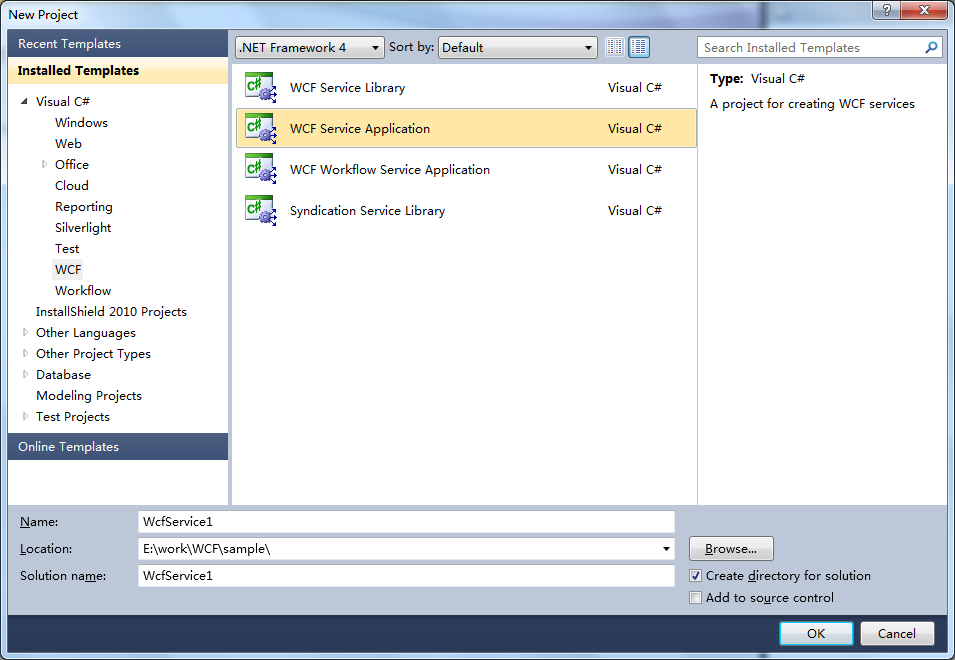
在VS2010里建立一个最简单的WCF服务，基本流程如下：

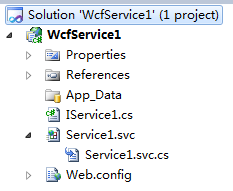
**一：新建WCF应用**

首先，新建一个WCF服务的应用（这里以framework 4.0为例），如下图所示，



建立完成之后，VS将自动生成一个最简单的WCF工程，在这个应用中，包含了最基本**“契约(Contract)”**以及**“服务功能实现(Service)”.**

工程如下：



不需要编辑任务文件，直接编译生成，得到一个WcfService1.dll文件

**二、WCF应用中的契约(Contract)**

在生成的WCF工程中，IService1.cs中为**“契约(Contract)”**（本例中契约和服务放在同一个工程下了，实际上也可以分为两个工程），代码如下：

**[**ServiceContract**]**

publicinterfaceIService1

**{**

**[**OperationContract**]**

    stringGetData**(**intvalue**);**

**[**OperationContract**]**

    CompositeTypeGetDataUsingDataContract**(**CompositeTypecomposite**);**

// TODO: Add your service operations here

**}**

// Use a data contract as illustrated in the sample below to add composite types to service operations.

**[**DataContract**]**

publicclassCompositeType

**{**

    boolboolValue=true**;**

    stringstringValue="Hello "**;**

**[**DataMember**]**

    publicboolBoolValue

**{**

        get**{**returnboolValue**; }**

        set**{**boolValue=value**; }**

**}**

**[**DataMember**]**

    publicstringStringValue

**{**

        get**{**returnstringValue**; }**

        set**{**stringValue=value**; }**

**}**

**}**

可以看到，声明了服务契约IService1，以接口形式声明，其中还包括两个操作契约GetData以及GetDataUsingDataContract。还声明了数据契约CompositeType，以类的形式声明，包含两个数据成员BoolValue和StringValue。

**三、WCF应用中的服务功能实现（Service）**

在生成的WCF应用中，Service1.svc.cs中为**“服务功能实现(Service)”**（本例中契约和服务放在同一个工程下了，实际上也可以分为两个工程），代码如下（Service1.svc还有其它作用，后面再说）：

publicclassService1**:**IService1

**{**

    publicstringGetData**(**intvalue**)**

**{**

        returnstring.Format**(**"You entered: {0}"**,**value**);**

**}**

    publicCompositeTypeGetDataUsingDataContract**(**CompositeTypecomposite**)**

**{**

        if**(**composite==null**)**

**{**

            thrownewArgumentNullException**(**"composite"**);**

**}**

        if**(**composite.BoolValue**)**

**{**

            composite.StringValue+="Suffix"**;**

**}**

        returncomposite**;**

**}**

**}**

可以看到，这个Service1类实现了在契约中声明的IService1接口（服务契约），也使用到了CompositeType类（数据契约）；实现了GetData以及GetDataUsingDataContract这两个服务契约中功能，这些功能即为WCF服务允许外部程序进行调用的功能。

