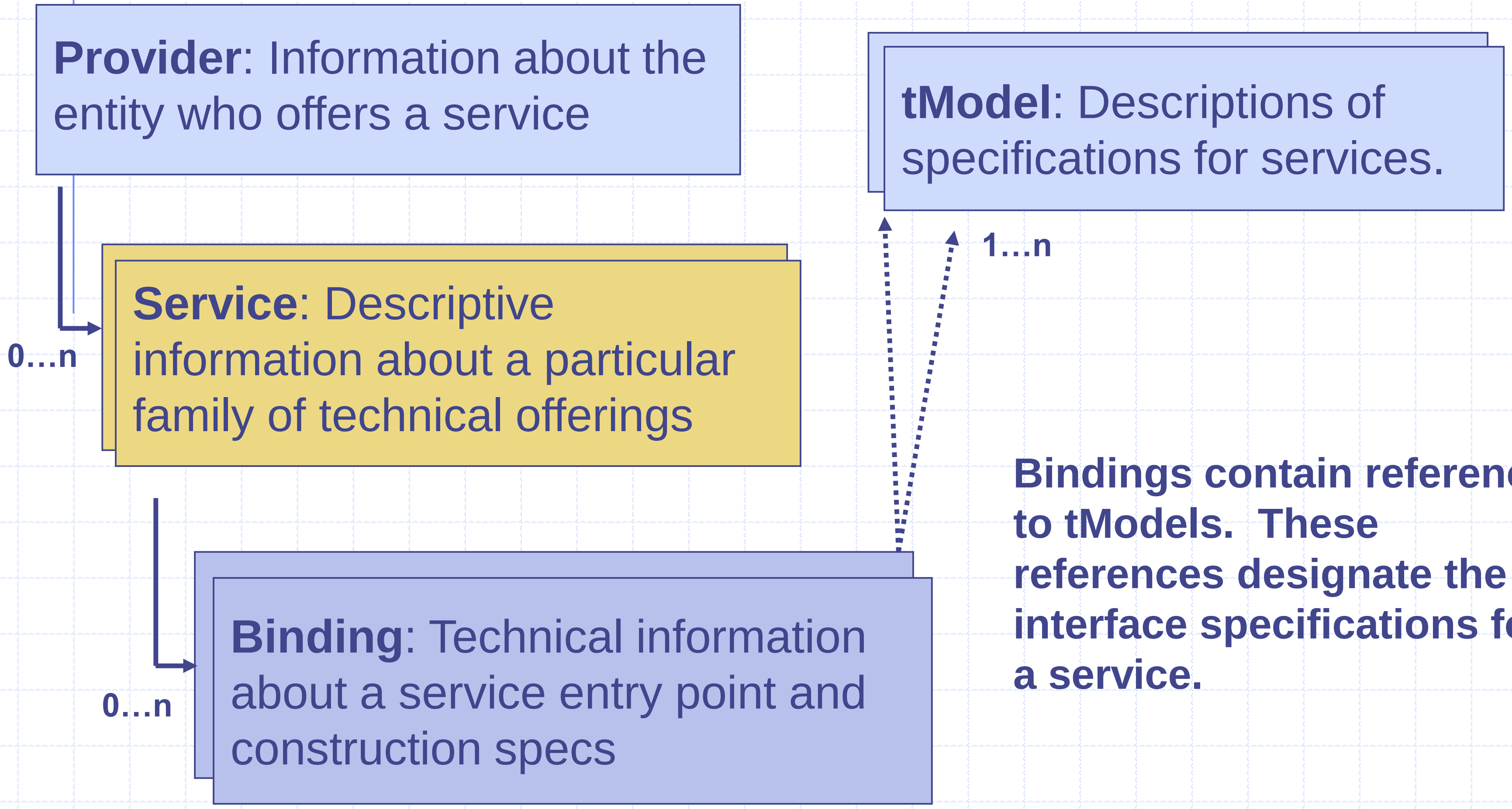
WSDL Schema



- <definitions> are root node of WSDL
- <import> allows other entities for inclusion
- <types> are data definitions - xsd
- <message> defines parameters of a Web Service function
- <portType> defines input and output operations
- <binding> specifies how each message is sent over the wire

UDDI

UDDI Information Model



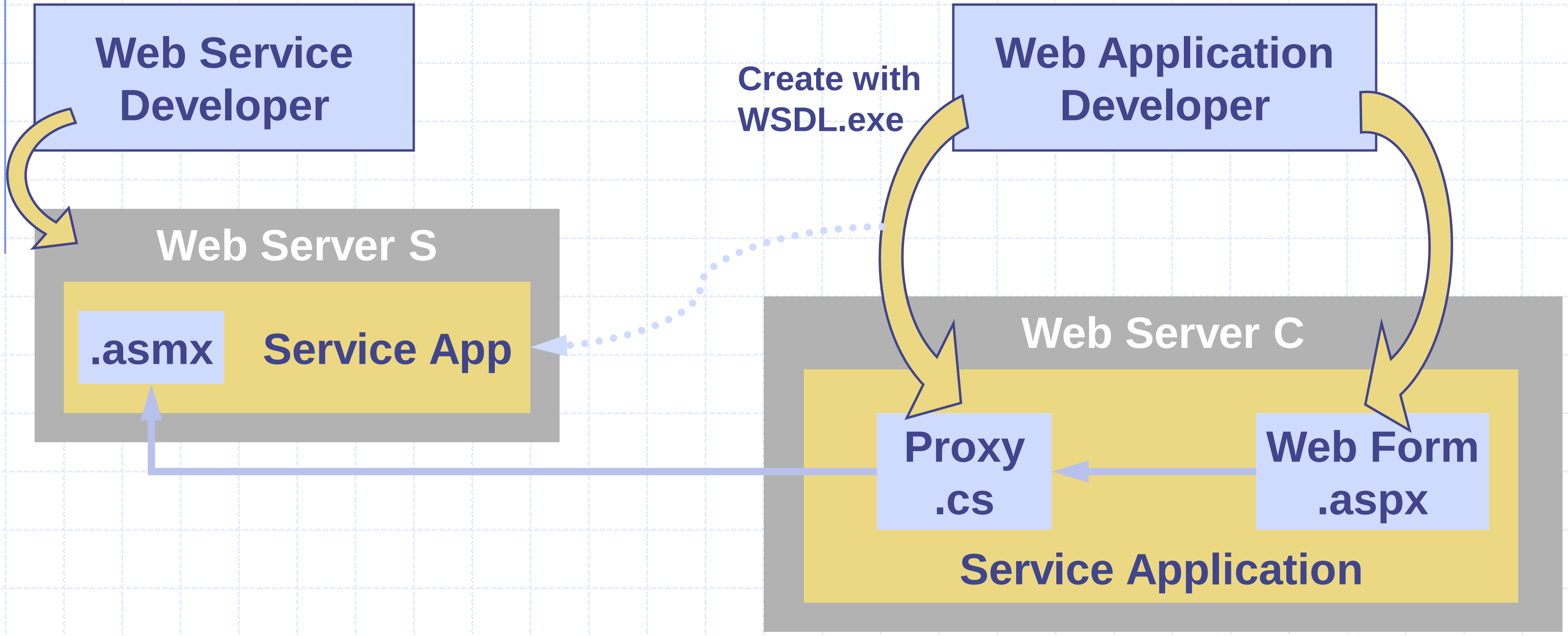
Provider: Information about the entity who offers a service
Service: Descriptive information about a particular family of technical offerings
Binding: Technical information about a service entry point and construction specs
tModel: Descriptions of specifications for services.
0…n
0…n
1…n

Bindings contain references to tModels. These references designate the interface specifications for a service.

Structure of a Microsoft WebService

- MyService.asmx, MyService.asmx.cs
  - Page Directive:
    <%@ Webservice Language="C#" Class="myService" %>
  - Class [derived from System.Web.Services.WebService]
  - Methods decorated with [WebMethod]
- Virtual Directory hosting this Application

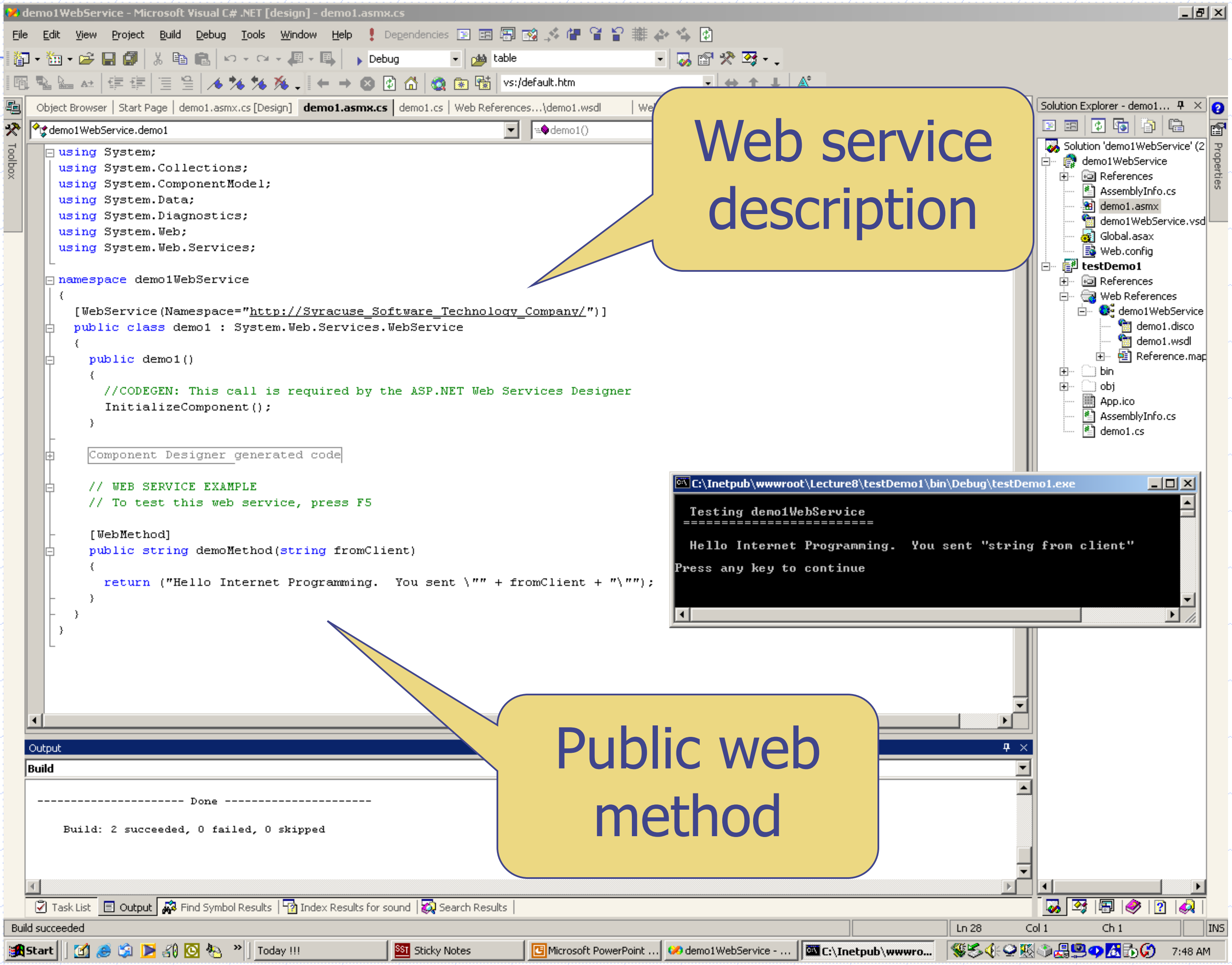
Consuming Web Services



Structure of WebService Client

- myService Proxy code
  - Generated using disco.exe and wsdl.exe (see CalcClient.cs code comments)
- myServiceClient code
  - Ordinary ASP or Winform application
  - myService Proxy = new myService();
  - Result = Proxy.myMethod(args);

DemoWebService Running


demo1WebService - Microsoft Visual C# .NET [design] - demo1.asmx.cs
Web service description
Public web method
Solution Explorer - demo1...
Output
Build succeeded

