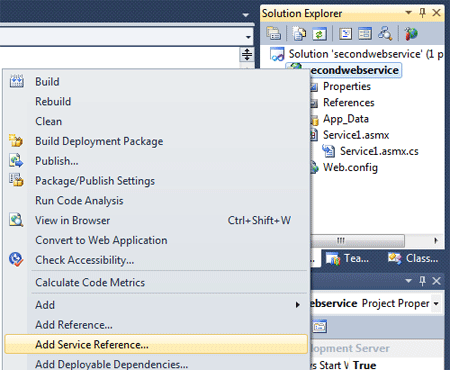
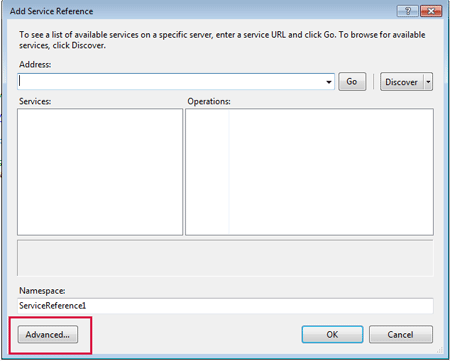
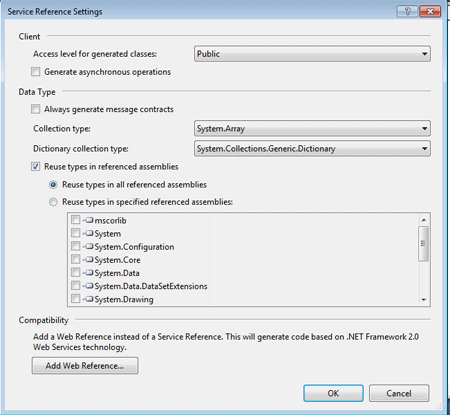
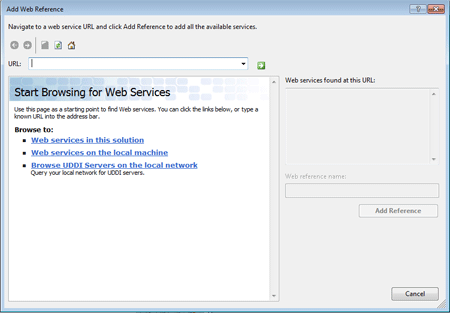
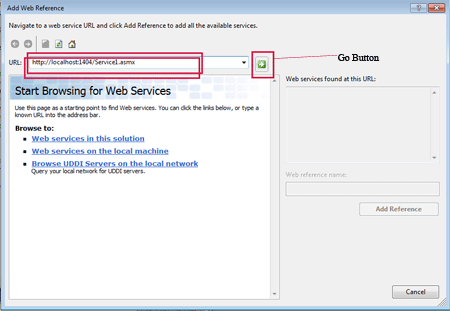
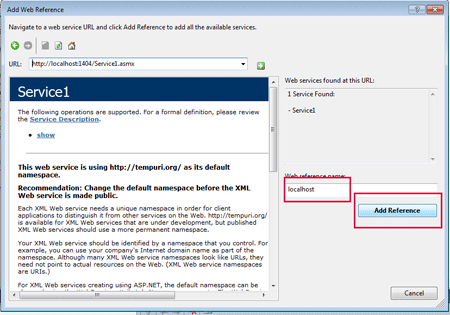
Add Exchange Rate web service in AX

We have used the kowabunga link to get the exchange rate and it is free. The link is : “<http://currencyconverter.kowabunga.net/converter.asmx>”

Reference link : <https://www.c-sharpcorner.com/UploadFile/718fc8/calling-a-web-service-into-another-web-service-application/>

1. Create web service Application :
   1. Go to Visual Studio 2010  and create a New Project
   2. Select an ASP.NET Web Service Application
   3. Go to Solution Explorer and right click at your project.  
        
      
   4. Select Add Service Reference. A new window will be open.

Here I have use this link : <http://currencyconverter.kowabunga.net/converter.asmx>  
  


* 1. Click at "Advance" button ( Given in Above figure.). A new window will open.  
       
