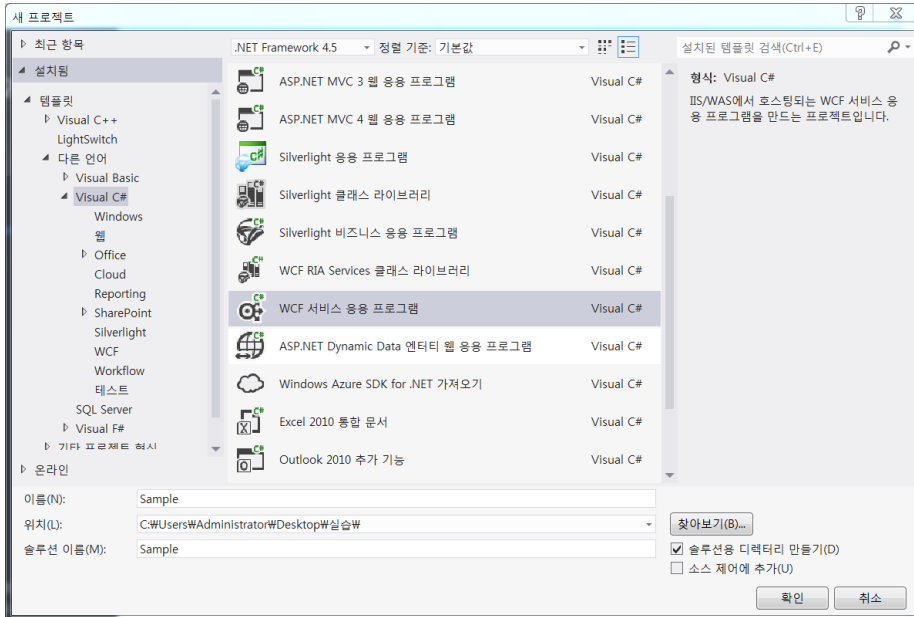


## [ Callback 기능을 갖는 WCF 구성]

### 1. 새 프로젝트 생성

솔루션 : WCF 서비스 응용프로그램

프로젝트 명 : Sample



### 2. 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);
    }
}
```

### 3. 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; }
    }
}
```

