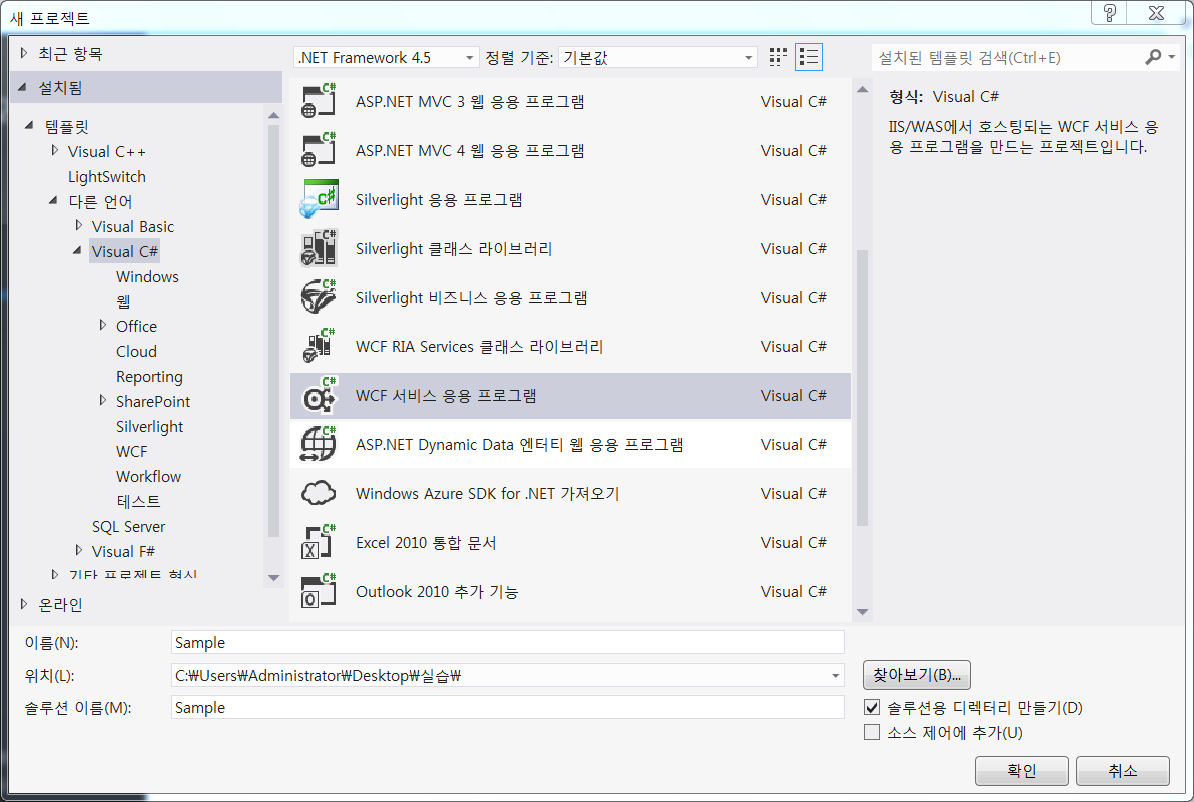
[ Callback 기능을 갖는 WCF 구성]

1. 새 프로젝트 생성

솔루션 : WCF 서비스 응용프로그램

프로젝트 명 : Sample



1. 인터페이스 구현

파일명 : ICal.cs(기존 생성된 파일명 수정)

|  |
| --- |
| namespace Sample  {  [ServiceContract]  public interface ICal  {  [OperationContract(IsOneWay=true)]  void Add(int num1, int num2);  [OperationContract(IsOneWay = true)]  void Sub(int num1, int num2);  [OperationContract(IsOneWay = true)]  void Mul(int num1, int num2);  [OperationContract(IsOneWay = true)]  void Div(int num1, int num2);  }  } |

1. 구현 객체 구현

파일명 : service1.svc

|  |
| --- |
| namespace Sample  {  public class CCal : ICal  {  private float m\_result;  public void Add(int num1, int num2) { m\_result = num1 + num2; }  public void Sub(int num1, int num2) { m\_result = num1 + num2; }  public void Mul(int num1, int num2) { m\_result = num1 \* num2; }  public void Div(int num1, int num2) { m\_result = (float)num1 / num2; }  }  } |

1. Callback 처리

4.1 Callback 인터페이스 등록

파일명 : ICal.cs

|  |
| --- |
| public interface ICallback  {  [OperationContract(IsOneWay= true)]  void Result(float result);  } |

4.2 이중 계약에 두 개의 인터페이스 연결 : 기존 인터페이스의 callbackcontrack 속성을 추가

파일명 : ICal.cs

|  |
| --- |
| [ServiceContract(CallbackContract=typeof(ICallback))]  public interface ICal  {  [OperationContract(IsOneWay=true)]  void Add(int num1, int num2);  … |

4.3 구현 객체 수정 : callback 인터페이스 추가

파일명 : service1.svc

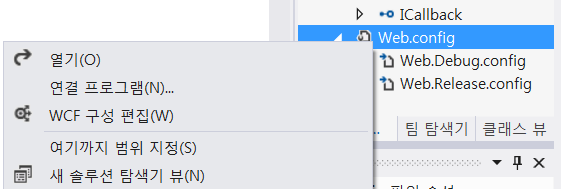
|  |
| --- |
| public class CCal : ICal  {  private float m\_result;  private ICallback callback = null; //🡸 추가  public CCal()  {  callback = OperationContext.Current.GetCallbackChannel<ICallback>(); //🡸 추가  } |