     
  2. Click at "Add Web Reference" button.  
       
     
  3. Copy the URL of first (Which has created at first) web service and paste it at the URL. Look at below screen - shot.  
       
     
  4. Click the Go button.  
       
     
  5. You can set Web reference name. the default name is "localhost". Click Add reference.
  6. Now a web reference has added into your web service application. Now replace the code with the following code.  
       
     using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

/// <summary>

/// Summary description for CurrencyRateConversion

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class CurrencyRateConversion : System.Web.Services.WebService

{

public CurrencyRateConversion()

{

}

[WebMethod]

public string CurrencyRateConvert(string CurrencyFrom, string CurrencyTo, DateTime RateDate)

{

try

{

decimal ConversionRate = 0;

currencyconverter.Converter cr = new currencyconverter.Converter();

return Convert.ToString(cr.GetConversionRate(CurrencyFrom, CurrencyTo, RateDate));

}

catch (Exception ex)

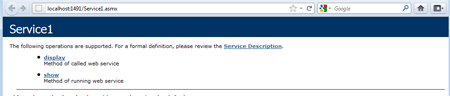
{

throw ex;

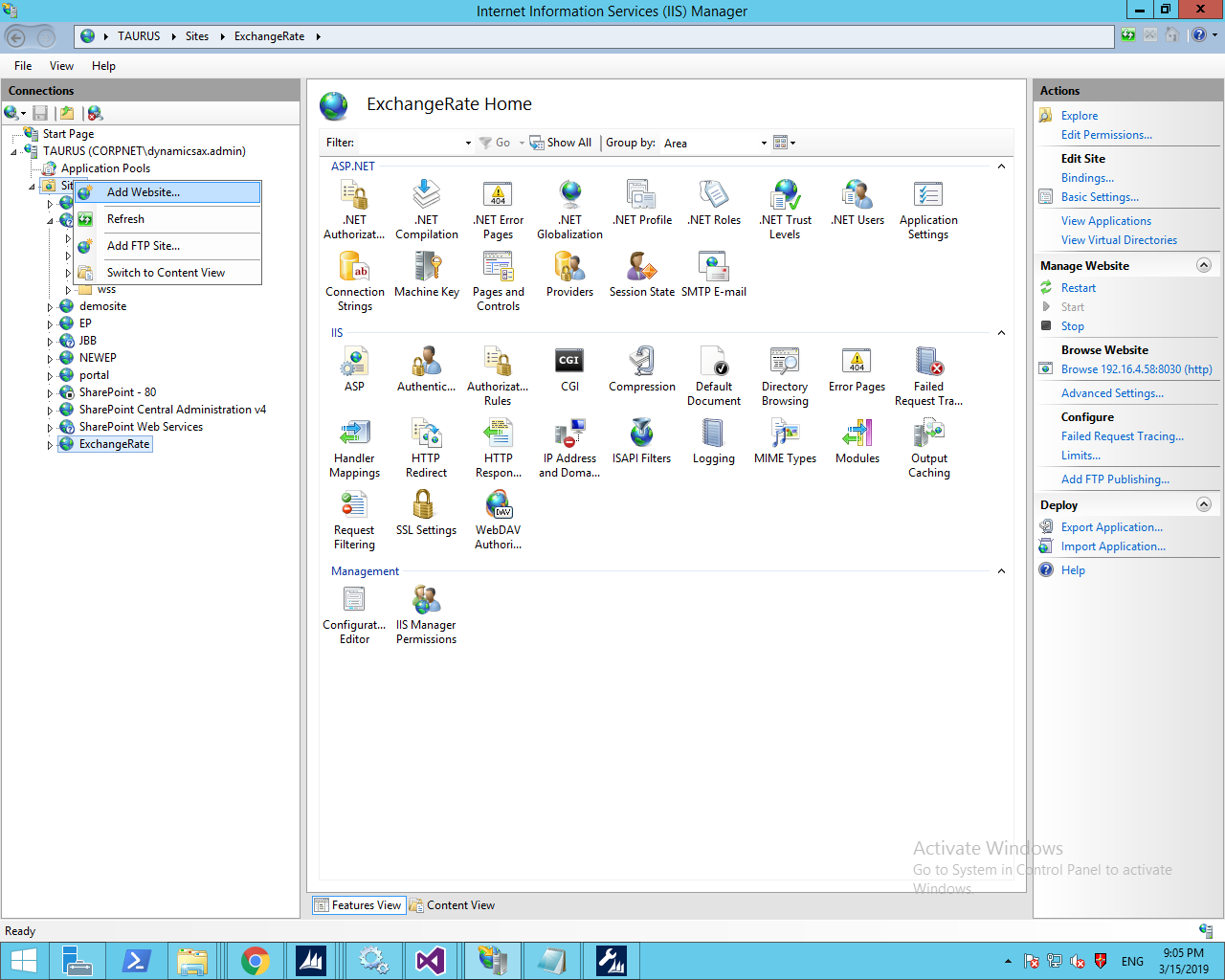
}

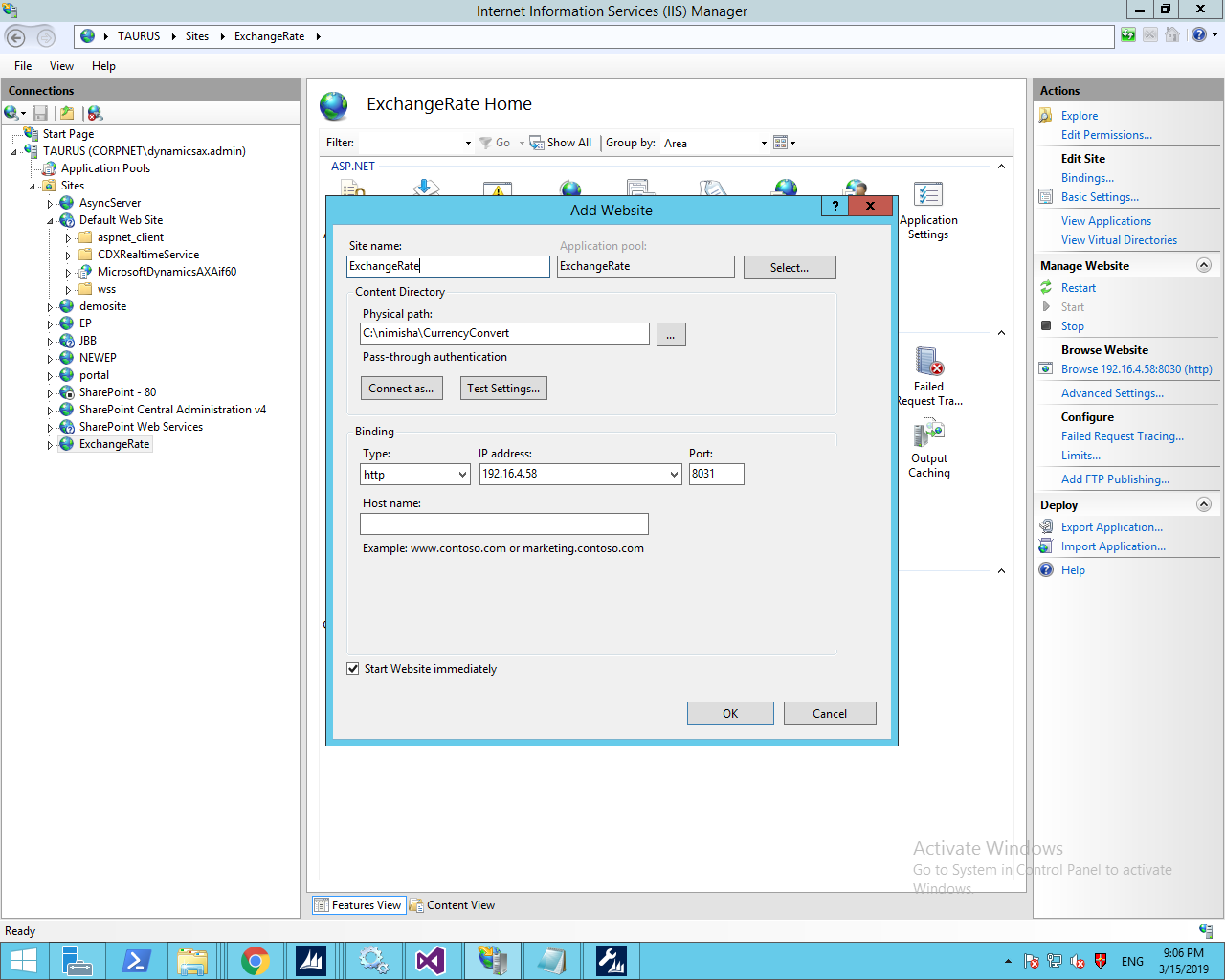
}

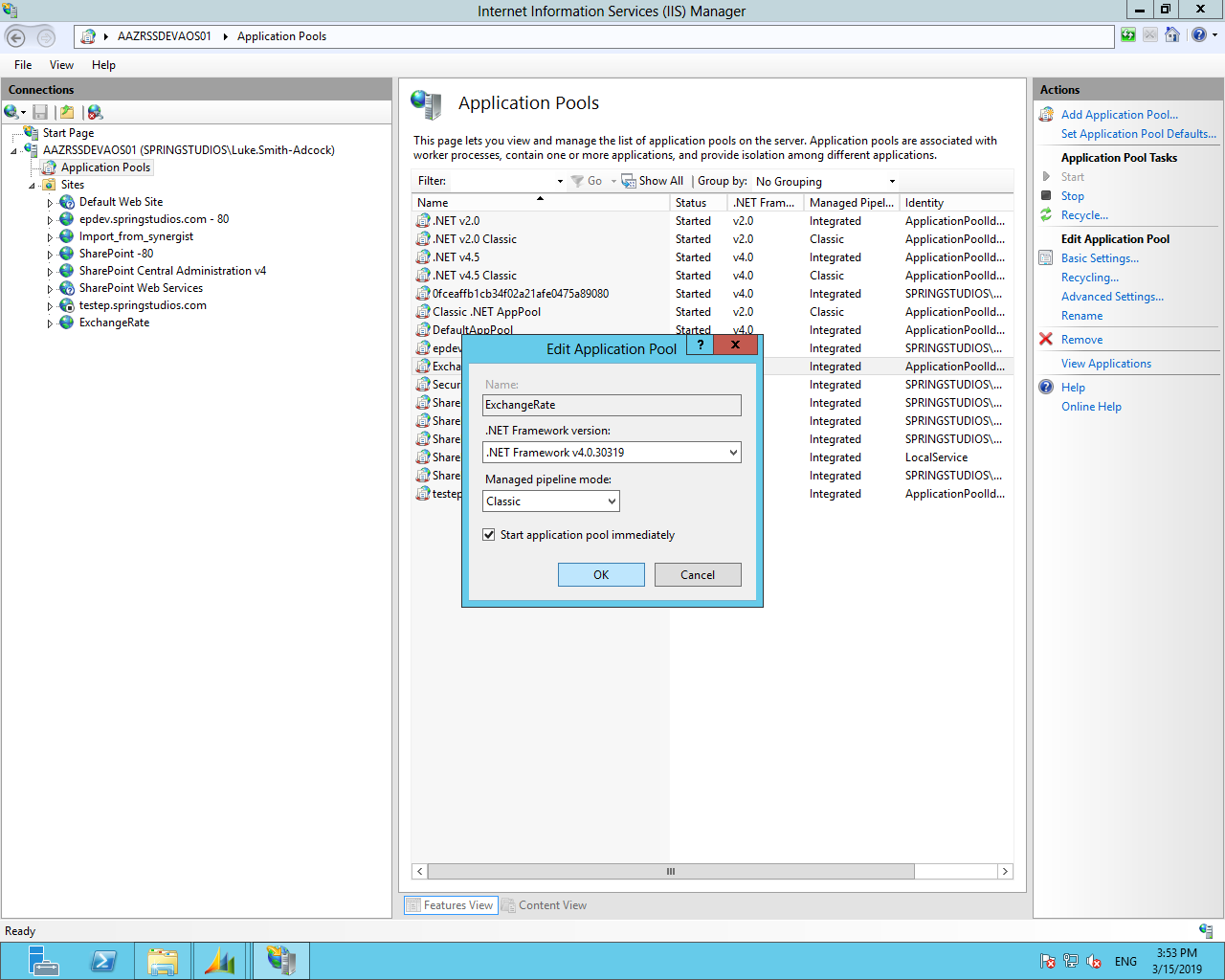
}

* 1. Run the service.   
       
