

Bài 1

GIỚI THIỆU TỔNG QUAN VỀ WINDOWS COMMUNICATION FOUNDATION

1. Mục lục

2.	Windows Communication Foundation (WCF) là gì?	2
3.	Tại sao sử dụng WCF?	2
4.	Kiến trúc của WCF	3
4.1	Các contracts (Các hiệp nghị).....	3
4.2	Runtime service (Dịch vụ thực thi).....	4
4.3	Message (Bản tin).....	5
4.4	Host and activation (Chứa và kích hoạt)	5
5.	Các tính năng của WCF	6
5.1	Transaction (Giao dịch).....	6
5.2	Host (Chứa)	6
5.3	Bảo mật.....	6
6.	Công cụ phát triển với WCF	7
7.	Ví dụ đầu tiên với WCF.....	7
8.	Sử dụng công cụ WCF Test Client để debug dịch vụ WCF	13
9.	Tài liệu tham khảo	15

Bài này giới thiệu tổng quan về công nghệ Windows Communication Foundation (WCF). Phần đầu nói về những công nghệ đã được triển khai trên thế giới nhằm giải quyết vấn đề liên lạc giữa máy tính với máy tính và những khó khăn thách thức trong việc triển khai các công nghệ trên. Từ đó dẫn đến sự ra đời của công nghệ nền tảng WCF, là nền tảng thống nhất cho việc xây dựng nhanh các ứng dụng cần giao tiếp. Sau đó sẽ giới thiệu cơ bản về kiến trúc của WCF, những thành phần hợp thành WCF. Phần cuối sẽ giới thiệu các công cụ cần thiết để xây dựng ứng dụng với WCF cùng với một ví dụ đơn giản với WCF để học viên dễ hình dung.

2. Windows Communication Foundation (WCF) là gì?

WCF là công nghệ nền tảng nhằm thống nhất nhiều mô hình lập trình giao tiếp được hỗ trợ trong .NET 2.0 thành một mô hình duy nhất. Vào tháng 11 năm 2005, .NET 2.0 được Microsoft phát hành trong đó có cung cấp các hàm API riêng biệt cho các liên lạc dựa trên SOAP để tối đa hoá sự làm việc giữa các nền tảng sử dụng Web Services, đồng thời .NET 2.0 còn cung cấp các API để tối ưu việc liên lạc dựa trên mã nhị phân giữa các ứng dụng chạy trên hệ thống Windows gọi là .NET Remoting, các API cho các giao dịch phân tán, và API cho liên lạc dị bộ. WCF thống nhất các API này thành một mô hình duy nhất nhằm đáp ứng mô hình lập trình hướng dịch vụ.

WCF có thể sử dụng các bản tin SOAP giữa hai tiến trình, do đó làm cho các ứng dụng dựa trên WCF có thể làm việc với các tiến trình khác thông qua việc giao tiếp sử dụng bản tin SOAP. Khi một tiến trình WCF liên lạc với một tiến trình không là WCF, các bản tin SOAP được mã hoá trên cơ sở XML, nhưng khi nó liên lạc với một tiến trình WCF khác, bản tin SOAP có thể được tối ưu hoá dựa trên mã hoá nhị phân.

3. Tại sao sử dụng WCF?

Như phần trên đã trình bày, .NET 2.0 hỗ trợ rất nhiều phương pháp liên lạc giữa các ứng dụng khác nhau nhằm vào các mục tiêu khác nhau. Các phương pháp liên lạc này khá phức tạp và phải mất nhiều thời gian để làm chủ được công nghệ. Tuy nhiên kiến thức thu được từ việc triển khai một phương pháp ít có khả năng dùng được khi làm việc với phương pháp khác.

Với việc ra đời của WCF, mọi phương pháp liên lạc trước kia đều có thể thực hiện trên WCF. Do vậy nhà phát triển chỉ cần làm chủ được công nghệ WCF là có thể xây dựng các ứng dụng một cách nhanh chóng.

WCF là một mô hình lập trình cho phép nhà phát triển xây dựng các giải pháp dịch vụ đảm bảo tính ổn định, và bảo mật và thậm chí là đảm bảo giao dịch. Nó làm đơn giản hoá việc

phát triển các ứng dụng nối kết và đưa ra cho nhà phát triển những giá trị mà có thể họ chưa nhận ra ngay, đó là cách tiếp cận phát triển hệ thống phân tán thống nhất, đơn giản, và quản lý được.

Do WCF được xây dựng trên cơ sở của .NET Framework 2.0 CLR, nó là tập các lớp cho phép các nhà phát triển xây dựng các ứng dụng hướng dịch vụ bằng môi trường lập trình quen thuộc của họ như VB.NET hay C#

4. Kiến trúc của WCF

Hình sau mô tả các lớp chủ yếu trong kiến trúc của Windows Communication Foundation (WCF)

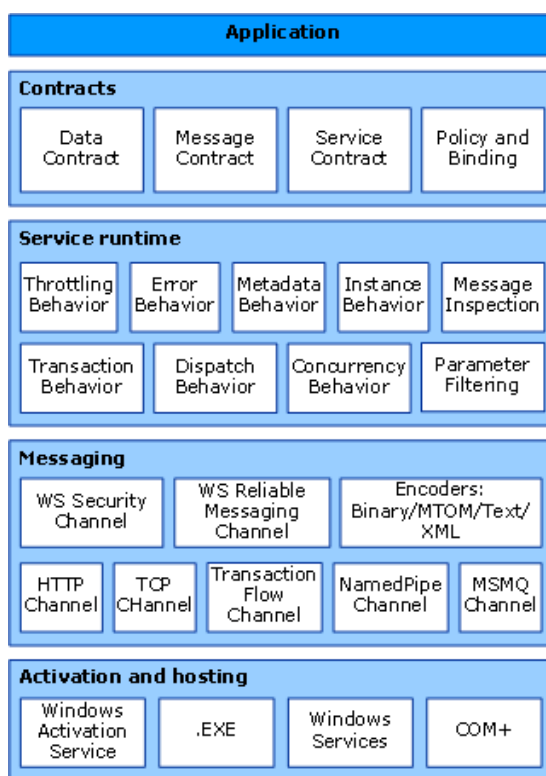


Figure 1 Kiến trúc của WCF (hình ảnh từ www.microsoft.com)