4.4 필요시 callback 호출 : 기존코드에 추가

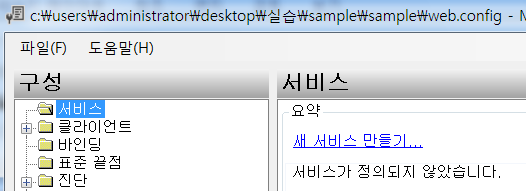
파일명 : service1.svc

|  |
| --- |
| public void Add(int num1, int num2) { m\_result = num1 + num2; callback.Result(m\_result); }  public void Sub(int num1, int num2) { m\_result = num1 + num2; callback.Result(m\_result); }  public void Mul(int num1, int num2) { m\_result = num1 \* num2; callback.Result(m\_result); }  public void Div(int num1, int num2) { m\_result = (float)num1 / num2; callback.Result(m\_result); } |

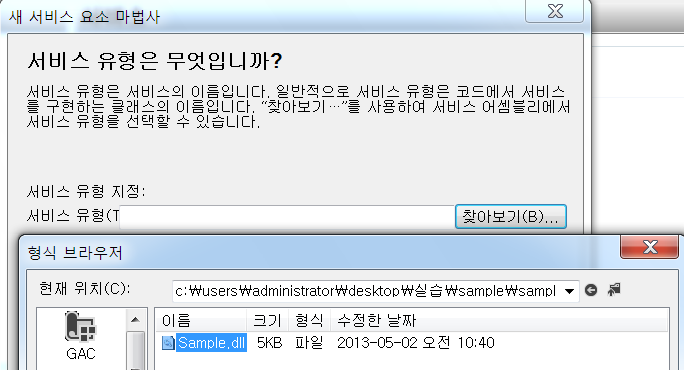
1. Binding 설정
   1. 구성편집 도구 이용

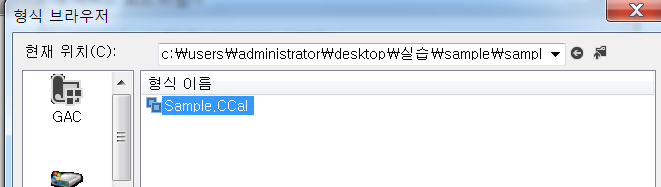


* 1. 서비스 >> 새 서비스 만들기

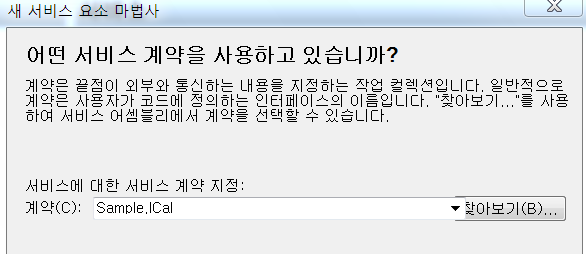


* 1. 서비스 유형 선택

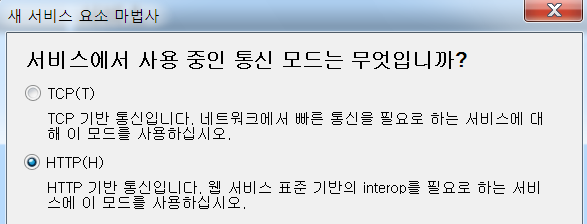




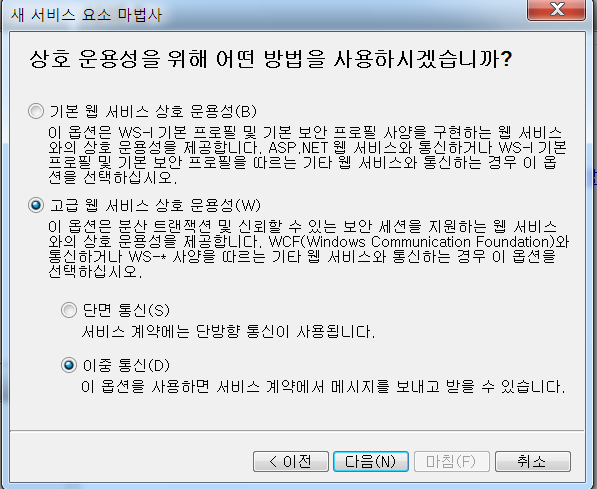
5.4 서비스 계약



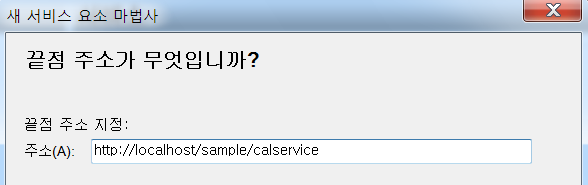
5.5 통신 모드 선택



5.6 상호 운영성 선택



5.7 주소 선택 : 주소 미지정 (IIS에서 자동 지정)



5.8 생성된 config 파일

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <configuration>  <appSettings>  <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />  </appSettings>  <system.web>  <compilation debug="true" targetFramework="4.5" />  <httpRuntime targetFramework="4.5"/>  </system.web>  <system.serviceModel>  <bindings>  </bindings>  <services>  <service name="Sample.CCal">  <endpoint address="" binding="wsDualHttpBinding"  bindingConfiguration="" contract="Sample.ICal" />  </service>  </services>    </system.serviceModel>    </configuration> |

5.4 실행 확인 및 게시

자동 게시

5.5 IIS 등록 및 확인

[ Callback 기능 연동 클라이언트]