WCF

WCF – a step ahead

- Windows Communication Foundation
- framework for building service-oriented applications.
- released for the first time in 2006 as a part of the .NET framework with Windows Vista.
- WCF 4.5 is the most recent version that is now widely used.

A WCF application consists of three components:

- WCF service,
- WCF service host, and
- WCF service client.

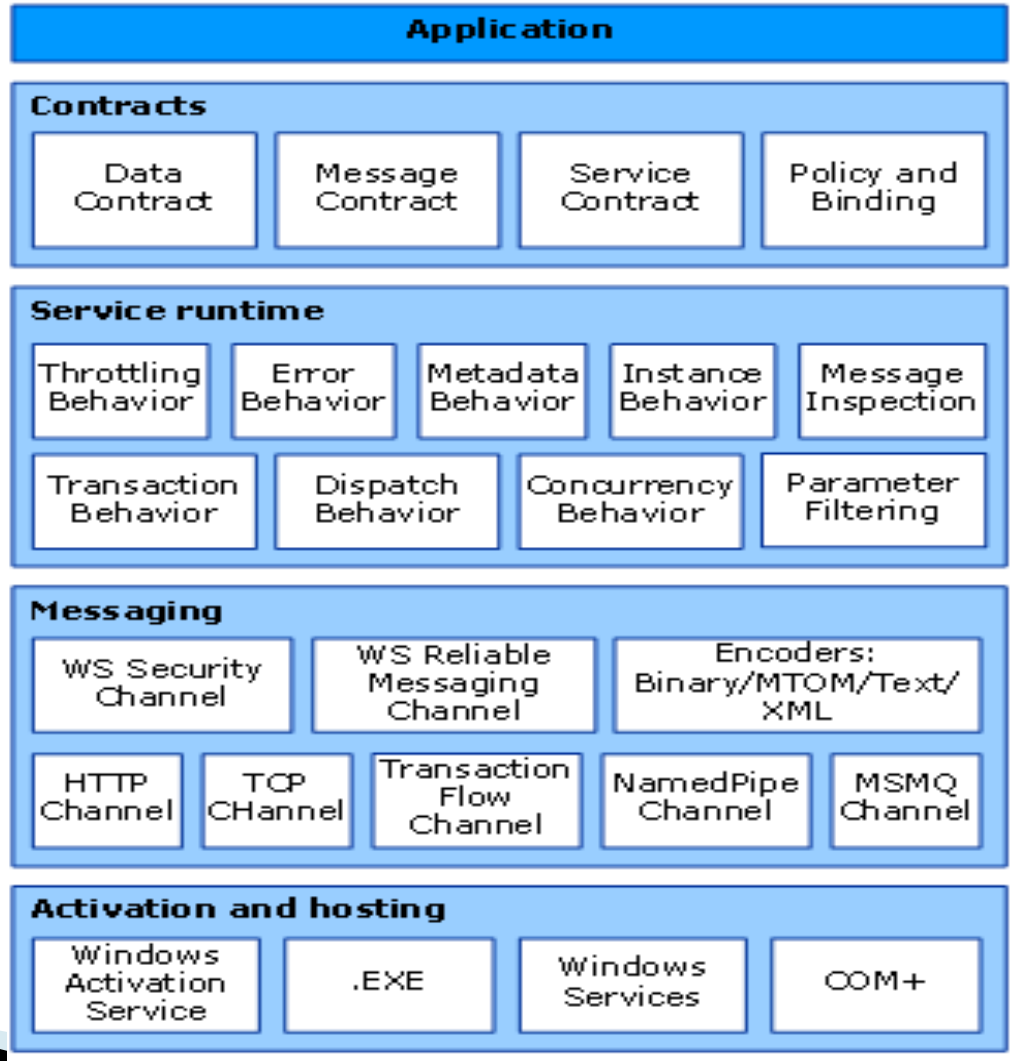
WCF platform is also known as the Service Model.

Limitations of web services

- The major limitation with web services is that the communication can happen over HTTP only.
- A second limitation with web services is that it provides simplex communication and there is no way to have half duplex or full duplex communication using web services.

