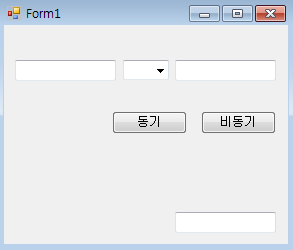
**WCF-계산기**

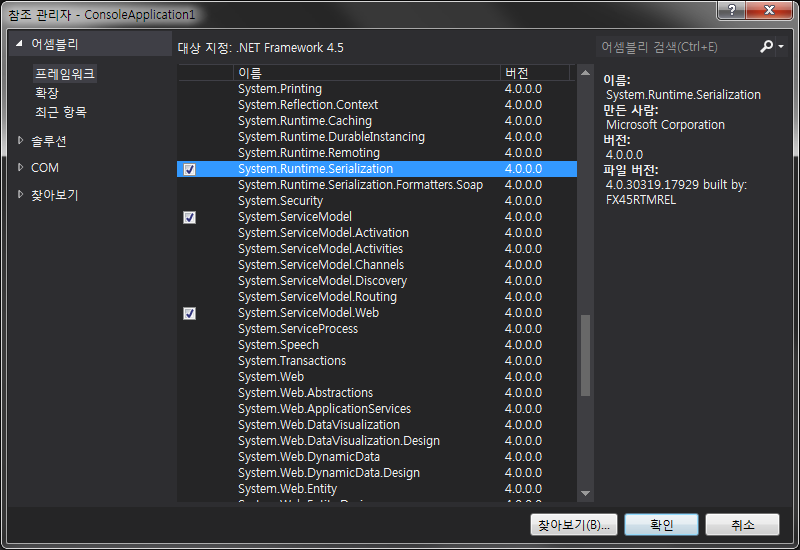
**< 계산기(Server) >**

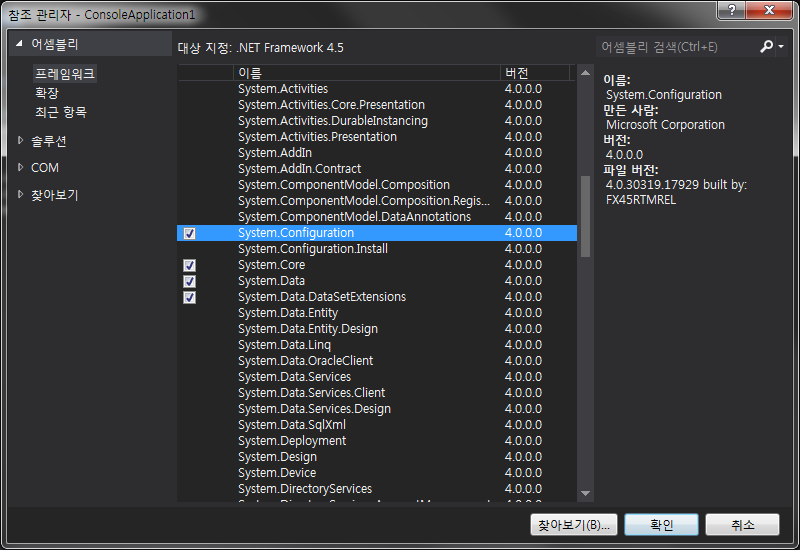
****

**< 참조추가 >**

System.Runtime.Serialization,System.Service.Model,

System.ServieceModel.Web ,System.Configuration **4**가지 추가

****

****

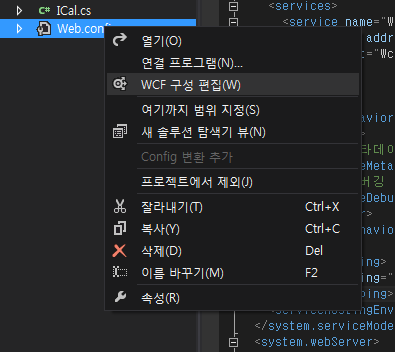
|  |
| --- |
| **Program** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Configuration;  using System.ServiceModel;  namespace ConsoleApplication4  {  class Program  {  static void Main(string[] args)  {  Uri uri = new Uri(ConfigurationManager.AppSettings["addr"]);  ServiceHost host = new ServiceHost(typeof(ConsoleApplication4.CCal), uri);  host.Open();  Console.WriteLine("채팅 서비스를 시작합니다. {0}", uri.ToString());  Console.WriteLine("멈추시려면 엔터를 눌러주세요.");  Console.ReadLine();  host.Abort();  host.Close();  }  }  } |