4.1 Các contracts (Các hiệp nghị)

Các contract trong WCF cũng giống như các hợp đồng/hiệp định mà bạn ký trong đời sống thật. Một hợp đồng bạn ký có thể chứa các thông tin như kiểu công việc bạn sẽ làm, và những thông tin mà bạn muốn đưa ra cho các bên khác. WCF contract cũng chứa các thông tin tương tự như vậy. Contract định nghĩa các đặc tả trong hệ thống bản tin. Thông thường có các loại contract sau:

- Contract dữ liệu mô tả các tham số cho các bản tin mà một dịch vụ có thể tạo ra hay sử dụng. Các tham số bản tin được định nghĩa bằng các tài liệu sử dụng ngôn ngữ đặc tả XML Schema (XSD), điều này cho phép các hệ thống hiểu XML có thể xử lý tài liệu dễ dàng. Các dịch vụ khi liên lạc với nhau có thể không cần đồng ý với nhau về các kiểu, nhưng cần đồng ý về contract dữ liệu, nghĩa là đồng ý về các tham số và các kiểu trả về.
- Contract bản tin định nghĩa các phần có trong bản tin sử dụng các giao thức SOAP, và nó cho phép điều khiển sâu hơn tới các phần trong bản tin khi có yêu cầu sự chính xác như vậy.
- Contract dịch vụ đặc tả chi tiết các phương thức của dịch vụ, và được phân phối như là một giao diện trong các ngôn ngữ lập trình như Visual Basic hay Visual C#. Có thể hình dung về contract dịch vụ một cách gián tiếp như sau: ‘Đây là các kiểu dữ liệu của các bản tin của tôi, đây là nơi tôi cung cấp, và đây là các giao thức mà tôi có thể liên lạc’
- Các chính sách và các kết nối (bindings) mô tả các điều kiện cần có để giao tiếp với một dịch vụ. Các chính sách sẽ bao gồm cả các yêu cầu về bảo mật và các điều kiện khác cần phải có khi kết nối với một dịch vụ.

4.2 Runtime service (Dịch vụ thực thi)

Lớp dịch vụ thực thi chứa các hành xử sẽ xảy ra trong quá trình thực hiện của dịch vụ, nghĩa là các hành xử thực thi của dịch vụ. Ta sẽ thấy một số các hành xử như sau:

- **Throttling behavior:** Điều khiển luồng nhằm quy định xem có bao nhiêu bản tin được xử lý
- **Error behavior:** Hành xử lỗi quy định những hành động khi lỗi xảy ra trong hệ thống
- **Metadata behavior:** Hành xử với các siêu dữ liệu quy định xem làm thế nào và khi nào thì các siêu dữ liệu được đưa ra bên ngoài dịch vụ
- **Instance behavior:** Hành xử thực thể quy định xem có bao nhiêu thực thể của dịch vụ đó được chạy
- **Transaction behavior:** Hành xử giao dịch cho phép việc rollback các giao dịch nếu xảy ra lỗi

- **Message inspection:** Kiểm tra bản tin đem lại cho dịch vụ khả năng kiểm tra tất cả hay một số phần của bản tin
- **Dispatch behavior:** Khi một bản tin được xử lý bởi nền tảng WCF, dịch vụ Dispatch behavior xác định xem bản tin được xử lý như thế nào.
- **Concurrency behavior:** Hành xử đồng thời xác định xem việc xử lý thế nào với việc đa luồng của mỗi dịch vụ hay mỗi thực thể của dịch vụ. Hành xử này giúp cho việc điều khiển số lượng luồng có thể truy nhập tới một thực thể của dịch vụ.
- **Parameter filtering:** Khi một bản tin được đưa tới một dịch vụ, sẽ xảy ra một số hành động dựa trên nội dung phần đầu đề của bản tin. Phần lọc tham số sẽ thực hiện lọc các đầu đề bản tin và thực hiện các hành động đặt sẵn dựa trên việc lọc đầu đề bản tin.

4.3 Message (Bản tin)

Lớp bản tin là tập hợp các kênh. Mỗi kênh là một thành phần xử lý bản tin theo một cách nào đó. Một tập các kênh thường được gọi là ngăn xếp kênh. Các kênh làm việc trên bản tin và trên đầu đề của bản tin. Lớp này khác với lớp thực thi dịch vụ chủ yếu bởi sự khác nhau trong việc xử lý nội dung bản tin.

Có hai kênh khác nhau là kênh vận chuyển (transport channel) và kênh điều khiển (control channel).

- Kênh vận chuyển phụ trách việc đọc và ghi các bản tin từ mạng (network) hoặc từ một số điểm giao dịch bên ngoài)
- Kênh điều khiển thực hiện xử lý bản tin theo giao thức, thông thường làm việc bằng cách đọc và ghi thêm các đầu đề cho bản tin.