Features	Web Service	WCF
Hosting	It can be hosted in IIS	It can be hosted in IIS, windows activation service, Self-hosting, Windows service
Programming	[WebService] attribute has to be added to the class	[ServiceContract] attribute has to be added to the class
Model	[WebMethod] attribute represents the method exposed to client	[OperationContract] attribute represents the method exposed to client
Operation	One-way, Request- Response are the different operations supported in web service	One-Way, Request-Response, Duplex are different type of operations supported in WCF
XML	System.Xml.serialization name space is used for serialization	System.Runtime.Serialization namespace is used for serialization
Encoding	XML 1.0, MTOM(Message Transmission Optimization Mechanism), DIME, Custom	XML 1.0, MTOM, Binary, Custom
Transports	Can be accessed through HTTP, TCP, Custom	Can be accessed through HTTP, TCP, Named pipes, MSMQ,P2P, Custom
Protocols	Security	Security, Reliable messaging, Transactions
Exception Handling	Returns to the client as SOAP Faults	Handled in a better way by using FaultControl
Multithreading	Does not support multithreading	Supports multithreading by using ServiceBehaviour class

WCF Architecture



WCF Architecture-Contracts

- The data exchanged by a service is defined by a data contract.
- The message contract defines specific message parts using SOAP protocols
- The service contract specifies the actual method signatures of the service.
- Policies and bindings stipulate the conditions required to communicate with a service.

WCF Architecture–Service Runtime

- **Throttling Behavior** – Manages the number of messages processed.
- **Error Behavior** – Defines the result of any internal service error occurrence.
- **Metadata Behavior** – Specifies the availability of metadata to the outside world.
- **Instance Behavior** – Defines the number of instances that needs to be created to make them available for the client.
- **Transaction Behavior** – Enables a change in transaction state in case of any failure.
- **Dispatch Behavior** – Controls the way by which a message gets processed by the infrastructure of WCF.
- **Concurrency Behavior** – Controls the functions that run parallel during a client-server communication.
- **Parameter Filtering** – Features the process of validation of parameters to a method before it gets invoked.

WCF Architecture-Messaging

This layer, composed of several channels, mainly deals with the message content to be communicated between two endpoints. The two major layers are:

- **Transport Channels** – These channels are present at the bottom of a stack and are accountable for sending and receiving messages using transport protocols like HTTP, TCP, Peer-to-Peer, Named Pipes, and MSMQ.
- **Protocol Channels** – Present at the top of a stack, these channels also known as layered channels, implement wire-level protocols by modifying messages.

WCF Architecture-Activation and Hosting

This layer is the place where services are actually hosted or can be executed for easy access by the client.

- **IIS** – IIS stands for Internet Information Service. It offers a myriad of advantages using the HTTP protocol by a service. Here, it is not required to have the host code for activating the service code; instead, the service code gets activated automatically.
- **Windows Activation Service** – This is popularly known as WAS and comes with IIS 7.0. Both HTTP and non-HTTP based communication is possible here by using TCP or Namedpipe protocols.
- **Self-hosting** – This is a mechanism by which a WCF service gets self-hosted as a console application. This mechanism offers amazing flexibility in terms of choosing the desired protocols and setting own addressing scheme.
- **Windows Service** – Hosting a WCF service with this mechanism is advantageous, as the services then remain activated and accessible to the client due to no runtime activation.

WCF Endpoints

WCF Endpoint is a portal for communicating with the world.

End point consists of three components- A B and C.

- **Address (Where)**

Basically URL, specifies where this WCF service is hosted .Client will use this url to connect to the service. e.g

http://localhost:8090/MyService/SimpleCalculator.svc

WCF Endpoints

- Binding (How)

Binding describes how client will communicate with service. There are different protocols available for the WCF to communicate to the Client.