|  |
| --- |
| **Service1** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.ServiceModel;  using System.Runtime.Serialization;  namespace ConsoleApplication4  {  [ServiceContract(CallbackContract = typeof(ICallback))]  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);  // TODO: 여기에 서비스 작업을 추가합니다.  }  public interface ICallback  {  [OperationContract(IsOneWay = true)]  void Result(float result);  }  // 아래 샘플에 나타낸 것처럼 데이터 계약을 사용하여 복합 형식을 서비스 작업에 추가합니다.  [DataContract]  public class CompositeType  {  bool boolValue = true;  string stringValue = "Hello ";  [DataMember]  public bool BoolValue  {  get { return boolValue; }  set { boolValue = value; }  }  [DataMember]  public string StringValue  {  get { return stringValue; }  set { stringValue = value; }  }  }  public class CCal : ICal  {  private float m\_result1;  private ICallback callback1 = null;  public CCal()  {  callback1 = OperationContext.Current.GetCallbackChannel<ICallback>(); // 추가  }  public void Add(int num1, int num2) { m\_result1 = num1 + num2; callback1.Result(m\_result1); }  public void Sub(int num1, int num2) { m\_result1 = num1 - num2; callback1.Result(m\_result1); }  public void Mul(int num1, int num2) { m\_result1 = num1 \* num2; callback1.Result(m\_result1); }  public void Div(int num1, int num2) { m\_result1 = (float)num1 / num2; callback1.Result(m\_result1); }  }  } |

|  |
| --- |
| **컨피그** |
| <?xml version="1.0" encoding="utf-8" ?>  <configuration>  <startup>  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />  </startup>  **<appSettings>**  **<add key="addr" value="net.tcp://localhost:7878/ConsoleApplication4"/>**  **</appSettings>**  <system.serviceModel>  <services>  <service name="ConsoleApplication4.CCal"  **behaviorConfiguration="FileServiceBehavior">**  **<host>**  **<baseAddresses>**  **<add baseAddress="http://localhost:9000/ConsoleApplication4"/>**  **</baseAddresses>**  **</host>**  <endpoint address="" binding="netTcpBinding" bindingConfiguration="**DuplexBinding**"  contract="ConsoleApplication4.ICal" />  </service>  </services>  **<bindings>**  **<netTcpBinding>**  **<binding name="DuplexBinding" maxConnections="100">**  **<reliableSession enabled="true"/>**  **<security mode="None"/>**  **</binding>**  **</netTcpBinding>**  **</bindings>**  **<behaviors>**  **<serviceBehaviors>**  **<behavior name="FileServiceBehavior">**  **<serviceMetadata httpGetEnabled="True"/>**  **<serviceDebug includeExceptionDetailInFaults="true"/>**  **</behavior>**  **</serviceBehaviors>**  **</behaviors>**  </system.serviceModel>  </configuration> |

**< 웹컨피그 설정 >**

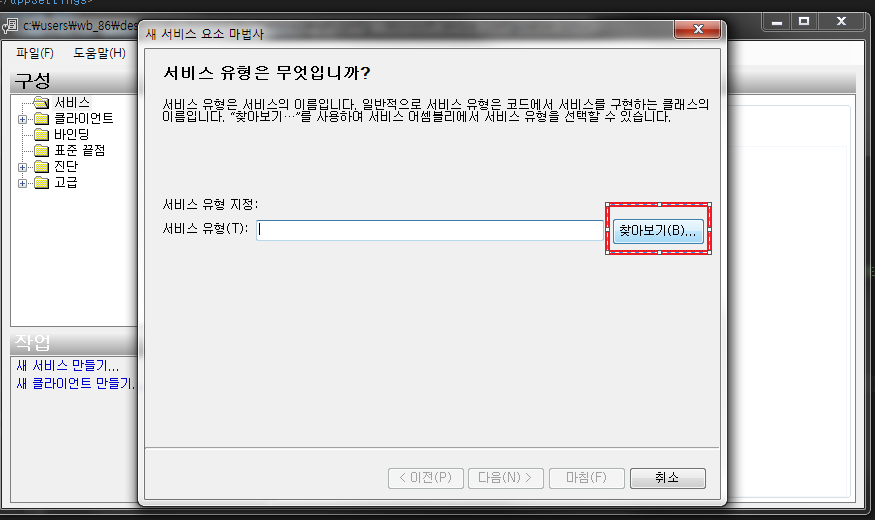
1. 먼저 WCF 구성 편집에 들어갑니다.



