

WCF Tutorial¹

WCF Overview

WCF stands for *Windows Communication Foundation*. The elementary feature of WCF is interoperability. It is one of the latest technologies of Microsoft that is used to build service oriented applications. Based on the concept of message-based communication, in which an HTTP request is represented uniformly, WCF makes it possible to have a unified API irrespective of diverse transport mechanisms.

WCF was released for the first time in 2006 as a part of the .NET framework with Windows Vista, and then got updated several times. 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.

Fundamental Concepts of WCF

Message

This is a communication unit that comprises of several parts apart from the body. Message instances are sent as well as received for all types of communication between the client and the service.

Endpoint

It defines the address where a message is to be sent or received. It also specifies the communication mechanism to describe how the messages will be sent along with defining the set of messages. A structure of an endpoint comprises of the following parts:

- **Address** – Address specifies the exact location to receive the messages and is specified as a Uniform Resource Identifier . It is expressed as `scheme://domain[:port]/[path]`. Take a look at the address mentioned below:

`net.tcp://localhost:9000/ServiceA`

Here, 'net.tcp' is the scheme for the TCP protocol. The domain is 'localhost' which can be the name of a machine or a web domain, and the path is 'ServiceA'.

- **Binding** – It defines the way an endpoint communicates. It comprises of some binding elements that make the infrastructure for communication. For example, a binding states the protocols used for transport like TCP, HTTP, etc., the format of message encoding, and the protocols related to security as well as reliability.
- **Contracts** – It is a collection of operations that specifies what functionality the endpoint exposes to the client. It generally consists of an interface name.

1 http://www.tutorialspoint.com/wcf/wcf_quick_guide.htm

Hosting

Hosting from the viewpoint of WCF refers to the WCF service hosting which can be done through many available options like selfhosting, IIS hosting, and WAS hosting.

Metadata

This is a significant concept of WCF, as it facilitates easy interaction between a client application and a WCF service. Normally, metadata for a WCF service is generated automatically when enabled, and this is done by inspection of service and its endpoints.

WCF Client

A client application that gets created for exposing the service operations in the form of methods is known as a WCF client. This can be hosted by any application, even the one that does service hosting.

Channel

Channel is a medium through which a client communicates with a service. Different types of channels get stacked and are known as Channel Stacks.

SOAP

Although termed as '*Simple Object Access Protocol*', SOAP is not a transport protocol; instead it is an XML document comprising of a header and body section.

Advantages of WCF

- It is interoperable with respect to other services. This is in sharp contrast to .NET Remoting in which both the client and the service must have .NET.
- WCF services offer enhanced reliability as well as security in comparison to ASMX *Active Server Methods* web services.
- Implementing the security model and binding change in WCF do not require a major change in coding. Just a few configuration changes is required to meet the constraints.
- WCF has builtin logging mechanism whereas in other technologies, it is essential to do the requisite coding.
- WCF has integrated AJAX and support for JSON *Javascript Object Notation*.
- It offers scalability and support for upcoming web service standards.
- It has a default security mechanism which is extremely robust.

WCF Versus Web Service

There are some major differences that exist between WCF and a Web service which are listed below.

- Attributes WCF: service is defined by ServiceContract and OperationContract attributes, whereas a web service is defined by WebService and WebMethod attributes.
- Protocols WCF: supports a range of protocols, i.e., HTTP, Named Pipes, TCP, and MSMQ, whereas a web service only supports HTTP protocol.

- **Hosting Mechanisms Various:** activation mechanisms are there for WCF hosting, i.e., IIS (*Internet Information Service*), WAS (*Windows Activation Service*), Self hosting and Windows Service, but a web service is hosted only by IIS.
- **Services WCF:** supports a robust security, trustworthy messaging, transaction and interoperability, while a web service only supports security services.
- **Serializer WCF:** Supports DataContract serializer by employing System.Runtime.Serialization, whereas a web service supports XML serializer by making use of System.Xml.Serialization.
- **Tools ServiceMetadata:** tool is used for client generation for a WCF service, while WSDL.EXE tool is used for generating the same for a web service.
- **Exception Handling In WCF,** unhandled exceptions are handled in a better way by making use of FaultContract. They do not return to the client like in a web service as SOAP faults.
- **Hash Table:** it is possible to serialize a Hash Table in WCF, but this is not the case in a web service.
- **Bindings WCF:** supports several types of bindings like BasicHttpBinding, WSDualHttpBinding, WSHttpBinding, etc., while a web service supports only SOAP or XML.
- **Multithreading WCF:** supports multithreading by using the ServiceBehavior Class, whereas this is not supported in a web service.
- **Duplex Service Operations WCF:** supports duplex service operations apart from supporting oneway and request-response service operations, whereas a web service does not support duplex service operations.