A binding has several characteristics, including the following:

- Transport –Defines the base protocol to be used like HTTP, Named Pipes, TCP, and MSMQ are some type of protocols.
- Encoding (Optional) – Three types of encoding are available–Text, Binary, or Message Transmission Optimization Mechanism (MTOM). MTOM is an interoperable message format that allows the effective transmission of attachments or large messages (greater than 64K).
- Protocol(Optional) – Defines information to be used in the binding such as Security, transaction or reliable messaging capability

WCF Endpoints

- **Contract**

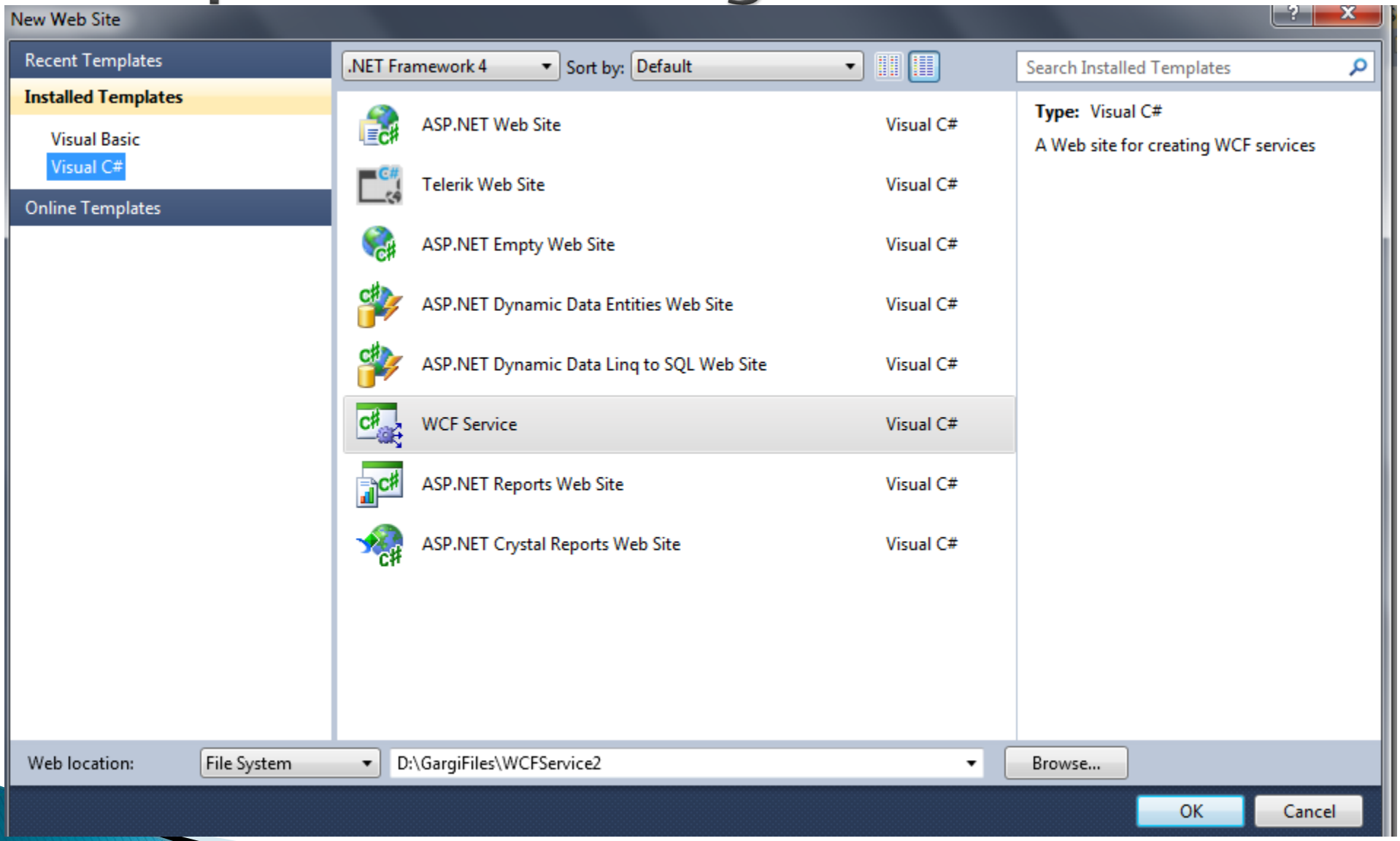
Collection of operation that specifies what the endpoint will communicate with outside world. Usually name of the Interface will be mentioned in the Contract, so the client application will be aware of the operations which are exposed to the client. Each operation is a simple exchange pattern such as one-way, duplex and request/reply.