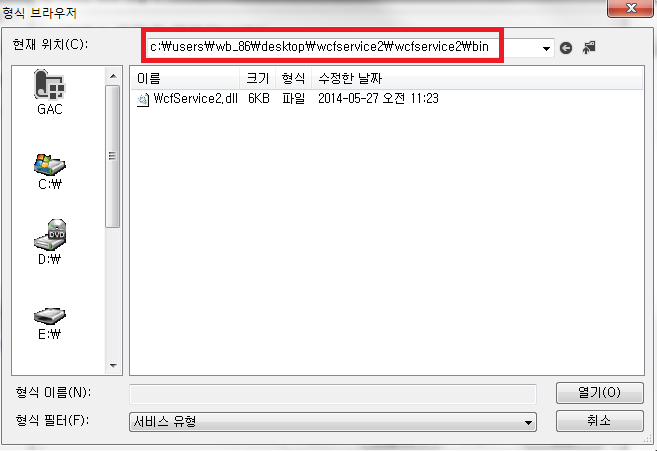
2.새 서비스 만들기를 선택 합니다.



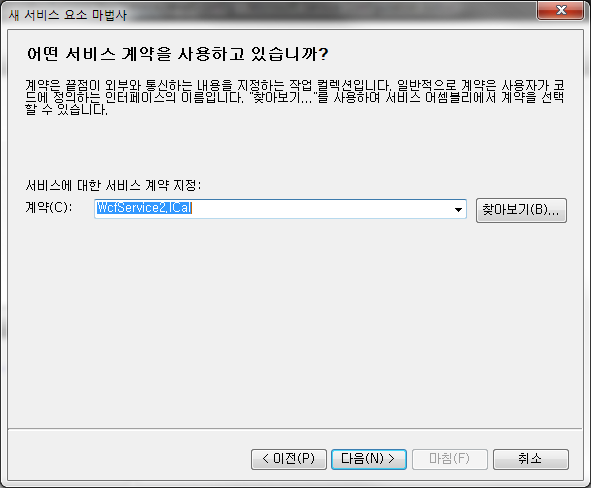
3.찾아보기를 누릅니다.



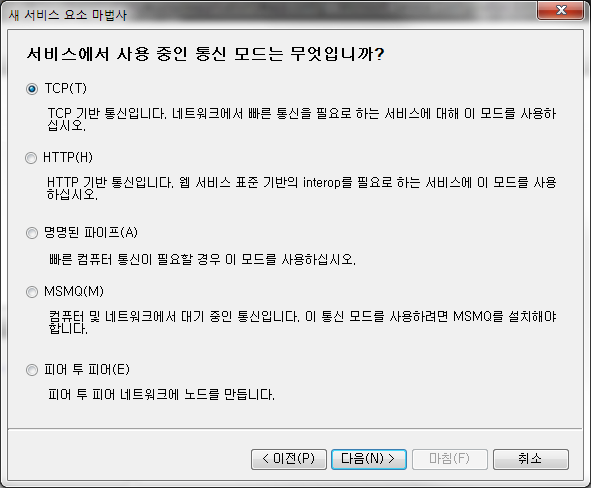
4.bin폴더에 들어갑니다.