#### 4. 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 callbackcontract 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>();    //←
    }
}
```

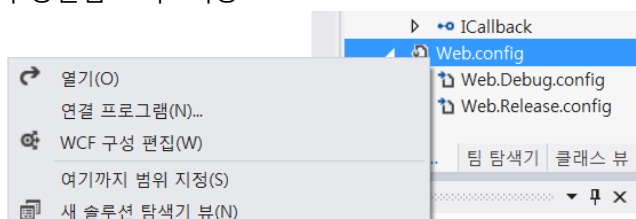
##### 4.4 필요시 callback 호출 : 기존코드에 추가

파일명 : service1.svc

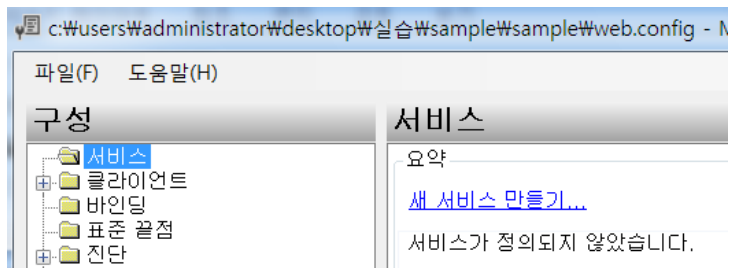
```
public void Add(int num1, int num2) { m_result = num1 + num2; callback.Result(m_result); }
public void Sub(int num1, int num2) { m_result = num1 + num2; callback.Result(m_result); }
public void Mul(int num1, int num2) { m_result = num1 * num2; callback.Result(m_result); }
public void Div(int num1, int num2) { m_result = (float)num1 / num2;
callback.Result(m_result); }
```