4.4 Host and activation (Chứa và kích hoạt)

Nhìn một cách tổng thể thì một dịch vụ thực chất là một chương trình. Cũng giống như các chương trình khác, một dịch vụ cần phải chạy trong một tệp thực thi. Dịch vụ này thường được gọi là dịch vụ tự chứa.

Các dịch vụ còn có thể được chứa, hoặc chạy trong một tệp thực thi được quản lý bởi một agent bên ngoài như IIS hay Windows Activation Services (WAS). WAS cho phép WCF được kích hoạt một cách tự động khi phân phối tới một máy tính có chạy WAS.

5. Các tính năng của WCF

WCF có nhiều tính năng và sẽ được mô tả chi tiết trong toàn bộ khoá học. Mục này chỉ nhằm mục đích thảo luận một số tính năng của WCF. Danh sách các tính năng ở đây không phải là danh sách hoàn chỉnh, mà chỉ hy vọng là danh sách các tính năng “đỉnh” nhất.

5.1 Transaction (Giao dịch)

Một giao dịch là một đơn vị của công việc. Một giao dịch đảm bảo chắc chắn rằng mọi thứ diễn ra trong giao dịch thành công hay thất bại đều là kết quả tổng thể. Ví dụ, nếu một giao dịch chứa ba mục công việc cần thực hiện, trong quá trình thực hiện giao dịch, một trong số các mục đó bị thất bại, khi đó cả ba mục sẽ là thất bại. Giao dịch chỉ thành công khi cả ba mục công việc đều thành công. Giao dịch thường thấy trong các thao tác với cơ sở dữ liệu.

WCF cho phép đưa vào việc xử lý giao dịch như trên với các liên lạc. Nhà phát triển có thể nhóm các liên lạc với nhau thành các giao dịch. Ở mức doanh nghiệp, tính năng này cho phép bạn thực hiện các công việc giao dịch qua các nền tảng khác nhau.

5.2 Host (Chứa)

WCF cho phép các dịch vụ được chứa trong một số lớn các môi trường khác nhau, như Windows NT Services, Windows Forms, và ứng dụng console, cũng như ở trên IIS (Internet Information Server) và WAS (Windows Activation Services).

Chứa ứng dụng trên IIS còn có thêm các lợi điểm khác là dịch vụ có thể nhận các ưu điểm của rất nhiều tính năng có sẵn trên IIS, ví dụ IIS có thể điều khiển một cách tự động việc bắt đầu hay kết thúc một dịch vụ.

5.3 Bảo mật

Bảo mật là tính năng không thể thiếu trong WCF nói riêng và trong liên lạc nói chung. Trong WCF, tất cả mọi thứ từ các bản tin tới các client hay server đều phải xác thực và WCF có tính năng để đảm bảo rằng các bản tin không bị lẩn trong quá trình vận chuyển. WCF bao gồm việc đảm bảo tính toàn vẹn và bảo mật của bản tin.

WCF còn cho phép bạn tích hợp ứng dụng của bạn với cơ sở hạ tầng bảo mật sẵn có, bao gồm cả các chuẩn bên ngoài môi trường Windows bằng cách sử dụng các bản tin SOAP bảo mật.

6. Công cụ phát triển với WCF

Để phát triển ứng dụng với WCF ta cần các phần mềm sau:

- **.NET Framework 3.5.** Học viên có thể tải về bộ cài của .NET Framework 3.5 tại trang web của Microsoft. Địa chỉ như sau: <http://www.microsoft.com/downloads/details.aspx?familyid=ab99342f-5d1a-413d-8319-81da479ab0d7&displaylang=en>
- **Microsoft Visual Studio 2008.** Bản dùng thử có thể tải về ở đây: <http://msdn.microsoft.com/en-us/visualc/aa700831.aspx>
- Hoặc **Microsoft Visual Studio Express Editions.** Địa chỉ tải về: <http://www.microsoft.com/express/>

7. Ví dụ đầu tiên với WCF

Xây dựng ứng dụng viết chữ Hello World ra màn hình sử dụng WCF. Các bước làm việc như sau:

1. Mở Visual Studio 2008, chọn Create New Project, trong phần WCF chọn WCF Service Library như hình dưới. Đặt tên project là HelloService.

