***Create a simple WCF Service and Client application.***

In this post, I will demonstrate how to create a simple WCF client and service console application. This post is intended for beginners who can follow the following steps to get a basic WCF Service up and running.

Minimum requirement: .Net framework 3.0

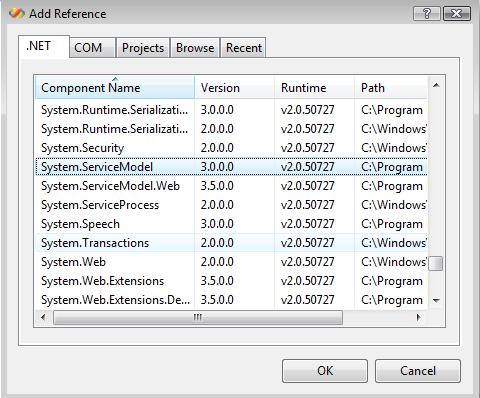
For this, we will create three projects.

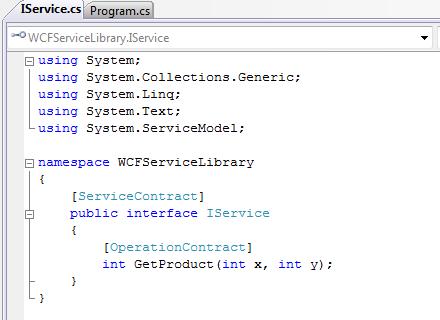
* ***A service class Library***: This library will contain the Service Contract, Operation Contracts and Data Contracts that will be shared by both WCF service and client applications.
* ***WCF service application***: A console application that will host the service.
* ***WCF client application***: A console application that will call a remote wcf service method and display the response from the service.

Lets start by creating the service class library.

1. Create a class library application and name it ***WCFServiceLibrary.***
2. Delete the default Class1.cs file and add new interface file named “***IService.cs***”.
3. Add reference to the **Syste.ServiceModel** in the project and then add the corresponding **using** directives.

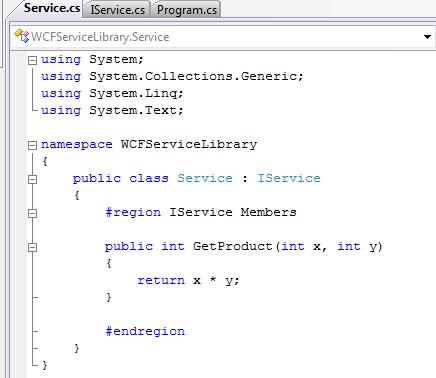
Then make the following change in the ***IService*** interface.

****

****

Here we add a definition for method ***GetProduct*** which will return the product of two numbers passed in as parameter.

1. Add another cs file named “***Service.cs***”. Implement the “***IService.cs***” interface and its method.



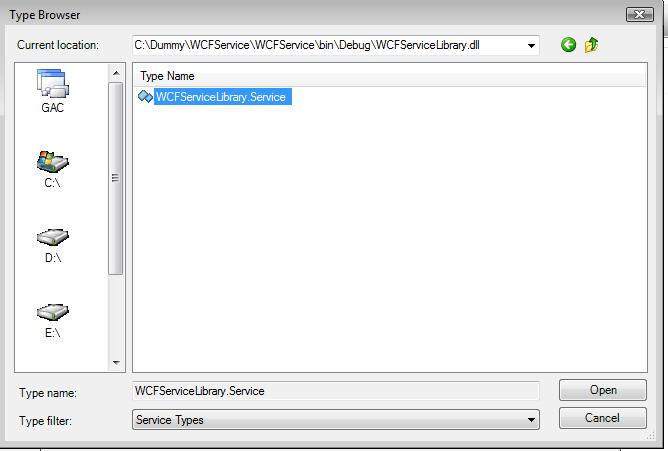
1. Now since we have the service library with a method exposed, lets create a service application now.
2. Create a console application named “***WCFService***” and add reference to the ***WCFServiceLibrary*** project.