#### 5. Binding 설정

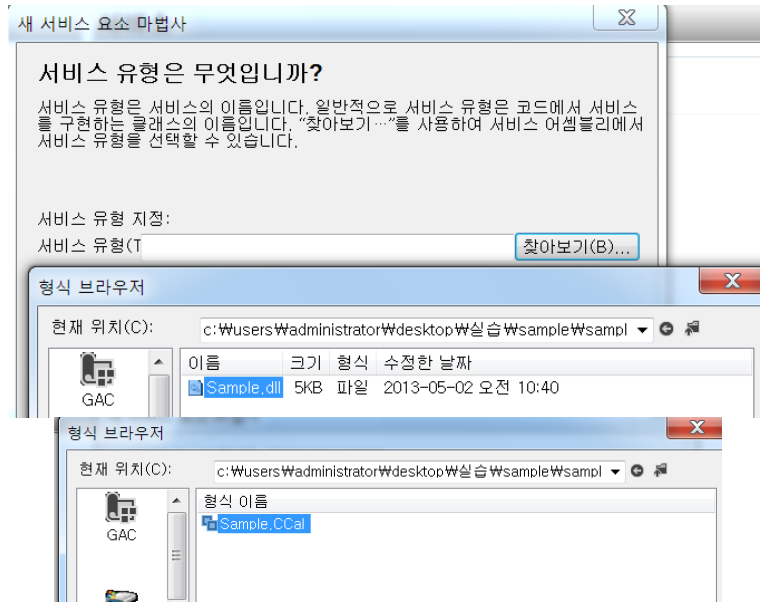
##### 5.1. 구성편집 도구 이용



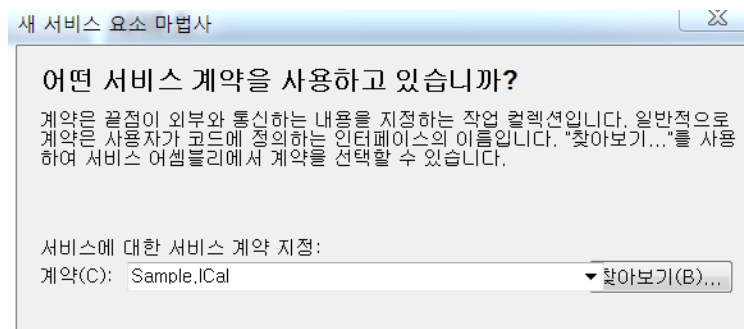
##### 5.2. 서비스 >> 새 서비스 만들기



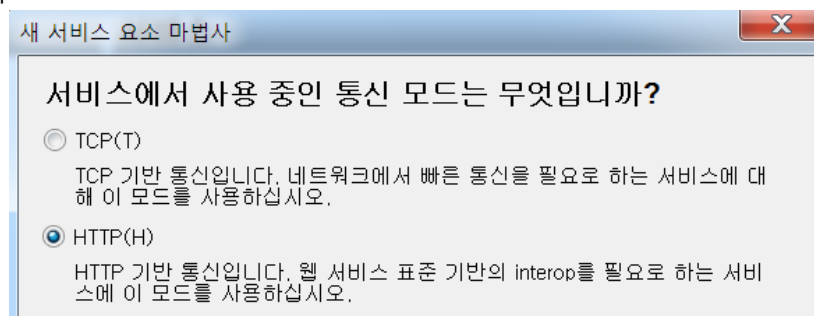
### 5.3. 서비스 유형 선택



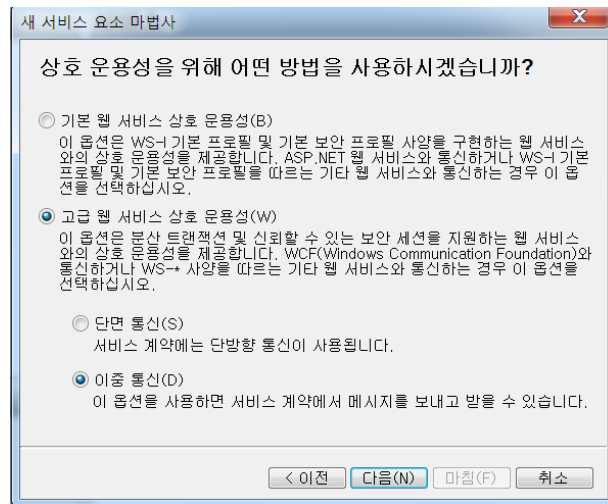
### 5.4 서비스 계약



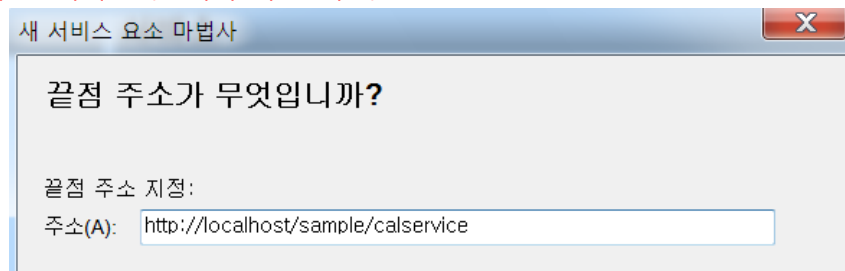
### 5.5 통신 모드 선택



### 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>

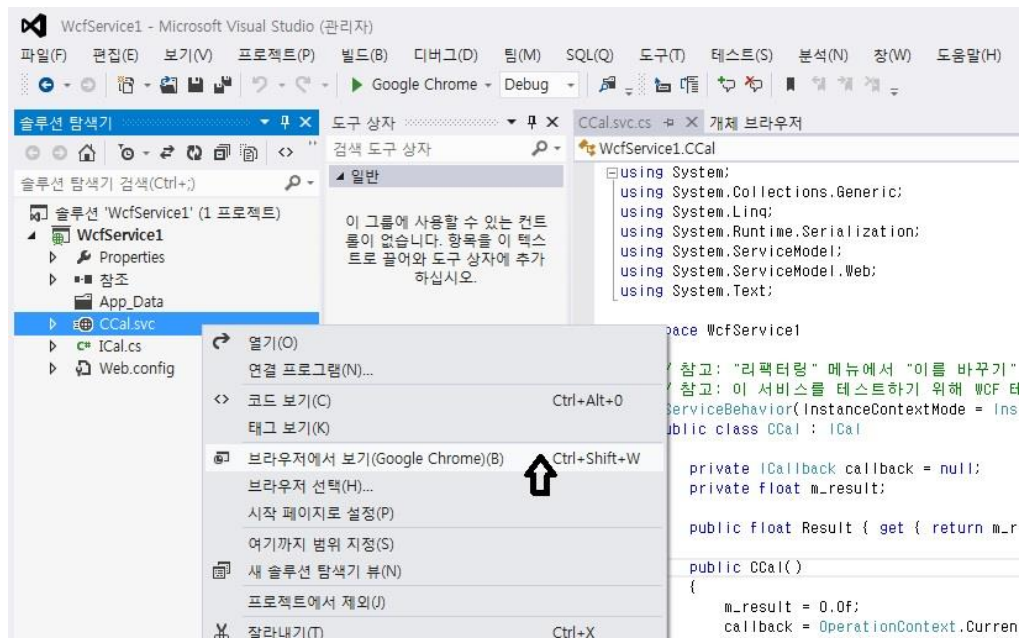
    <services>
      <service name="Sample.CCal">
        <endpoint address="" binding="wsDualHttpBinding"
          bindingConfiguration="" contract="Sample.ICal" />
      </service>
    </services>

  </system.serviceModel>

</configuration>
```

