# How to create and host a WCF Duplex Service using wsDualHttpBinding

## Introduction

This sample shows how to create and host a WCF Duplex Service using wsDualHttpBinding.

WSDualHttpBinding supports duplex services. A duplex service is a service that uses duplex message patterns.

These patterns provide the ability for a service to communicate back to the client via a callback.

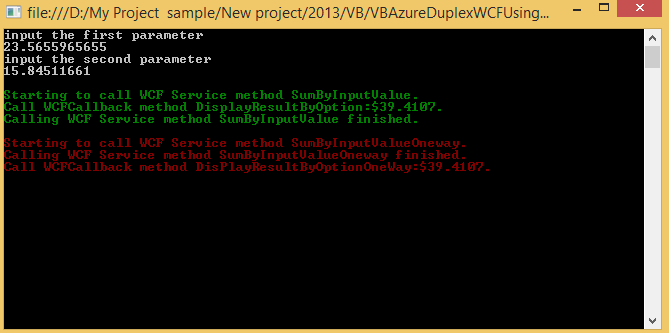
## Running the Sample

Step 1: Run VS2013 as administrator, then open the solution.

Step 2: Press Ctrl+F5 start the work role on compute emulator.

Step 3: Right click Client project, choose debug->start new instance.

Step 4: This is the screenshot when the solution is ran successfully.



## Using the Code

Step 1:  Create the WCF service class library.

|  |
| --- |
| -Code block start-  --C# code snippet start--  [ServiceContract(CallbackContract = typeof(IServiceCalulateCallBack), SessionMode = SessionMode.Required)]  public interface IServiceCalculate  {  /// <summary>  /// Get a value dbeOneParameter plus dbeTwoParameter when the mehtod finishes waiting for an underlying response message.  /// </summary>  /// <param name="dbeOneParameter"></param>  /// <param name="dbeTwoParameter"></param>  /// <param name="strSymbol"></param>  [OperationContract]  void SumByInputValue(double dbeOneParameter, double dbeTwoParameter, string strSymbol);  /// <summary>  /// Get a value dbeOneParameter plus dbeTwoParameter without the method finishing waiting for an underlying response message.  /// </summary>  /// <param name="dbeOneParameter"></param>  /// <param name="dbeTwoParameter"></param>  /// <param name="strSymbol"></param>  [OperationContract(IsOneWay = true)]  void SumByInputValueOneway(double dbeOneParameter, double dbeTwoParameter, string strSymbol);  }  public interface IServiceCalulateCallBack  {  /// <summary>  /// The value will be displayed according to the format value chosen when the method finishes waiting for an underlying response message.  /// </summary>  /// <param name="strSymbol"></param>  /// <param name="dbeSumValue"></param>  [OperationContract]  void DisplayResultByOption(string strSymbol, double dbeSumValue);    /// <summary>  /// The value will be displayed according to the format value chosen without the method finishing waiting for an underlying response message.  /// </summary>  /// <param name="strSymbol"></param>  /// <param name="dbeSumValue"></param>  [OperationContract(IsOneWay=true)]  void DisPlayResultByOptionOneWay(string strSymbol, double dbeSumValue);  }  --C# code snippet end--  --VB code snippet start--  <ServiceContract(CallbackContract:=GetType(IServiceCalulateCallBack), SessionMode:=SessionMode.Required)>  Public Interface IServiceCalculate  ''' <summary>  ''' Get a value dbeOneParameter plus dbeTwoParameter when the mehtod finishes waiting for an underlying response message.  ''' </summary>  ''' <param name="dbeOneParameter"></param>  ''' <param name="dbeTwoParameter"></param>  ''' <param name="strSymbol"></param>  ''' <remarks></remarks>  <OperationContract>  Sub SumByInputValue(ByVal dbeOneParameter As Double, ByVal dbeTwoParameter As Double, ByVal strSymbol As String)  ''' <summary>  ''' Get a value dbeOneParameter plus dbeTwoParameter without the method finishing waiting for an underlying response message.  ''' </summary>  ''' <param name="dbeOneParameter"></param>  ''' <param name="dbeTwoParameter"></param>  ''' <param name="strSymbol"></param>  ''' <remarks></remarks>  <OperationContract(IsOneWay:=True)>  Sub SumByInputValueOneway(ByVal dbeOneParameter As Double, ByVal dbeTwoParameter As Double, ByVal strSymbol As String)  End Interface  Public Interface IServiceCalulateCallBack  ''' <summary>  ''' The value will be displayed according to the format value chosen when the method finishes waiting for an underlying response message.  ''' </summary>  ''' <param name="strSymbol"></param>  ''' <param name="dbeSumValue"></param>  ''' <remarks></remarks>  <OperationContract>  Sub DisplayResultByOption(ByVal strSymbol As String, ByVal dbeSumValue As Double)  ''' <summary>  ''' The value will be displayed according to the format value chosen without the method finishing waiting for an underlying response message.  ''' </summary>  ''' <param name="strSymbol"></param>  ''' <param name="dbeSumValue"></param>  ''' <remarks></remarks>  <OperationContract(IsOneWay:=True)>  Sub DisPlayResultByOptionOneWay(ByVal strSymbol As String, ByVal dbeSumValue As Double)  --VB code snippet end--  -Code block end- |