     **Output:**

1. Host the custom web service (created in Step -1) on IIS :

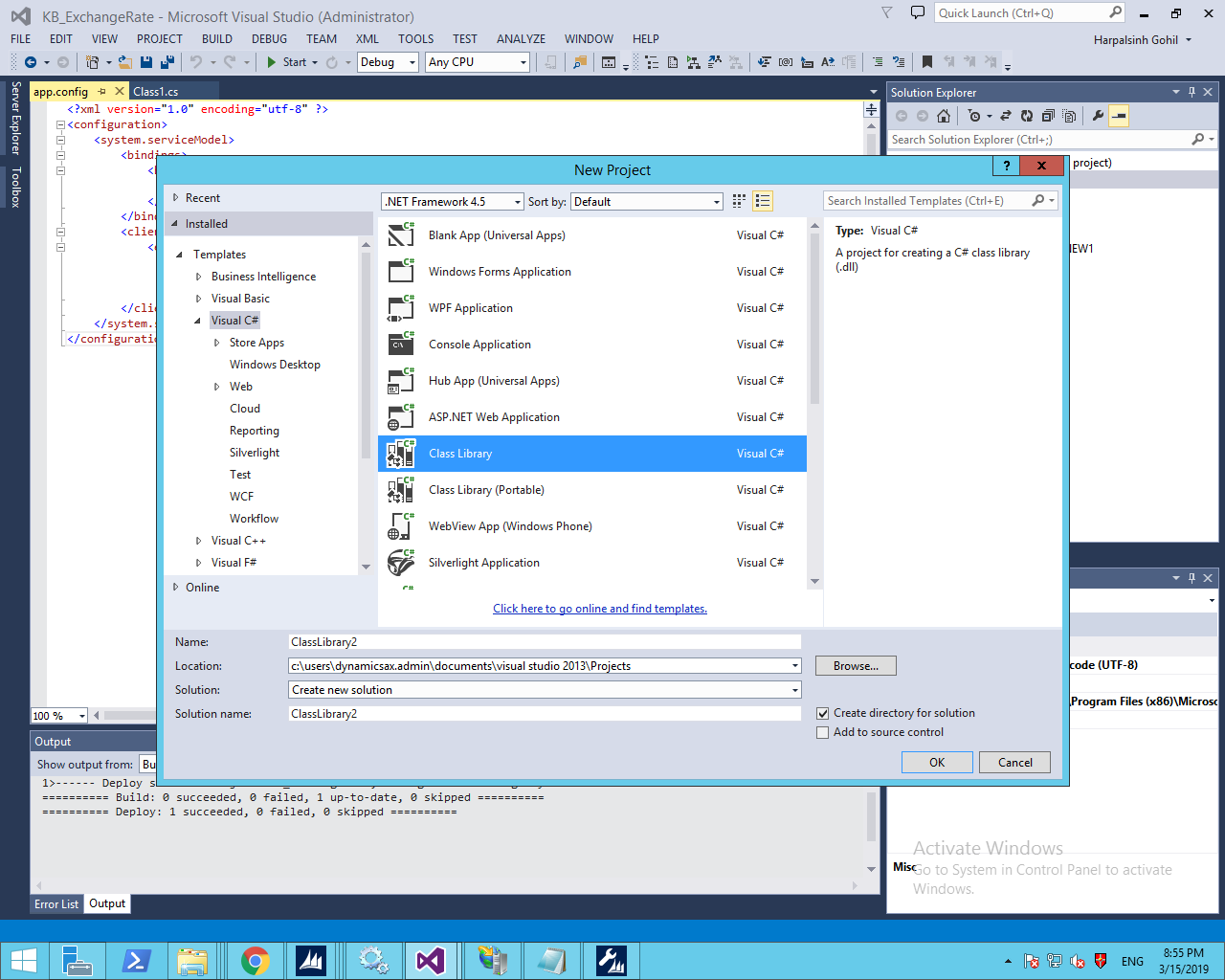


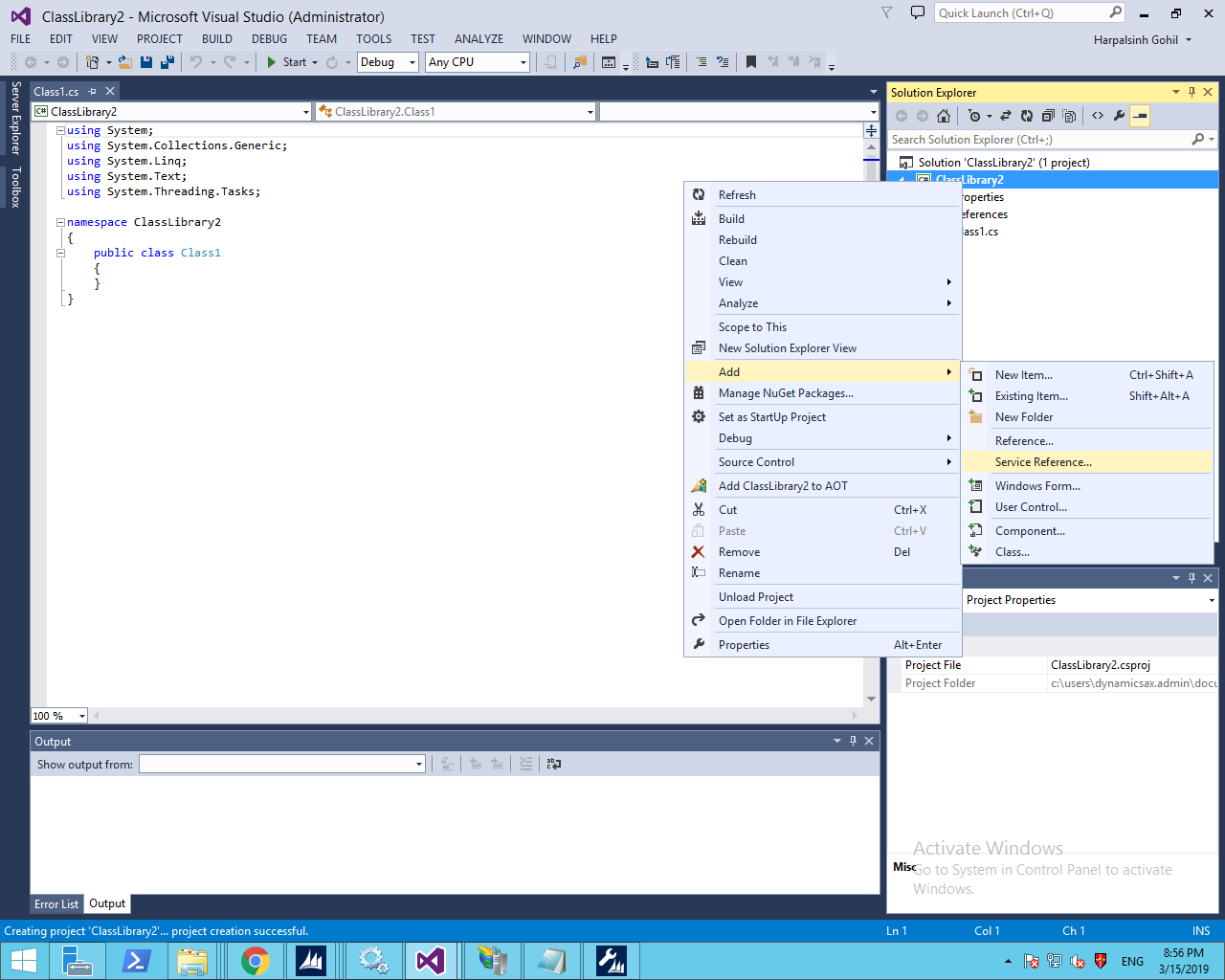