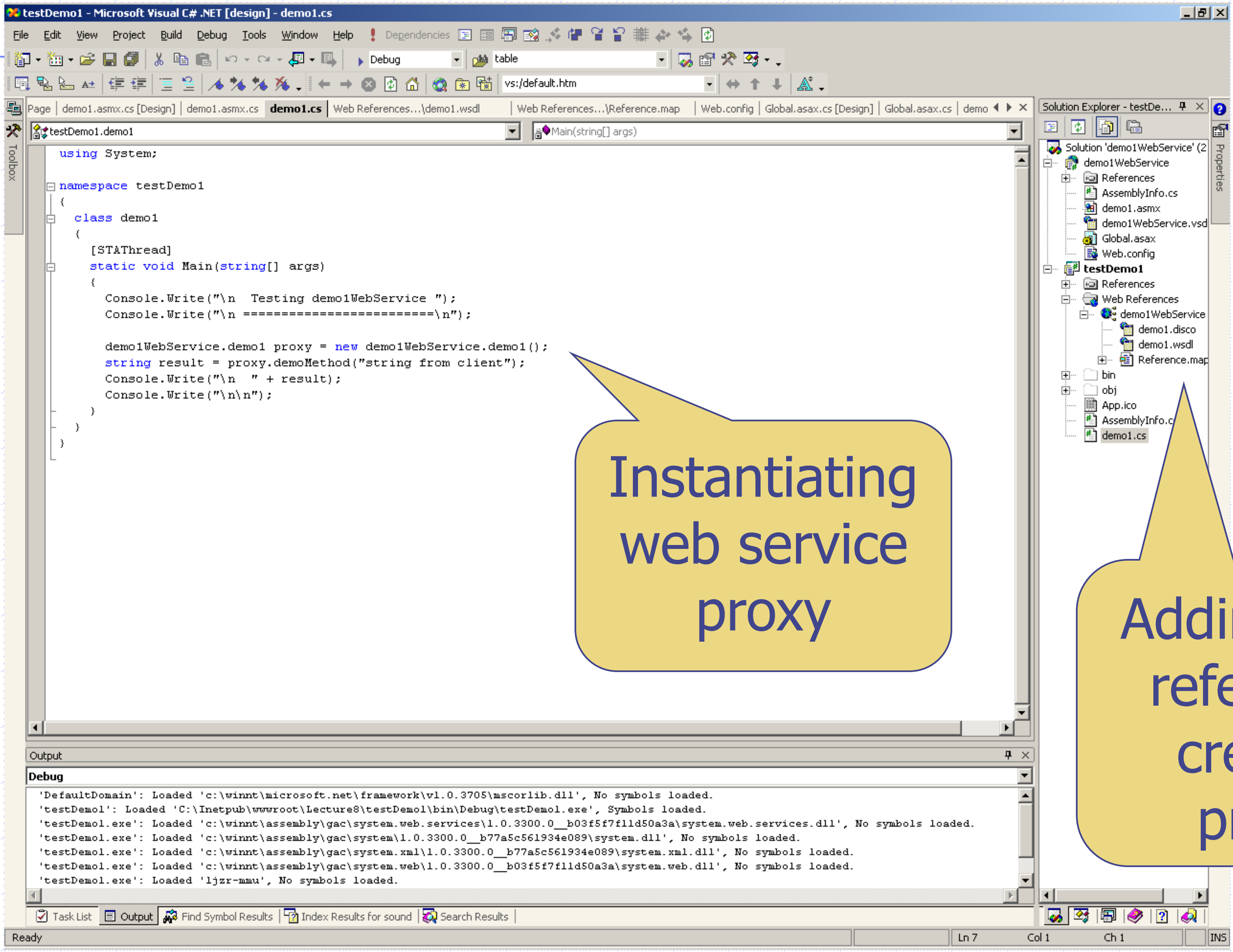
```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;

namespace demo1WebService
{
  [WebService(Namespace="http://Syracuse_Software_Technology_Company/")]
  public class demo1 : System.Web.Services.WebService
  {
    public demo1()
    {
      //CODEGEN: This call is required by the ASP.NET Web Services Designer
      InitializeComponent();
    }

    Component Designer generated code

    // WEB SERVICE EXAMPLE
    // To test this web service, press F5

    [WebMethod]
    public string demoMethod(string fromClient)
    {
      return ("Hello Internet Programming.  You sent \"" + fromClient + "\"");
    }
  }
}
```

C:\Inetpub\wwwroot\Lecture8\testDemo1\bin\Debug\testDemo1.exe

```
 Testing demo1WebService
=========================

 Hello Internet Programming.  You sent "string from client"
Press any key to continue
```

```
---------------------- Done ----------------------

    Build: 2 succeeded, 0 failed, 0 skipped
```

Client of DemoWebService


testDemo1 - Microsoft Visual C# .NET [design] - demo1.cs
File Edit View Project Build Debug Tools Window Help
Debug
table
vs:/default.htm
Page | demo1.asmx.cs [Design] | demo1.asmx.cs | demo1.cs | Web References...\demo1.wsdl | Web References...\Reference.map | Web.config | Global.asax.cs [Design] | Global.asax.cs | demo
testDemo1.demo1
Main(string[] args)
Solution Explorer - testDe...
Solution 'demo1WebService' (2
demo1WebService
References
AssemblyInfo.cs
demo1.asmx
demo1WebService.vsd
Global.asax
Web.config
testDemo1
References
Web References
demo1WebService
demo1.disco
demo1.wsdl
Reference.map
bin
obj
App.ico
AssemblyInfo.c
demo1.cs
Toolbox
Properties
Output
Debug
Task List | Output | Find Symbol Results | Index Results for sound | Search Results
Ready | Ln 7 | Col 1 | Ch 1 | INS

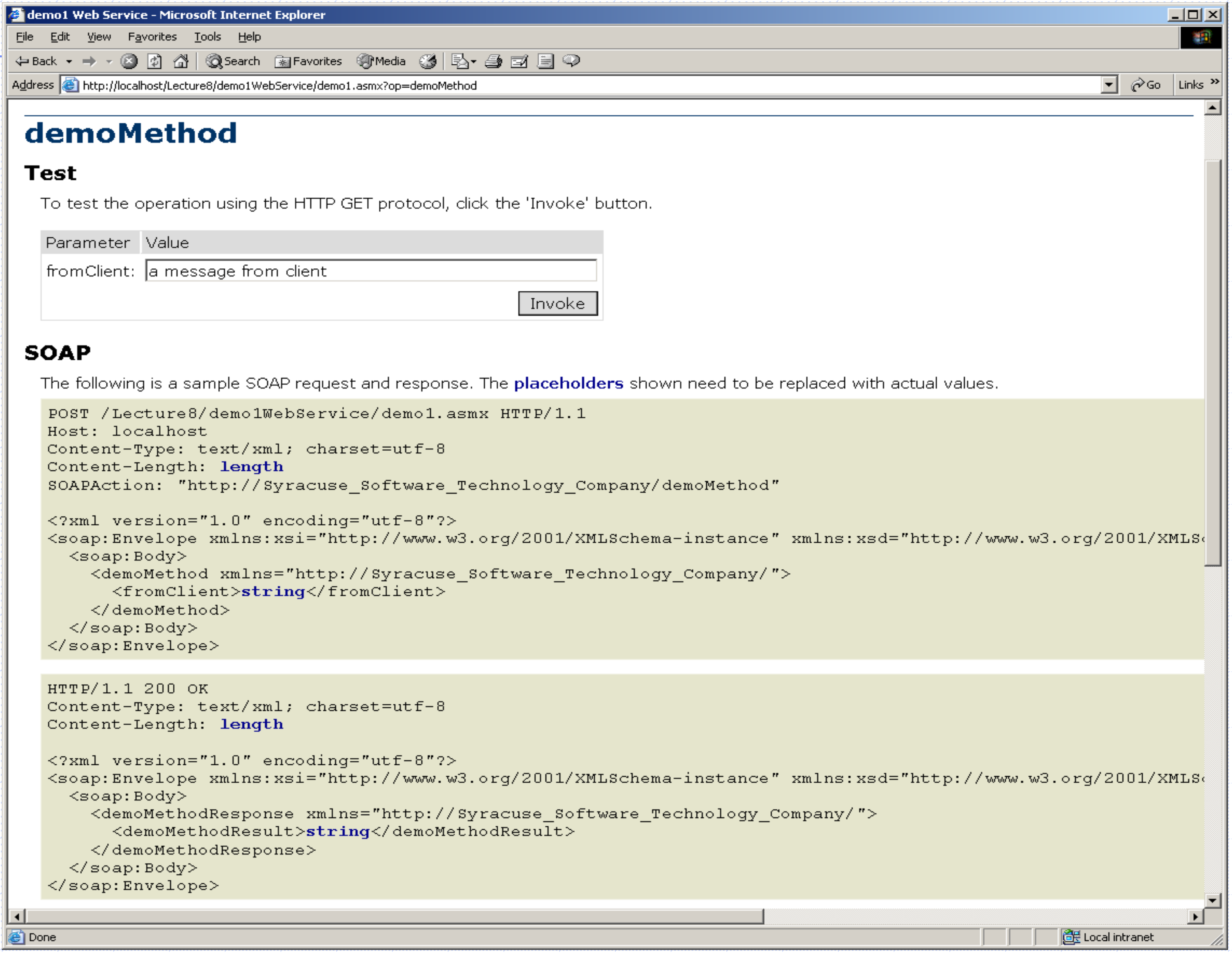
```
using System;

namespace testDemo1
{
  class demo1
  {
    [STAThread]
    static void Main(string[] args)
    {
      Console.Write("\n  Testing demo1WebService ");
      Console.Write("\n ========================\n");

      demo1WebService.demo1 proxy = new demo1WebService.demo1();
      string result = proxy.demoMethod("string from client");
      Console.Write("\n  " + result);
      Console.Write("\n\n");
    }
  }
}
```

```
'DefaultDomain': Loaded 'c:\winnt\microsoft.net\framework\v1.0.3705\mscorlib.dll', No symbols loaded.
'testDemo1': Loaded 'C:\Inetpub\wwwroot\Lecture8\testDemo1\bin\Debug\testDemo1.exe', Symbols loaded.
'testDemo1.exe': Loaded 'c:\winnt\assembly\gac\system.web.services\1.0.3300.0__b03f5f7f11d50a3a\system.web.services.dll', No symbols loaded.
'testDemo1.exe': Loaded 'c:\winnt\assembly\gac\system\1.0.3300.0__b77a5c561934e089\system.dll', No symbols loaded.
'testDemo1.exe': Loaded 'c:\winnt\assembly\gac\system.xml\1.0.3300.0__b77a5c561934e089\system.xml.dll', No symbols loaded.
'testDemo1.exe': Loaded 'c:\winnt\assembly\gac\system.web\1.0.3300.0__b03f5f7f11d50a3a\system.web.dll', No symbols loaded.
'testDemo1.exe': Loaded 'ljzr-mmu', No symbols loaded.
```

