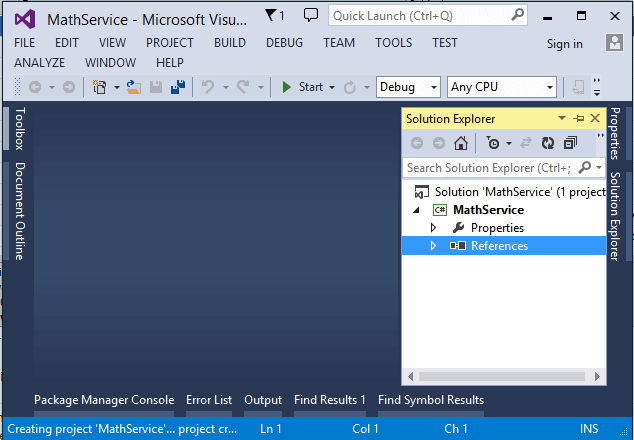
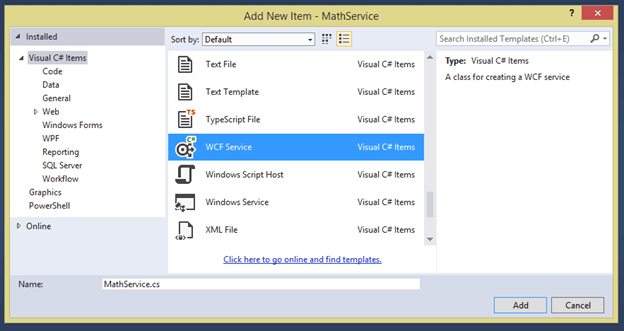
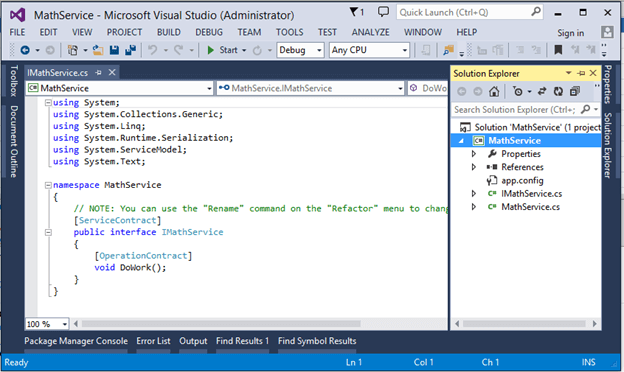
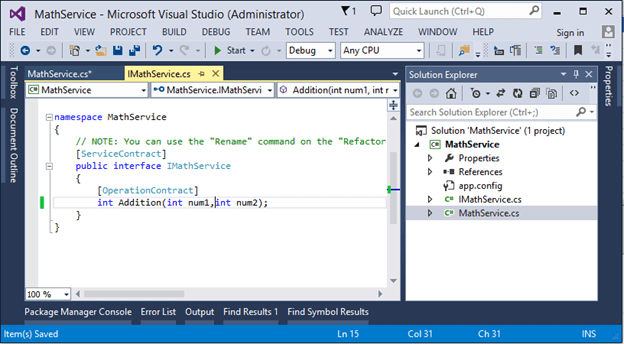
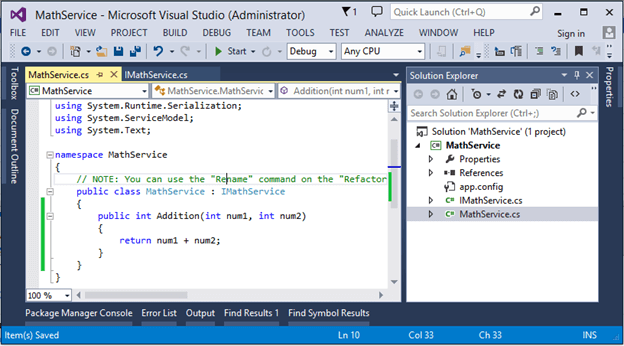
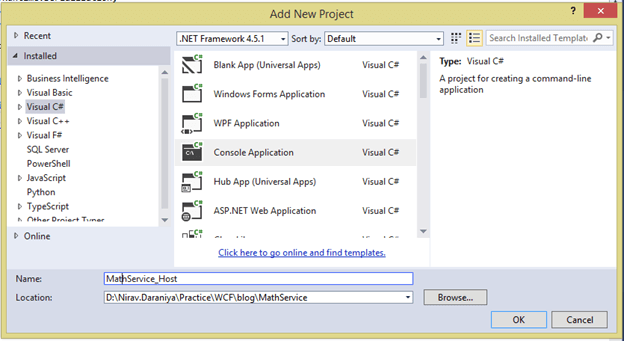
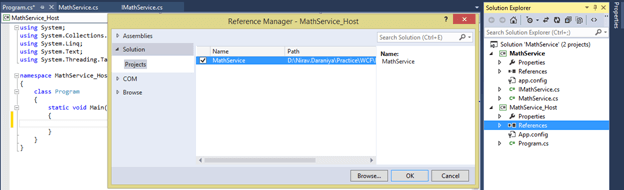
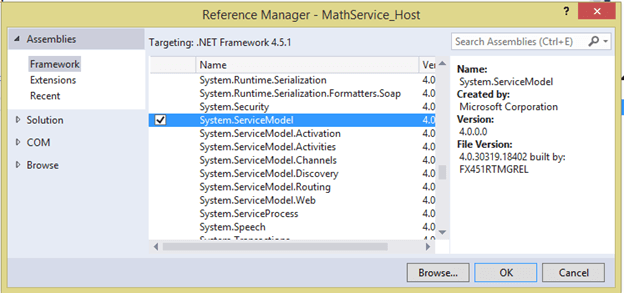
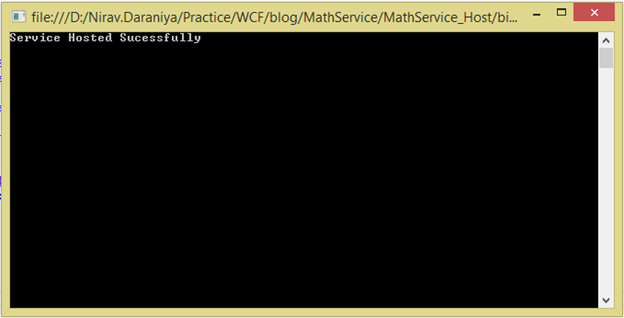
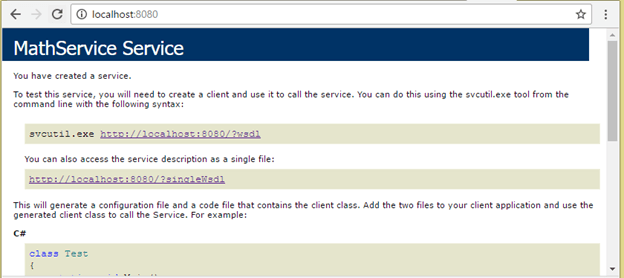
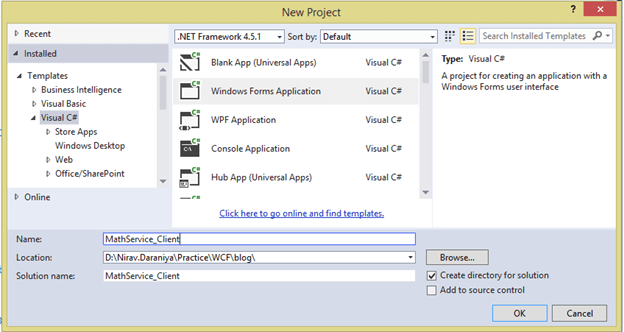
