WCF
Windows Communication Foundation

Haitao Cheng
What Is Windows Communication Foundation?

- Windows Communication Foundation (WCF) is designed to offer a manageable approach to distributed computing, broad interoperability, and direct support for service orientation.
The most important aspects of WCF

- Unification of existing .NET Framework communication technologies.
- Interoperability with Applications Built on Other Technologies
Unification of Microsoft Distributed Computing Technologies

- ASP.NET Web services (ASMX).
- .NET Framework remoting.
- Enterprise Services.
- Microsoft Message Queuing (MSMQ).
Interoperability

- Because WCF can communicate using Web services, interoperability with other platforms that also support SOAP, such as the leading J2EE-based application servers, is straightforward.

- You can also configure and extend WCF to communicate with Web services using messages not based on SOAP, for example, simple XML formats like RSS.
Interoperability

Because WCF’s fundamental communication mechanism is SOAP-based Web services, WCF-based applications can communicate with other software running in a variety of contexts. An application built on WCF can interact with all of the following:

- WCF-based applications running in a different process on the same Windows machine.
- WCF-based applications running on another Windows machine.
- Applications built on other technologies, such as J2EE application servers, that support standard Web services. These applications can be running on Windows machines or on machines running other operating systems.
Basic Programming Steps

- Define the service contract. A service contract specifies the signature of a service, the data it exchanges, and other contractually required data.

- Implement the contract. To implement a service contract, create the class that implements the contract and specify custom behaviors that the runtime should have. For information.

- Configure the service by specifying endpoint information and other behavior information.

- Host the service in an application.

- Build a client application.
Define Service contract

```csharp
[ServiceContract]
public interface IMyService
{

[OperationContract]
Person GetPersonData();

// TODO: Add your service operations here
}
```
Define Data contract

```csharp
[DataContract]
public class Person
{
    [DataMember]
    public string Sex {get;set;}

    [DataMember]
    public string Name { get; set; }
}
```
public class MyServiceImp : IMyService
{
    public Person GetPersonData()
    {
        return new Person
        {
            Sex = "Female",
            Name = "Claire Redfield"
        };
    }
}
"ABC": address, binding, contract

<endpoint address="" binding="wsHttpBinding"
  contract="WcfService.IMyService">
  <identity>
    <dns value="localhost" />
  </identity>
</endpoint>

<endpoint address="mex" binding="mexHttpBinding"
  contract="IMetadataExchange" />
class Program
{
    static void Main(string[] args)
    {
        MyServiceClient client = new MyServiceClient();
        Person person = client.GetPersonData();
        Console.WriteLine("Person Data:{0},{1}", person.Name, person.Sex);
        Console.ReadLine();
    }
}