**四、寄宿（Host）WCF服务**

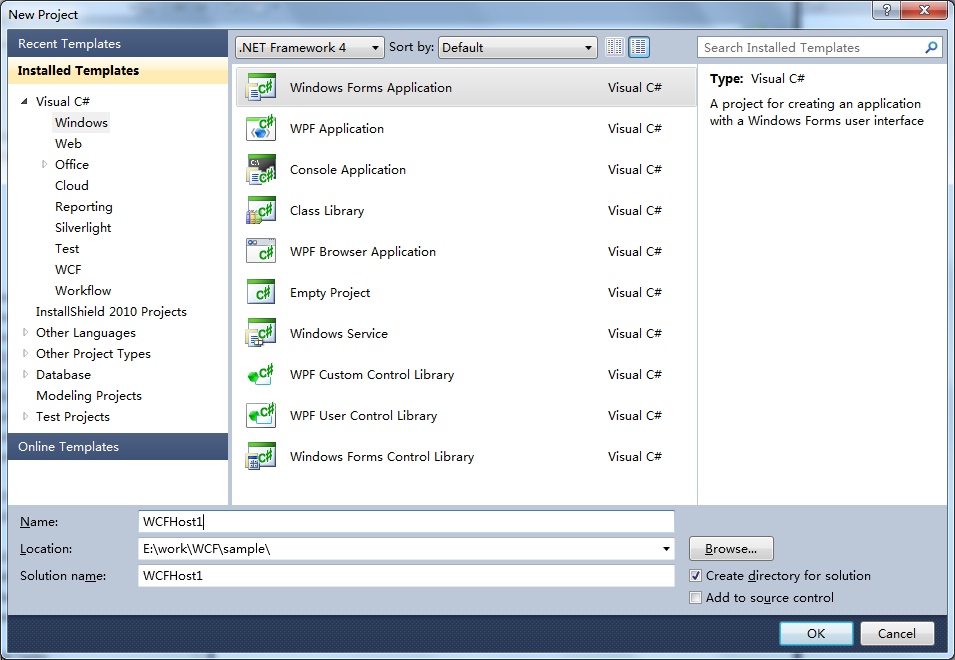
有2种常见的寄宿方式：

        1）一种是为一组WCF服务创建一个托管的应用程序，通过手工启动程序的方式对服务进行寄宿，所有的托管的应用程序均可作为WCF服务的宿主，比如 Console应用、Windows Forms应用和ASP.NET应用等，我们把这种方式的服务寄宿方式称为自我寄宿（Self Hosting）。

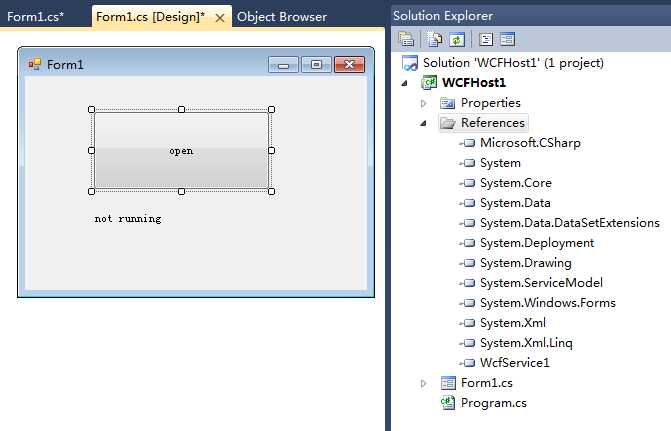
        2）另一种则是通过操作系统现有的进程激活方式为WCF服务提过宿主，Windows下的进程激活手段包括IIS、Windows Service或者WAS（Windows Process Activation Service）等

无论采用哪种寄宿方式，在为某个服务创建 ServiceHost的过程中，WCF框架内部会执行一系列的操作，其中最重要的步骤就是为服务创建服务描述（Service Description）

