
THE USE OF WEB SERVICES TECHNOLOGY IN .Net

Prof. Ms. Shabnam S. Mahat

Assistant Professor

Institute of Management and Rural Development Administration, Sangli.

Abstract:

Web Services is a distributed computing technology that offers interaction and collaboration among vendors and customers, with the vision of providing universal computing. web services programming in software industry have become a key role in providing business need based services. Web Services will make computing resources, both hardware and software, accessible through the Internet. They would inspire a pay-per-usage model and make dynamic collaborations possible. One of the significant classifications of Web Services is: "Web Services are loosely coupled software components delivered over Internet-standard technologies." [1] This paper reflects the importance of web service, Web Service History, the success, the advantages and limitations of web service. This paper will make a good awareness for those new to web services as well as make more clarification to those already aware with the technology particularly those developing applications for different services.

Keywords: XML, SOAP, WSDL, UDDI, web services.

I. Introduction:

The term Web services describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. A Web Service is a software program that uses XML to exchange information with other software via common internet protocols. In a simple sense, Web Services are a way for interacting with objects over the Internet. It is also Describe as Web Services are self-contained, modular, distributed, dynamic applications that can be described, published, located, or invoked over the network to create products, processes, and supply chains. These applications can be local, distributed, or Web-based. Web services are built on top of open standards such as TCP/IP, HTTP, Java, HTML, and XML.

- XML: is used to tag the data
- SOAP: is used to transfer the data,
- WSDL: is used for describing the services available and
- UDDI: is used for listing what services are available.

A web service is

- Language, Protocol and Platform Independent.
- It accepts a state less service architecture.
- Scalable (e.g. multiplying two numbers together to an entire customer-relationship management system).
- Programmable (encapsulates a task).
- Based on XML (open, text-based standard).
- Self-describing (metadata for access and use).
- Discoverable (search and locate in registries)- ability of applications and developers to search for and locate desired Web services through registries. This is based on UDDI.

1) Web Service History

- Microsoft coined the term "Web services" in June 2000, when the company introduced Web services as a key component of its .Net initiative, a broad new vision for embracing the Internet in the development, engineering and use of software.
- Web services encompass a set of related standards that can enable any two computers to communicate and exchange data via a network, such as the Internet. The primary standard used in Web services is the Extensible Markup Language (XML) developed by the World Wide Web Consortium (W3C).
- Developers use XML tags to describe individual pieces of data, forming XML documents, which are text-based and can be processed on any platform. XML provides the foundation for many core Web services standards (SOAP, WSDL, and UDDI) and vocabularies (XML-based markup for a specific industry or purpose).
- Almost every type of business can benefit from Web services such as expediting software development, integrating applications and databases, and automating transactions with suppliers, partners, and clients.

2) Web services advantages

- Promote a modular approach to programming, so multiple organizations can communicate with the same Web service.

- Comparatively easy and inexpensive to implement, because they employ an existing infrastructure and because most applications can be repackaged as Web services.
- Significantly reduce the costs of enterprise application (EAI) integration and B2B communications.
- Implemented incrementally, rather than all at once which lessens the cost and reduces the organizational disruption from an abrupt switch in technologies.

3) **Web Services Limitations**

- SOAP, WSDL, UDDI- require further development.
- Interoperability.
- Royalty fees.
- Too slow for use in high-performance situations.
- Increase traffic on networks.
- The lack of security standards for Web services.

II. **The Web Services Model:**

The Web Services architecture is based upon service provider, service registry and service requestor. The interactions involve the publish, find and bind operations. The service provider defines a service description for the Web service and publishes it to a service requestor or service registry. The service requestor uses a find operation to retrieve the service description locally or from the service registry and uses the service description to bind with the service provider and invoke or interact with the Web service implementation. Service provider and service requestor roles are logical constructs and a service can exhibit characteristics of both.

1. **Service provider:** From a business perspective, this is the owner of the service. From an architectural perspective, this is the platform that hosts access to the service.
2. **Service requestor:** This is the business application that looking for and invoking or initiating an interaction with a service. The service requestor role can be played by a browser driven by a person or a program without a user interface, for example another Web service.
3. **(Discovery Agencies) Service registry:** This is a searchable registry of service descriptions where service providers publish their service descriptions. Service requestors find services and obtain binding information (in the service descriptions) for services during development for static

binding or during execution for dynamic binding. For statically bound service requestors, the service registry is an optional role in the architecture, because a service provider can send the description directly to service requestors. Likewise, service requestors can obtain a service description from other sources besides a service registry, such as a local file, FTP site, Web site, Advertisement and Discovery of Services (ADS) or Discovery of Web Services (DISCO).

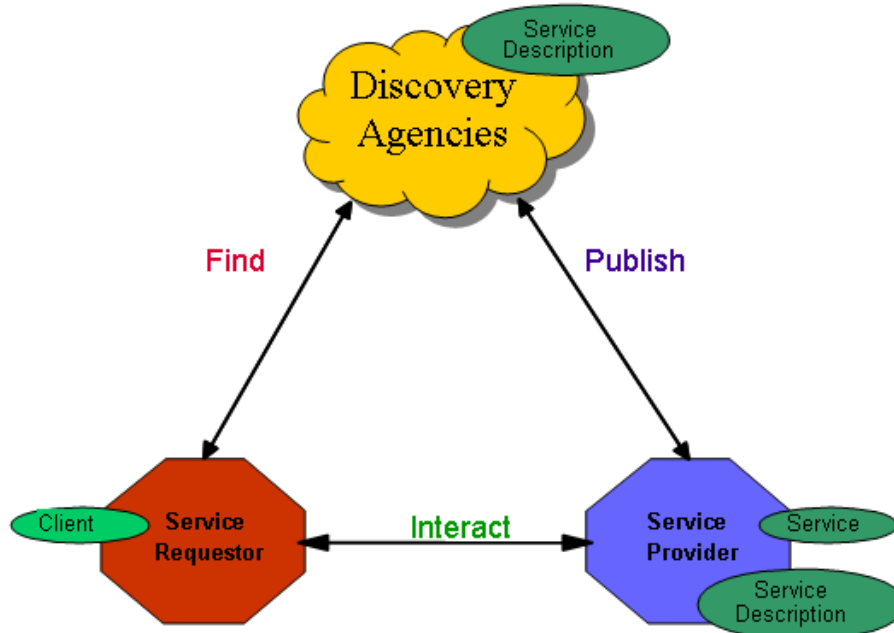


Figure 1: Service Oriented Architecture

Operations in Web Service Architecture:

- Publish: Service descriptions need to be published in order for service requestor to find them
- Find: Service requestor retrieves a service description directly or queries the service registry for the service required
- Bind: Service requestor invokes or initiates an interaction with the service at runtime.

III. Web Service Architecture:

Web services rely on a set of standards to support interoperability among applications developed in different languages and running on different platforms or operating systems. One way to understand Web services is to understand Web services standards [4]

Web services standards including:

- 1) SOAP (Simple Object Access Protocol)
- 2) WSDL (Web Services Description Language)
- 3) UDDI (Universal Description, Discovery, and Integration)

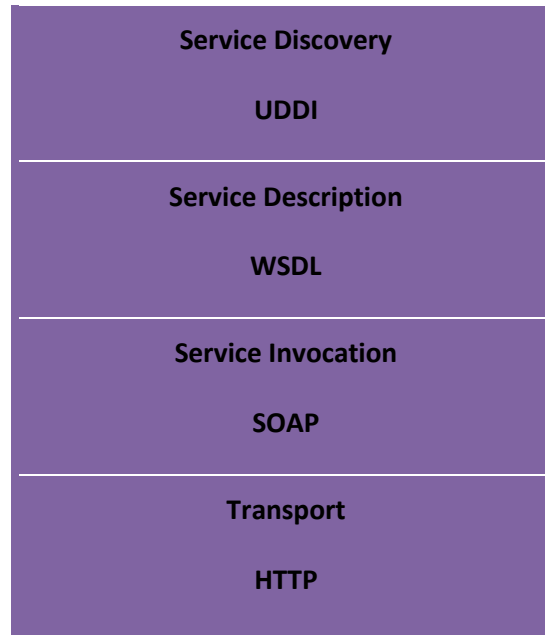


Figure 1: Web Service Architecture

IV. Use of web services for solutions in the following areas:

- **Enterprise Application Integration (EAI).** A web service could allow multiple business partners to exchange inventory, order, or financial data, for example, without specifically knowing the precise data layout in which data is stored for each partner. For instance, many customer relationship management (CRM) or other front-end applications store customer data in a format that is not entirely compatible with the way a back-end enterprise resource planning (ERP) system stores its financial or inventory information. However, a sales organization may wish to use its CRM solution to process real-time orders with up-to-date inventory information from the ERP system. A web service could be a solution to managing the transformation of CRM requests to ERP storage and from ERP responses to CRM confirmations.
- **Business-to-business (B2B) integration.** Similar to the EAI solution, a B2B solution could take advantage of a Web Services capability to provide cached data for large orders. B2B transactions, unlike business-to-consumer (B2C) transactions, often consist of high-volume transactions that would be prohibitive to execute at the level of a B2C transaction. For instance, a consumer might order one box of pencils from an online stationery store, but a

business might order a thousand boxes monthly, with multiple shipping addresses. The scale and complexity of a B2B transaction requires the intervention of a web service to help simplify and process the transaction quickly and with consistency.

- **Business-to-consumer integration.** B2C web services typically manage web-based transactions. For example, a web service that allows you to look up postal codes eliminates the need for businesses to create a new program every time the service is included on a web site. Some commerce sites might use web services to help manage currency conversion when taking international sales orders.
- **Mobile (Smart client applications).** Web services play a vital role in providing lightweight but powerful applications to mobile devices. Web services allow mobile device users to perform a variety of tasks which require little more than data input at the device and data display of the results. All processing can occur on a remote web service, thus decreasing bandwidth requirements on the mobile device itself.
- **Distributed and Peer-to-Peer.** If you use distributed computing over an uncontrolled network (such as the Internet) rather than over a LAN or corporate network, you might use web services. Web services do not require state maintenance, thus offering potentially improved performance, particularly where a request-response behavior is not absolutely required. For applications that require strict request-response behavior and high security, you should consider using an older, more controlled model, such as COM or .NET remoting.

V. Web Service Example

- **Web Portal-** A web portal might obtain top news headlines from an associated press web service.
- **Weather Reporting-** You can use Weather Reporting web service to display weather information in your personal website.
- **Stock Quote-** You can display latest update of Share market with Stock Quote on your web site.
- **News Headline:** You can display latest news update by using News Headline Web Service in your website.
- You can make your own web service and let others use it. For example you can make Free SMS Sending Service with footer with your companies advertisement, so whosoever uses this service indirectly advertises your company. You can apply your ideas in N no. of ways to take advantage of it.

VI. Web Services have Two Types of Uses:

1. Reusable application-components.

- There are things applications need very repeatedly. So why make these over and over again?
- Web services can offer application-components like: currency conversion, weather reports, or even language translation as services.

2. Connect existing software.

- Web services can help to solve the interoperability problem by giving different applications a way to link their data.
- With Web services you can exchange data between different applications and different platforms.