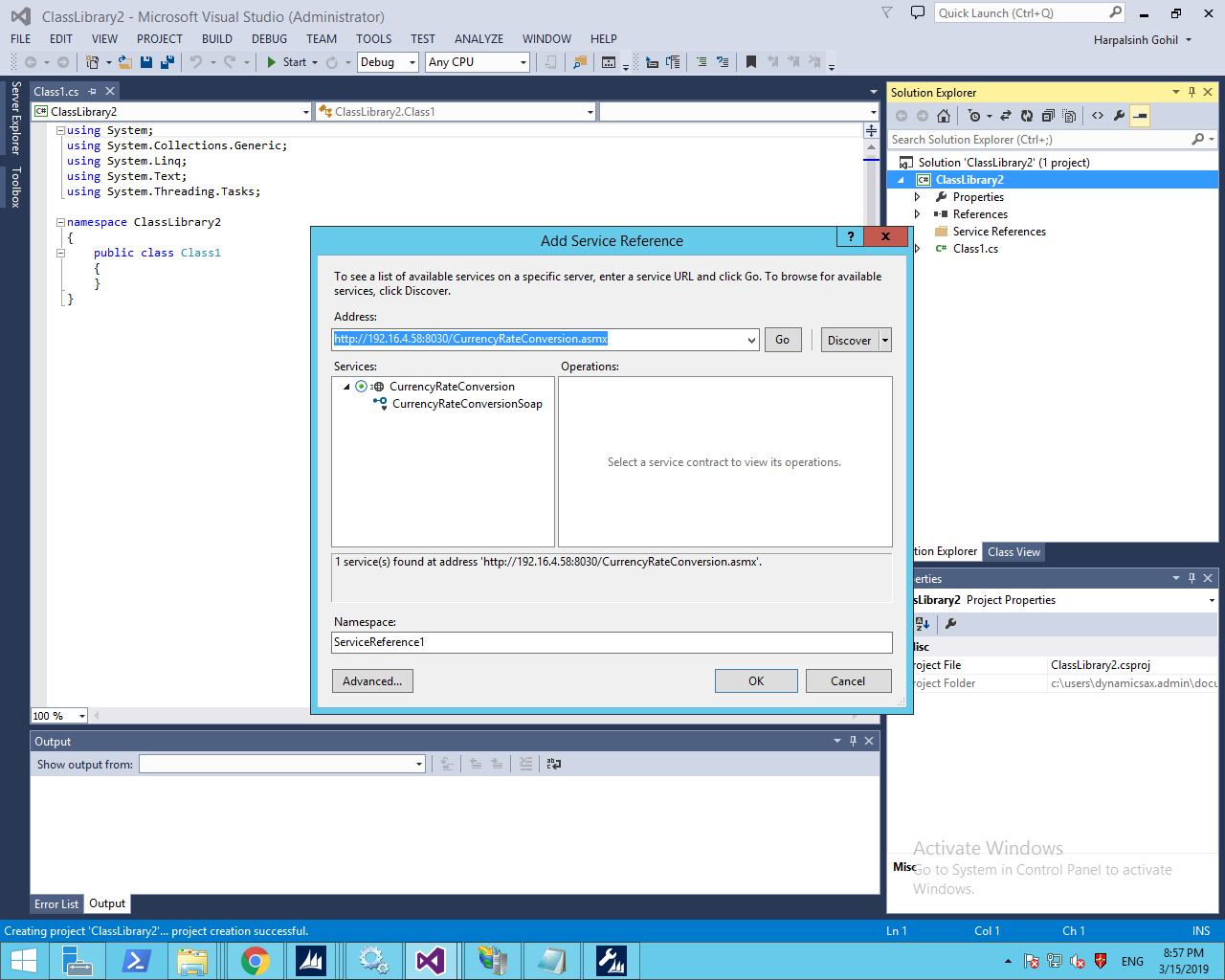