Instantiating web service proxy

Adding web reference creates proxy

SOAP Request and Response


demo1 Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/Lecture8/demo1WebService/demo1.asmx?op=demoMethod
Go Links
Done
Local intranet

demoMethod

Test

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

Parameter	Value
fromClient:	a message from client

Invoke

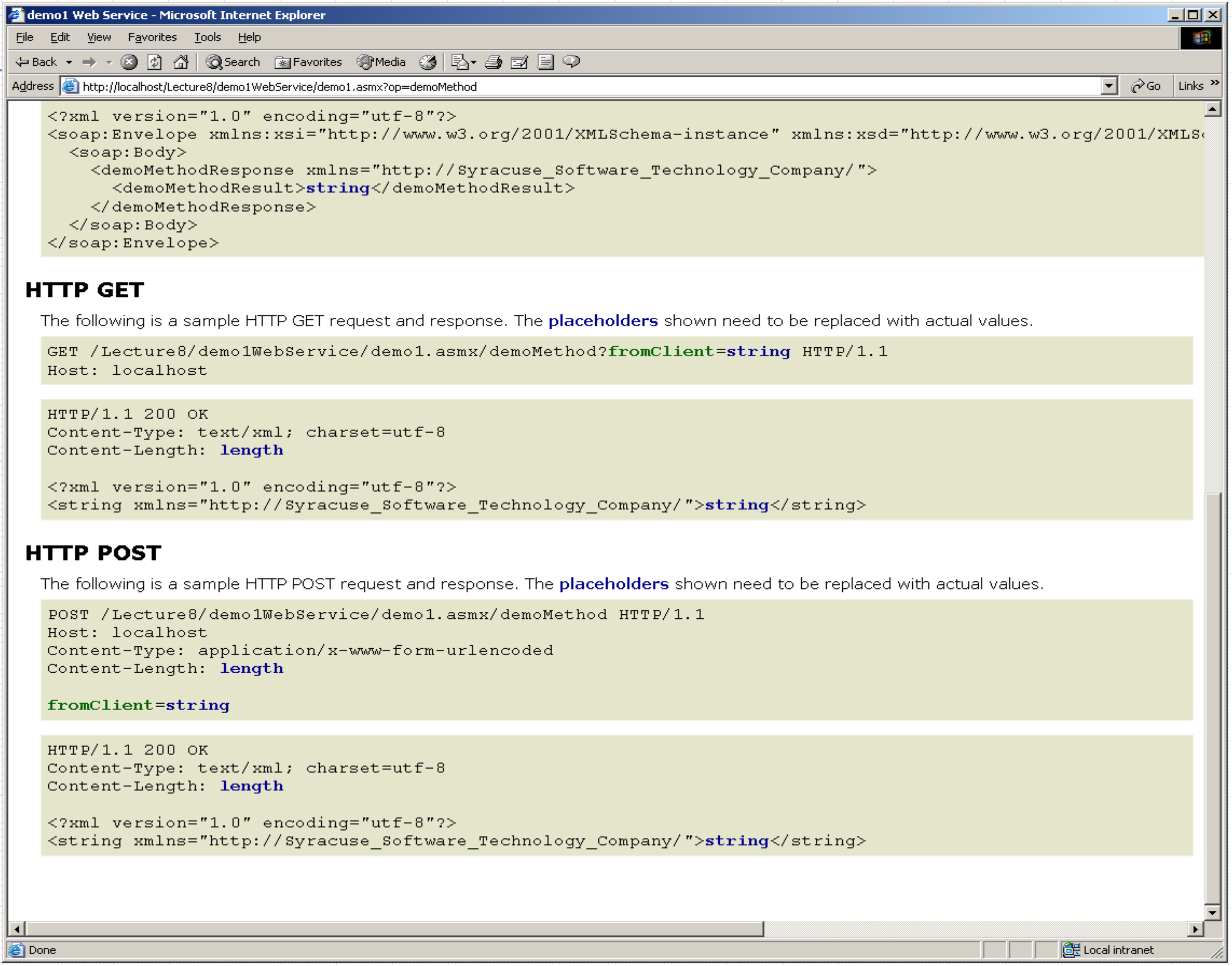
SOAP

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

```
POST /Lecture8/demo1WebService/demo1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Syracuse_Software_Technology_Company/demoMethod"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLS
  <soap:Body>
    <demoMethod xmlns="http://Syracuse_Software_Technology_Company/">
      <fromClient>string</fromClient>
    </demoMethod>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLS
  <soap:Body>
    <demoMethodResponse xmlns="http://Syracuse_Software_Technology_Company/">
      <demoMethodResult>string</demoMethodResult>
    </demoMethodResponse>
  </soap:Body>
</soap:Envelope>
```

HTTP GET and POST exchanges


demo1 Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/Lecture8/demo1WebService/demo1.asmx?op=demoMethod
Go Links
Done
Local intranet

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLS
  <soap:Body>
    <demoMethodResponse xmlns="http://Syracuse_Software_Technology_Company/">
      <demoMethodResult>string</demoMethodResult>
    </demoMethodResponse>
  </soap:Body>
</soap:Envelope>
```

HTTP GET

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

```
GET /Lecture8/demo1WebService/demo1.asmx/demoMethod?fromClient=string HTTP/1.1
Host: localhost
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://Syracuse_Software_Technology_Company/">string</string>
```

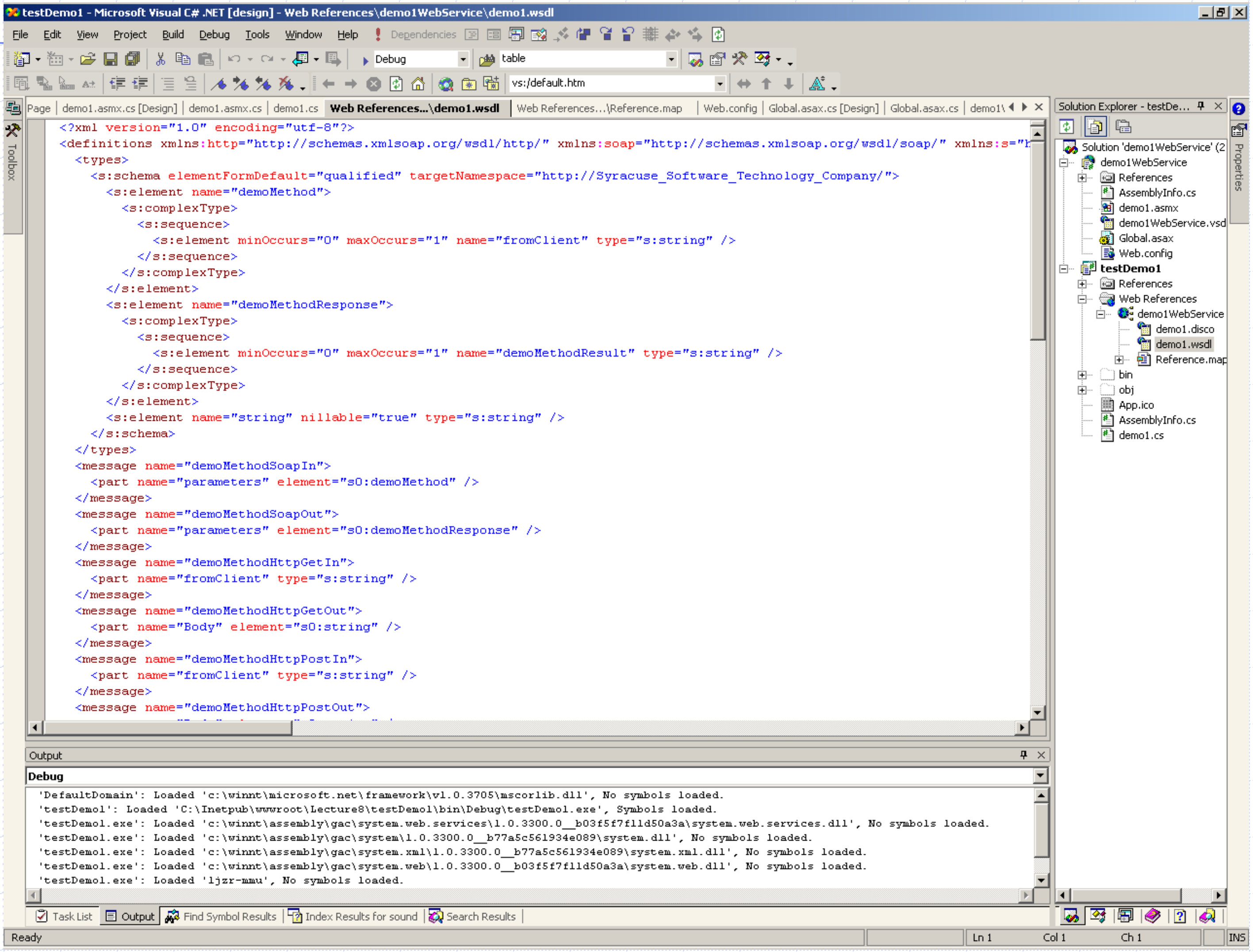
HTTP POST

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

```
POST /Lecture8/demo1WebService/demo1.asmx/demoMethod HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

fromClient=string
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://Syracuse_Software_Technology_Company/">string</string>
```

demo1.wsdl

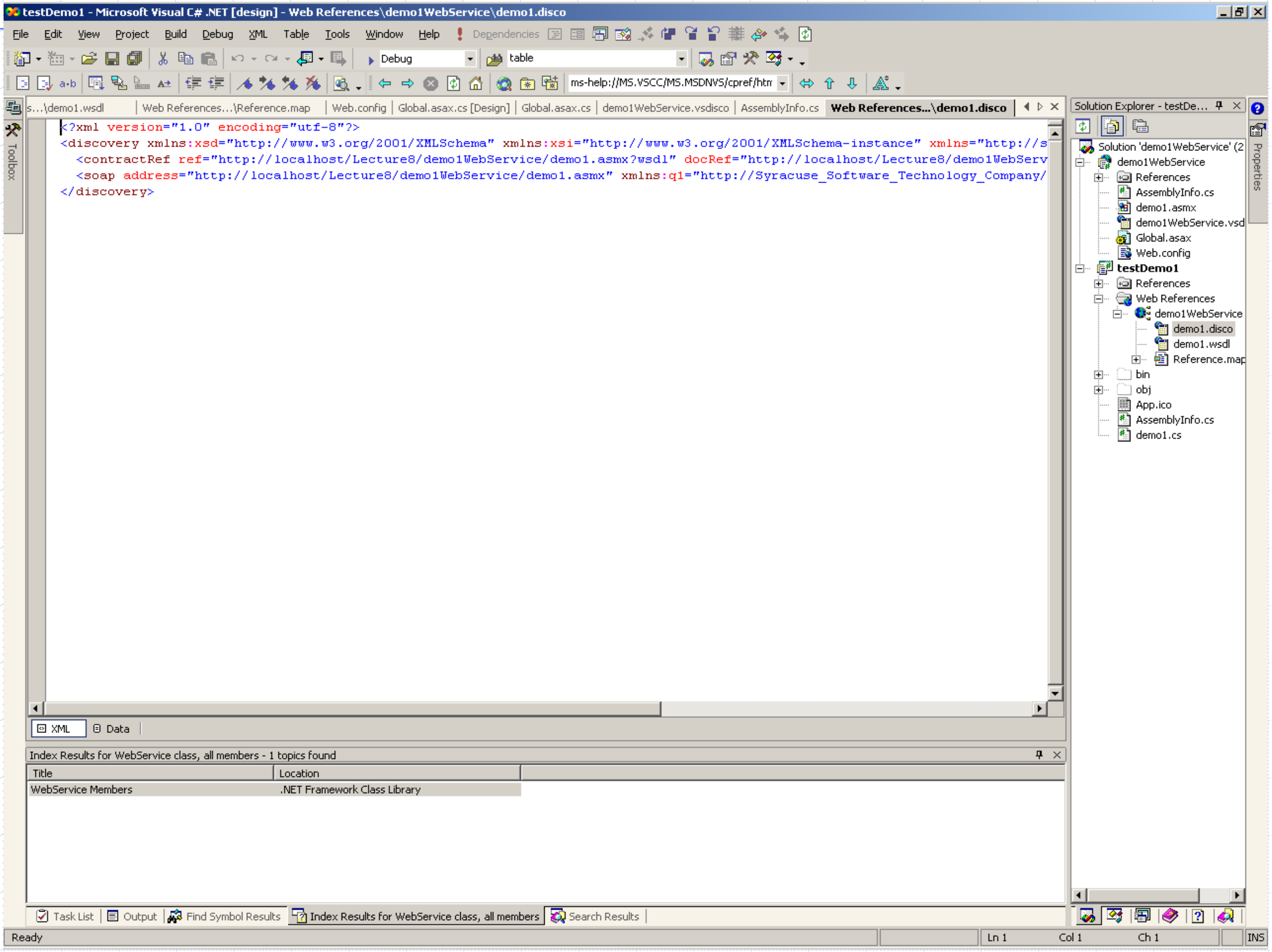


```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="h
  <types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://Syracuse_Software_Technology_Company/">
      <s:element name="demoMethod">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="fromClient" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="demoMethodResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="demoMethodResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="string" nillable="true" type="s:string" />
    </s:schema>
  </types>
  <message name="demoMethodSoapIn">
    <part name="parameters" element="s0:demoMethod" />
  </message>
  <message name="demoMethodSoapOut">
    <part name="parameters" element="s0:demoMethodResponse" />
  </message>
  <message name="demoMethodHttpGetIn">
    <part name="fromClient" type="s:string" />
  </message>
  <message name="demoMethodHttpGetOut">
    <part name="Body" element="s0:string" />
  </message>
  <message name="demoMethodHttpPostIn">
    <part name="fromClient" type="s:string" />
  </message>
  <message name="demoMethodHttpPostOut">
```