VII. ASP.NET Web Services Files :

Certain files are automatically generated when you create applications with ASP.NET Web Services. These files enable the ASP.NET Web Services to render their services through a web server. The following table lists the files and their descriptions.

| File | Description |
|--------------------|--|
| .asmx | When you create an ASP.NET Web Services application, a text file is automatically generated with the .asmx extension. The required Web Services directive is placed at the top of this file to correlate between the URL address of the web service and its implementation. Within the .asmx file, you add Web Services logic to the methods visible by the client application. The .asmx file acts as the base URL for clients calling the XML web service. This file is compiled into an assembly, along with other files, for deployment. |
| code-behind | When you create an ASP.NET Web Service application, a code-behind file is generated with a language-specific extension. You add your Web Services logic to the public method to process Web Services requests and responses. |
| compiled DLL files | Web Services DLL files provide dynamic services on the web server. |

| | |
|-------|---|
| .wsdl | This file is generated when you click the Add Web Reference feature to add the web service to your client application. It describes the Web Services interface available to the client. |
| .map | This file enables the discovery of a web service that is exposed on a given server. It also contains links to other resources that describe the web service. |

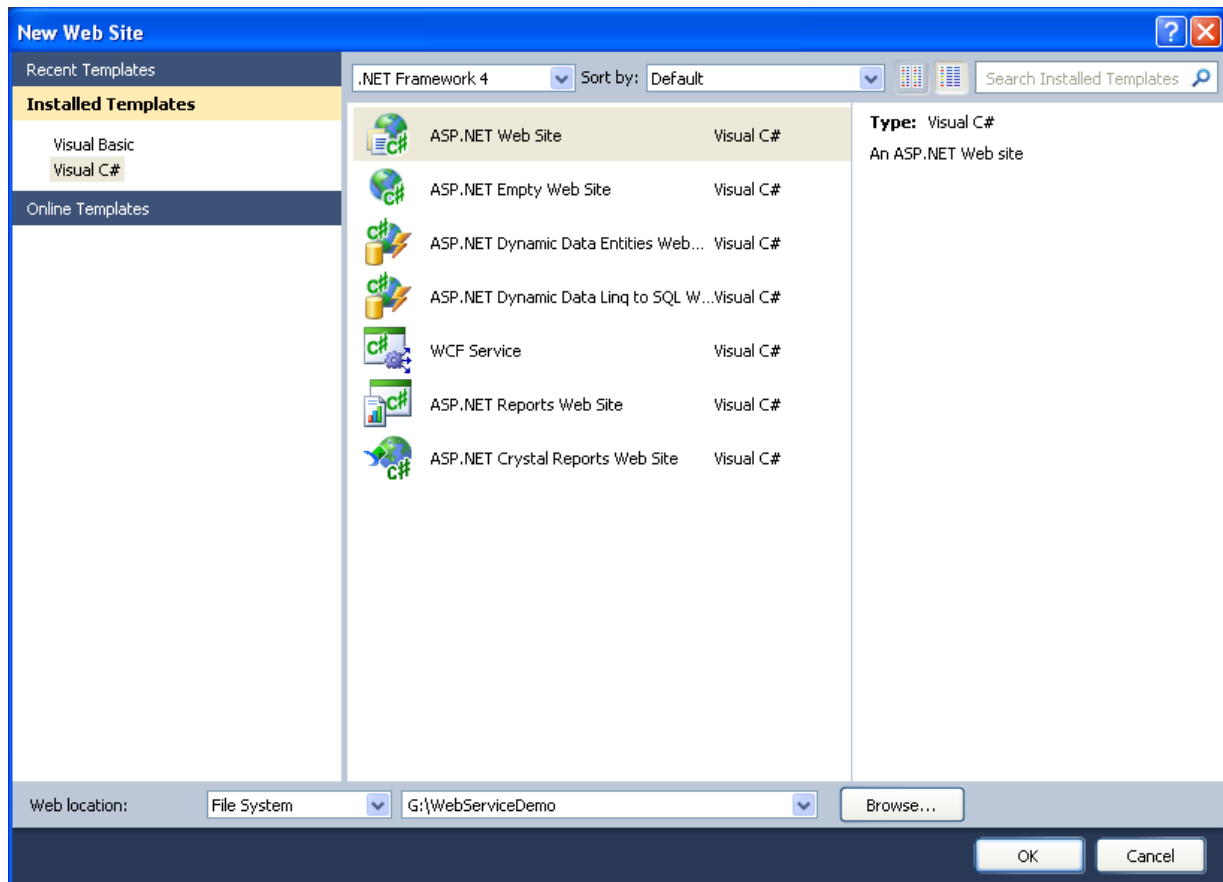
VIII. Example of Creating Web Service in .Net

Here are samples codes which I use to create and consume ASP.NET Web Service in two phases:

Phases 1: Create Web Service Source File

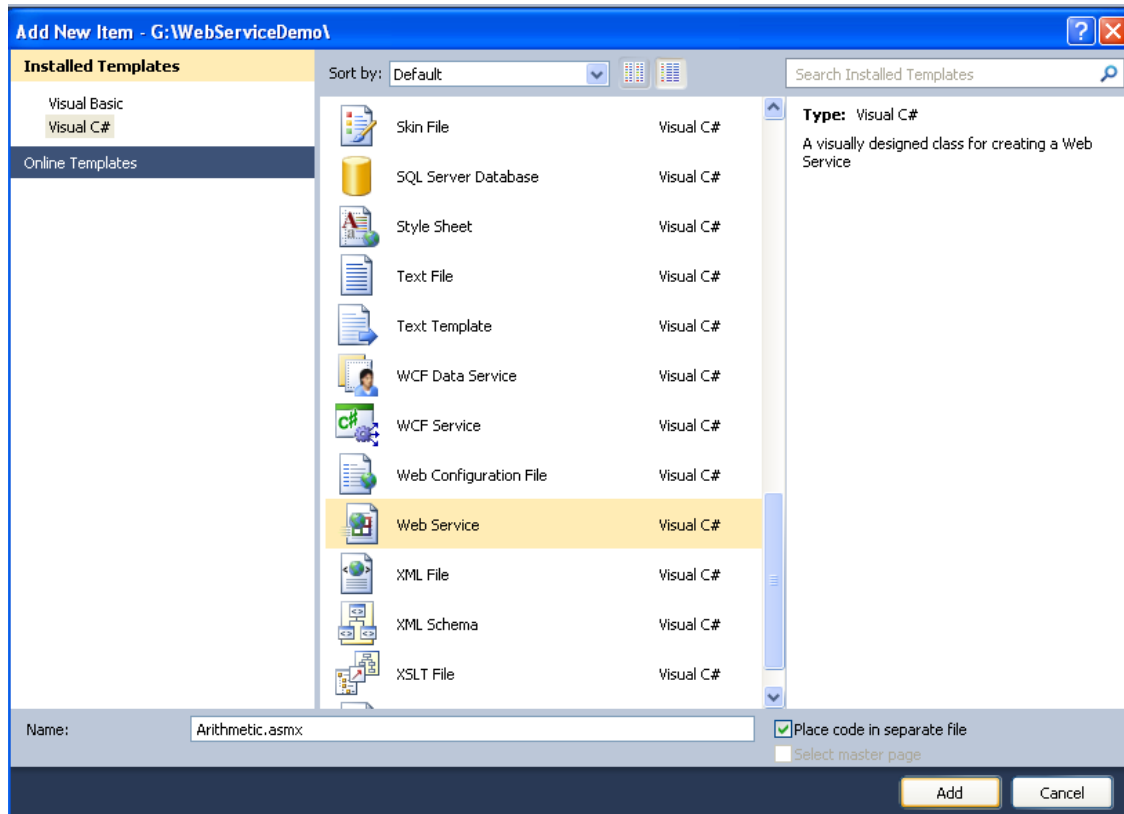
- **Step 1- Create the ASP.NET Web Service Source File**

Open Visual Studio 2010 and create a new web site.->Select .Net Framework 3.5. ->Select ASP.NET Web Service page ->Then, you have to give the name of your service. In this example I am giving it's name "**WebServiceDemo**". Then Click the ok Button. A screen-shot of these activity is given below.



- **Step 2- Add web Service file to the ASP.NET Web site**

Right click on **WebServiceDemo web site**->Click on Add New Item.->Select Web service. ->Change the Name of WebServices.asmx to **Arithmetic.asmx** ->Then,Click on Add Button. A screen-shot of these activity is given below.



- **Step 3- Open the Arithmetic.asmx file, this will look like as follows**

```
App_Code/Arithmetic.cs x
Arithmetic Arithmetic()
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

/// <summary>
/// Summary description for Arithmetic
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following
// [System.Web.Script.Services.ScriptService]
public class Arithmetic : System.Web.Services.WebService {

    public Arithmetic () {

        //Uncomment the following line if using designed components
        //InitializeComponent();
    }

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

Here (in the above figure), you will note that there is predefined method "**HelloWorld**" which returns the string "**Hello World**". You can use your own method and can perform various operations. Here I made a simple method which returns the Addition, Subtraction, multiplication and Division of two numbers using the code.

Arithmetic.cs File is as below.

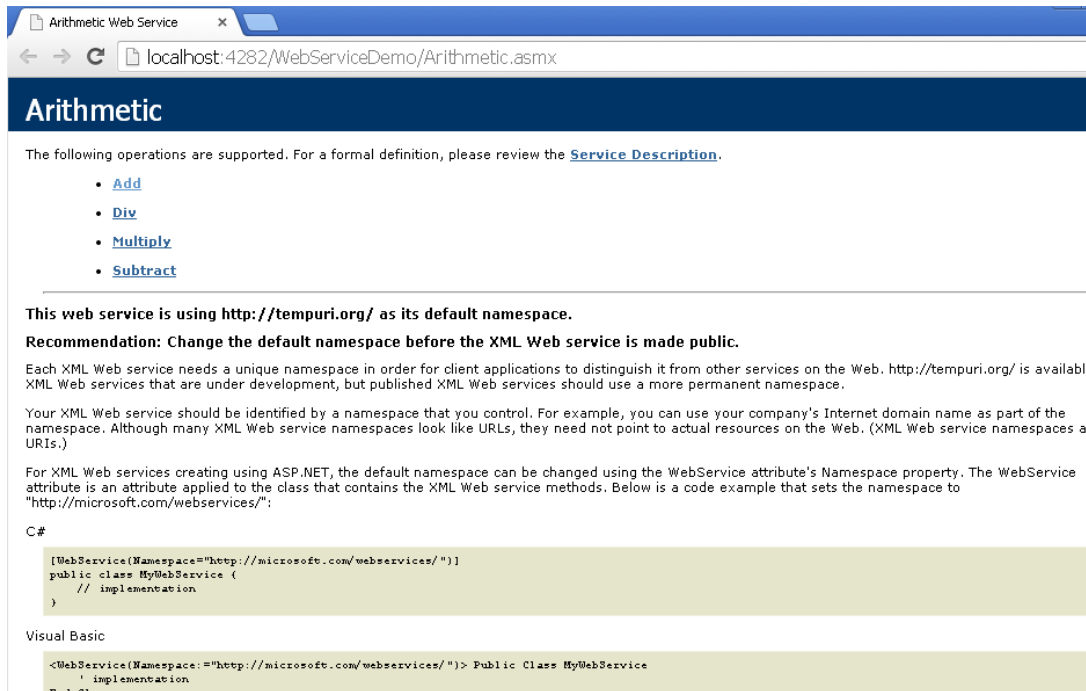
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
///<summary>
/// Summary description for Arithmetic
///</summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// [System.Web.Script.Services.ScriptService]
public class Arithmetic : System.Web.Services.WebService
{
    [WebMethod]
    public int Add(int x, int y)
    {
        return x + y;
    }
    public int Subtract(int x, int y)
    {
        return x - y;
    }
    public int Multiply(int x, int y)
    {
        return x * y;
    }
    public int Div(int x, int y)
    {
        return x / y;
    }
}
```

The [WebMethod] attribute

The Service class exposes a single method, the public method **Multiplication**, which takes two integer arguments and returns the multiplication of two number as integer. To expose a method as a part of a web service, you must decorate it with the WebMethod attribute, which tells the compiler to treat it as

such. Any method marked with the WebMethod attribute must be defined as public. Class methods exposed as web services follow the same object-oriented rules as any other class, and therefore methods marked private, protected, or internal are not accessible and will return an error if you attempt to expose them using the WebMethod attribute.

