

3 Ways to generate proxy for WCF Service



Imran Abdul Ghani, 16 Jun 2014

CPOL



0.00 (No votes)

In Windows Communication Foundation, for a client application to communicate with a WCF Service, we have following options: Using ChannelFactory Generating Proxies I have already discussed about difference between ChannelFactory and Proxies in one of my previous WCF Tutorial on this blog. In this ar

In Windows Communication Foundation, for a client application to communicate with a WCF Service, we have following options:

- Using ChannelFactory
- Generating Proxies

I have already discussed about [difference between ChannelFactory and Proxies](#) in one of my previous [WCF Tutorial](#) on this blog. In this article, we will focus only on multiple ways to generate proxy in Windows Communication Foundation.

"A Proxy in Windows Communication Foundation is a class that enables client applications to communicate with a service by sending and receiving messages. It actually encapsulates a number of service details like service path, service implementation technology, platform being used, communication protocol etc. as well as all the methods (signature only) of the Service Contract."

Windows Communication Foundation supports following three ways to generate proxy for a WCF Service.

- Adding Service Reference
- Implementing ClientBase
- Using Tool i.e. SvcUtil.exe

So, in this WCF Tutorial, we will carry out all three possible ways to generate proxy for a WCF service. Let's create a simple WCF Service first.

Create WCF Service

I have already provided implementation for a new WCF Service creation and hosting in a separate WCF Tutorial [here](#). Please follow the steps to create a "[StudentService](#)" and [host in a console application](#).

Option 1: Generate Proxy Adding Service Reference

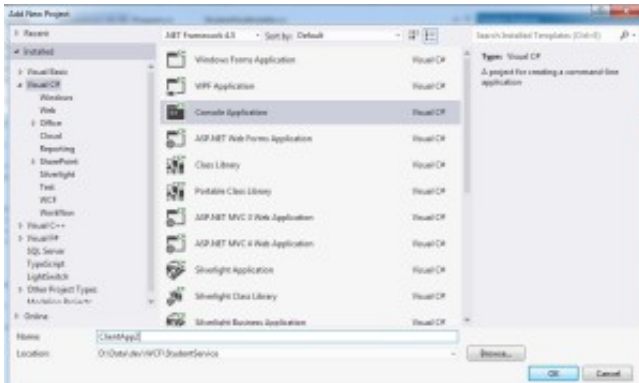
Implementation of generating proxy by adding a service reference is also [available here](#) for the same WCF service i.e. "StudentService".

Option 2: Generate Proxy by implementing ClientBase<T> class

Generating proxy by using ClientBase<T> class option has an advantage that it creates proxy at run time, so

it will accommodate service implementation changes. Let's follow the steps to generate proxy.

- Add a Client Project to solution named as "ClientApp2" that is basically a Console Application.



- Add reference of StudentService Project to ClientApp2.
 - Add following proxy class using **ClientBase<T>** as:

```
public class StudentServiceProxy : ClientBase<IStudentService>, IStudentService
{
    public string GetStudentInfo(int studentId)
    {
        return base.Channel.GetStudentInfo(studentId);
    }
}
```

Note: Don't forget to add **"using StudentService"** to class.

- Following is the code for *program.cs* class in ClientApp2. We are using above created proxy class to communicate with WCF Service **"StudentService"**.

```
class Program
{
    static void Main(string[] args)
    {
        StudentServiceProxy myclient;
        myclient = new StudentServiceProxy();

        int studentId = 1;
        Console.WriteLine("Calling StudentService with StudentId = 1....");
        Console.WriteLine("Student Name = {0}", myclient.GetStudentInfo(studentId));
        Console.ReadLine();
    }
}
```

Note: Don't forget to add **"using System.ServiceModel"** to class.

- App.Config file will have following configuration:

```
<system.serviceModel>
```

```
<bindings>
  <wsHttpBinding>
    <binding name="WSHttpBinding_IStudentService" />
  </wsHttpBinding>
</bindings>
<client>
  <endpoint address="http://localhost:4321/StudentService"
    binding="wsHttpBinding"
    bindingConfiguration="WSHttpBinding_IStudentService"
    contract="StudentService.IStudentService"
    name="WSHttpBinding_IStudentService">
  </endpoint>
</client>
</system.serviceModel>
```

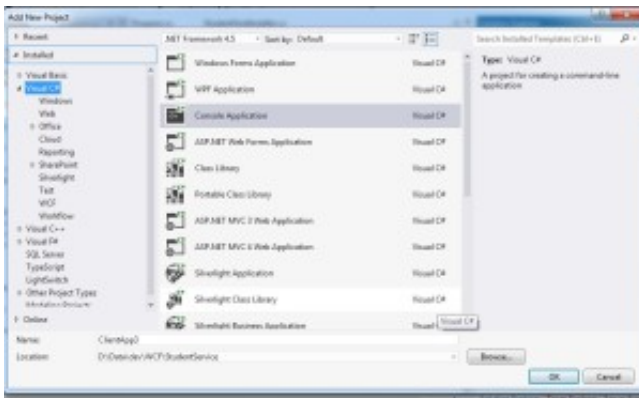
Now, when we run the client application, we will receive the following same output as we get in earlier option "Adding Service Reference".



Option 3: Generate Proxy by using SvcUtil.exe Tool

Let's generate proxy by using third option i.e. SvcUtil.exe Tool by following step by step approach.

- Add a Client Project to solution named as "ClientApp3" that is basically a Console Application.



- Our WCF Service must be running, so let's run our service.



- Open Visual Studio Command Prompt and generate proxy using svcutil.exe tool as follows:




```
svcutil http://localhost:4321/StudentService /out:StudentServiceProxy.cs
```


Share

About the Author



Imran Abdul Ghani

  Software Developer (Senior) Emaratech
United Arab Emirates 

Imran Abdul Ghani has more than 10 years of experience in designing/developing enterprise level applications. He is Microsoft Certified Solution Developer for .NET(MCSD.NET) since 2005. You can reach his blogging at [WCF Tutorials](#), [Web Development](#), [SharePoint for Dummies](#).