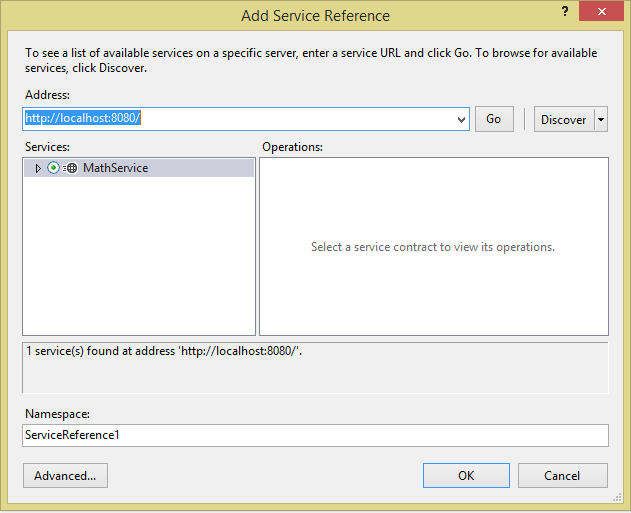
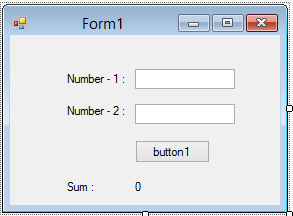
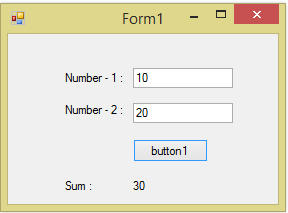
WCF stands for Windows Communication Foundation. Using the WCF Service, you can communicate with multiple types of clients. Here, you can say that WCF Service will communicate with both the Windows and Linux client. First, we make one simple WCF Service for “addition” and host it on console application; then, consume it from one Windows Form application.