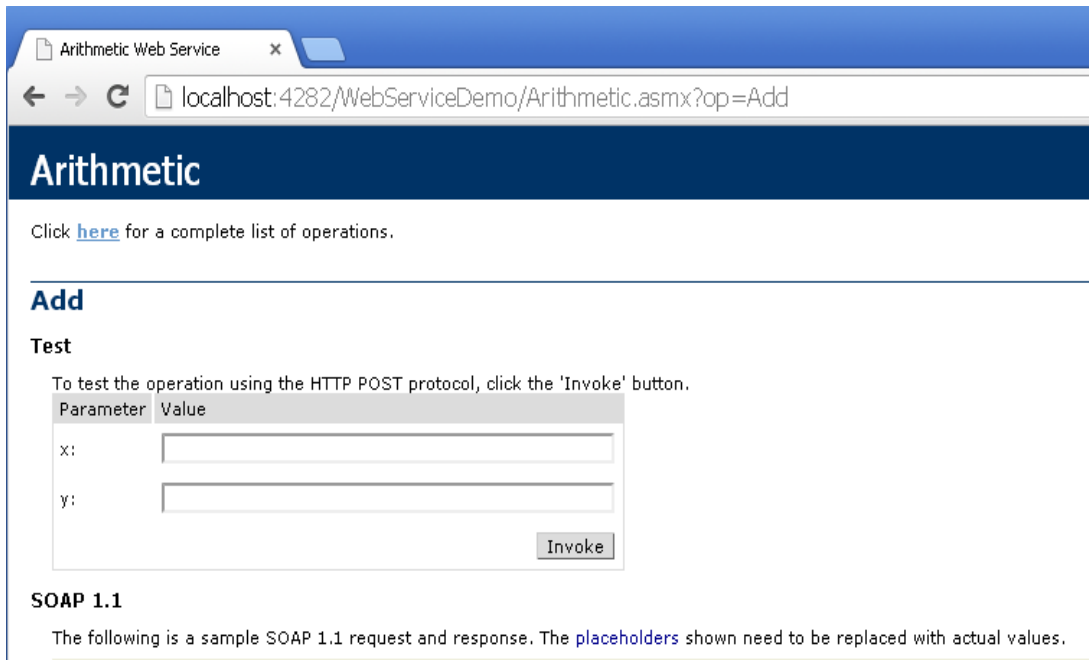
- **Step 4-** Build Web Service (Arithmetic.cs file) and Run the Web Service for testing by pressing F5 function key.you will see



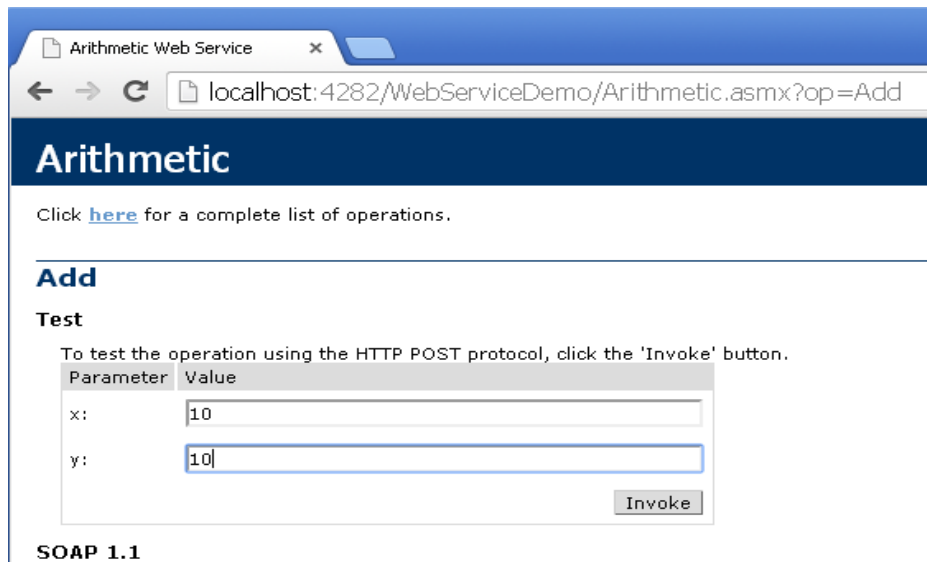
- **Step 5-** Copy the url of this web service for further use.

e.i **<http://localhost:4282/WebServiceDemo/Arithmetic.asmx>**

Click on the Add button to test the web service. this will open the following page



Enter the value of x and y.



By pressing the "Invoke" button a XML file is generated.



Now our web service is ready to use; we just need to create a new web site to consume the web service.