```
'DefaultDomain': Loaded 'c:\winnt\microsoft.net\framework\v1.0.3705\mscorlib.dll', No symbols loaded.
'testDemol': Loaded 'C:\Inetpub\wwwroot\Lecture8\testDemol\bin\Debug\testDemol.exe', Symbols loaded.
'testDemol.exe': Loaded 'c:\winnt\assembly\gac\system.web.services\1.0.3300.0__b03f5f7f11d50a3a\system.web.services.dll', No symbols loaded.
'testDemol.exe': Loaded 'c:\winnt\assembly\gac\system\1.0.3300.0__b77a5c561934e089\system.dll', No symbols loaded.
'testDemol.exe': Loaded 'c:\winnt\assembly\gac\system.xml\1.0.3300.0__b77a5c561934e089\system.xml.dll', No symbols loaded.
'testDemol.exe': Loaded 'c:\winnt\assembly\gac\system.web\1.0.3300.0__b03f5f7f11d50a3a\system.web.dll', No symbols loaded.
'testDemol.exe': Loaded 'ljzr-mmu', No symbols loaded.
```

testDemo1 - Microsoft Visual C# .NET [design] - Web References\demo1WebService\demo1.wsdl
File Edit View Project Build Debug Tools Window Help
Debug
table
vs:/default.htm
Page | demo1.asmx.cs [Design] | demo1.asmx.cs | demo1.cs | Web References...\demo1.wsdl | Web References...\Reference.map | Web.config | Global.asax.cs [Design] | Global.asax.cs
Solution Explorer - testDe...
Solution 'demo1WebService' (2
demo1WebService
References
AssemblyInfo.cs
demo1.asmx
demo1WebService.vsd
Global.asax
Web.config
testDemo1
References
Web References
demo1WebService
demo1.disco
demo1.wsdl
Reference.map
bin
obj
App.ico
AssemblyInfo.cs
demo1.cs
Output
Debug
Task List | Output | Find Symbol Results | Index Results for sound | Search Results
Ready
Ln 1
Col 1
Ch 1
INS

demo1.disco



testDemo1 - Microsoft Visual C# .NET [design] - Web References\demo1WebService\demo1.disco

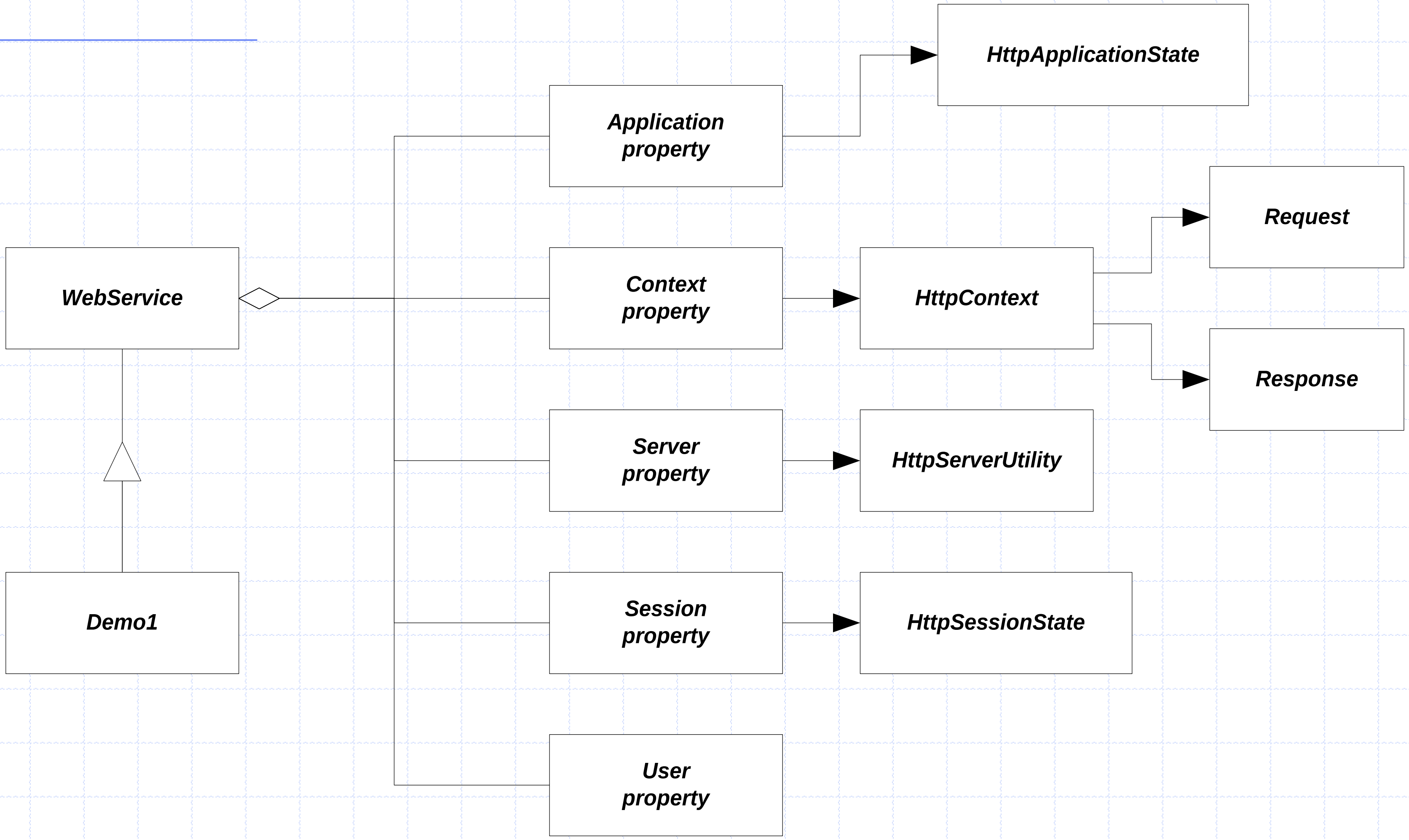
```
<?xml version="1.0" encoding="utf-8"?>
<discovery xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://s
  <contractRef ref="http://localhost/Lecture8/demo1WebService/demo1.asmx?wsdl" docRef="http://localhost/Lecture8/demo1WebServ
  <soap address="http://localhost/Lecture8/demo1WebService/demo1.asmx" xmlns:q1="http://Syracuse_Software_Technology_Company/
</discovery>
```

Index Results for WebService class, all members - 1 topics found

Title	Location
WebService Members	.NET Framework Class Library

File Edit View Project Build Debug XML Table Tools Window Help
Debug
table
s...\demo1.wsdl | Web References...\Reference.map | Web.config | Global.asax.cs [Design] | Global.asax.cs | demo1WebService.vsdisco | AssemblyInfo.cs | Web References...\demo1.disco
XML Data
Solution Explorer - testDe...
Solution 'demo1WebService' (2
demo1WebService
References
AssemblyInfo.cs
demo1.asmx
demo1WebService.vsd
Global.asax
Web.config
testDemo1
References
Web References
demo1WebService
demo1.disco
demo1.wsdl
Reference.map
bin
obj
App.ico
AssemblyInfo.cs
demo1.cs
Task List | Output | Find Symbol Results | Index Results for WebService class, all members | Search Results
Ready
Ln 1 Col 1 Ch 1 INS

Web Service Application Structure



WebService Properties

- HttpApplicationState
  - Share state among all users of an application.
- HttpSessionState
  - Share state from page to page for one user.
- HttpContext
  - Provides access to the server Request and Response objects.
- HttpServerUtility
  - Provides CreateObject, Execute, and MapPath methods.
- User
  - Supports authentication of user.

WebMethods

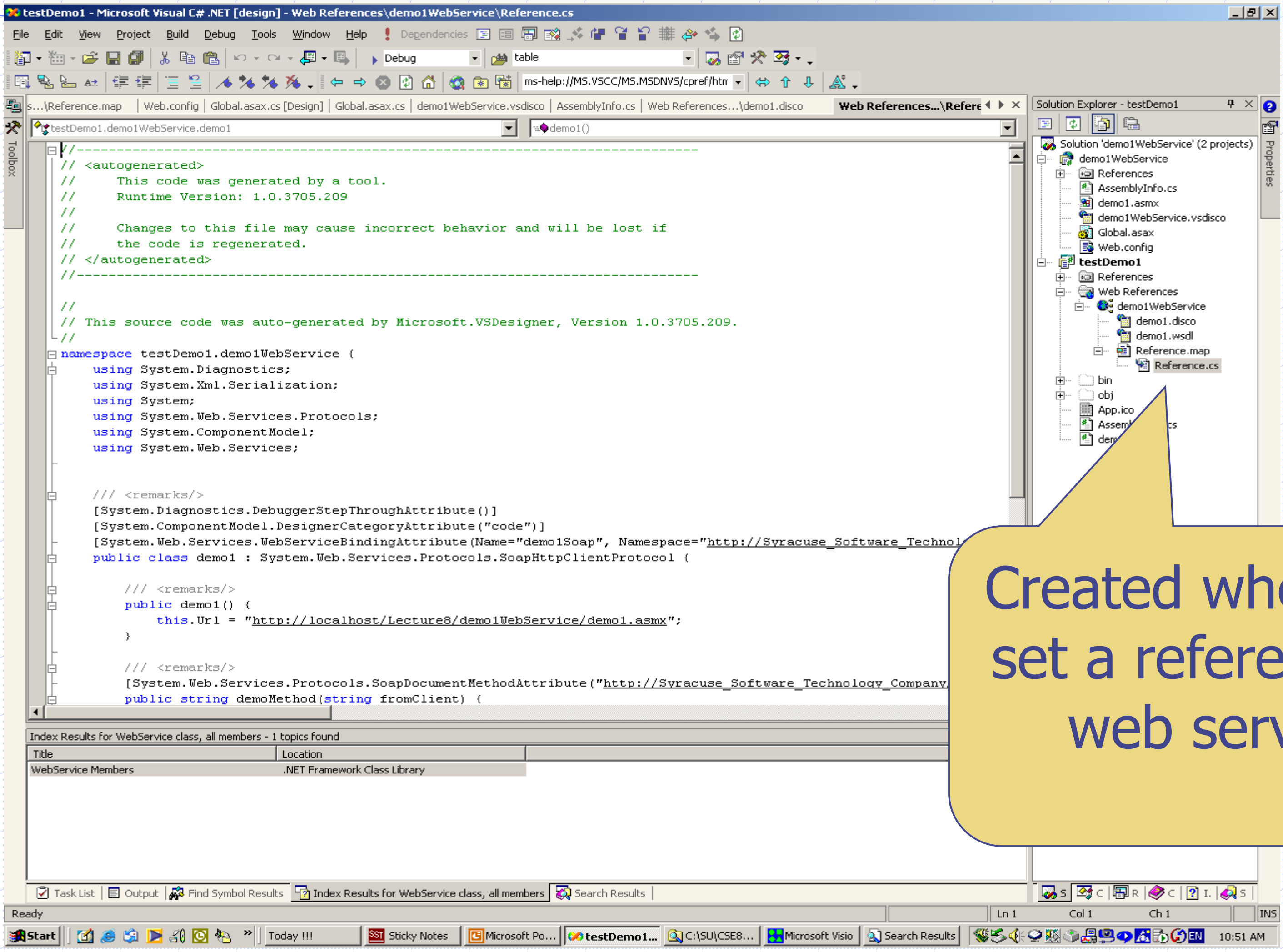
- WebMethod methods can pass many of the C# and CLR types
- User defined objects can also be passed if they are serializable:
  - .Net XML serializer will not serialize non-public members
    - Due to limitations of WSDL language
  - User defined types can only be passed with SOAP. GET and POST won’t work.
  - The WSDL contract contains a schema description of any user defined objects passed by a WebMethod

Web Service Clients

- Web Service Clients use Web Service proxies to communicate with the remote service:

```
// create proxy instance
demo1WebService.demo1 proxy = new demo1WebService.demo1();

// use proxy
string result = proxy.demoMethod("string from client");
```

AutoGenerated Proxy


testDemo1 - Microsoft Visual C# .NET [design] - Web References\demo1WebService\Reference.cs
File Edit View Project Build Debug Tools Window Help
Solution Explorer - testDemo1
Solution 'demo1WebService' (2 projects)
demo1WebService
References
AssemblyInfo.cs
demo1.asmx
demo1WebService.vsdisco
Global.asax
Web.config
testDemo1
References
Web References
demo1WebService
demo1.disco
demo1.wsdl
Reference.map
Reference.cs
bin
obj
App.ico

