**초보자를 위한 자습서**

이 단원에 포함된 항목에서는 WCF(Windows Communication Foundation) 프로그래밍 기능에 대해 간략하게 설명합니다. 이 항목 아래쪽에 나열된 순서대로 진행하십시오. 이 자습서를 수행하면 WCF 서비스 및 클라이언트 응용 프로그램을 만드는 데 필요한 단계에 대한 기초적인 이해를 할 수 있습니다. 서비스는 하나 이상의 끝점을 노출하는 구문이며 각 끝점은 하나 이상의 서비스 작업을 노출합니다. 서비스의 끝점은 서비스를 찾을 수 있는 주소, 클라이언트가 서비스와 통신해야 하는 정보가 포함된 바인딩 및 서비스에서 클라이언트에 제공하는 기능을 정의하는 계약을 지정합니다.

이 자습서의 항목을 순서대로 수행하면 서비스를 실행하고 클라이언트에서 서비스의 작업을 호출할 수 있습니다. 첫 번째 세 개 항목에서는 계약을 사용하여 서비스를 정의하는 방법, 서비스를 구현하는 방법 및 서비스를 코드에서 구성, 호스팅 및 실행하는 방법에 대해 설명합니다. 만들어진 서비스는 자체 호스팅되며 클라이언트와 서비스는 동일한 컴퓨터에서 실행됩니다. 서비스는 구성 대신 코드를 사용하여 구성됩니다. IIS(인터넷 정보 서비스)에서 서비스를 호스팅할 수도 있습니다. 구성 파일 내에 서비스를 구성할 수도 있습니다.

다음 세 개 항목에서는 클라이언트 프록시를 만드는 방법, 클라이언트 응용 프로그램을 구성하는 방법 및 서비스의 기능에 액세스할 수 있는 클라이언트를 만들고 사용하는 방법에 대해 설명합니다. 서비스는 클라이언트 응용 프로그램이 서비스 작업과 통신하는 데 필요한 구문을 정의하는 액세스 가능한 메타데이터를 게시합니다. WCF에서는 ServiceModel Metadata Utility Tool (Svcutil.exe)를 통해 이 게시된 메타데이터에 액세스하고 해당 메타데이터를 사용하여 서비스에 대한 클라이언트 응용 프로그램을 생성 및 구성하는 프로세스를 자동화합니다.

이 단원에 있는 모든 항목에서는 Visual Studio 2008을 개발 환경으로 사용하는 것으로 간주합니다

참고:

Windows Vista를 실행하는 경우 시작 메뉴로 이동하여 Microsoft Visual Studio 2008을 마우스 오른쪽 단추로 클릭하고 관리자 권한으로 실행을 선택하여 Visual Studio를 시작해야 합니다. Visual Studio 2008을 항상 관리자로 시작하려면 바로 가기를 만들어 마우스 오른쪽 단추로 클릭한 다음 속성을 선택하십시오. 여기서 호환성 탭을 선택하고 관리자 권한으로 이 프로그램 실행 확인란을 선택하십시오. 이 바로 가기로 Visual Studio 2008을 시작하면 프로그램이 항상 관리자로 실행됩니다.

**방법: Windows Communication Foundation 서비스 계약 정의**

이 작업은 기본 WCF(Windows Communication Foundation) 서비스와 이 서비스를 호출할 수 있는 클라이언트를 만드는 데 필요한 6가지 작업 중 첫 번째입니다.

기본 WCF 서비스를 만들 때 첫 번째 작업은 계약을 정의하는 것입니다. 계약은 서비스가 지원하는 작업을 지정합니다. 작업은 웹 서비스 메서드로 간주될 수 있습니다. C++, C# 또는 VB 인터페이스를 정의하여 계약을 만듭니다. 인터페이스의 각 메서드는 특정 서비스 작업에 해당합니다. 각 인터페이스에는 ServiceContractAttribute가 적용되어야 하고 각 작업에는 OperationContractAttribute가 적용되어야 합니다. ServiceContractAttribute를 포함하는 인터페이스 내에 있는 메서드에 OperationContractAttribute가 없으면 해당 메서드가 노출되지 않습니다.