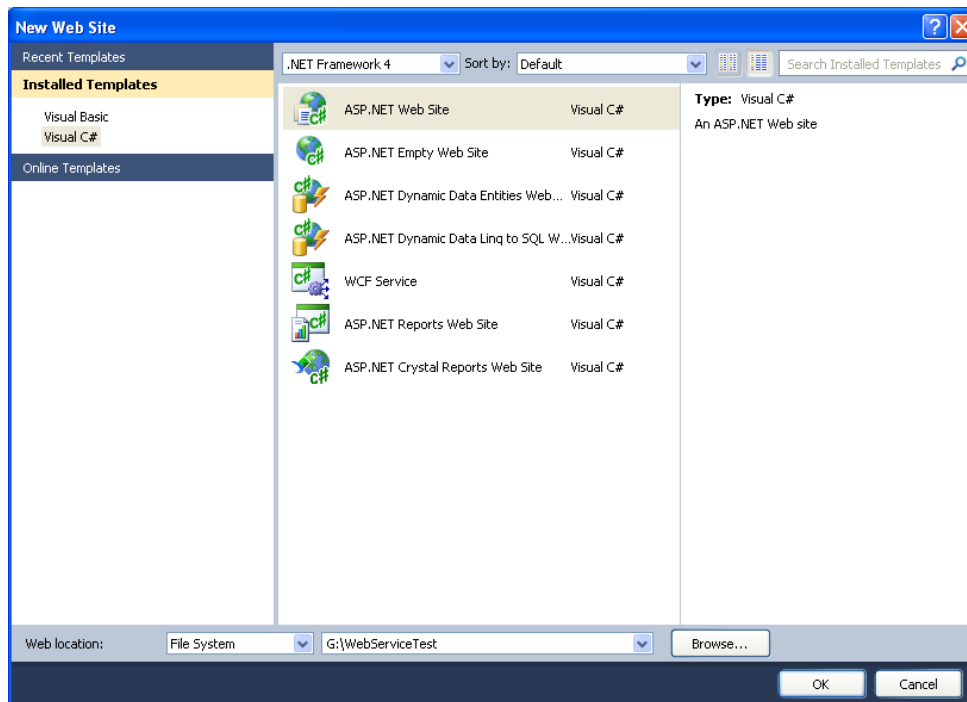
Phases 2: Testing Web Service in .Net

A Monthly Double-Blind Peer Reviewed Refereed Open Access International e-Journal - Included in the International Serial Directories

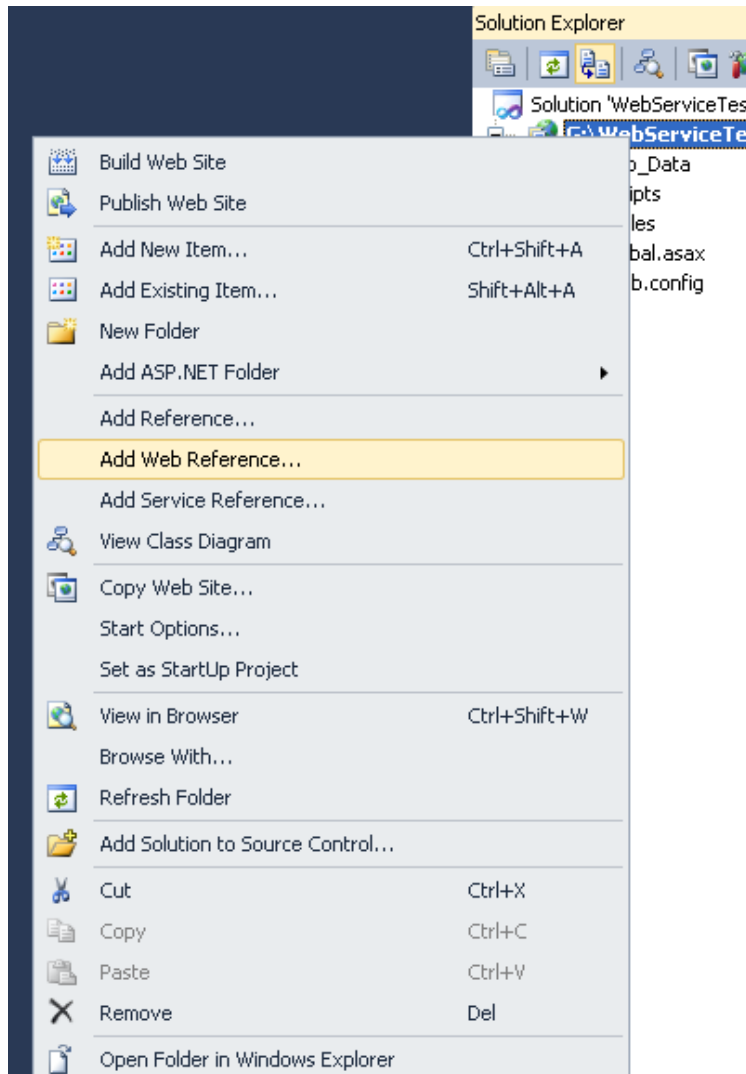
International Journal in IT and Engineering

<http://www.ijmr.net.in> email id- irjmss@gmail.com

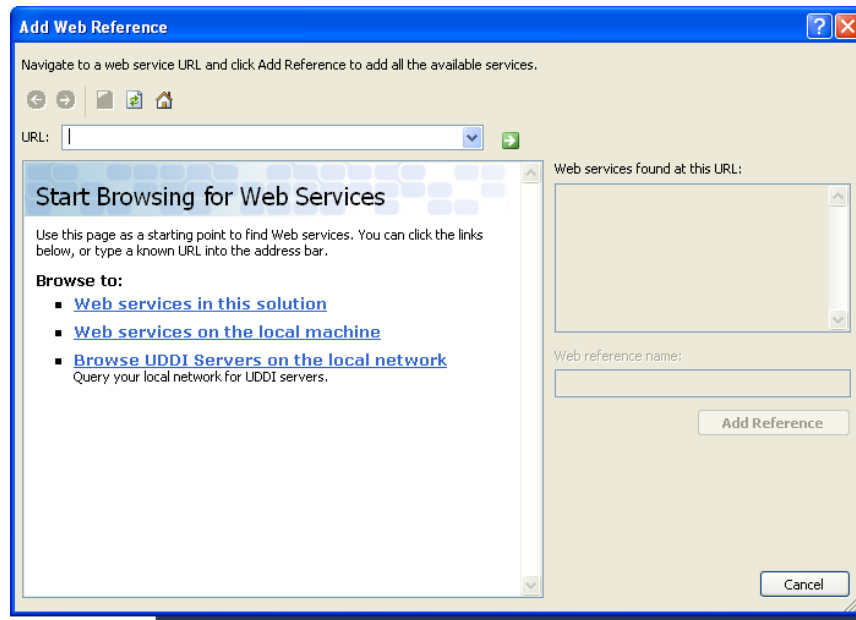
- **Step 1-** Create a **WebServiceTest** Web Site by File > New > Web Site > Asp.net Web Site