WCF Developers Tools

For developing a WCF service application, there are mainly two tools – Microsoft Visual Studio and CodePlex. Microsoft Visual Studio is a complete package of development tools, necessary for developing a large number of diverse applications like ASP.NET web applications, desktop applications, mobile applications, and many more.

Microsoft Visual Studio uses the .NET framework functionality. CodePlex on the other hand is an opensource project hosting site of Microsoft that offers several free tools for WCF service application development.

Microsoft Visual Studio

There are many editions of Microsoft Visual Studio and initially it was not an ardent supporter of WCF development. Nowadays, the latest version of Microsoft Visual Studio is also a preferred tool for developing WCF service application. There is also a readymade template in Visual Studio for developing WCF service application.

Selection of such a template leads to addition of files for the following purposes:

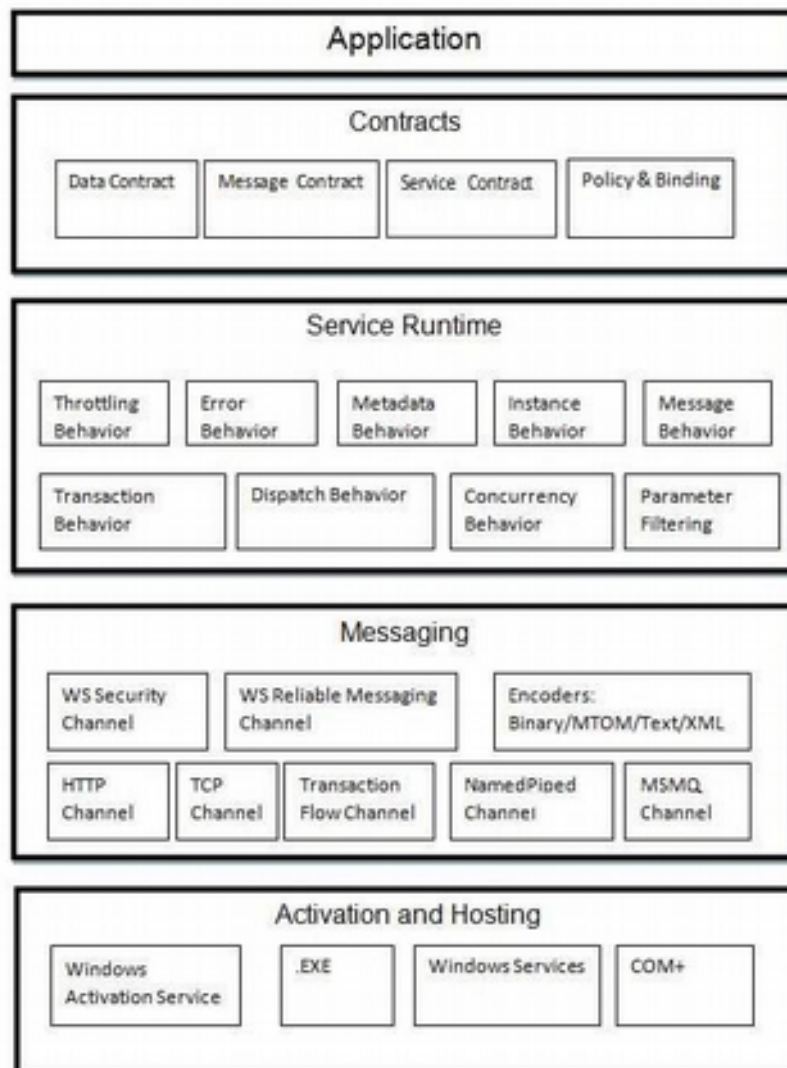
- Service contract
- Service implementation

- Service configuration

Requisite attributes get added automatically and a simple “Hello World” service gets created by Microsoft Visual Studio without even writing any code.

WCF Architecture

WCF has a layered architecture that offers ample support for developing various distributed applications. The architecture is explained below in detail.



Contracts

The contracts layer is just next to the application layer and contains information similar to that of a real-world contract that specifies the operation of a service and the kind of accessible information it will make. Contracts are basically of four types discussed below in brief:

- **Service contract** - This contract provides information to the client as well as to the outer world about the offerings of the endpoint, and the protocols to be used in the communication process.