```
//------------------------------------------------------------------------------
// <autogenerated>
//     This code was generated by a tool.
//     Runtime Version: 1.0.3705.209
//
//     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 1.0.3705.209.
//
namespace testDemo1.demo1WebService {
    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="demo1Soap", Namespace="http://Syracuse_Software_Technol
    public class demo1 : System.Web.Services.Protocols.SoapHttpClientProtocol {

        /// <remarks/>
        public demo1() {
            this.Url = "http://localhost/Lecture8/demo1WebService/demo1.asmx";
        }

        /// <remarks/>
        [System.Web.Services.Protocols.SoapDocumentMethodAttribute("http://Syracuse_Software_Technology_Company
        public string demoMethod(string fromClient) {
```

Index Results for WebService class, all members - 1 topics found

Title	Location
WebService Members	.NET Framework Class Library

Created when you set a reference to web service

Web Services versus Remoting

- Web Services:
  - Can be used by any platform that understands XML, SOAP, and WSDL.
    - Metadata (types) provided by WSDL
  - Hosted by IIS and inherits ASP's security model.
  - Uses HTTP protocol so accessible by web pages and can pass through most firewalls.
  - Can only pass a limited set of user-defined objects:
    - Can't serialize an object graph or all .Net containers.

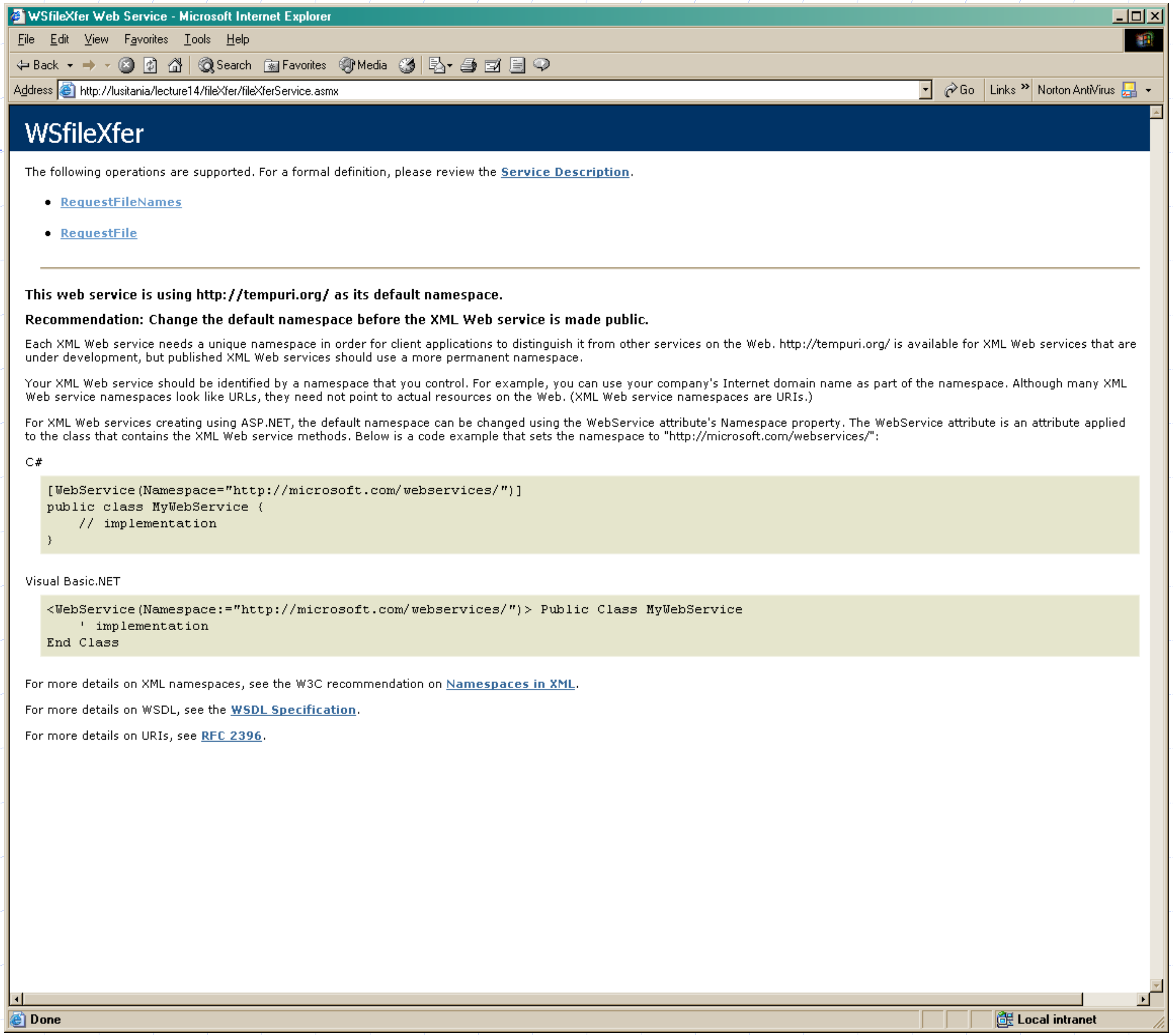
Web Services versus Remoting

- Remoting:
  - Requires .Net platform on client as well as server.
  - Requires custom security (notoriously hard to get right).
  - Metadata provided by assembly.
    - Can pass any .Net type, including object graphs and all .Net containers.
    - Rich, but none portable types.

An Example

- FileXferService
  - Public Interface:
    - string[] RequestFileNames();
    - Byte[] RequestFile(string FileName);

Browser View


WSfileXfer Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://lusitania/lecture14/fileXfer/fileXferService.asmx
Go Links Norton AntiVirus
Done
Local intranet

WSfileXfer

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

- RequestFileNames
- RequestFile

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.NET

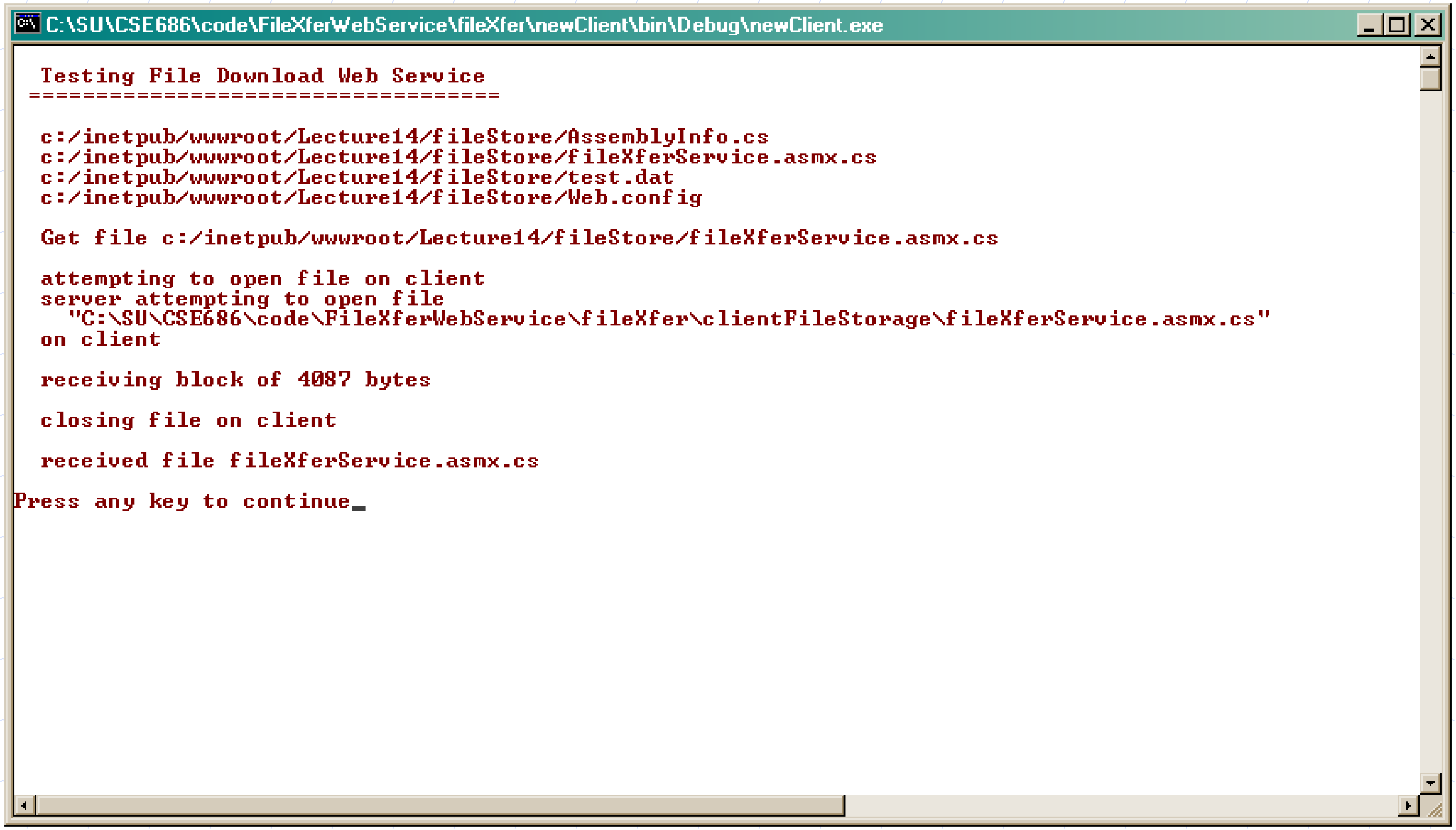
```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

For more details on XML namespaces, see the W3C recommendation on Namespaces in XML.

For more details on WSDL, see the WSDL Specification.

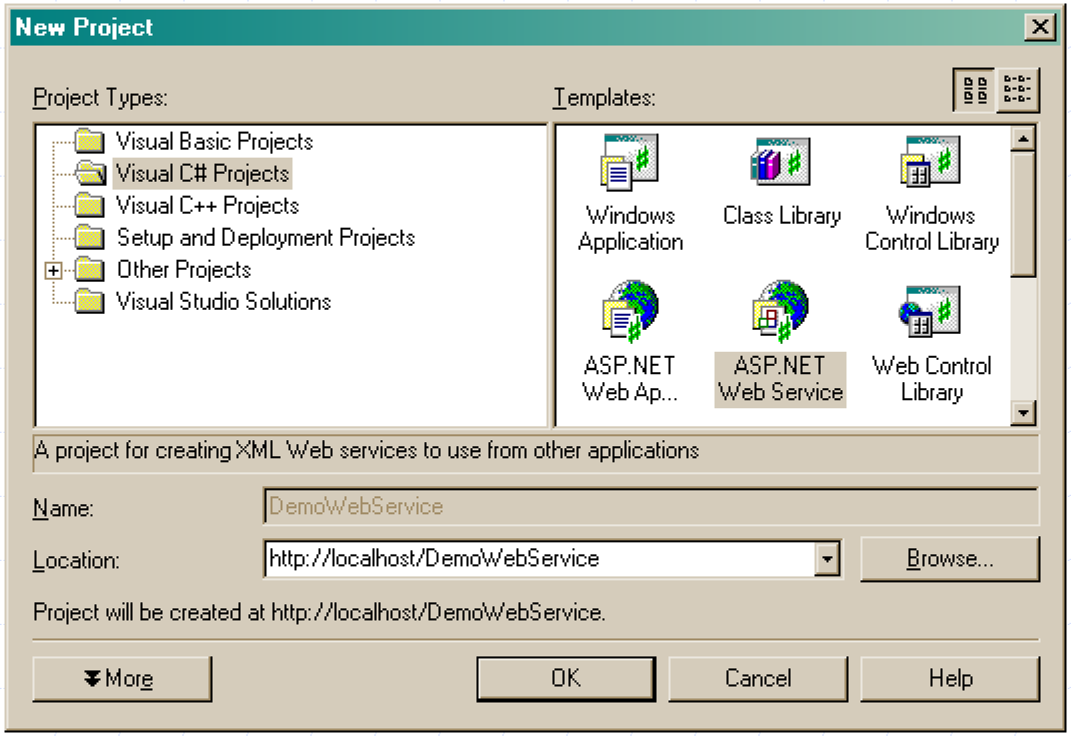
For more details on URIs, see RFC 2396.