## 6. 실행 확인 및 게시

### 6.1 웹으로 실행 확인



## 6.2 정상적인 실행 결과

### CCal 서비스

서비스를 만들었습니다.

이 서비스를 테스트하려면 클라이언트를 만들고 이 클라이언트를 사용하여 서비스를 호출해야 합니다. 다음 구문의 명령줄에서 svcutil.exe 도구를 사용하여 이 작업을 할 수 있습니다.

```
svcutil.exe http://localhost:50113/CCal.svc?wsdl
```

또한 단일 파일로 서비스 설명에 액세스할 수도 있습니다.

```
http://localhost:50113/CCal.svc?singleWsdl
```

그러면 클라이언트 클래스가 포함된 구성 파일 및 코드 파일이 생성됩니다. 두 파일을 클라이언트 응용 프로그램에 추가하고 생성된 클라이언트 클래스를 사용하여 서비스를 호출하십시오. 예를 들면, 다음과 같습니다.

C#

```
class Test
{
    static void Main()
    {
        CalClient client = new CalClient();

        // 서비스에서 작업을 호출하려면 'client' 변수를 사용하십시오.

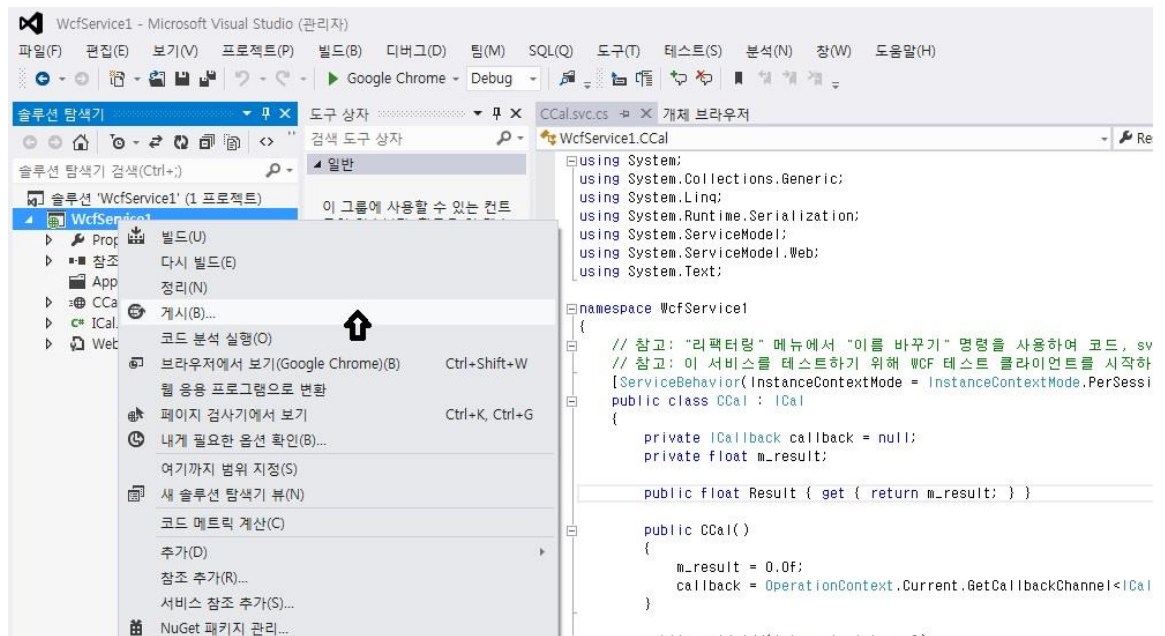
        // 항상 클라이언트를 닫으십시오.
        client.Close();
    }
}
```

Visual Basic

```
Class Test
    Shared Sub Main()
        Dim client As CalClient = New CalClient()
        ' 서비스에서 작업을 호출하려면 'client' 변수를 사용하십시오.

        ' 항상 클라이언트를 닫으십시오.
        client.Close()
    End Sub
End Class
```