- **Data contract** - The data exchanged by a service is defined by a data contract. Both the client and the service has to be in agreement with the data contract.
- **Message contract** - A data contract is controlled by a message contract. It primarily does the customization of the type formatting of the SOAP message parameters. Here, it should be mentioned that WCF employs SOAP format for the purpose of communication.
- **Policy and Binding** - There are certain pre-conditions for communication with a service, and such conditions are defined by policy and binding contract. A client needs to follow this contract.

Service Runtime

The service runtime layer is just below the contracts layer. It specifies the various service behaviors that occur during runtime. There are many types of behaviors that can undergo configuration and come under the 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.

Messaging

This layer, composed of several channels, mainly deals with the message content to be communicated between two endpoints. A set of channels form a channel stack and the two major types of channels that comprise the channel stack are the following ones:

- **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.

Activation and Hosting

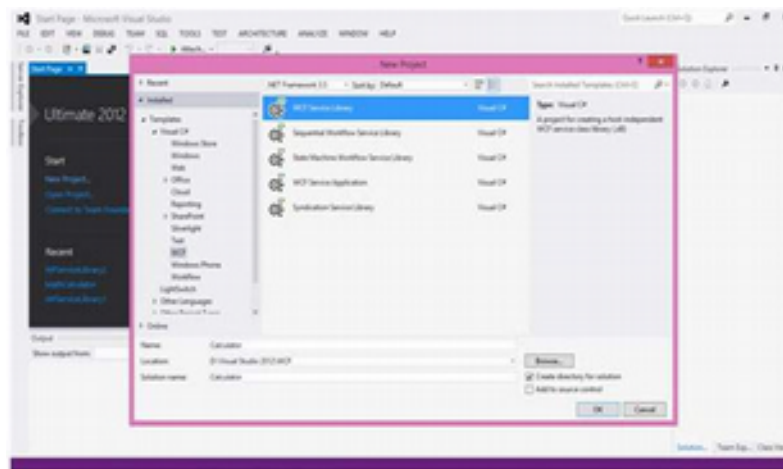
The last layer of WCF architecture is the place where services are actually hosted or can be executed for easy access by the client. This is done by various mechanisms discussed below in brief.

- **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 - Creating WCF Service

Creating a WCF service is a simple task using Microsoft Visual Studio 2012. Given below is the step-by-step method for creating a WCF service along with all the requisite coding, to understand the concept in a better way.

- Launch Visual Studio 2012.
- Click on new project, then in Visual C# tab, select WCF option.



A WCF service is created that performs basic arithmetic operations like addition, subtraction, multiplication, and division. The main code is in two different files – one interface and one class.

A WCF contains one or more interfaces and its implemented classes.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace WcfServiceLibrary1
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to
    // change the interface name "IService1" in both code and config file // together.

```

```

[ServiceContract]
public interface IService1
{
    [OperationContract]
    int sum(int num1, int num2);

    [OperationContract]
    int Subtract(int num1, int num2);

    [OperationContract]
    int Multiply(int num1, int num2);

    [OperationContract]
    int Divide(int num1, int num2);
}

// 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]
    public string StringValue
    {
        get { return stringValue; }
        set { stringValue = value; }
    }
}
}

```

The code behind its class is given below.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace WcfServiceLibrary1
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to // change the
class name "Service1" in both code and config file
    // together.

    public class Service1 : IService1
    {
        /// This Function Return summation of two integer numbers

        public int sum(int num1, int num2)
        {