인터페이스를 사용하여 Windows Communication Foundation 계약을 만들려면

1. 시작 메뉴에서 Visual Studio 2008를 마우스 오른쪽 단추로 클릭한 다음 관리자 권한으로 실행을 선택하여 해당 프로그램을 관리자로 엽니다.
2. 콘솔 응용 프로그램 프로젝트를 새로 만듭니다. 새 프로젝트 대화 상자에서 Visual Basic 또는 Visual C#을 선택하고 콘솔 응용 프로그램 템플릿을 선택한 다음 이름을 Service로 지정합니다. 기본 위치를 사용합니다.
3. 기본 Service 네임스페이스를 Microsoft.ServiceModel.Samples로 변경합니다.
4. 다음과 같이 System.ServiceModel.dll에 대한 참조를 프로젝트에 추가합니다.

솔루션 탐색기의 프로젝트 폴더 아래에서 참조 폴더를 마우스 오른쪽 단추로 클릭하고 참조 추가를 선택합니다.

참조 추가 대화 상자에서 찾아보기 탭을 클릭하고

c:\Windows\Microsoft.Net\Framework\v3.0\Windows Communication Foundation

을 탐색하고 System.ServiceModel.dll을 선택한 다음 확인을 클릭합니다.

참고:

Csc.exe 또는 Vbc.exe와 같은 명령 줄 컴파일러를 사용하는 경우 어셈블리의 경로도 제공해야 합니다. 기본적으로 예를 들어 Windows Vista을 실행하는 컴퓨터에서 경로는

Windows\Microsoft.NET\Framework\v3.0\Windows Communication Foundation

입니다.

1. System.ServiceModel 네임스페이스에 대한 using 문(Visual Basic의 경우 Imports)을 추가합니다.
2. ICalculator라고 하는 새 인터페이스를 정의하고 ServiceContractAttribute 특성을 Namespace 값이 "http://Microsoft.ServiceModel.Sample"인 인터페이스에 적용합니다. 네임스페이스를 명시적으로 지정하면 기본 네임스페이스 값이 계약 이름에 추가되지 않기 때문에 되도록이면 명시적으로 지정하는 것이 좋습니다.

참고:

특성을 사용하여 인터페이스 또는 클래스에 주석을 달 때 특성 이름에서 "특성" 부분을 삭제할 수 있습니다. 따라서 ServiceContractAttribute가 [ServiceContract]가 됩니다.

1. ICalculator 계약이 인터페이스 내에서 노출(더하기, 빼기, 곱하기, 나누기)하는 각 작업에 대한 메서드를 선언하고 공용 WCF 계약의 일부로 노출할 각 메서드에 OperationContractAttribute 특성을 적용합니다.

|  |
| --- |
| // Define a service contract.  [ServiceContract(Namespace="http://Microsoft.ServiceModel.Sample")]  public interface ICalculator  {  [OperationContract]  double Add(double n1, double n2);  [OperationContract]  double Subtract(double n1, double n2);  [OperationContract]  double Multiply(double n1, double n2);  [OperationContract]  double Divide(double n1, double n2);  } |

**방법: Windows Communication Foundation 서비스 계약 구현**

이 작업은 기본 WCF(Windows Communication Foundation) 서비스와 이 서비스를 호출할 수 있는 클라이언트를 만드는 데 필요한 6가지 작업 중 두 번째입니다.

WCF 서비스를 만들려면 먼저 인터페이스를 사용하여 정의된 계약을 만들어야 합니다. 이 예제에서처럼 다음 단계에서는 인터페이스를 구현합니다. 여기에는 사용자 정의 ICalculator 인터페이스를 구현하는 CalculatorService 클래스의 생성이 포함됩니다. 이 작업에 사용되는 코드는 이 절차 다음에 나오는 예제에 제공되어 있습니다.

WCF 서비스 계약을 구현하려면

1. ICalculator 인터페이스를 정의한 동일한 파일에서 CalculatorService라는 새 클래스를 만듭니다.

CalculatorService은 ICalculator 인터페이스를 구현합니다.