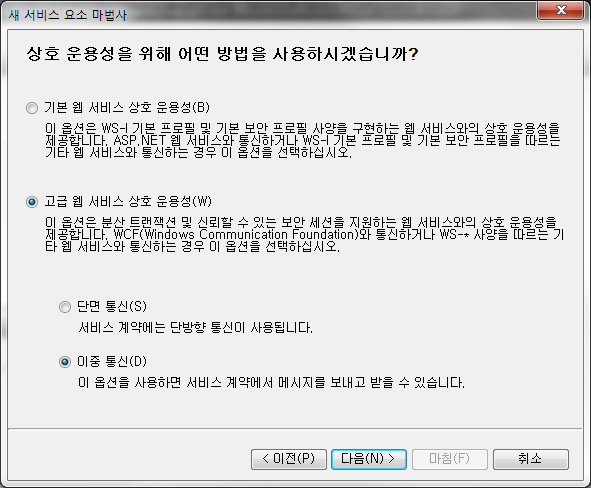
5.



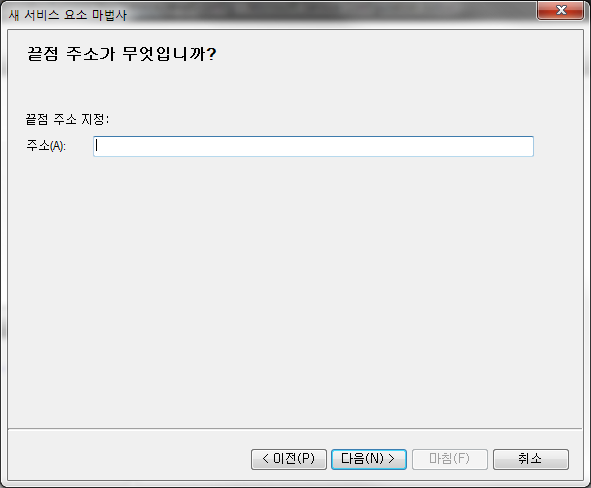
6.



7.



8. 끝점 주소는 null로 지정해 주면 알아서 지정이 됩니다.

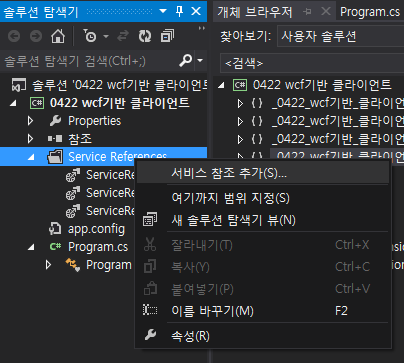


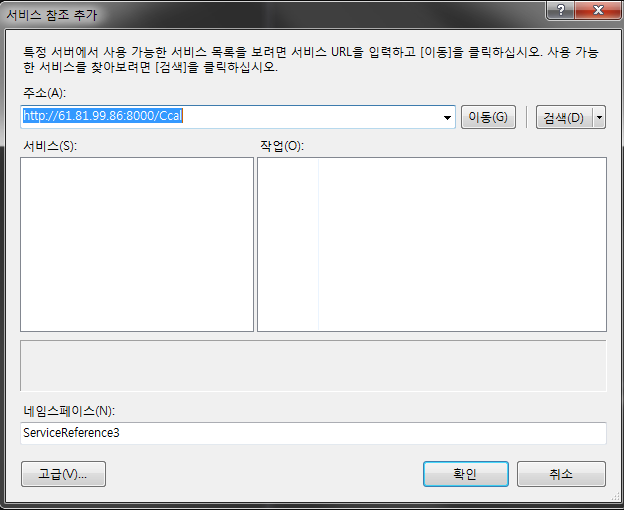
**< 계산기(Client) >**

|  |
| --- |
| **Form1** |
| 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);  }  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);  }  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 && Text\_Num2.Text != "0")  temp.Invoke(int.Parse(Text\_Num1.Text), int.Parse(Text\_Num2.Text));  }  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 && Text\_Num2.Text != "0")  temp.BeginInvoke(int.Parse(Text\_Num1.Text), int.Parse(Text\_Num2.Text), Callback, null);  }  } |

|  |
| --- |
| **Program** |
| static class Program  {  /// <summary>  /// 해당 응용 프로그램의 주 진입점입니다.  /// </summary>  [STAThread]  static void Main()  {  Application.EnableVisualStyles();  Application.SetCompatibleTextRenderingDefault(false);  Application.Run(new Form1());  }  } |

* **주의사항 – 서비스참조를 받아야함**

****

****

**마지막으로 using처리를 해줍니다.**