Client Application View


C:\SU\CSE686\code\FileXferWebService\fileXfer\newClient\bin\Debug\newClient.exe

```
  Testing File Download Web Service
 ===================================

  c:/inetpub/wwwroot/Lecture14/fileStore/AssemblyInfo.cs
  c:/inetpub/wwwroot/Lecture14/fileStore/fileXferService.asmx.cs
  c:/inetpub/wwwroot/Lecture14/fileStore/test.dat
  c:/inetpub/wwwroot/Lecture14/fileStore/Web.config

  Get file c:/inetpub/wwwroot/Lecture14/fileStore/fileXferService.asmx.cs

  attempting to open file on client
  server attempting to open file
    "C:\SU\CSE686\code\FileXferWebService\fileXfer\clientFileStorage\fileXferService.asmx.cs"
  on client

  receiving block of 4087 bytes

  closing file on client

  received file fileXferService.asmx.cs

Press any key to continue_
```

Creating a Web Service Project



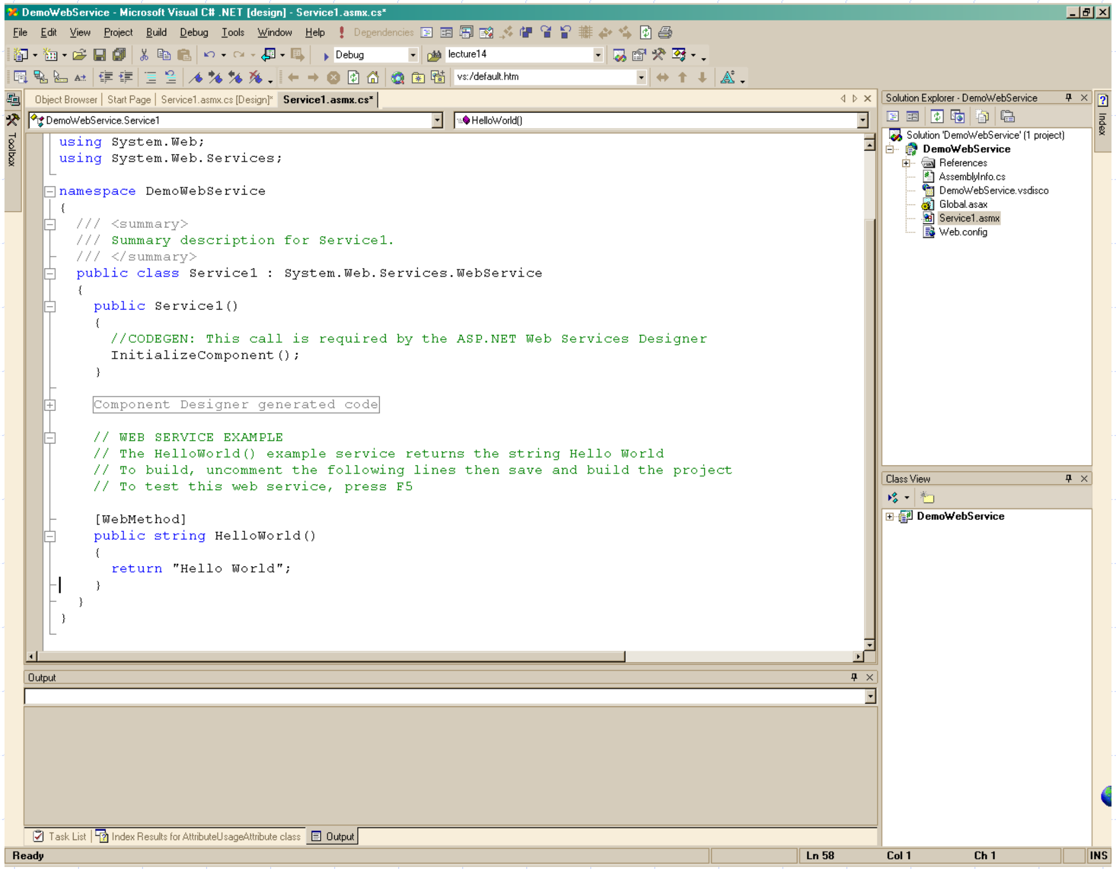
A project for creating XML Web services to use from other applications

Name: DemoWebService

Location: http://localhost/DemoWebService

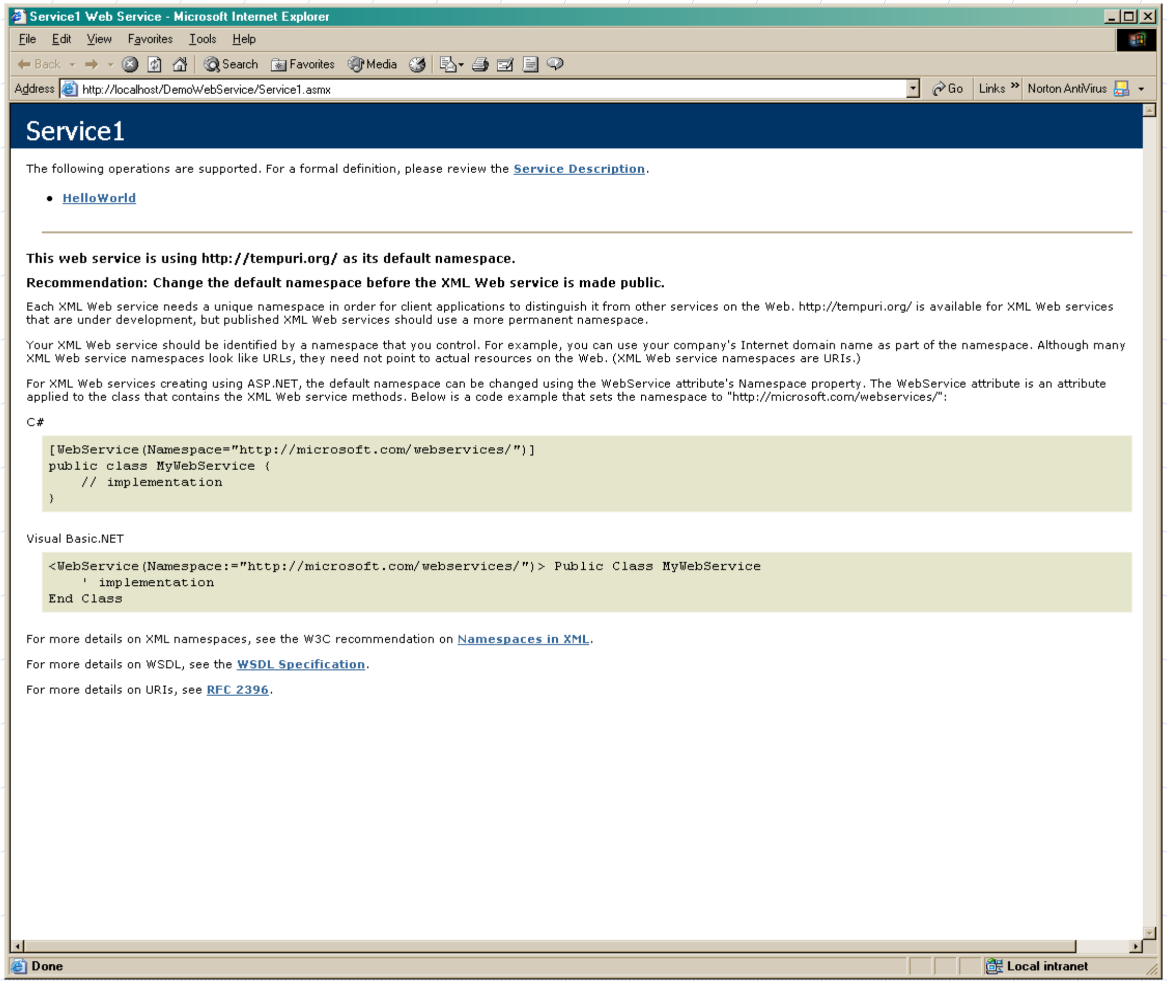
Project will be created at http://localhost/DemoWebService.

Resulting "Generic" Web Service


DemoWebService - Microsoft Visual C# .NET [design] - Service1.asmx.cs*
File Edit View Project Build Debug Tools Window Help
Object Browser | Start Page | Service1.asmx.cs [Design]* | Service1.asmx.cs*
DemoWebService.Service1
HelloWorld()
Solution Explorer - DemoWebService
Solution 'DemoWebService' (1 project)
DemoWebService
References
AssemblyInfo.cs
DemoWebService.vsdisco
Global.asax
Service1.asmx
Web.config
Class View
DemoWebService
Output
Task List | Index Results for AttributeUsageAttribute class | Output
Ready Ln 58 Col 1 Ch 1 INS

```
using System.Web;
using System.Web.Services;

namespace DemoWebService
{
  /// <summary>
  /// Summary description for Service1.
  /// </summary>
  public class Service1 : System.Web.Services.WebService
  {
    public Service1()
    {
      //CODEGEN: This call is required by the ASP.NET Web Services Designer
      InitializeComponent();
    }

    Component Designer generated code

    // WEB SERVICE EXAMPLE
    // The HelloWorld() example service returns the string Hello World
    // To build, uncomment the following lines then save and build the project
    // To test this web service, press F5

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

Resulting "Generic" Test


Service1 Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/DemoWebService/Service1.asmx
Go Links Norton AntiVirus
Done
Local intranet

Service1

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
}
```

Visual Basic.NET

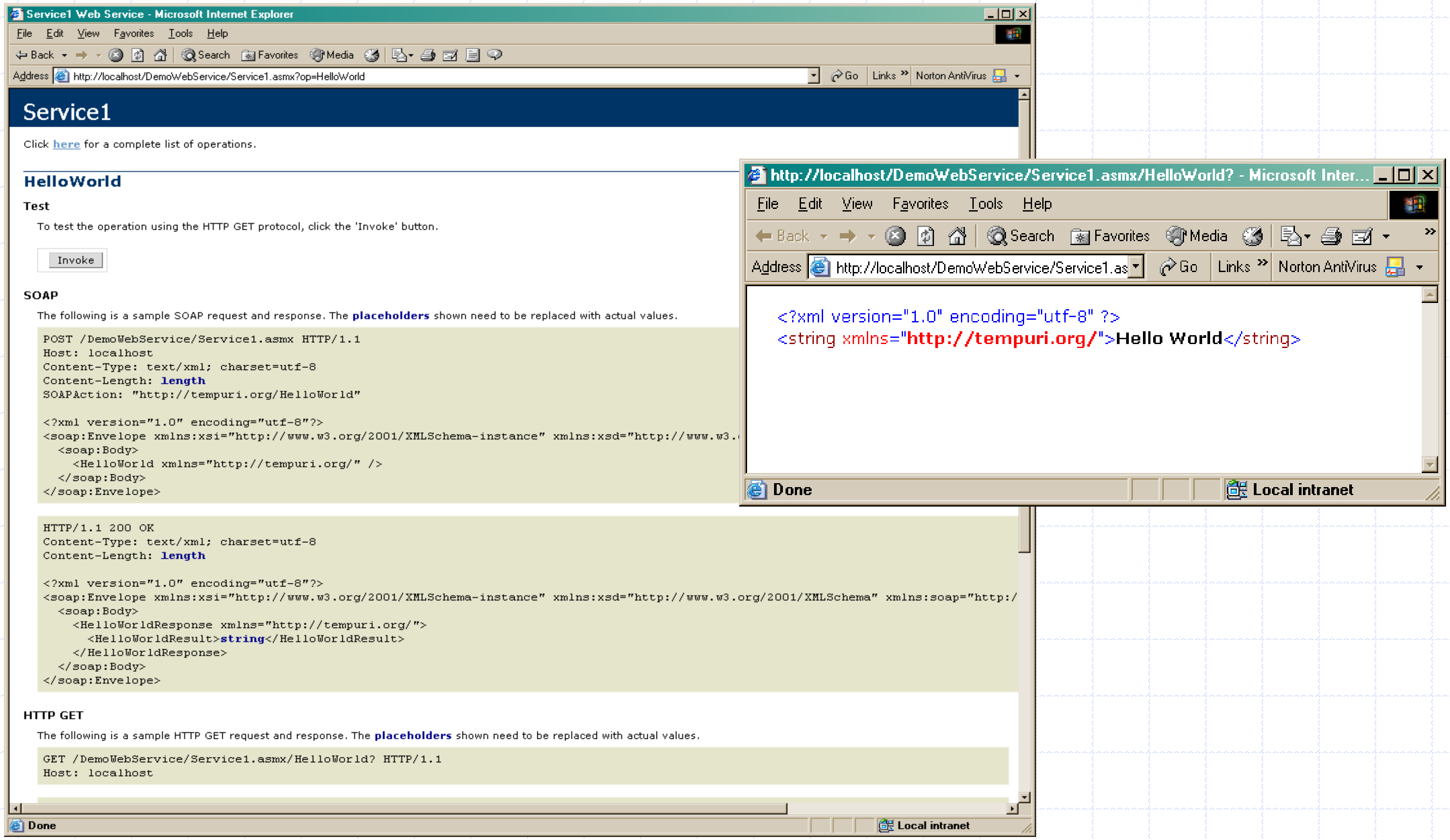
```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

For more details on XML namespaces, see the W3C recommendation on Namespaces in XML.

For more details on WSDL, see the WSDL Specification.

For more details on URIs, see RFC 2396.

Sample Soap Request


Service1 Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/DemoWebService/Service1.asmx?op=HelloWorld
Go Links Norton AntiVirus

Service1

Click here for a complete list of operations.

HelloWorld

Test

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

Invoke

SOAP

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