1. CalculatorService 클래스 내에서 ICalculator 인터페이스에 정의한 각 메서드를 구현합니다.

|  |
| --- |
| // Service class that implements the service contract.  public class CalculatorService : ICalculator  {  public double Add(double n1, double n2)  {  return n1 + n2;  }  public double Subtract(double n1, double n2)  {  return n1 - n2;  }  public double Multiply(double n1, double n2)  {  return n1 \* n2;  }  public double Divide(double n1, double n2)  {  return n1 / n2;  }  } |

**방법: 기본 Windows Communication Foundation 서비스 호스트 및 실행**

이 작업은 기본 WCF(Windows Communication Foundation) 서비스와 이 서비스를 호출할 수 있는 클라이언트를 만드는 데 필요한 6가지 작업 중 세 번째입니다.

이 항목은 기본 WCF(Windows Communication Foundation) 서비스를 실행하는 방법을 설명합니다. 이 절차는 다음 단계로 구성됩니다.

* 서비스에 대한 기본 주소를 만듭니다.
* 서비스에 대한 서비스 호스트를 만듭니다.
* 메타데이터 교환을 사용하도록 설정합니다.
* 서비스 호스트를 엽니다.

이 작업에서 작성된 전체 코드 목록은 이 절차 다음에 나오는 다음 예제에 제공되어 있습니다. Program 클래스에 정의된 Main() 메서드에 다음 코드를 추가합니다. 이 클래스는 Service 솔루션을 만들 때 생성되었습니다.

서비스에 대한 기본 주소를 구성하려면

1. 서비스의 기본 주소에 대한 Uri 인스턴스를 만듭니다. 이 URI는 HTTP 구성표, 사용자의 로컬 컴퓨터, 포트 번호 8000 및 서비스 계약의 서비스 네임스페이스에 대해 지정된 서비스에 대한 ServiceModelSample/Service 경로를 지정합니다.

서비스를 호스팅하려면

1. System.ServiceModel.Description 네임스페이스를 가져옵니다. 이 코드 줄은 using 또는 imports 문의 나머지 부분과 함께 Program.cs/Program.vb 파일의 맨 위에 배치되어야 합니다.
2. 서비스를 호스팅할 새 ServiceHost 인스턴스를 만듭니다. 서비스 계약을 구현하는 형식 및 기본 주소를 지정해야 합니다. 이 샘플에서는 기본 주소가 http://localhost:8000/ServiceModelSample/Service이며 CalculatorService는 서비스 계약을 구현하는 형식입니다.
3. CommunicationException을 catch하는 try-catch 문을 추가하고 다음 세 단계에서 try 블록에 코드를 추가합니다. catch 절은 오류 메시지를 표시한 다음 selfHost.Abort()를 호출해야 합니다.
4. 서비스를 노출하는 끝점을 추가합니다. 이렇게 하려면 끝점이 노출하는 계약, 바인딩 및 끝점에 대한 주소를 지정해야 합니다. 이 샘플에서는 ICalculator를 계약으로, WSHttpBinding을 바인딩으로, CalculatorService를 주소로 지정합니다. 여기에서는 끝점 주소가 상대 주소입니다. 끝점에 대한 전체 주소는 기본 주소와 끝점 주소의 조합입니다. 이 경우 전체 주소는

http://localhost:8000/ServiceModelSample/Service/CalculatorService입니다.

1. 메타데이터 교환을 사용하도록 설정합니다. 이렇게 하려면 서비스 메타데이터 동작을 추가합니다. 먼저 ServiceMetadataBehavior 인스턴스를 만들고 HttpGetEnabled 속성을 true로 설정한 다음 새 동작을 서비스에 추가합니다. 메타데이터를 게시하는 경우 보안 문제에 대한 자세한 내용은 메타데이터 관련 보안 고려 사항을 참조하십시오.
2. ServiceHost를 열고 들어오는 메시지를 기다립니다. 사용자가 Enter 키를 누르면 ServiceHost가 닫힙니다.