```

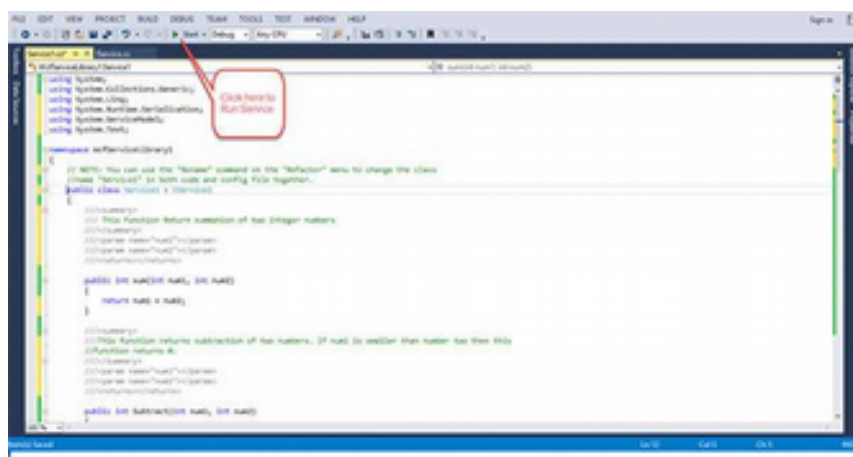
```
        return num1 + num2;
    }
    // This function returns subtraction of two numbers.
    // If num1 is smaller than number two then this function returns 0

    public int Subtract(int num1, int num2)
    {
        if (num1 > num2)
        {
            return num1 - num2;
        }
        else
        {
            return 0;
        }
    }
    // This function returns multiplication of two integer numbers.

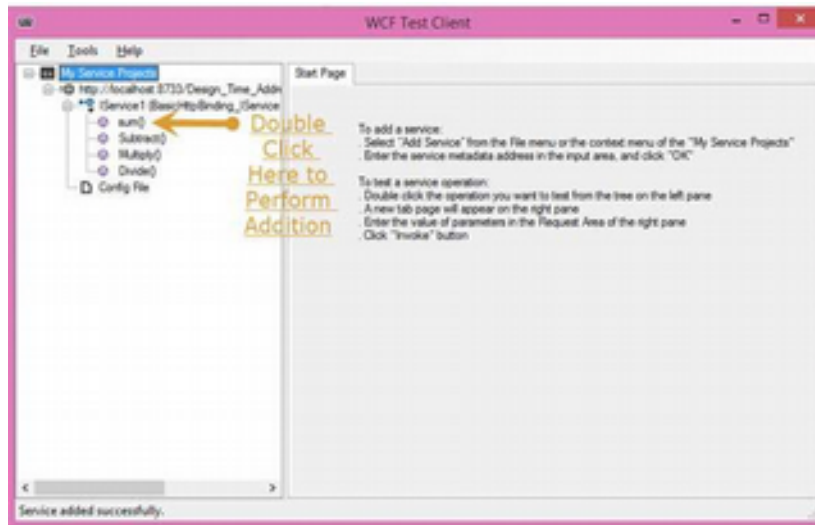
    public int Multiply(int num1, int num2)
    {
        return num1 * num2;
    }
    // This function returns integer value of two integer number.
    // If num2 is 0 then this function returns 1.

    public int Divide(int num1, int num2)
    {
        if (num2 != 0)
        {
            return (num1 / num2);
        }
        else
        {
            return 1;
        }
    }
}
}
```

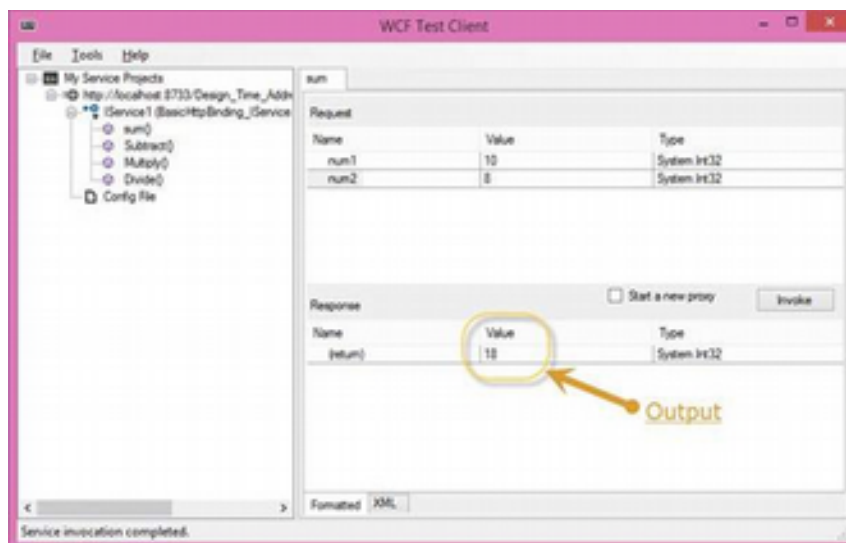
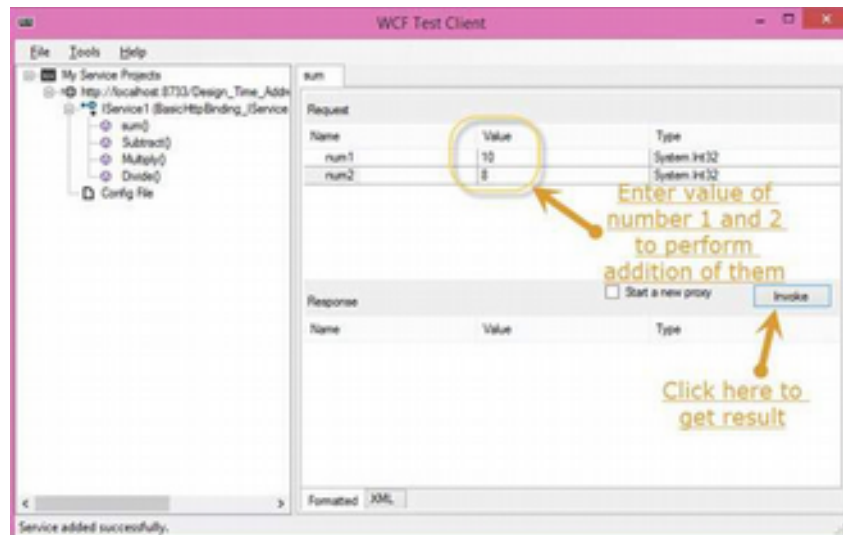
To run this service, click the Start button in Visual Studio.



