**WCF原数据**

* 所谓获取WCF的服务元数据（Metadata），归根结点，实际上就是获取服务的终结点(Endpoint)的信息，这是服务公开在外的数据信息，包括Address、Binding与Contract，也就是所谓的ABC。
* 元数据遵循Web服务描述语言(WSDL)标准，所以可被多种语言支持，除WCF的svcutil外，Java程序员也可使用诸如WSDL2Java的工具生成Java语言的客户端代理类。

如：svcutil http://localhost:8000/?wsdl

命令中http://localhost:8000/?wsdl连接返回一个XML，该XML即为元数据。

**元数据的二种方案**

**一、在<behavior>下添加子节点<serviceMetadata> 将httpGetEnabled属性设为true，通过此属性启用HTTP-GET元数据。**



**二、已终结点方式公开服务的元数据**



**使用编程的方式配置服务**

//启动

private void button1\_Click(object sender, EventArgs e)

{

host = new ServiceHost(typeof(WcfServiceLibrary2.Service1));

host.AddServiceEndpoint(typeof(WcfServiceLibrary2.IService1), new NetTcpBinding(), new Uri("net.tcp://localhost:3200"));

host.Opened += delegate { label1.Text = "服务已启动"; };

host.Open();

}

两种创建客户端代理程序的方法

1. 通过服务提供的元数据，由svcutil.exe命令自动生成
2. 通过使用ChannelFactory对象，动态创建代理类。

第一种方式，上面已经说明，下面为第二种方式的代码

1. 添加servicemodel引用，以及契约的引用。
2. 编写客户端代码

//加载

private void button1\_Click(object sender, EventArgs e)

{

//通道工厂

NetTcpBinding bind = new NetTcpBinding();

EndpointAddress address = new EndpointAddress("net.tcp://localhost:3200/emp");

ChannelFactory<WcfServiceLibrary1.IService1> factory = new ChannelFactory<WcfServiceLibrary1.IService1>(bind);

WcfServiceLibrary1.IService1 service1Channel =factory.CreateChannel(address);

dataGridView1.DataSource = service1Channel.getEmps();

//关闭通道工厂

factory.Close();

}