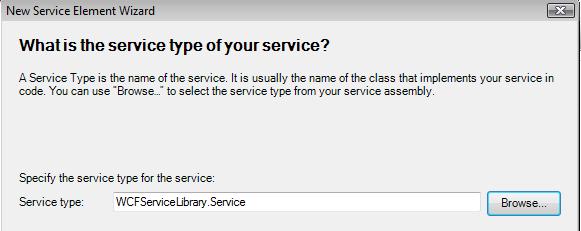
Also add reference to System.ServiceModel. Add the following using directive.

using System.ServiceModel;

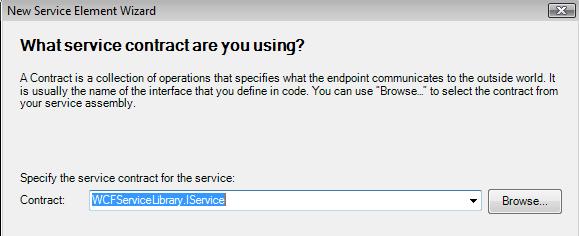
using WCFServiceLibrary;

1. Lets now create a configuration file for the WCF Service. This configuration file will define the Endpoint binding, endpoint address and behaviors for the service. Add a new ***app.config*** file to the ***WCFService*** project.
2. Open the ***WCF Service Configuration Editor***, Tools -> WCF Service Configuration Editor. Then select the app.config file created in above step. You can now follow the wizard to create the binding for the service.
3. Since we are creating a config file for service, select the “***Create a New Service***” link.
4. In ***Service Type*** step***,*** browse to the ***WCFServiceLibrary.dll*** andselect the type “***WCFServiceLibrary.Service***”.

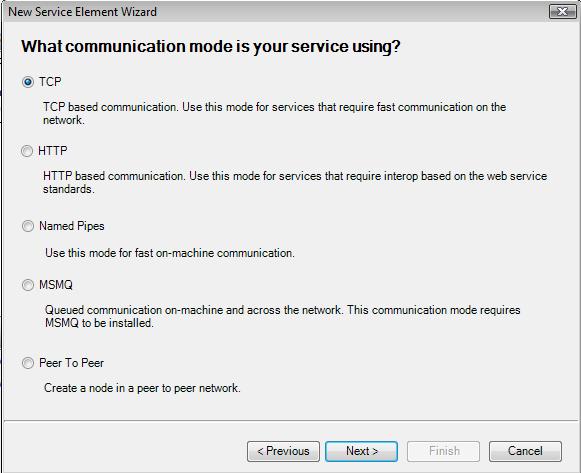
******

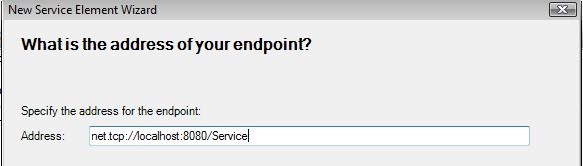
******

1. Click ***Next,*** and you will see the Contract for the service.

******

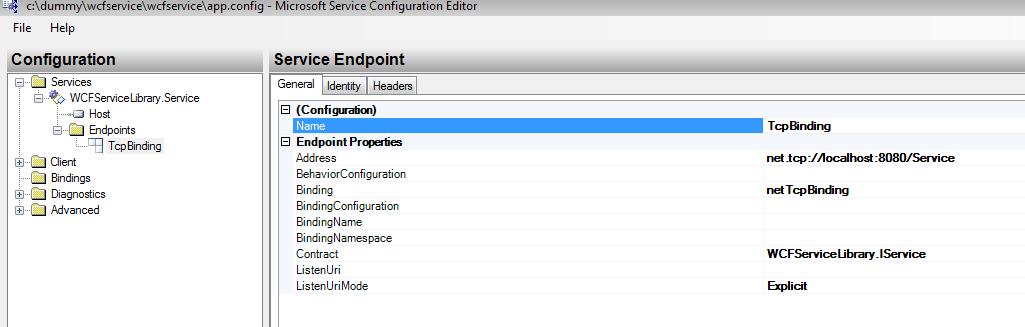
1. Click next; here you need to select the mode you want to use for binding, Http, Tcp, Named Pipes etc. Select Tcp and click next. Now set the endpoint address like below:

******

******

Click Next and finish the wizard.