WCF Endpoints

Endpoints will be mentioned in the web.config file on the created service.

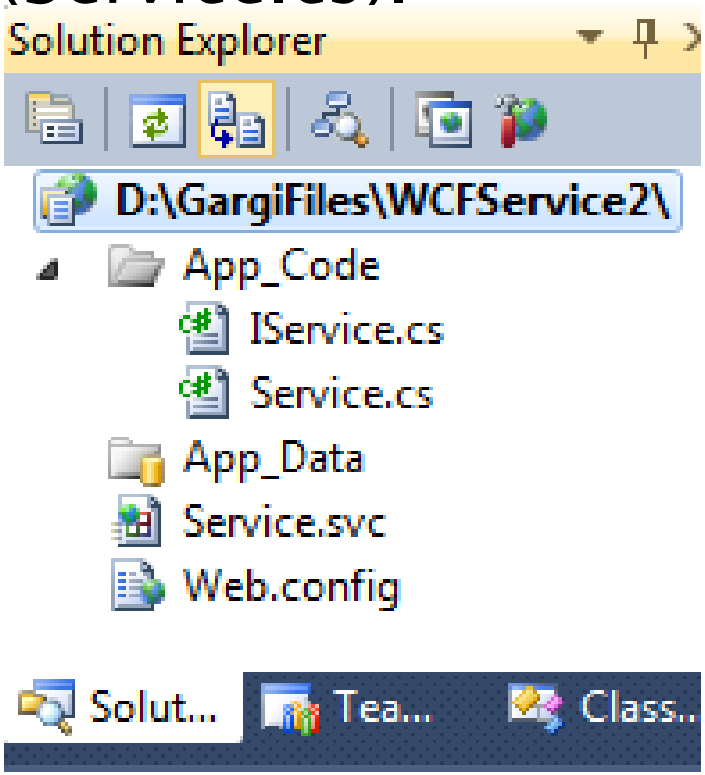
```
<system.serviceModel>
<services>
    <service name="MathService"
     behaviorConfiguration="MathServiceBehavior">
     <endpoint
      address="http://localhost:8090/MyService/MathService.svc"
contract="IMathService"
       binding="wsHttpBinding"/>
    </service>
  </services>
  <behaviors>
    <serviceBehaviors>
      <behavior name="MathServiceBehavior">
        <serviceMetadata httpGetEnabled="True"/>
        <serviceDebug includeExceptionDetailInFaults="true" />
      </behavior>
    </serviceBehaviors>
  </behaviors>
</system.serviceModel>
```

Steps for creating WCF service



Steps for creating WCF service

Once the project is created the following list of files will be added. Inside App_Code folder, there are two files added – one interface (IService.cs) and code-behind(Service.cs).



Steps for creating WCF service

- The .svc file will have the code behind file name mentioned in it.

```
<%@ ServiceHost Language="C#" Debug="true" Service="Service" CodeBehind="~/App_Code/Service.cs" %>
```

Steps for creating WCF service

```
[ServiceContract]
public interface IService
{

    [OperationContract]
    string GetData(int value);

    [OperationContract]
    CompositeType GetDataUsingDataContract(CompositeType composite);

    // TODO: Add your service operations here
}

// Use a data contract as illustrated in the sample below to add composite types to service operations.
[DataContract]
public class CompositeType
{
    bool boolValue = true;
    string stringValue = "Hello ";

    [DataMember]
    public bool BoolValue
    {
        get { return boolValue; }
        set { boolValue = value; }
    }

    [DataMember]
```