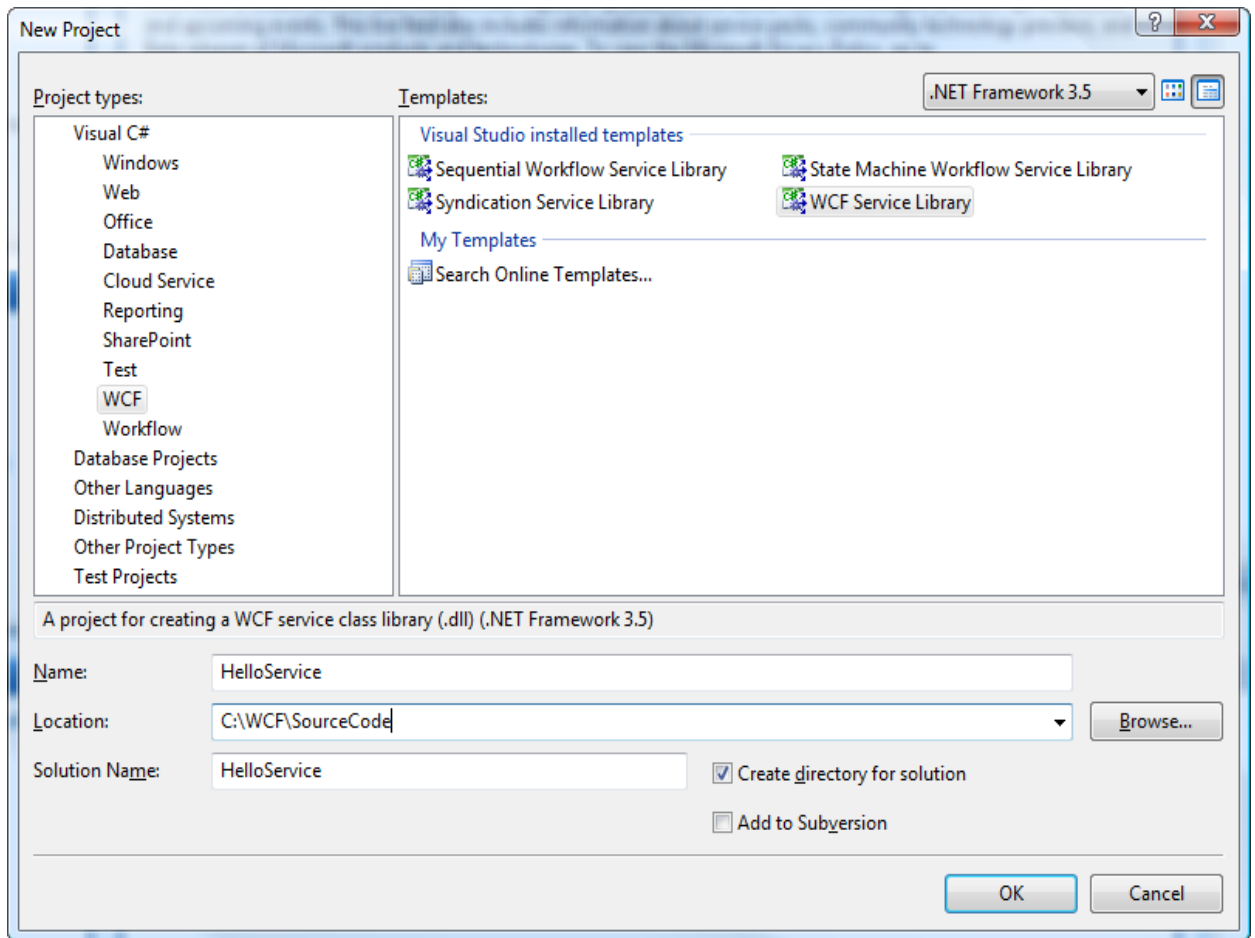


Figure 2 Tạo WCF project

2. Sau khi project được tạo ra, xóa 2 tệp IService1.cs và Service1.cs. Đồng thời làm sạch tệp app.config thành như sau:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <compilation debug="true" />
  </system.web>
  <system.serviceModel>
    <services>
    </services>
    <behaviors>
    </behaviors>
  </system.serviceModel>
</configuration>
```


3. Kích chuột phải vào project HelloService, chọn “Add → New Item”. Sau đó chọn thêm một WCF Service, đặt tên nó là HelloWorld như hình dưới. Hệ thống sẽ thêm vào 2 tệp cho bạn:

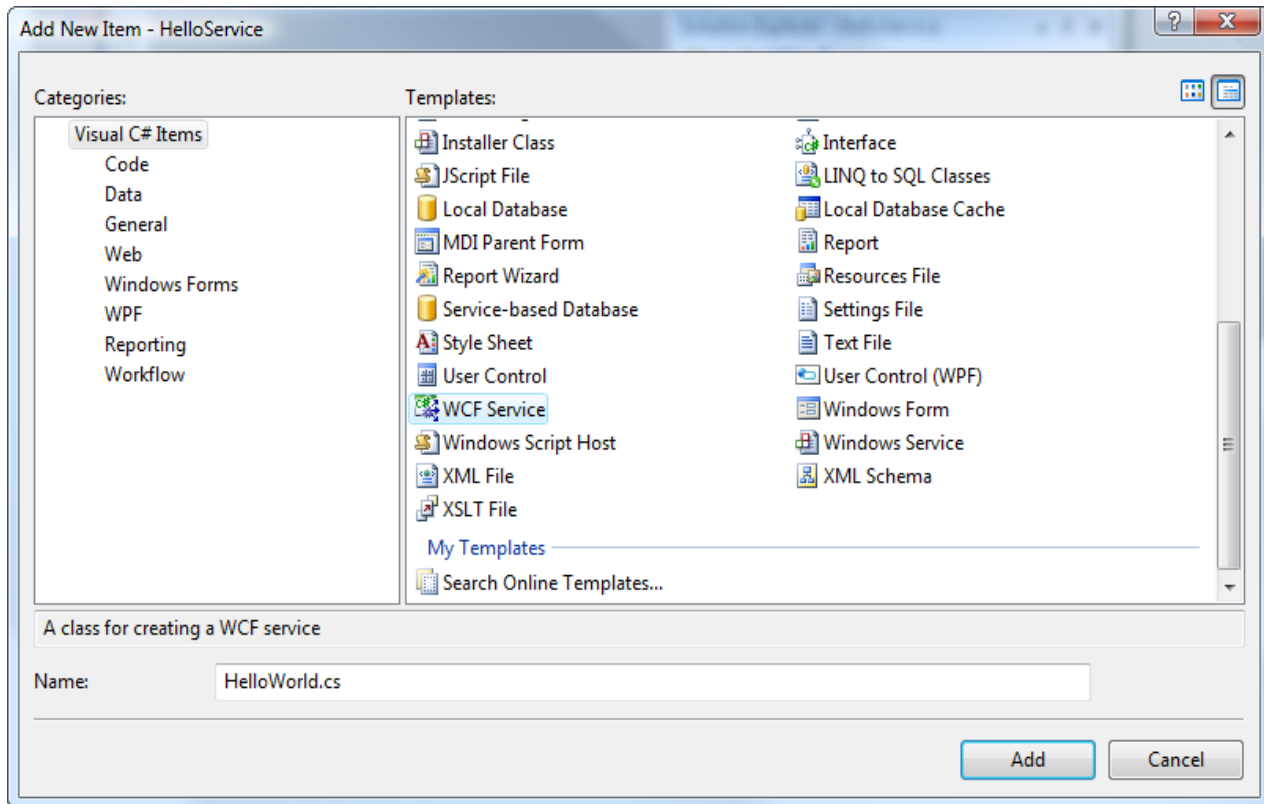


Figure 3 Thêm HelloWorld service

a. Tệp IHelloworld.cs chứa một giao diện gọi là IHelloworld, đây là contract cho dịch vụ của bạn. Thay hàm DoWork thành SayHello như sau:

```
[ServiceContract]
public interface IHelloworld
{
    [OperationContract]
    string SayHello(string inputName);
}
```

b. Tệp HelloWorld.cs, chứa lớp gọi là HelloWorld, lớp này sẽ cài đặt contract cho dịch vụ của bạn. Thay hàm DoWork thành SayHello như sau:

```
public class HelloWorld : IHelloworld
{
    #region IHelloworld Members
    public string SayHello(string inputName)
```

```

    {
        return "Chào bạn " + inputName;
    }
#endregion
}

```

Khi thực hiện cài đặt một dịch vụ WCF, thông thường bạn định nghĩa phân giao diện trước (các interface) sau đó mới thực hiện thông qua một lớp để cài đặt giao diện đó. Bạn hoàn toàn có thể làm bằng tay để viết lại phần khai báo các phương thức đã có trong giao diện. Tuy vậy, có một cách nhanh hơn giúp bạn khai báo các phương thức cũng như thuộc tính của giao diện bằng Visual Studio. Cách làm như sau:

Bước 1. Mở file chứa class cần cài đặt. Di chuyển con trỏ tới tên giao diện như hình dưới

```

namespace HelloService
{
    public class HelloWorld : IHelloWorld
    {
    }
}

```

Figure 4 Di chuyển con trỏ tới tên giao diện

Bước 2. Bấm chuột vào hình chữ nhật nhỏ dưới chữ IHelloWorld (giống smart tag trong Microsoft Word), xem hình dưới

```

namespace HelloService
{
    public class HelloWorld : IHelloWorld
    {
    }
}

```


interface HelloService.IHelloWorld

Options to implement interface (Shift+)

Figure 5 Kích hoạt SmartTag

Bước 3. Mở menu popup và chọn một trong 2 mục.

```
namespace HelloService
{
    public class HelloWorld : IHelloWorld
    {
    }
}
```

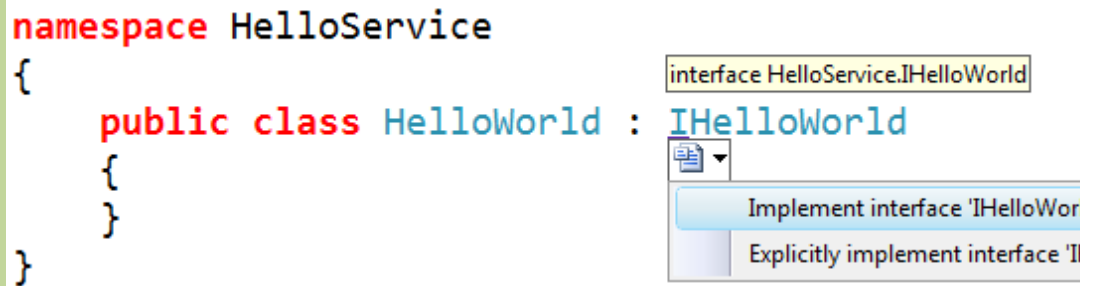


Figure 6 Chọn mục menu để cài đặt giao diện

Kết quả. Sau khi chọn một mục menu ta sẽ được kết quả như hình dưới.

```
namespace HelloService
{
    public class HelloWorld : IHelloWorld
    {
        #region IHelloWorld Members

        public string SayHello(string inputName)
        {
            throw new NotImplementedException();
        }