本例以第一种为例，建立一个WinForm应用作为托管程序，在VS2010中，建立一个普通的WinForm程序（以frameword4.0为例），如下图：

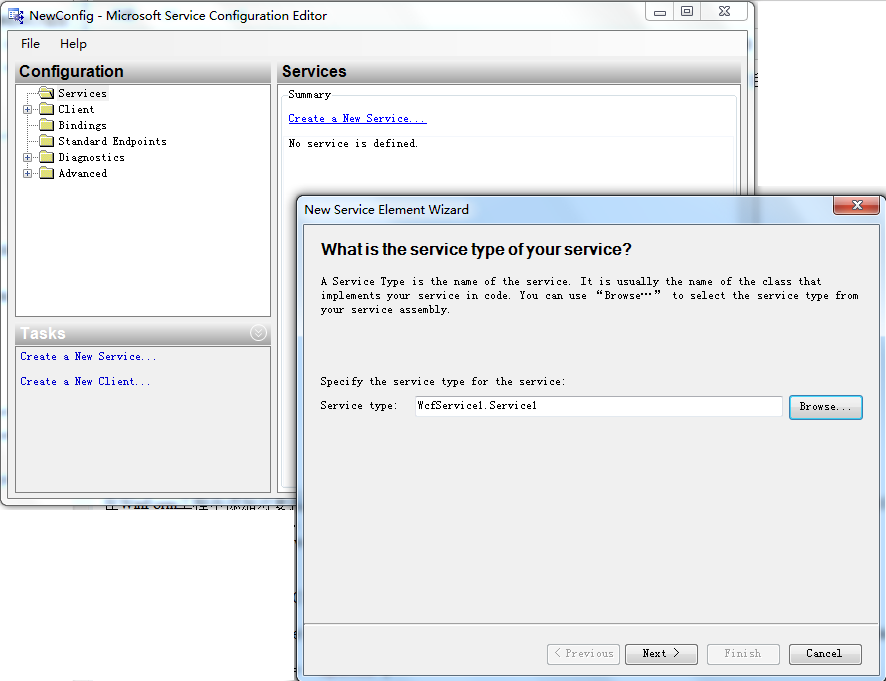


在工程中，添加对System.ServiceModel的引用，并添加对刚才生成的WcfService1.dll的引用，然后在Form上添加一个Button和一个Label控件，如下图：

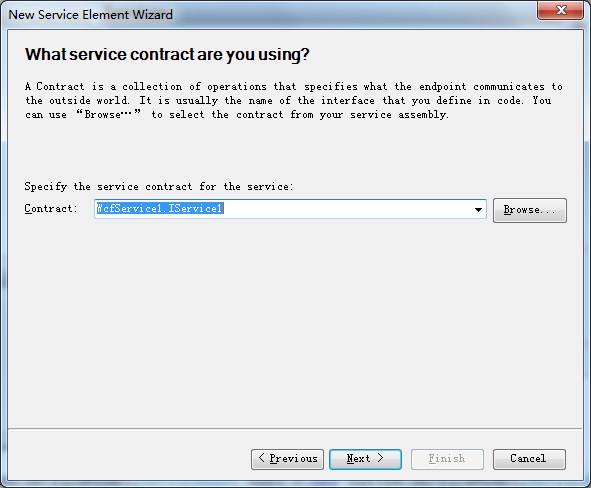


在WinForm工程中添加对要托管的WCF服务的描述，既可以通过代码实现，也可以通过工具生成。本例在VS2010下，通过Tools-WCF Service Configeration Editor工具生成。

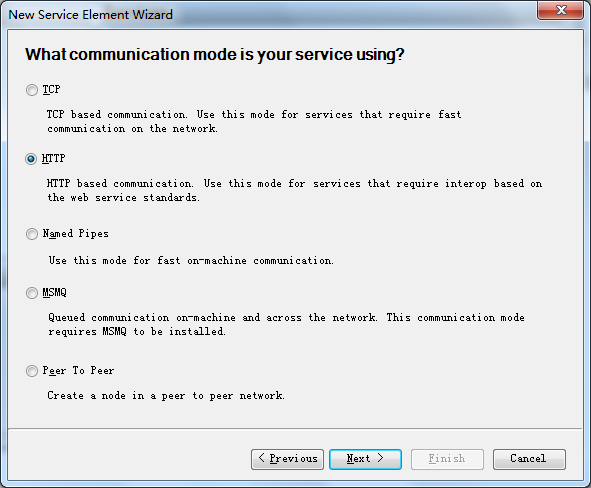
点开后，通过菜单的File-New Config新建一个描述文件，然后点击Service页面的Create a New Service，弹出如下界面：