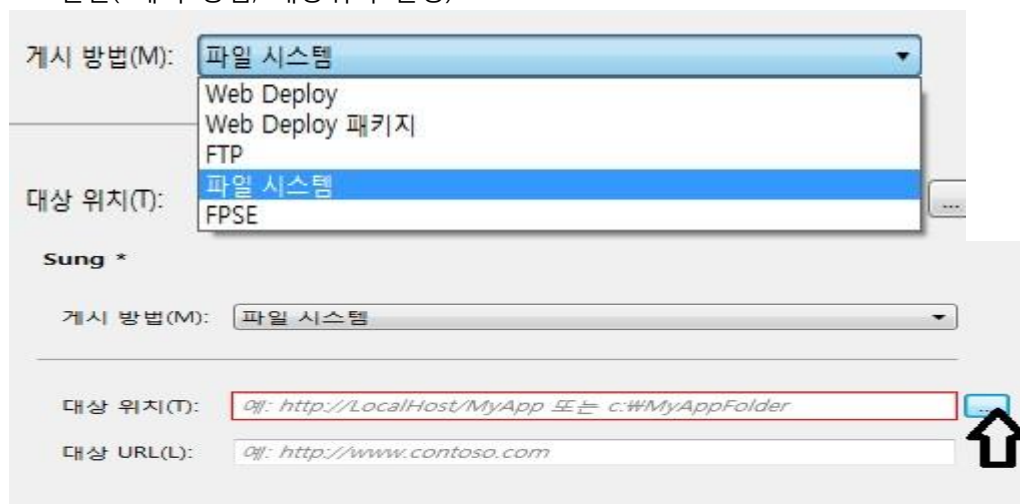
## 6.3 솔루션 탐색기 >> 현재 실행 프로젝트 우클릭 >> 게시



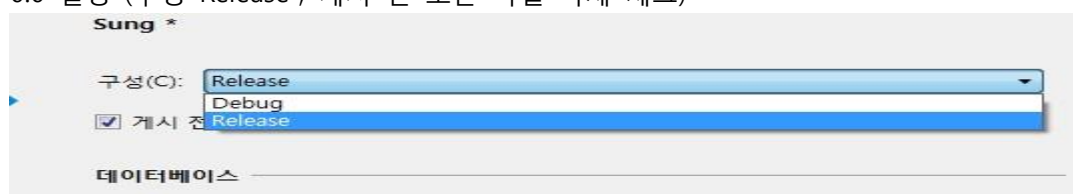
#### 6.4 프로필 등록

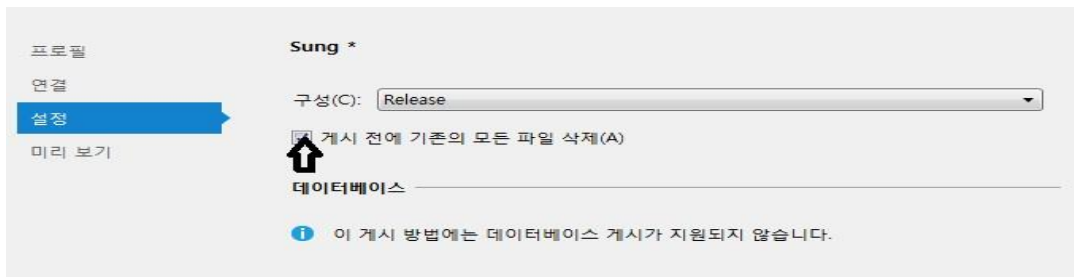


#### 6.5 연결( 게시 방법, 대상위치 설정)

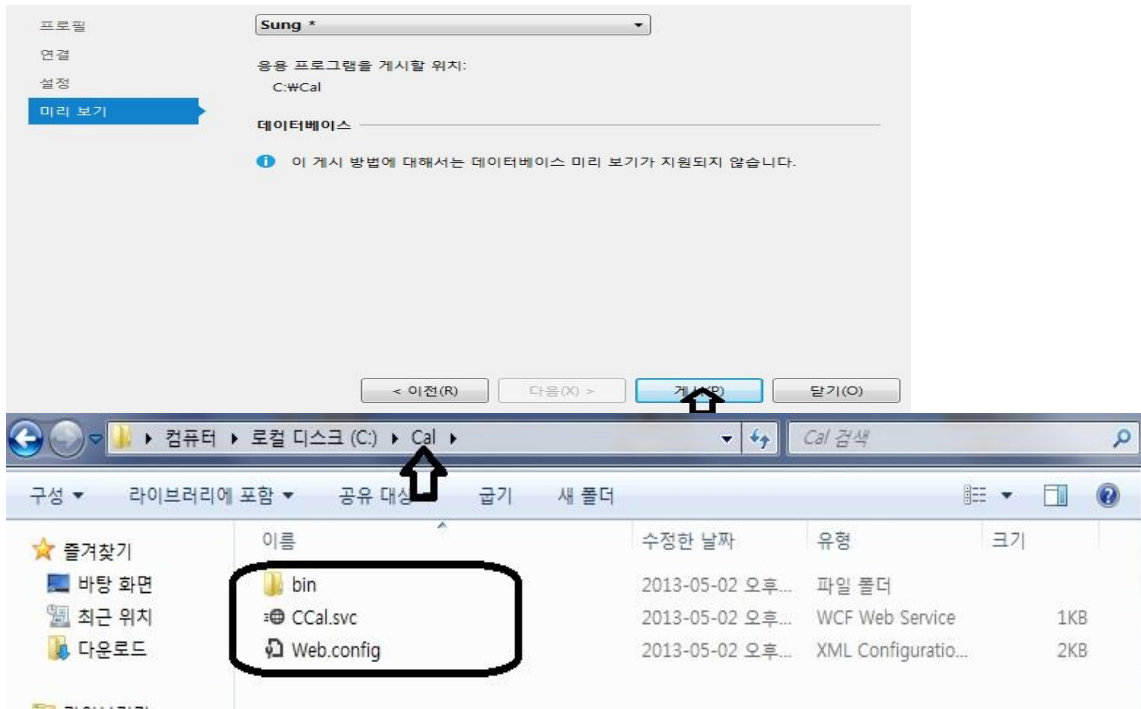