        #endregion
    }
}
```

Figure 7 Kết quả

Như các bạn đã thấy, sử dụng tính năng này của Visual Studio cho phép chúng ta thực hiện tạo nhanh các khai báo cần thiết để cài đặt giao diện.

4. Vậy là chúng ta đã có một dịch vụ hoàn chỉnh. Bấm F5 để chạy. Visual Studio sẽ tự động gọi chương trình WCF Test Client (là chương trình client để test dịch vụ của WCF). Xem hình dưới.

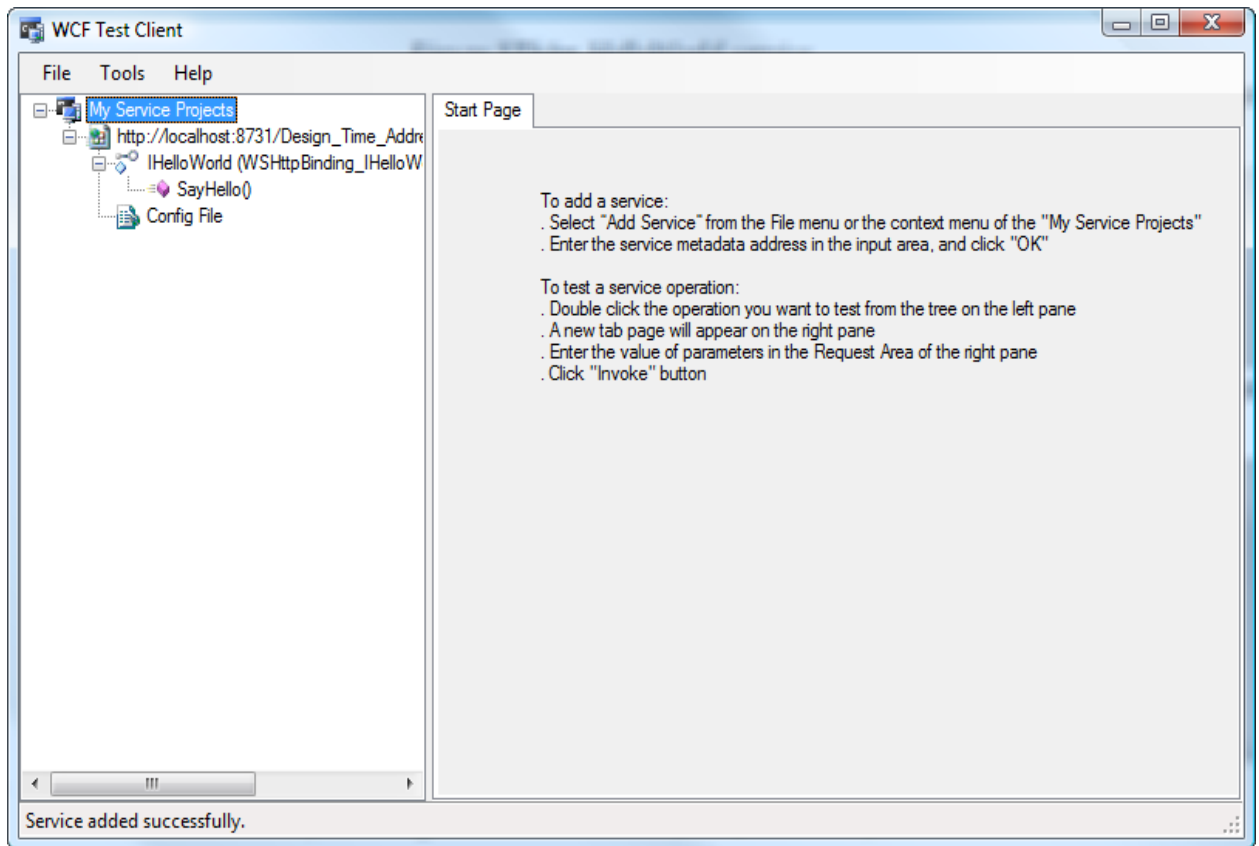


Figure 8 WCF Test Client

5. Chuyển tới phương thức SayHello, đưa vào tham số, giả sử là tên của bạn, sau đó bấm nút Invoke. Kết quả xem hình dưới.

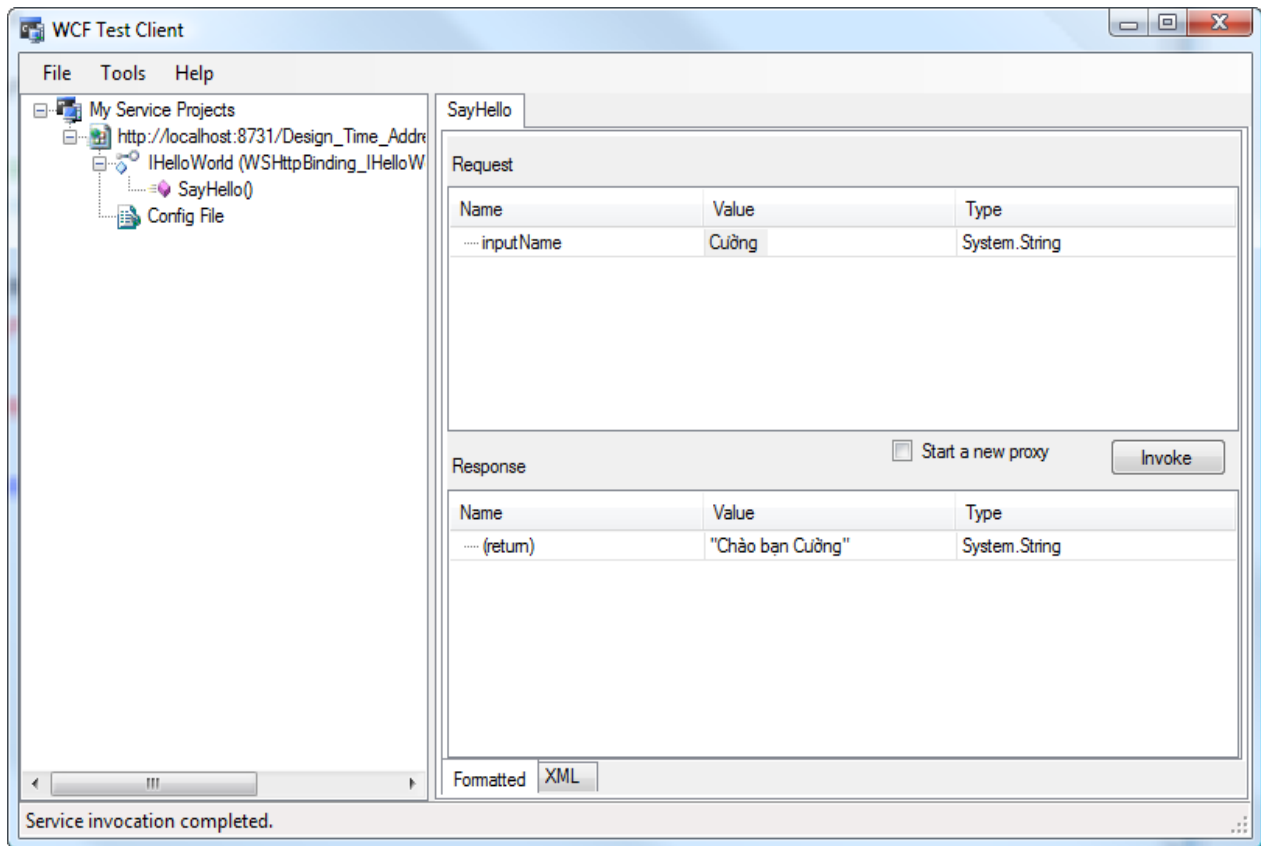


Figure 9 Kết quả thực thi dịch vụ HelloWorld

8. Sử dụng công cụ WCF Test Client để debug dịch vụ WCF

Công cụ WCF Test Client có thể được kích hoạt theo 2 cách.