```
POST /DemoWebService/Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/HelloWorld"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.
  <soap:Body>
    <HelloWorld xmlns="http://tempuri.org/" />
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http:/
  <soap:Body>
    <HelloWorldResponse xmlns="http://tempuri.org/">
      <HelloWorldResult>string</HelloWorldResult>
    </HelloWorldResponse>
  </soap:Body>
</soap:Envelope>
```

HTTP GET

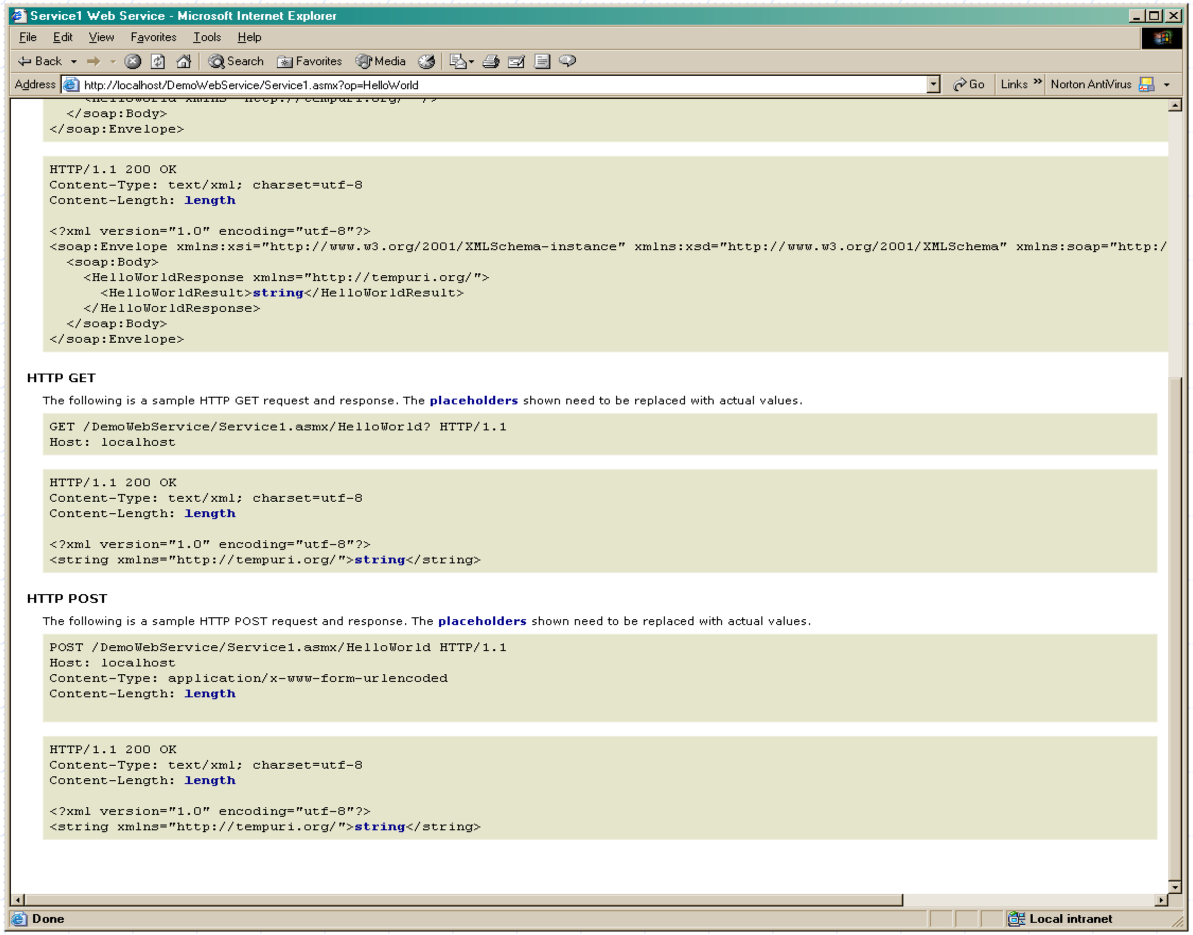
The following is a sample HTTP GET request and response. The **placeholders** shown need to be replaced with actual values.

```
GET /DemoWebService/Service1.asmx/HelloWorld? HTTP/1.1
Host: localhost
```

Done
Local intranet
http://localhost/DemoWebService/Service1.asmx/HelloWorld? - Microsoft Inter...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/DemoWebService/Service1.as
Go Links Norton AntiVirus

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://tempuri.org/">Hello World</string>
```

Sample “Generic” HTTP GET and POST


Service1 Web Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/DemoWebService/Service1.asmx?op=HelloWorld
Go Links Norton AntiVirus
Done
Local intranet

```
    </soap:Body>
  </soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http:/
  <soap:Body>
    <HelloWorldResponse xmlns="http://tempuri.org/">
      <HelloWorldResult>string</HelloWorldResult>
    </HelloWorldResponse>
  </soap:Body>
</soap:Envelope>
```

HTTP GET

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

```
GET /DemoWebService/Service1.asmx/HelloWorld? HTTP/1.1
Host: localhost
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

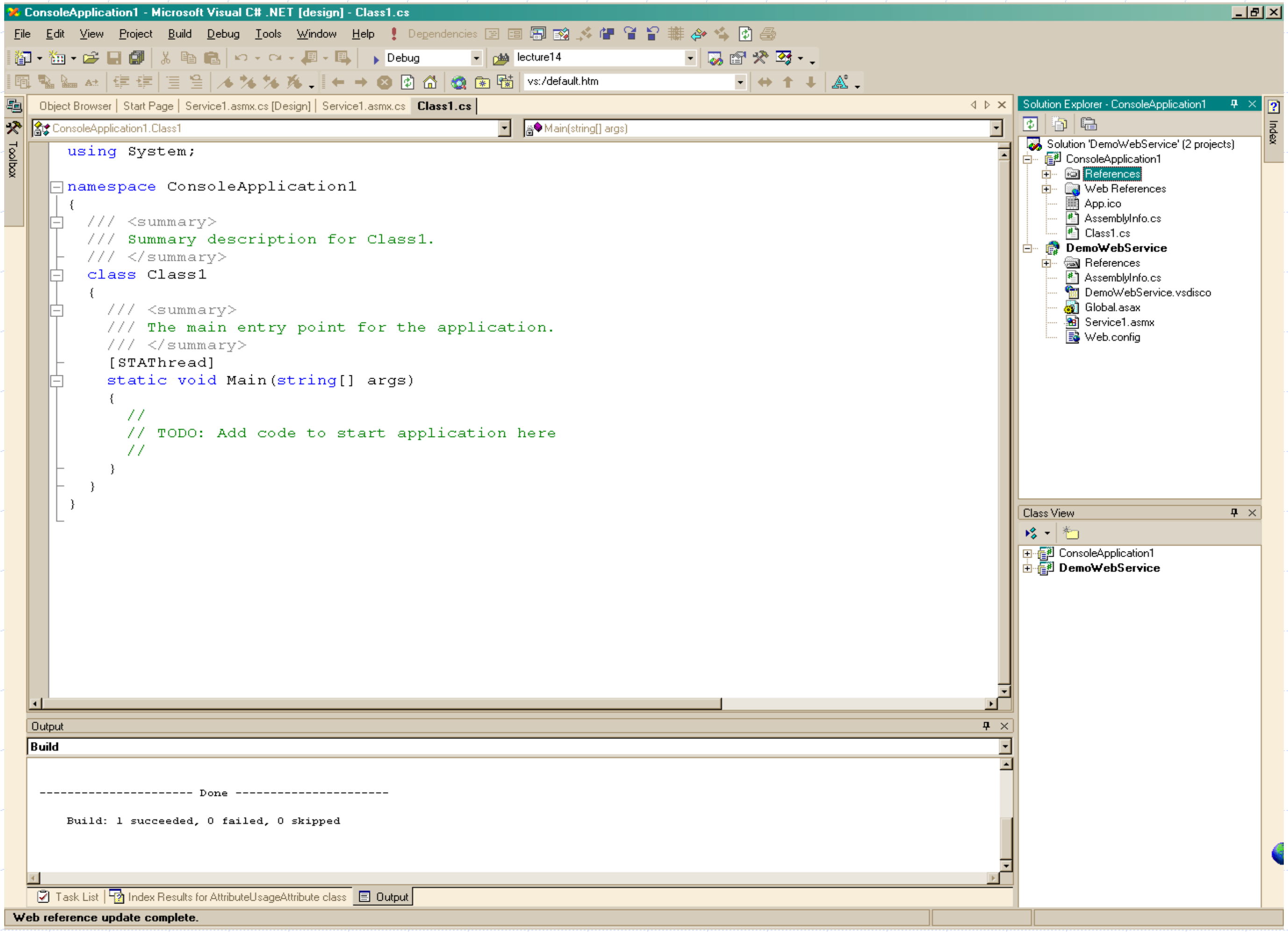
HTTP POST

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

```
POST /DemoWebService/Service1.asmx/HelloWorld HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

Create Console Client


ConsoleApplication1 - Microsoft Visual C# .NET [design] - Class1.cs
File Edit View Project Build Debug Tools Window Help
Debug
lecture14
vs:/default.htm
Object Browser | Start Page | Service1.asmx.cs [Design] | Service1.asmx.cs | Class1.cs
ConsoleApplication1.Class1
Main(string[] args)
Solution Explorer - ConsoleApplication1
Solution 'DemoWebService' (2 projects)
ConsoleApplication1
References
Web References
App.ico
AssemblyInfo.cs
Class1.cs
DemoWebService
References
AssemblyInfo.cs
DemoWebService.vsdisco
Global.asax
Service1.asmx
Web.config
Class View
ConsoleApplication1
DemoWebService
Output
Build
Task List
Index Results for AttributeUsageAttribute class
Output