#### 6.6 설정 (구성 Release , 게시 전 모든 파일 삭제 체크)

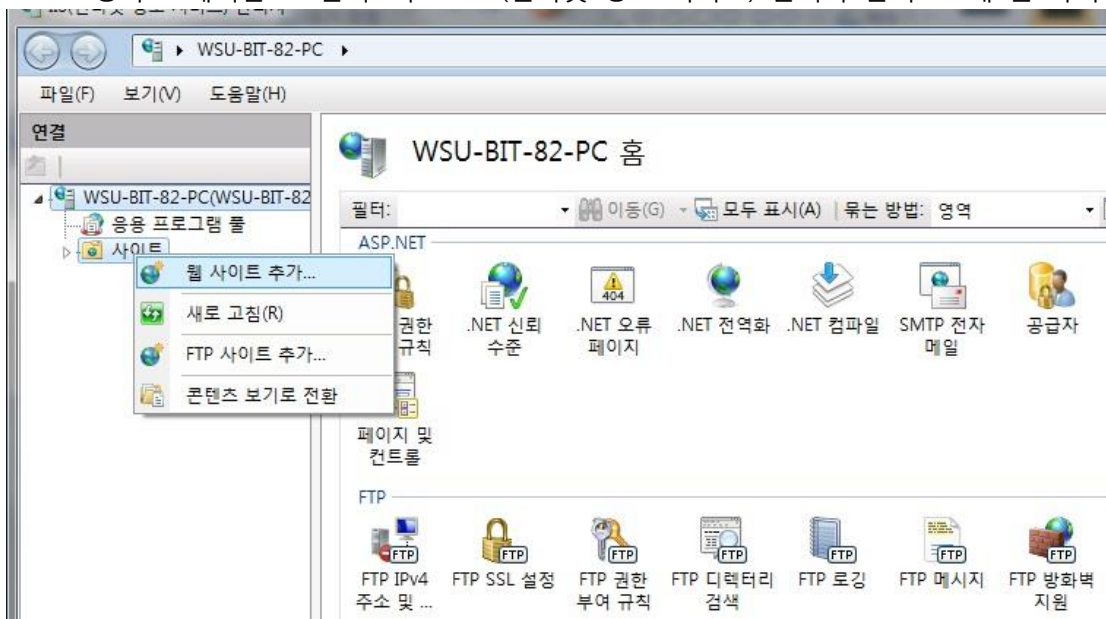




#### 6.7 정해진 경로에 웹 서비스 파일 게시( DLL 파일까지 자동등록됨 )



#### 6.8 IIS 등록 - 제어판>> 관리도구>> IIS(인터넷 정보 서비스) 관리자 클릭>> 새 웹 사이트 추가



사이트 이름, 경로, 포트 번호 설정



사이트 이름(S):  응용 프로그램 클(L):

콘텐츠 디렉터리

실제 경로(P):  