- Cách thứ nhất như các bạn đã thấy ở ví dụ trên. Công cụ này được kích hoạt một cách tự động khi ta thực hiện debug một WCF Service Library.
- Cách thứ 2 là chạy trực tiếp và thêm vào các dịch vụ cần debug. Cách này thông thường hay sử dụng để debug dịch vụ khi host dịch vụ bởi IIS. Công cụ WCF Test Client có thể tìm thấy ở thư mục sau:

C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\WcfTestClient.exe

Khi kích hoạt trực tiếp WCF Test Client, sẽ thấy màn hình khởi động như sau:

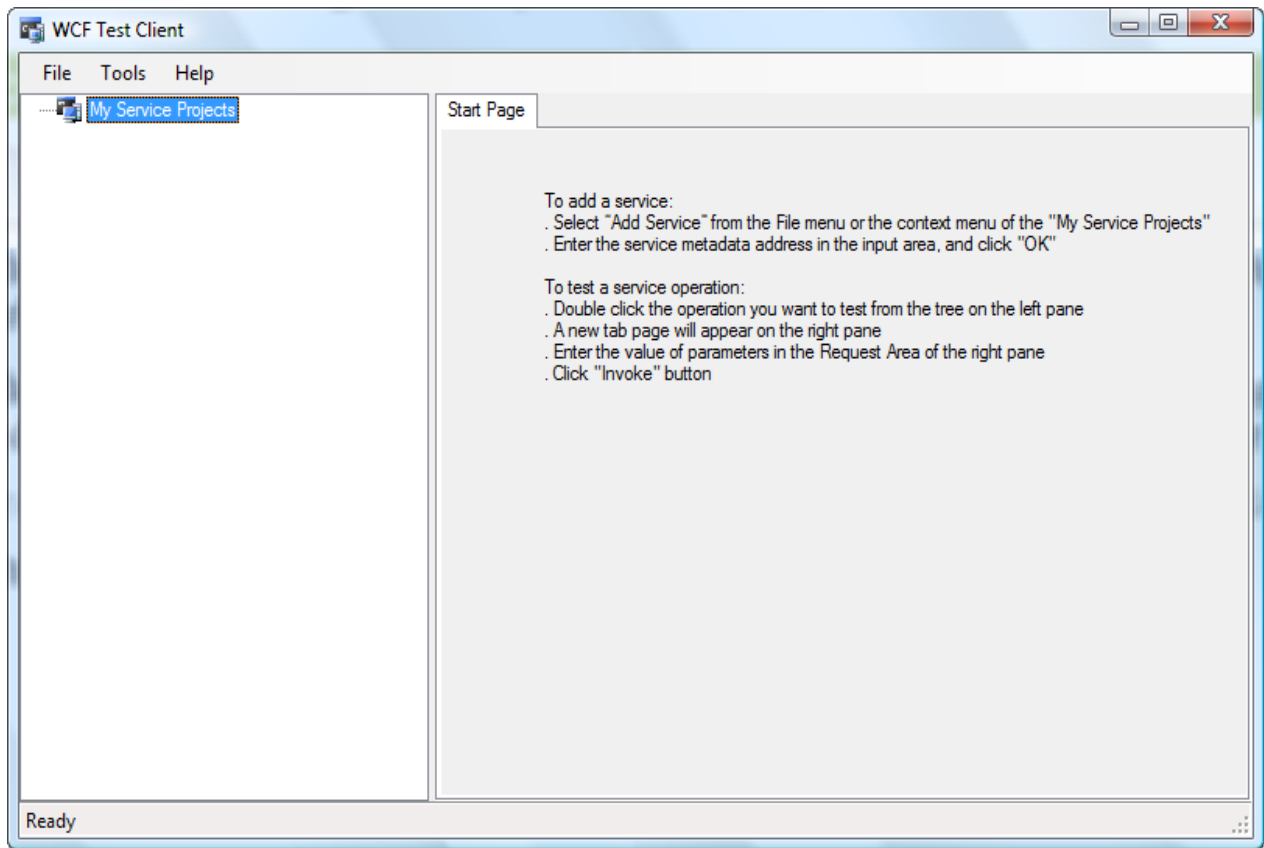


Figure 10 Màn hình khởi động của WCT Test Client

Bấm chuột phải vào My Service Projects, bạn có thể thêm vào các dịch vụ WCF để thực hiện debug

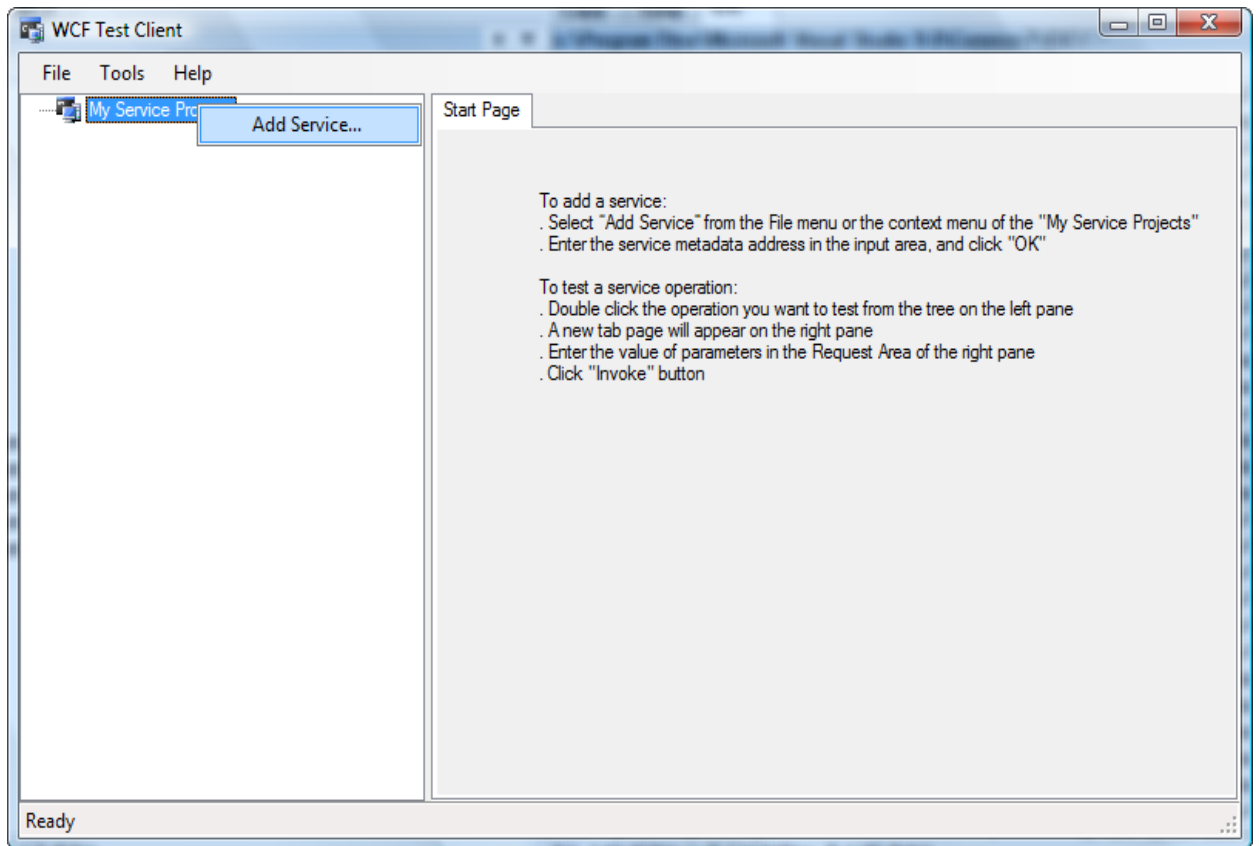