While we run this service, the following screen appears.

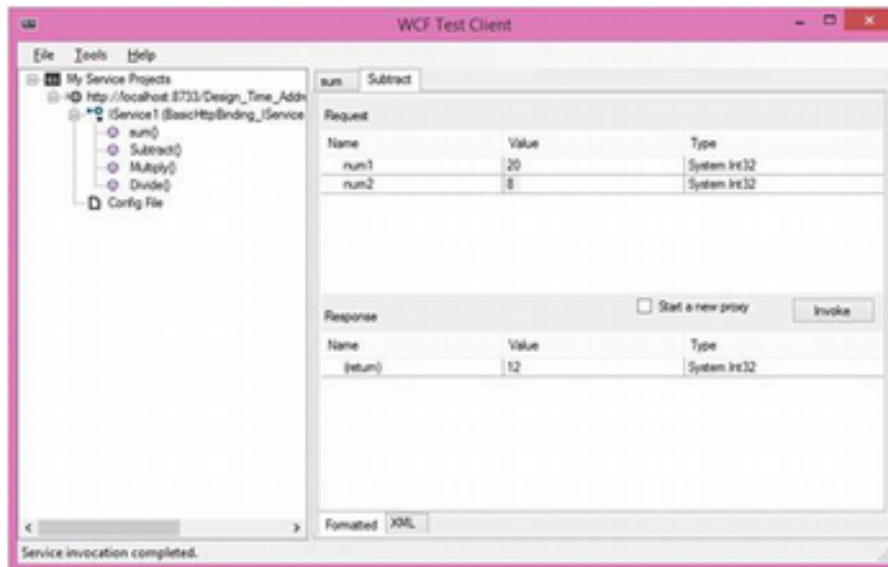


On clicking the sum method, the following page opens. Here, you can enter any two integer numbers and click on the Invoke button. The service will return the summation of those two numbers.



Like summation, we can perform all other arithmetic operations which are listed in the menu. And here are the snaps for them.

The following page appears on clicking the Subtract method. Enter the integer numbers, click the Invoke button, and get the output as shown here:



WCF - Hosting WCF Service

After creating a WCF service, the next step is to host it so that the client applications can consume it. This is known as WCF service hosting. A WCF service can be hosted by using any of the four ways given below:

- **IIS Hosting** - IIS stands for *Internet Information Services*. Its working model is similar to that of ASP.NET while hosting a WCF service. The best feature of IIS hosting is that the service activation is processed automatically. IIS hosting also offers process health monitoring, idle shutdown, process recycling, and many more features to facilitate a WCF service hosting.
- **Self-Hosting** - When a WCF service is hosted in a managed application, it is known as self-hosting. It requires a developer to write the requisite coding for ServiceHost initialization. In self-hosting, a WCF service can be hosted in a variety of applications like Console application, Windows form, etc.
- **WAS Hosting** - Hosting a WCF service in *Windows Activation Service (WAS)* is most advantageous because of its features such as process recycling, idle time management, common configuration system, and support for HTTP, TCP, etc.
- **Windows Service Hosting** - For local system clients, it is best to host WCF service as a window service and this is known as *Window Service Hosting*. All the versions of Windows support this type of hosting and here, the Service Control Manager can control the process lifecycle of the WCF service.

Creating a Self Hosted WCF Service

<http://www.codeproject.com/Articles/650869/Creating-a-Self-Hosted-WCF-Service>