Step2:  Host the WCF on the work role.

The following code shows how to run the service on work role.

|  |
| --- |
| -Code block start-  --C# code snippet start--  public override void Run()  {  Trace.TraceInformation("HostWCFService is running");  Trace.TraceInformation(" Try to start hosting WCF service...");  this.serviceHost = new ServiceHost(typeof(ServiceCalculate.ServiceCalculate));  try  {    this.serviceHost.Open();  Trace.TraceInformation("WCF service hosting started successfully.");  }  catch (TimeoutException timeoutException)  {  Trace.TraceError("The service operation timed out. {0}",  timeoutException.Message);  }  catch (CommunicationException communicationException)  {  Trace.TraceError("Could not start WCF service host. {0}",  communicationException.Message);  }  try  {  this.RunAsync(this.cancellationTokenSource.Token).Wait();  }  finally  {  this.runCompleteEvent.Set();  }  }  --C# code snippet end--  --VB code snippet start--  Public Overrides Sub Run()  Trace.TraceInformation("HostWCFService is running")  Trace.TraceInformation(" Try to start WCF service host...")  Me.serviceHost = New ServiceHost(GetType(ServiceCalculate.ServiceCalculate))  Try  Me.serviceHost.Open()  Trace.TraceInformation("WCF service host started successfully.")  Catch timeoutException As TimeoutException  Trace.TraceError("The service operation timed out. {0}", timeoutException.Message)  Catch communicationException As CommunicationException  Trace.TraceError("Could not start WCF service host. {0}", communicationException.Message)  End Try  Try  Me.RunAsync(Me.cancellationTokenSource.Token).Wait()  Finally  Me.runCompleteEvent.Set()  End Try  End Sub  --VB code snippet end--  -Code block end- |

Step3:  The following code shows the config file on the work role.

|  |
| --- |
| -Code block start-  --XML code snippet start--  <system.serviceModel>  <services>  <service name="ServiceCalculate.ServiceCalculate" behaviorConfiguration="CalculatorServiceBehavior" >  <host>  <baseAddresses>  <add baseAddress="http://localhost/CalculatorService"/>  </baseAddresses>  </host>  <endpoint address="" binding="wsDualHttpBinding" contract="ServiceCalculate.IServiceCalculate"> </endpoint>  </service>  </services>  <behaviors>  <serviceBehaviors >  <behavior name="CalculatorServiceBehavior">  <serviceMetadata httpGetEnabled="true"/>  </behavior>  </serviceBehaviors>  </behaviors>  </system.serviceModel>  --XML code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

Step3:  Call WCF Service on the client project.