1. Open Visual Studio instance with "Run as administrator" and create one class library project. Name it “MathService”. It will generate one default class file. Now, you have to delete this “Class1.cs” file.   
     
   
2. Now, right click on the project and add a new item. It will open a new window. Select the “WCF Service” and give it the name “MathService”.  
     
   
3. It will generate 2 files -  interface and class file.  
     
   
4. Now, open the interface file and add a new function. Name it “Addition.” It will accept 2 integers - input parameter and return type are also integers, as show in the below figure.  
     
   
5. Now, go to “MathService.cs” file and implement this interface, as shown in the below image.  
     
   
6. Now, build the project. Our WCF service is ready now. We have to host it so that the client will consume it.
7. Now, in same solution, add new project as console application and name it as ‘MathService\_Host’.  
     
   
8. Now, give solution reference of WCF Service to console application project and also give “System.ServiceModel” namespace reference to host project.  
     
     
     
   
9. Now, open ‘Program.cs’ file and write the below code.
   1. ServiceHost host = **new** ServiceHost(**typeof**(MathService.MathService));
   2. host.Open();
   3. Console.WriteLine("Service Hosted Sucessfully");
   4. Console.Read();
10. Now, for communicating with client end-point, we have to add configuration file (if already in project, then no need to add). So, we first add App.config file to project, then write the below code to it.
    1. <?xml version="1.0" encoding="utf-8" ?>
    2. <configuration>
    3. <system.serviceModel>
    4. <behaviors>
    5. <serviceBehaviors>
    6. <behavior name="mexBehaviour">
    7. <serviceMetadata httpGetEnabled="true" />
    8. </behavior>
    9. </serviceBehaviors>
    10. </behaviors>
    11. <services>
    12. <service name="MathService.MathService" behaviorConfiguration="mexBehaviour">
    13. <endpoint address="MathService" binding="basicHttpBinding" contract="MathService.IMathService">
    14. </endpoint>
    15. <endpoint address="MathService" binding="netTcpBinding" contract="MathService.IMathService">
    16. </endpoint>
    17. <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />
    18. <host>
    19. <baseAddresses>
    20. <add baseAddress="http://localhost:8080/" />
    21. <add baseAddress="net.tcp://localhost:8090" />
    22. </baseAddresses>
    23. </host>
    24. </service>
    25. </services>
    26. </system.serviceModel>
    27. </configuration>

Here, in the above code, we have to specify the binding configuration, end-point details, binding protocols, service behaviors, base address etc.  
  
Here, we specify the “MathService.MathService” means you have to give full name of the service, like ‘Namespace Name.Service Name’

1. Now, our host project is ready. Let's build the host project and run it. Check the output in console window.  
     
   
2. Now, we will check this service in browser. So, open the http://localhost:8080/ URL in browser and the output will be as shown below.  
     
   
3. Now, we have to create client. For that, open another instance of Visual Studio and run it as administrator. Create new Windows Form application named as “MachService\_Client” and add service reference to it.  
     
     
     
   *Give Service Reference.*  
     
   
4. Now, design the client form as shown in the below image.  
     
   
5. Write the below code on ButtonClick event.
   1. ServiceReference1.MathServiceClient \_obj = **new** ServiceReference1.MathServiceClient("NetTcpBinding\_IMathService");
   2. lblResult.Text = \_obj.Addition(Convert.ToInt32(textBox1.Text), Convert.ToInt32(textBox2.Text)).ToString();
6. Now, run the project and check the output.



Thanks for reading my article. If you have any confusion regarding this article, please post your comment in the comment box.