서비스가 작동하는지 확인하려면

1. Visual Studio 내에서 service.exe를 실행합니다. Windows Vista에서 실행하는 경우 이 서비스는 관리자 권한으로 실행해야 합니다. Visual Studio가 관리자 권한으로 실행되었기 때문에 service.exe도 관리자 권한으로 실행됩니다. 또한 관리자 권한으로 서비스를 실행하는 새 명령 프롬프트를 시작하고 해당 명령 프롬프트 내에서 service.exe를 실행할 수 있습니다.
2. Internet Explorer를 열고 서비스의 디버그 페이지인 http://localhost:8000/ServiceModelSamples/Service를 찾습니다.

|  |
| --- |
| static void Main(string[] args)  {  Uri uri = new Uri("http://localhost:8000/ServiceModelSample/Service");  using (ServiceHost serviceHost =  new ServiceHost(typeof(CalculatorService), uri))  {  serviceHost.AddServiceEndpoint(typeof(ICalculator), new WSHttpBinding(),  "http://localhost:8000/ServiceModelSample/Service/CalculatorService");  ServiceMetadataBehavior metadata = new ServiceMetadataBehavior();  metadata.HttpGetEnabled = true;  metadata.HttpGetUrl = uri;  serviceHost.Description.Behaviors.Add(metadata);  serviceHost.Open();  Console.WriteLine("The service is ready.");  Console.WriteLine("Press <ENTER> to terminate service.");  Console.WriteLine();  Console.ReadLine();  }  } |

**방법: Windows Communication Foundation 클라이언트 만들기**

이 작업은 기본 WCF(Windows Communication Foundation) 서비스와 이 서비스를 호출할 수 있는 클라이언트를 만드는 데 필요한 6가지 작업 중 네 번째입니다.

이 항목에서는 WCF 서비스에서 메타데이터를 검색하고 이를 사용하여 서비스에 액세스할 수 있는 WCF 프록시를 만드는 방법에 대해 설명합니다. 이 작업은 WCF에서 제공하는 ServiceModel Metadata Utility Tool (Svcutil.exe)를 사용하여 수행합니다. 이 도구는 서비스로부터 메타데이터를 가져와 사용자가 선택한 언어를 사용하여 프록시에 대한 관리되는 소스 코드 파일을 생성합니다. 클라이언트를 만드는 것 이외에 해당 도구는 클라이언트에 대한 구성 파일도 만듭니다. 이 구성 파일을 사용하여 클라이언트 응용 프로그램에서 해당 끝점 중 하나의 서비스에 연결할 수 있습니다.

클라이언트 응용 프로그램은 생성된 프록시를 사용하여 WCF 클라이언트 개체를 만듭니다. 이 절차에 대해서는 방법: Windows Communication Foundation 클라이언트 사용에서 설명합니다.

이 작업에서 생성된 클라이언트 코드는 다음 예제 절차에서 제공됩니다.

Windows Communication Foundation 클라이언트를 만들려면

1. Visual Studio 2008에서 다음 단계를 수행하여 클라이언트의 현재 솔루션에서 새 프로젝트를 만듭니다.

서비스가 포함된 동일한 솔루션 내의 오른쪽 위에 있는 솔루션 탐색기에서 프로젝트가 아니라 현재 솔루션을 마우스 오른쪽 단추로 클릭하고 추가를 선택한 다음 새 프로젝트를 선택합니다.

새 프로젝트 추가 대화 상자에서 Visual Basic 또는 Visual C#을 선택하고 콘솔 응용 프로그램 템플릿을 선택한 다음 이름을 Client로 지정합니다. 기본 위치를 사용합니다.

확인을 클릭합니다.

1. 다음과 같이 System.ServiceModel.dll에 대한 참조를 프로젝트에 추가합니다.

솔루션 탐색기의 클라이언트 프로젝트 아래에서 참조 폴더를 마우스 오른쪽 단추로 클릭하고 참조 추가를 선택합니다.

최근에 사용한 파일 탭을 선택하고 목록 상자에서 System.ServiceModel.dll을 선택한 다음 확인을 클릭합니다. 이 자습서의 첫 단계에서 이 어셈블리에 대한 참조를 이미 추가했으므로 최근 사용한 파일 탭에 이 어셈블리가 나열되어 있을 것입니다. 최근 사용한 파일 탭에서 보이지 않으면 찾아보기 탭을 클릭하고 C:\Windows\Microsoft.NET\Framework\v3.0\Windows Communication Foundation으로 이동하여 어셈블리를 선택합니다.