|  |
| --- |
| -Code block start-  --C# code snippet start--  static void Main(string[] args)  {  InstanceContext instance = new InstanceContext(new Program());  CalculatorServiceClient.ServiceCalculateClient client = new CalculatorServiceClient.ServiceCalculateClient(instance);  try  {  WSDualHttpBinding binding = client.Endpoint.Binding as WSDualHttpBinding;  binding.ClientBaseAddress = new Uri("http://localhost:8081/client");  double dbeOneParameter = 0;  double dbeTwoParameter = 0;  Console.WriteLine("input the first parameter.");  inputParameter(out dbeOneParameter);  Console.WriteLine("input the second parameter.");  inputParameter(out dbeTwoParameter);  Console.WriteLine();  Console.ForegroundColor = ConsoleColor.DarkGreen;  Console.WriteLine("Starting to call WCF Service method SumByInputValue.");  client.SumByInputValue(dbeOneParameter, dbeTwoParameter, strSymbol);  Console.WriteLine("Calling WCF Service method SumByInputValue finished.");  Console.WriteLine();  Console.ResetColor();  Console.ForegroundColor = ConsoleColor.DarkRed;  Console.WriteLine("Starting to call WCF Service method SumByInputValueOneway.");  client.SumByInputValueOneway(dbeOneParameter, dbeTwoParameter, strSymbol);  Console.WriteLine("Calling WCF Service method SumByInputValueOneway finished.");  }  catch (Exception ex)  {  }  Console.ReadLine();  Console.ResetColor();  }    public void DisplayResultByOption(string strSymbol, double dbeSumValue)  {  Console.Write("Call WCFCallback method DisplayResultByOption: ");  Console.WriteLine(string.Format(strSymbol, dbeSumValue)+".");  }  public void DisPlayResultByOptionOneWay(string strSymbol, double dbeSumValue)  {  Console.Write("Call WCFCallback method DisPlayResultByOptionOneWay:");  Console.WriteLine(string.Format(strSymbol, dbeSumValue) + ".");  }  --C# code snippet end--  --VB code snippet start--  Sub CallWCF()  Dim instance As New InstanceContext(New Program())  Dim client As New CalculatorServiceClient.ServiceCalculateClient(instance)  Try  Dim binding As WSDualHttpBinding = TryCast(client.Endpoint.Binding, WSDualHttpBinding)  binding.ClientBaseAddress = New Uri("http://localhost:8081/client")  Dim dbeOneParameter As Double = 0  Dim dbeTwoParameter As Double = 0  Console.WriteLine("input the first parameter")  inputParameter(dbeOneParameter)  Console.WriteLine("input the second parameter")  inputParameter(dbeTwoParameter)  Console.WriteLine()  Console.ForegroundColor = ConsoleColor.DarkGreen  Console.WriteLine("Starting to call WCF Service method SumByInputValue.")  client.SumByInputValue(dbeOneParameter, dbeTwoParameter, strSymbol)  Console.WriteLine("Calling WCF Service method SumByInputValue finished.")  Console.WriteLine()  Console.ResetColor()  Console.ForegroundColor = ConsoleColor.DarkRed  Console.WriteLine("Starting to call WCF Service method SumByInputValueOneway.")  client.SumByInputValueOneway(dbeOneParameter, dbeTwoParameter, strSymbol)  Console.WriteLine("Calling WCF Service method SumByInputValueOneway finished.")  Catch ex As Exception  End Try  Console.ReadLine()  Console.ResetColor()  End Sub    Public Sub DisplayResultByOption(ByVal strSymbol As String, ByVal dbeSumValue As Double) Implements CalculatorServiceClient.IServiceCalculateCallback.DisplayResultByOption  Console.Write("Call WCFCallback method DisplayResultByOption:")  Console.WriteLine(String.Format(strSymbol, dbeSumValue) + ".")  End Sub  Public Sub DisPlayResultByOptionOneWay(ByVal strSymbol As String, ByVal dbeSumValue As Double) Implements CalculatorServiceClient.IServiceCalculateCallback.DisPlayResultByOptionOneWay  Console.Write("Call WCFCallback method DisPlayResultByOptionOneWay:")  Console.WriteLine(String.Format(strSymbol, dbeSumValue) + ".")  End Sub  --VB code snippet end--  -Code block end- |

## More Information

<https://msdn.microsoft.com/en-us/library/ms731354(v=vs.110).aspx>

<https://msdn.microsoft.com/en-us/library/ms731360(v=vs.110).aspx>

<https://msdn.microsoft.com/en-us/library/ms731298(v=vs.110).aspx>