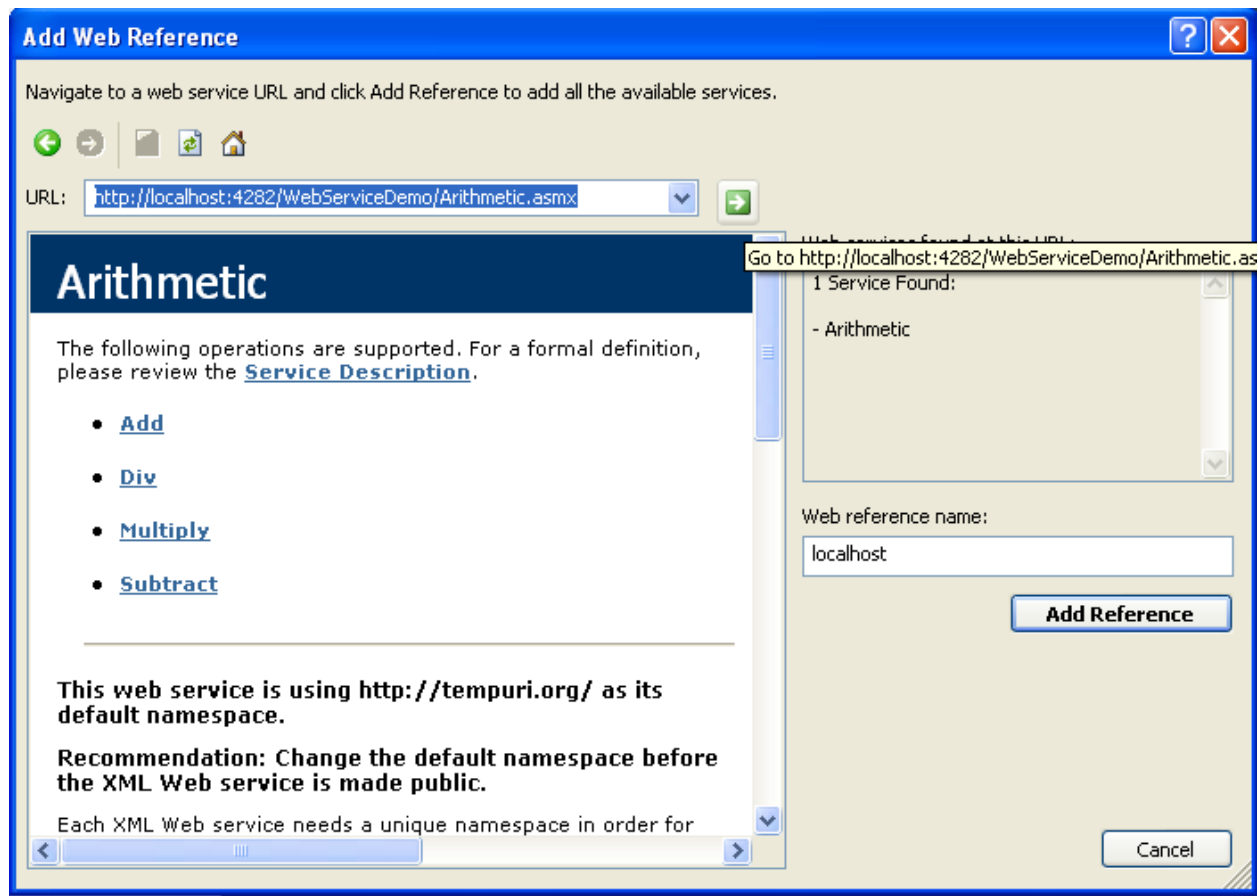
- **Step 2-** Right-click Solution Explorer and choose "Add Web Reference".



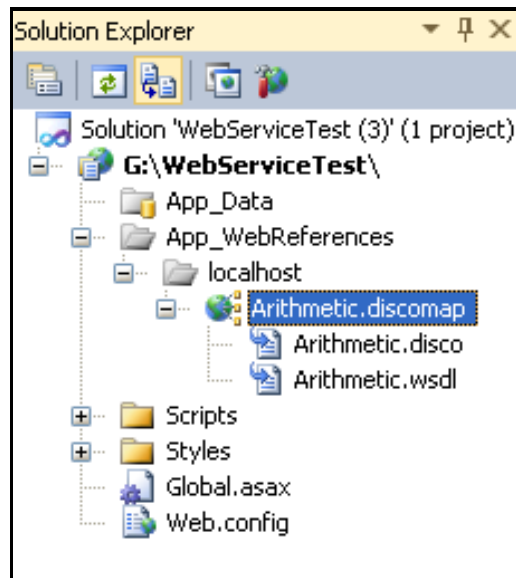
This will open the following window.