참고:

Csc.exe 또는 Vbc.exe와 같은 명령줄 컴파일러를 사용하는 경우 어셈블리의 경로도 제공해야 합니다. 기본적으로 예를 들어 Windows Vista를 실행하는 컴퓨터에서 경로는

Windows\Microsoft.NET\Framework\v3.0\Windows Communication Foundation입니다.

1. 생성된 Program.cs 또는 Program.vb 파일에 있는 System.ServiceModel 네임스페이스에 using 문(Visual Basic에서는 Imports)을 추가합니다.

using System.ServiceModel;

1. 이전 단계에서 만든 서비스를 시작합니다.
2. 적절한 스위치로 Service Model Metadata Utility Tool (SvcUtil.exe)를 실행하고 다음 단계를 수행하여 클라이언트 코드 및 구성 파일을 만듭니다.

시작 메뉴의 Microsoft Windows SDK 항목 아래에서 CMD 셸을 선택하여 Windows SDK 콘솔 세션을 시작합니다.

클라이언트 노드를 배치할 디렉터리로 이동합니다. 기본값을 사용하여 클라이언트 프로젝트를 만든 경우 디렉터리는 C:\Users\<사용자 이름>\Documents\Visual Studio 2005\Projects\Service\Client입니다.

적절한 스위치로 명령줄 도구 Service Model Metadata Utility Tool (SvcUtil.exe)를 사용하여 클라이언트 코드를 만듭니다. 다음 예제에서는 서비스에 대해 코드 파일 및 구성 파일을 생성합니다.

svcutil.exe /language:cs /out:generatedProxy.cs

/config:app.config <http://localhost:8000/ServiceModelSample/service>

기본적으로 클라이언트 프록시 코드 이름은 서비스 이름에 따라 지정됩니다. 예를 들어 이 경우에는 CalculatorService.cs 또는 CalculatorService.vb가 되며, 여기서 확장명은 프로그래밍 언어에 해당합니다( .vb는 Visual Basic, .cs는 C#). /out 스위치는 클라이언트 프록시 파일의 이름을 generatedProxy.cs로 변경하고 /config 스위치는 클라이언트 구성 파일의 이름을 기본 output.config에서 app.config로 변경합니다. 이 두 파일은 C:\Users\<사용자 이름>\Documents\Visual Studio 2005\Projects\Service\Client 디렉터리에 생성됩니다.

1. 생성된 프록시를 Visual Studio의 클라이언트 프로젝트에 추가하려면 솔루션 탐색기에서 클라이언트 프로젝트를 마우스 오른쪽 단추로 클릭하고 추가를 선택한 다음 기존 항목을 선택합니다. 이전 단계에서 생성된 generatedProxy.cs 파일을 선택합니다.

**방법: 기본 Windows Communication Foundation 클라이언트 구성**

이 작업은 기본 WCF(Windows Communication Foundation) 서비스와 이 서비스를 호출할 수 있는 클라이언트를 만드는 데 필요한 6가지 작업 중 다섯 번째입니다.

이 항목에서는 Service Model Metadata Utility (Svcutil.exe)를 사용하여 생성된 클라이언트 구성 파일을 클라이언트 프로젝트에 추가하고 클라이언트 구성 요소의 내용을 자세히 설명합니다. 클라이언트를 구성하려면 클라이언트에서 서비스에 액세스하는 데 사용하는 끝점을 지정해야 합니다. 끝점에는 주소, 바인딩 및 계약이 포함되어 있으며 이러한 각 항목은 클라이언트 구성 프로세스에서 지정되어야 합니다.

클라이언트에 대해 생성된 구성 파일의 컨텐츠는 해당 절차 이후의 예제에서 제공됩니다.