통과 인증

바인딩

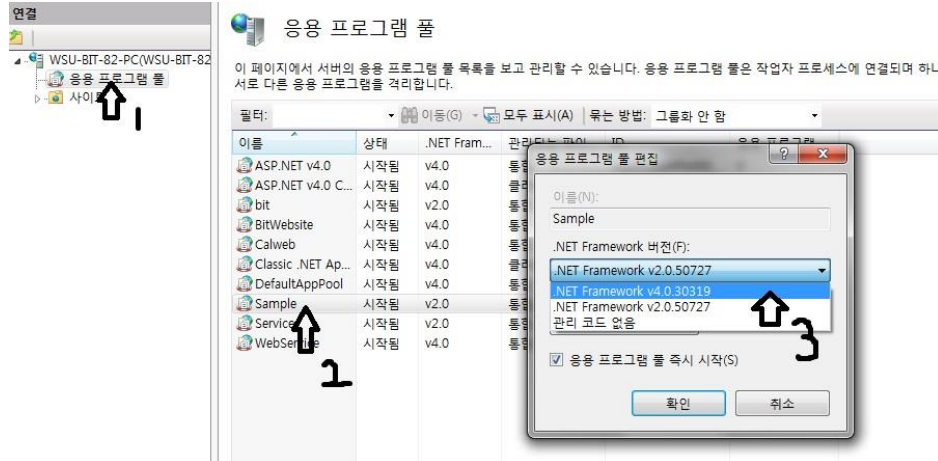
종류(T):  IP 주소(I):  포트(O):

호스트 이름(H):

예: www.contoso.com 또는 marketing.contoso.com

☒ 웹 사이트 즉시 시작(M)

## 6.9 응용 프로그램 풀에서 방금 생성한 사이트 .net 버전 최신버전으로 설정



## 6.10 정상적인 실행 확인 (사진 6.2 와 같은 화면이 나와야함)

<http://localhost:8080/CCal.svc>

[ Callback 기능 연동 클라이언트 ]