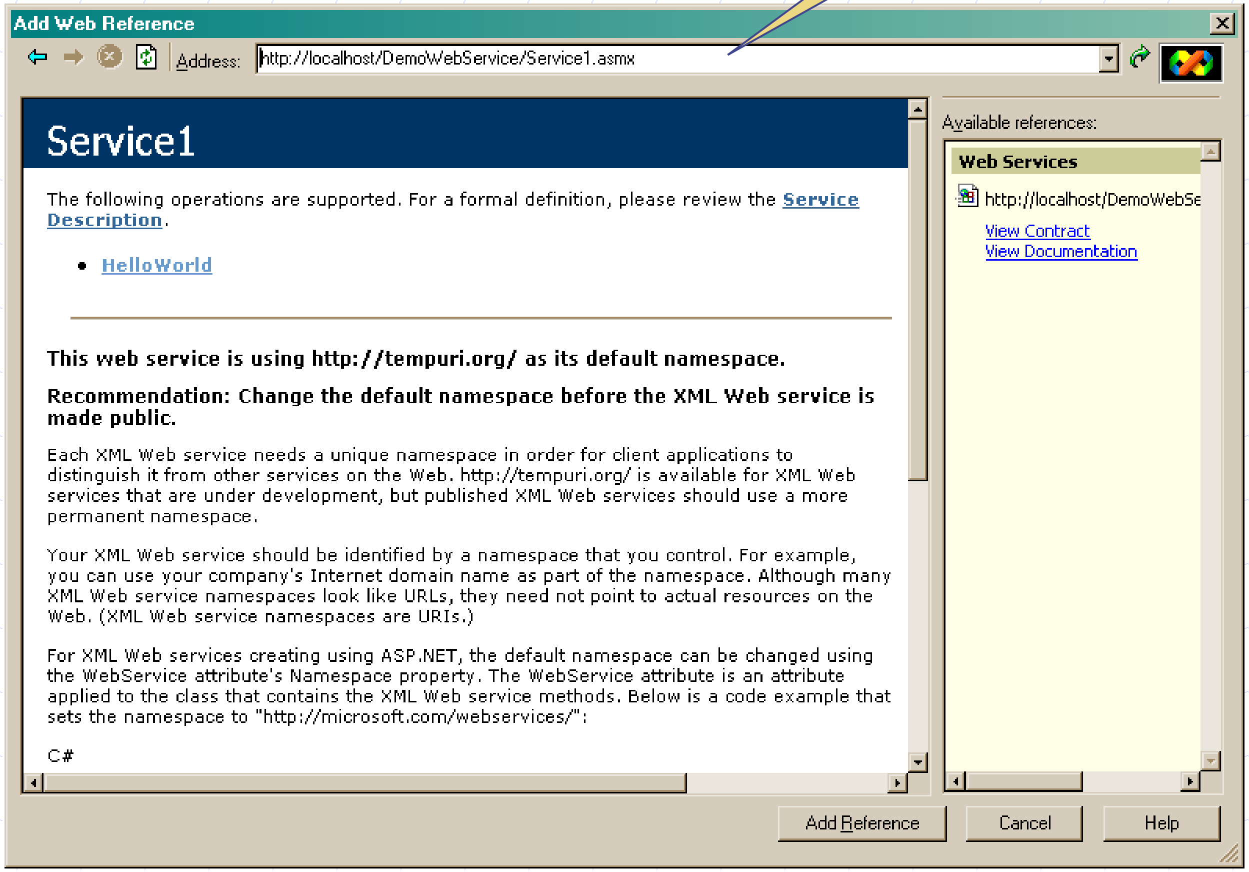
```
using System;

namespace ConsoleApplication1
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main(string[] args)
        {
            //
            // TODO: Add code to start application here
            //
        }
    }
}
```

```
--------------------- Done ---------------------

    Build: 1 succeeded, 0 failed, 0 skipped
```

Web reference update complete.

Adding a Web Reference

You have to locate the folder and asmx file, using explorer, then type in path here.



Service1

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.

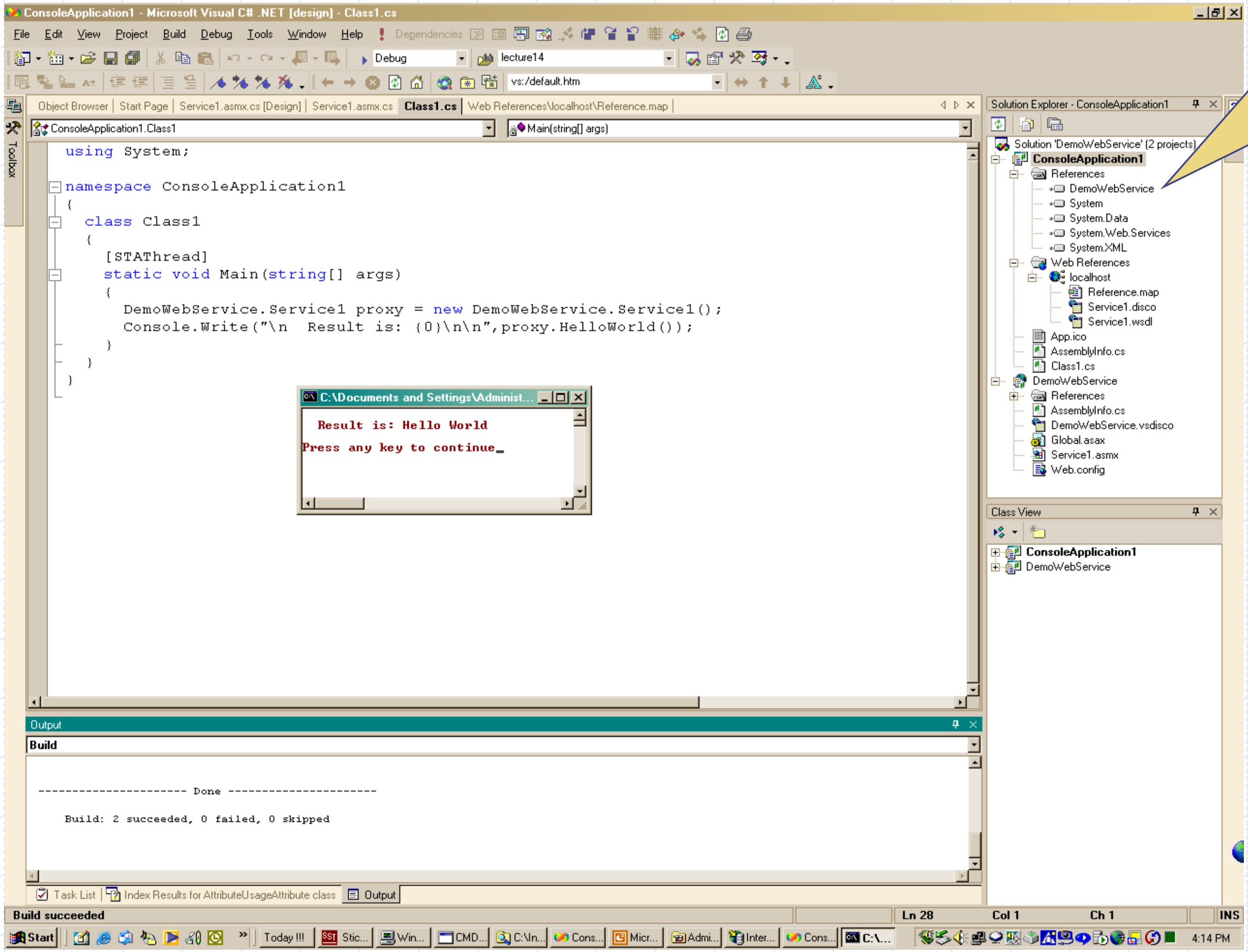
Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

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#

Client Accessing Web Service


ConsoleApplication1 - Microsoft Visual C# .NET [design] - Class1.cs
File Edit View Project Build Debug Tools Window Help
Debug
lecture14
vs:/default.htm
Object Browser | Start Page | Service1.asmx.cs [Design] | Service1.asmx.cs | Class1.cs | Web References\localhost\Reference.map
ConsoleApplication1.Class1
Main(string[] args)
Solution Explorer - ConsoleApplication1
Solution 'DemoWebService' (2 projects)
ConsoleApplication1
References
DemoWebService
System
System.Data
System.Web.Services
System.XML
Web References
localhost
Reference.map
Service1.disco
Service1.wsdl
App.ico
AssemblyInfo.cs
Class1.cs
DemoWebService
References
AssemblyInfo.cs
DemoWebService.vsdisco
Global.asax
Service1.asmx
Web.config
Class View
ConsoleApplication1
DemoWebService
C:\Documents and Settings\Administ...
Output
Build
Task List | Index Results for AttributeUsageAttribute class | Output
Build succeeded
Ln 28 Col 1 Ch 1 INS
Start
4:14 PM

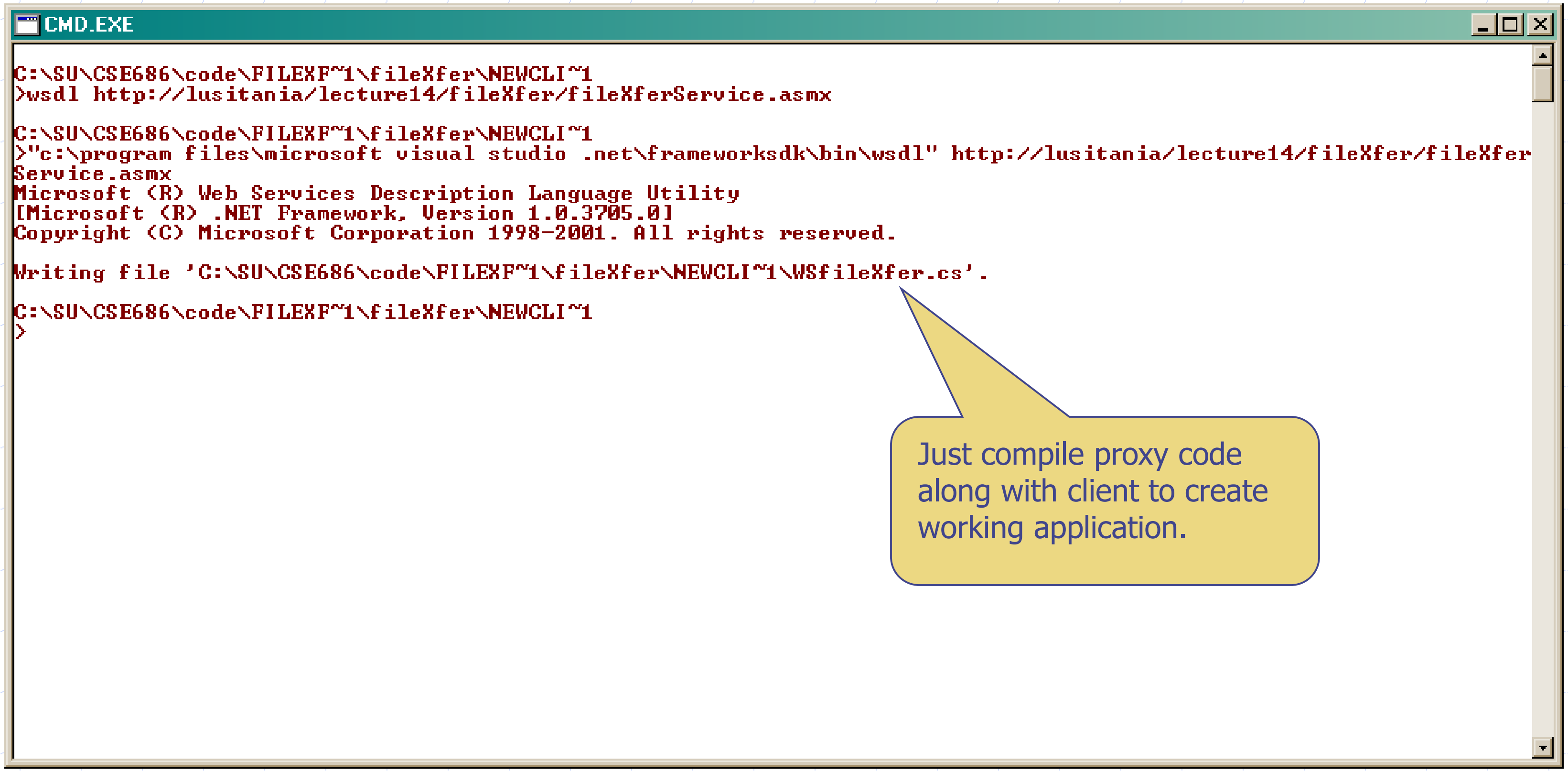
```
using System;

namespace ConsoleApplication1
{
  class Class1
  {
    [STAThread]
    static void Main(string[] args)
    {
      DemoWebService.Service1 proxy = new DemoWebService.Service1();
      Console.Write("\n  Result is: {0}\n\n",proxy.HelloWorld());
    }
  }
}
```

```
 Result is: Hello World
Press any key to continue_
```

```
--------------------- Done ----------------------

    Build: 2 succeeded, 0 failed, 0 skipped
```

Note:
You also need a reference to project so client knows about web service type.

Create Proxy Source Code with WDSL.exe - alternative to adding web reference



CMD.EXE

```
C:\SU\CSE686\code\FILEXF~1\fileXfer\NEWCLI~1
>wsdl http://lusitania/lecture14/fileXfer/fileXferService.asmx

C:\SU\CSE686\code\FILEXF~1\fileXfer\NEWCLI~1
>"c:\program files\microsoft visual studio .net\frameworksdk\bin\wsdl" http://lusitania/lecture14/fileXfer/fileXfer
Service.asmx
Microsoft (R) Web Services Description Language Utility
[Microsoft (R) .NET Framework, Version 1.0.3705.0]
Copyright (C) Microsoft Corporation 1998-2001. All rights reserved.

Writing file 'C:\SU\CSE686\code\FILEXF~1\fileXfer\NEWCLI~1\WSfileXfer.cs'.

C:\SU\CSE686\code\FILEXF~1\fileXfer\NEWCLI~1
>
```

Just compile proxy code along with client to create working application.

Web Services

The End