Figure 11 Thêm một dịch vụ vào WCF Test Client

Màn hình dưới đây là ví dụ thêm vào một dịch vụ được host trên IIS

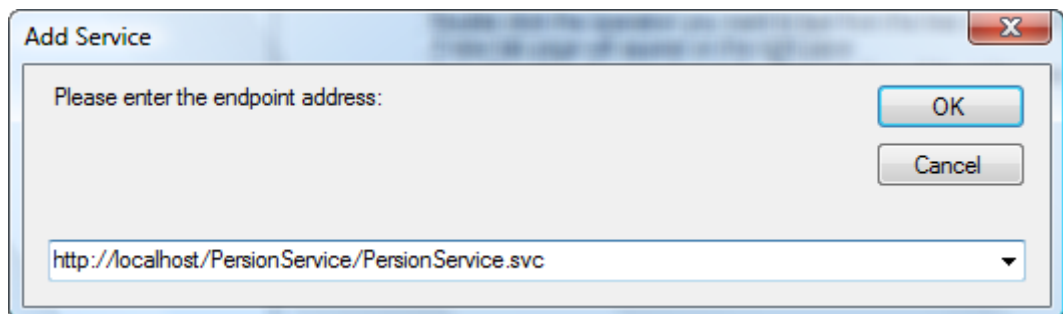


Figure 12 Thêm vào một dịch vụ được host trên IIS

9. Tài liệu tham khảo

1. Windows Communication Foundation (<http://msdn.microsoft.com/en-au/library/ms735119.aspx>)
2. What Is Windows Communication Foundation? (<http://msdn.microsoft.com/en-us/library/ms731082.aspx>)

3. Windows Communication Foundation

(http://en.wikipedia.org/wiki/Windows_Communication_Foundation)

4. Writing the WCF Hello World App ([http://blah.winsmarts.com/2008-4-](http://blah.winsmarts.com/2008-4-Writing_the_WCF_Hello_World_App.aspx)

[Writing_the_WCF_Hello_World_App.aspx](http://blah.winsmarts.com/2008-4-Writing_the_WCF_Hello_World_App.aspx))