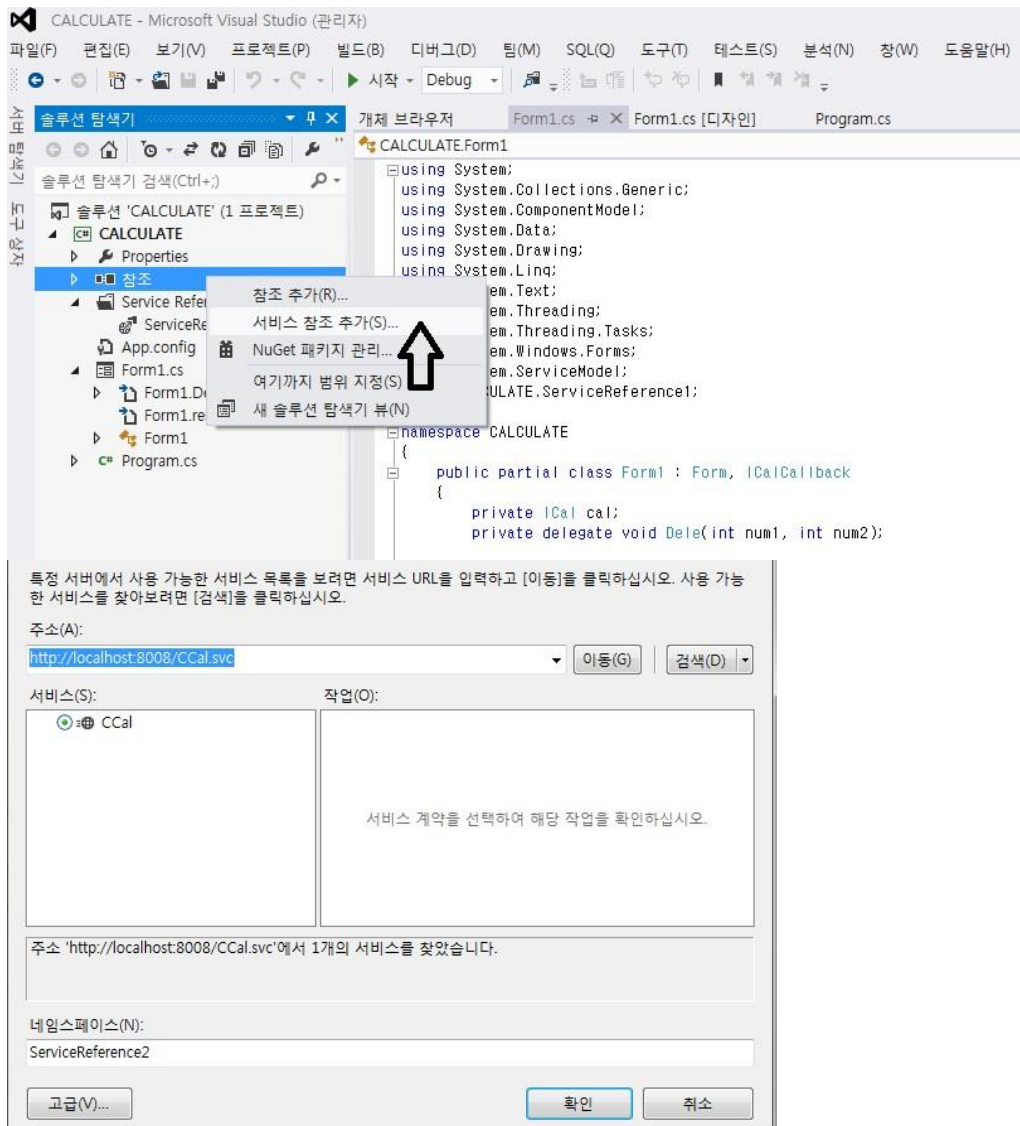
## 7. 새 프로젝트 생성

솔루션 : WindowsForm 프로그램

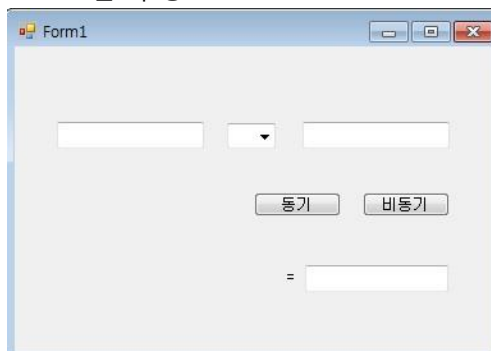
프로젝트 명 : Client

## 7.1 웹서비스 참조 추가 ( IIS에 등록된 웹 서비스 참조 )





## 7.2 UI 폼 구성



## 7.3 인터페이스 참조 ,Callback 인터페이스 상속 및 델리게이트 선언

```
public partial class Form1 : Form, ICalCallback
{
    private ICal cal;
    private delegate void Dele(int num1, int num2);

    public Form1()
    {
```

```

        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        InstanceContext site = new InstanceContext(this);
        cal = new CalClient(site);
    }
}

```

#### 7.4 ICalCallback 인터페이스 의 Result 함수 및 비동기 Callback 함수 생성

```

public void Result(float result)
{
    ResultBox.Text = result.ToString();
}

private void Callback(IAsyncResult result)
{
    var asyncResult = (System.Runtime.Remoting.Messaging.AsyncResult)result;
    var testDelegate = (Dele)asyncResult.AsyncDelegate;
    testDelegate.EndInvoke(result);
}

```

#### 7.4 동기식 버튼 컨트롤 설정

```

private void button1_Click(object sender, EventArgs e)
{
    Dele temp = null;
    try
    {
        switch (comboBox1.SelectedItem.ToString())
        {
            case "+": temp = new Dele(cal.Add); break;
            case "-": temp = new Dele(cal.Sub); break;
            case "*": temp = new Dele(cal.Mul); break;
            case "/": temp = new Dele(cal.Div); break;
        }
    }
    catch
    {
        MessageBox.Show("정상적으로 입력해 주세요");
    }
    if (temp != null && textBox2.Text != "0")
        temp.Invoke(int.Parse(Text_Num1.Text), int.Parse(Text_Num2.Text));
}

```

#### 7.5 비동기식 버튼 컨트롤 설정

```

private void button2_Click(object sender, EventArgs e)
{

```

```
Dele temp = null;
try
{
    switch(comboBox1.SelectedItem.ToString())
    {
        case "+": temp = new Dele(cal.Add); break;
        case "-": temp = new Dele(cal.Sub); break;
        case "*": temp = new Dele(cal.Mul); break;
        case "/": temp = new Dele(cal.Div); break;
    }
}
catch
{
    MessageBox.Show("정상적으로 입력해 주세요");
}
if(temp != null && textBox2.Text != "0")
    temp.BeginInvoke(int.Parse(Text_Num1.Text), int.Parse(Text_Num2.Text), Callback,
null);
}
```