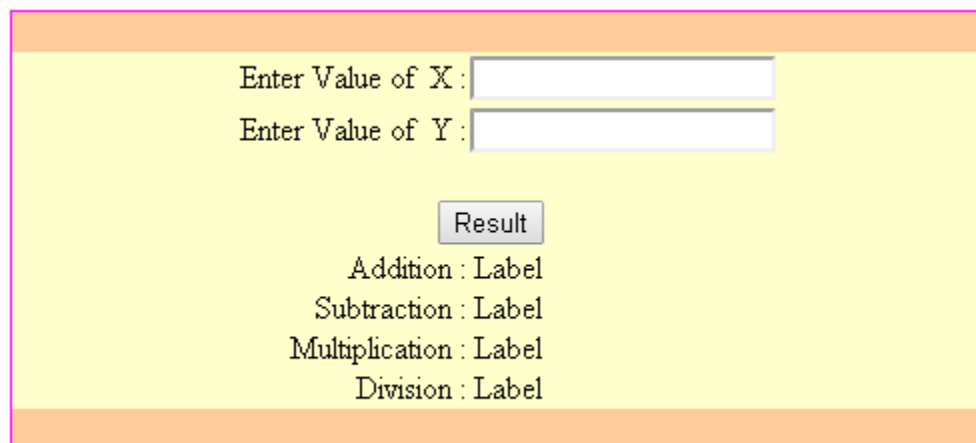
- **Step 3-** Paste the url of the web service (Copied by step 4 i.e. <http://localhost:4282/WebServiceDemo/Arithmetic.asmx>) and click on 'Go' button and then 'Add reference'.



- **Step 4-** Now your web service is ready to use in the Solution Explorer you will see.



- **Step 5-** Go to the design of the Default.aspx page; drag and drop 2 Textboxes, 4 Label and one button.

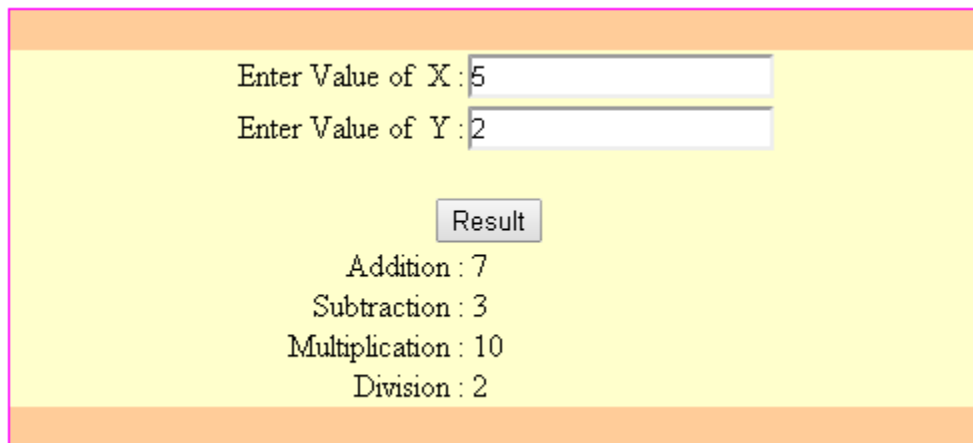


- **step 6-** Go to Default.cs page and on the button click event use the following code.

```
protected void Button1_Click(object sender, EventArgs e)
{
    localhost.Arithmetic WS = newlocalhost.Arithmetic();
    int x = Convert.ToInt32(TextBox1 .Text);
    int y = Convert.ToInt32(TextBox2 .Text);
```

```
Label1.Text = WS.Add(x, y).ToString();  
Label2.Text = WS.Subtract(x, y).ToString();  
Label3.Text = WS.Multiply(x, y).ToString();  
Label4.Text = WS.Div(x, y).ToString();  
}
```

Enter the values and test the web service as shown below.



Enter Value of X : 5
Enter Value of Y : 2

Result

Addition : 7
Subtraction : 3
Multiplication : 10
Division : 2

Conclusion:

Developing Web Services can be confusing because of the numerous technologies (XML, SOAP, UDDI , and WSDL). Using Visual Studio .NET to create and consume the Web Service makes the process less difficult. When the Web Service is compiled Visual Studio .NET creates the WSDL document, the SOAP request and response messages for the exposed methods. Visual Studio .NET also creates the Web Discovery Service file that is used to discover Web Services on your server. Developers should study each of these technologies to gain a through understanding of Web Services development.

References:

- [1] ON THE USE OF WEB SERVICES TECHNOLOGY IN E-HEALTH APPLICATIONS John B. Oladosu, Funmilola A. Ajala, And Olukunle O. Popoola
- [2] <http://www.c-sharpcorner.com/UploadFile/1d42da/web-service-basics/>
- [3] Web Services Conceptual Architecture (WSCA 1.0)
- [4] "The Implications and Impacts of Web Services to Electronic Commerce Research and Practices" Minder Chen, Andrew N. K. Chen, Benjamin B. M. Shao.
- [5] Wikipaedia (2008)
- [6] <http://www.dev2dev.bea.com>. Official website for the Developer.
- [7] V. Vasudaran (2001): A Web Services Primer, available online at <http://webservices.xml.com/pub/a/ws/2001/04/04/webservices/>

[1] Lakshmi Ananthamurthy (2008): "Introduction to Web Service." Available at <http://www.developer.com/services/article.php/1485821>

Acknowledgment:

I express my deep sense of gratitude to the all above references who provide a graceful information to me about web services. I express my deep sense of gratitude to my parents and my husband for his valuable moral support. Their inspiration and kind co-operation helped to shape this study properly.

About the Author:

Prof. Shabnam Sikandar Mahat, (B.Sc-electronics, MBA-IT, M.Phil, Ph.D (Appearing)

Department of Management Studies and Computer Applications, Bharati Vidyapeeth Deemed University, Pune, Institute of Management and Rural Development Administration, Sangli, India.