Steps for creating WCF service

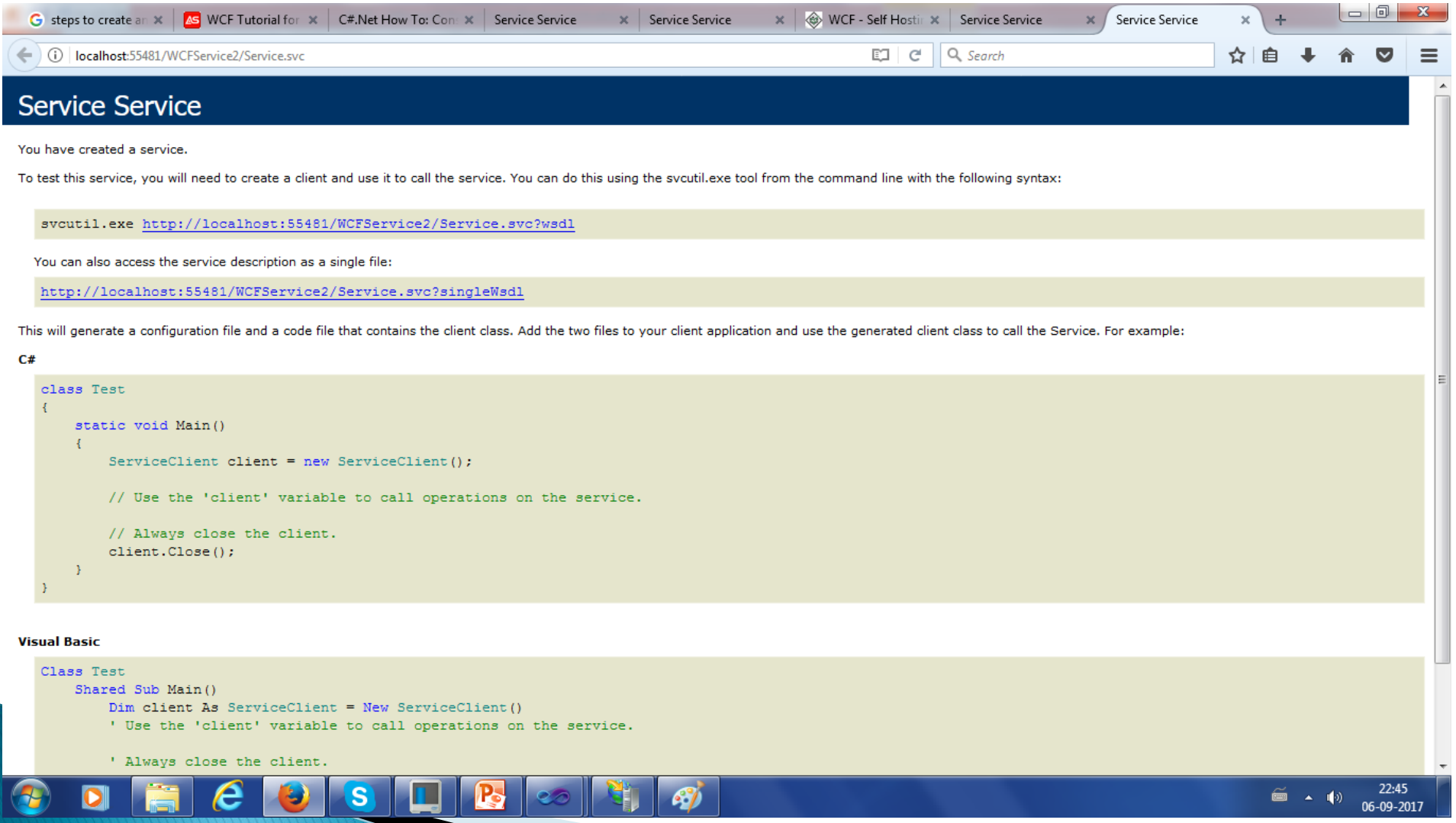
- The Service.cs will implement the IService interface.

```
public class Service : IService
{
    public string GetData(int value)
    {
        return string.Format("You entered: {0}", value);
    }

    public CompositeType GetDataUsingDataContract(CompositeType composite)
    {
        if (composite == null)
        {
            throw new ArgumentNullException("composite");
        }
        if (composite.BoolValue)
        {
            composite.StringValue += "Suffix";
        }
        return composite;
    }
}
```