点击Browse...按钮，选中刚才生成的WcfService1.dll，并选中其中的Service1服务，点击下一步，弹出如下界面：

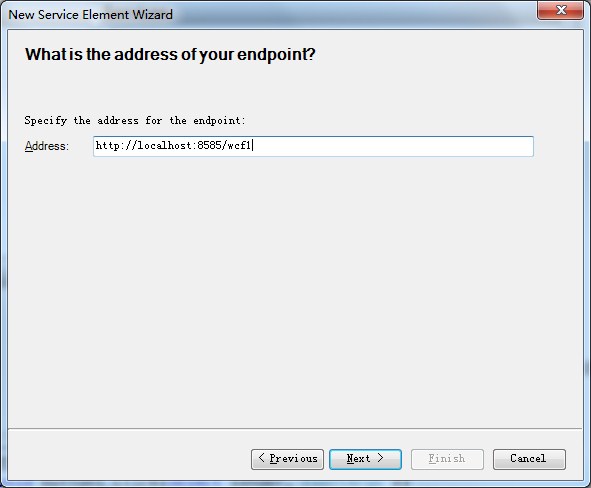


这个界面中要选择使用的契约，默认即可，点击下一步，弹出如下界面：



这个界面中要选择使用的网络通讯协议，选择最通用的HTTP协议，并在下一个界面中选择基本的Http服务（Basic WebService interoperablitity）,

在接下来的界面中，输入访问地址，其它程序可以通过这个地址来访问本WCF服务。



完成这一步之后，接下来还有一些设置，我也不清楚用途，略去，最后点击File-Save，生成一个App.config文件。这里把该文件贴上来：

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

<configuration>

    <system.serviceModel>

        <behaviors>

            <serviceBehaviors>

                <behavior name="NewBehavior0">

                    <serviceMetadata httpGetEnabled="true" httpGetUrl="http://localhost:8585/wcf1/metadata" />

                </behavior>

            </serviceBehaviors>

        </behaviors>

        <services>

            <service behaviorConfiguration="NewBehavior0" name="WcfService1.Service1">

                <endpoint address="" binding="basicHttpBinding"

                    bindingConfiguration="" name="ep1" contract="WcfService1.IService1" />

                <host>

                    <baseAddresses>

                        <add baseAddress="http://localhost:8585/wcf1" />

                    </baseAddresses>

                </host>

            </service>

        </services>

    </system.serviceModel>

</configuration>

把该App.config文件放在与WinForm工程的根目录下（与Form1.cs同一目录），还需要在VS中将该文件加入到工程中。

添加Button控件的处理函数：

privatevoidbutton1\_Click**(**objectsender**,**EventArgse**)**

**{**

    System.ServiceModel.ServiceHosthost=newSystem.ServiceModel.ServiceHost**(**typeof**(**WcfService1.Service1**));**

    host.Open**();**

    this.label1.Text="opened"**;**

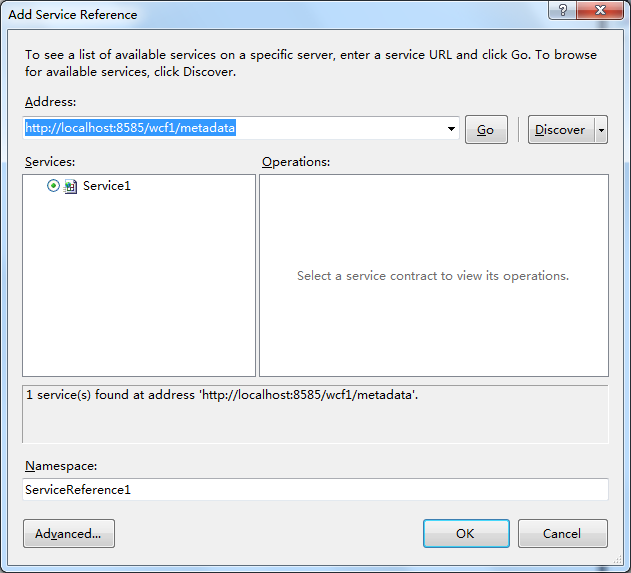
**}**

编译运行，点击界面中的Button控件，即可以将该WCF服务寄宿到该Frm中（关闭窗体后WCF服务结束）。

**注：如果运行时报这个错误（HTTP could not register URL http://+:8585/wcf1/. Your process does not have access rights to this namespace），有可能是权限不足导致，需要使用管理员权限运行。**

**五、在客户端访问并调用WCF服务**

新建一个WinFrm工程（不要关闭刚才的寄宿界面），在VS菜单中点击Project-Add Service Reference..项，输入刚才在寄宿界面中定义的地址，就可以添加对该WCF服务的引用，如下图：



在程序中使用如下代码，即可以调用WCF服务中的GetData方法：

ServiceReference1.Service1Clientaa=newServiceReference1.Service1Client**();**

MessageBox.Show**(**aa.GetData**(**2**));**