Windows Communication Foundation 클라이언트를 구성하려면

1. 이전 방법: Windows Communication Foundation 클라이언트 만들기 절차에서 생성된 App.config 구성 파일을 Visual Studio의 클라이언트 프로젝트에 추가합니다. 솔루션 탐색기에서 클라이언트 프로젝트를 마우스 오른쪽 단추로 클릭하고 추가를 선택한 다음 기존 항목을 선택합니다. 그런 다음 C:\Users\<사용자 이름>\Documents\Visual Studio 2005\Projects\Service\Client 디렉터리에서 App.config 구성 파일을 선택합니다. Svcutil.exe 도구를 사용하여 생성할 때 /config:app.config 스위치를 사용했기 때문에 이 구성 파일의 이름은 App.config 파일로 지정됩니다. 확인을 클릭합니다. 기본적으로 기존 항목 추가 대화 상자에서는 .config 확장명을 가진 파일을 모두 필터링합니다. 이러한 파일을 표시하려면 기존 항목 추가 대화 상자의 오른쪽 아래에 있는 드롭다운 목록 상자에서 모든 파일(\*.\*)을 선택합니다.
2. 생성된 구성 파일을 엽니다. Svcutil.exe는 바인딩에 대한 모든 설정 값을 생성합니다. 다음 예제에서는 생성된 구성 파일을 보여 줍니다. <system.serviceModel> 섹션 아래에서 <endpoint> 요소를 찾습니다. 다음 구성 파일은 생성된 파일의 단순화된 버전입니다.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <configuration>  <system.serviceModel>  <bindings>  <wsHttpBinding>  <binding name="WSHttpBinding\_ICalculator">  </binding>  </wsHttpBinding>  </bindings>  <client>  <endpoint  address="http://localhost:8000/ServiceModelSamples/Service/CalculatorService"  binding="wsHttpBinding"  bindingConfiguration="WSHttpBinding\_ICalculator"  contract="Microsoft.ServiceModel.Samples.ICalculator"  name="WSHttpBinding\_ICalculator">  </endpoint>  </client>  </system.serviceModel>  </configuration> |

이 예제에서는 다음 주소에 위치한 서비스에 액세스하도록 클라이언트가 사용하는 끝점을 구성합니다. http://localhost:8000/ServiceModelSample/service

끝점 요소는 Microsoft.ServiceModel.Samples.ICalculator 계약이 시스템 제공 WsHttpBinding을 통해 구성된 통신에 사용되도록 지정합니다. 이 바인딩은 HTTP를 전송, 상호 운용 가능한 보안 및 기타 구성 세부 사항으로 지정합니다.

**방법: Windows Communication Foundation 클라이언트 사용**

이 작업은 기본 WCF(Windows Communication Foundation) 서비스와 이 서비스를 호출할 수 있는 클라이언트를 만드는 데 필요한 6가지 작업 중 여섯 번째입니다.

WCF(Windows Communication Foundation) 프록시를 만들고 구성한 후에는 클라이언트 인스턴스를 만들고 클라이언트 응용 프로그램을 컴파일하여 WCF 서비스와 통신하는 데 사용할 수 있습니다. 이 항목에서는 WCF 클라이언트를 만들고 사용하는 절차에 대해 설명합니다. 이 절차에서는 WCF 클라이언트 만들기, 생성된 프록시에서 서비스 작업 호출, 작업 호출 완료 후 클라이언트 닫기의 세 가지 작업에 대해 설명합니다.

또한 이 절차에서 설명하는 코드는 절차 다음에 나오는 예제에 제공됩니다. 이 작업의 코드는 클라이언트 프로젝트에서 생성된 Program 클래스의 Main() 메서드에 배치해야 합니다.

Windows Communication Foundation 클라이언트를 사용하려면

1. 호출할 서비스의 기본 주소에 대한 EndpointAddress 인스턴스를 만든 다음 WCF Client 개체를 만듭니다.
2. Client 내에서 클라이언트 작업을 호출합니다.
3. WCF 클라이언트에서 Close를 호출하고 사용자가 Enter 키를 눌러 응용 프로그램을 종료할 때까지 기다립니다.

|  |
| --- |
| static void Main(string[] args)  {  Uri uri = new Uri("http://localhost:8000/ServiceModelSample/Service");  EndpointAddress address = new EndpointAddress(uri);  //app.config  CalculatorClient client = new CalculatorClient("WSHttpBinding\_ICalculator");  double value1 = 100.0D;  double value2 = 15.99D;  double result = client.Add(value1, value2);  Console.WriteLine(result);  } |