1. Then go to “***Endpoints***” node and give the name for configuration as ***TcpBinding.*** Then save the project.



1. Select root ***Services*** node and then create a binding configuration by selecting “***(Default )Click to create***”. Supply the name of the binding as ***NetTcpBinding.*** With rest set to default, save the config file.
2. Now add the following lines of code to the ***program.cs*** file of ***WCFService*** project.

namespace WCFService

{

class Program

{

static void Main(string[] args)

{

try

{

ServiceHost host = new ServiceHost(typeof(WCFServiceLibrary.Service));

host.Open();

Console.WriteLine(string.Format("The service is running....."));

}

catch (Exception exc)

{

Console.WriteLine(string.Format("Error: {0}",exc.Message));

}

Console.ReadLine();

}

}

When you run the service you should see a console open up with message “**The service is running…..**”

1. Now since the service is running, lets create a client to access the remote service. Create a console application named “***WCFClient***” and add the reference to the **System.ServiceModel**, **WCFServiceLibrary** project. Then add the following using directives in the program.cs file.

using System.ServiceModel;

using System.ServiceModel.Channels;

using WCFServiceLibrary;

1. Now let’s create the app.config file for the client. Again we are going to use the “***WCF Service Configuration Editor***”. The catch here is that, now we can just specify the app.config file of the service application and it will automatically generate the settings for the client according to the service. So follow the steps below to get this done.
   1. Add a new app.config file to the ***WCFClient*** application.
   2. Open the ***WCF Service Configuraiton Editor***and select to open this app.config file.
   3. Select the ***Client*** node in the left tree and on the right side you will see a link label ***Create a New Client***. Click it.
   4. Select the “***From service config***” option and browse to the app.config file of the WCFService application which we created at the end of step 8.
   5. Proceed by selecting next, and provide “***ClientTcpBinding***” as name when you are prompted for the client configuration name. Proceed to finish and then save the config file.

Remove the <identity> section if exists. The configuration file should look like this.

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<system.serviceModel>

<bindings>

<netTcpBinding>

<binding name="NetTcpBinding" />

</netTcpBinding>

</bindings>

<client>

<endpoint address="net.tcp://localhost:8080/Service" binding="netTcpBinding"

bindingConfiguration="NetTcpBinding" contract="WCFServiceLibrary.IService"

name="ClientTcpBinding">

</endpoint>

</client>

</system.serviceModel>

</configuration>

1. Now add the following codes of line to the program.cs file of ***WCFClient*** project.

namespace WCFClient

{

class Program

{

static void Main(string[] args)

{

try

{

ChannelFactory<WCFServiceLibrary.IService> factory = new ChannelFactory<IService>("ClientTcpBinding");

factory.Open();

IService remoteService = factory.CreateChannel();

int x=4;

int y=4;

int retVal = remoteService.GetProduct(x, y);

Console.WriteLine(string.Format("The product of {0} and {1} is {2} ",x,y,retVal));

}

catch (Exception exc)

{

Console.WriteLine(string.Format("Error:{0}",exc.Message));

}

Console.ReadLine();

}

}

}

1. Now run the ***WCFService*** application first. Right Click WCFService->Debug->Start New Instance.

***13.jpg***

1. Run the ***WCFClient*** project with WCFClient->Debug->Start New Instance. You should see the following output.

14.jpg

So here we learned how to create a WCF Service, host it in console application, program the client to access this remote service exposed method. We also learned how we use the ***WCF Service Configuration Editor*** tool which saved us from manually editing the configuration files and save our time.

Enjoy!!!:-)