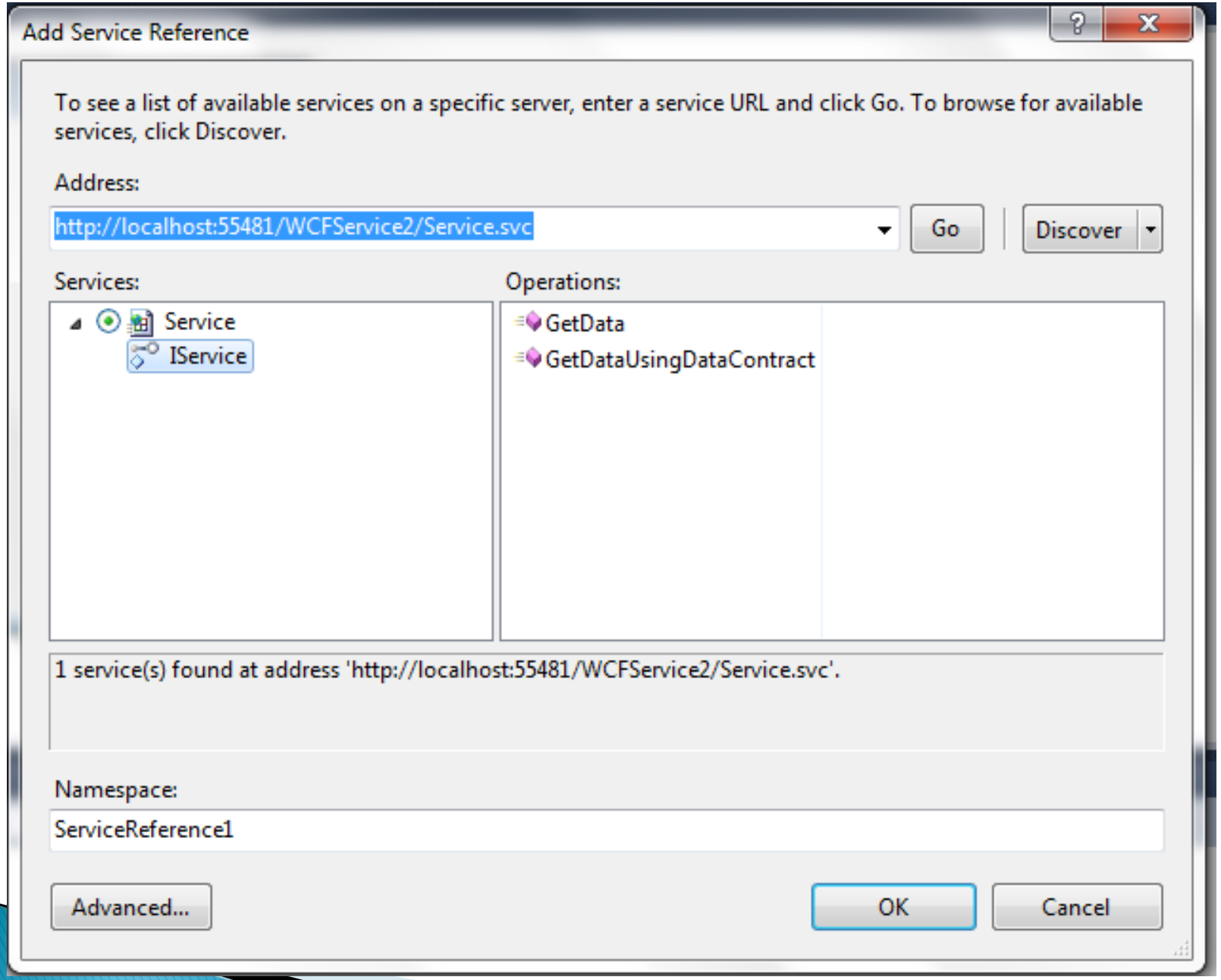
Steps for creating WCF service

- Run this service.



Steps for consuming WCF service

- Open a Console Application or Windows Form Application.
- Add Service Reference by right clicking on Reference.
- The following screen will appear.


Add Service Reference
To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.
Address:
http://localhost:55481/WCFService2/Service.svc
Go
Discover
Services:
Service
IService
Operations:
GetData
GetDataUsingDataContract
1 service(s) found at address 'http://localhost:55481/WCFService2/Service.svc'.
Namespace:
ServiceReference1
Advanced...
OK
Cancel

Steps for consuming WCF service

- Inside the code file access the WCF service in the following manner.

```
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            ServiceReference1.ServiceClient cl = new ServiceReference1.ServiceClient();
            cl.GetData(1);
        }
    }
}
```