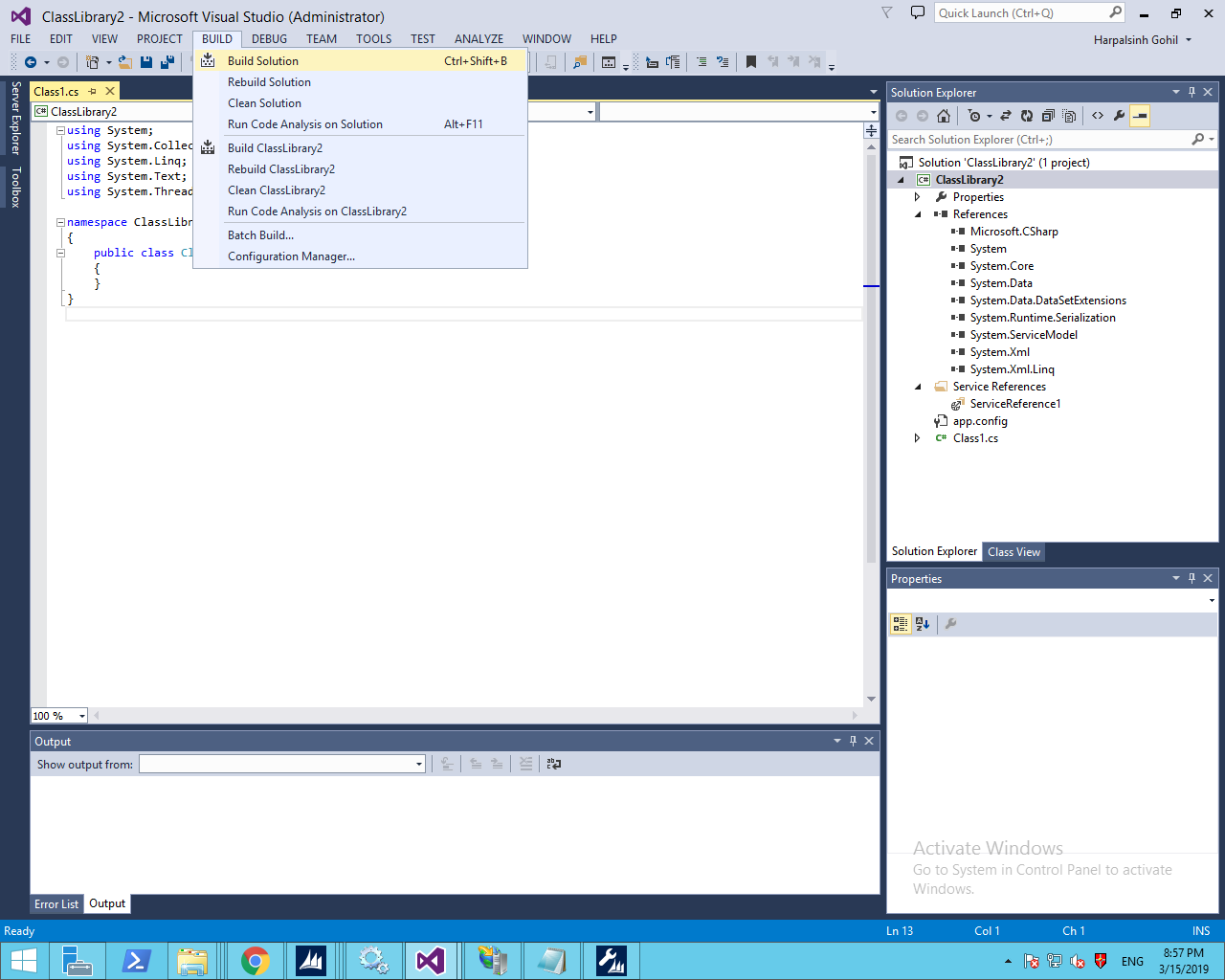
<http://192.16.4.58:8031/CurrencyRateConversion.asmx>

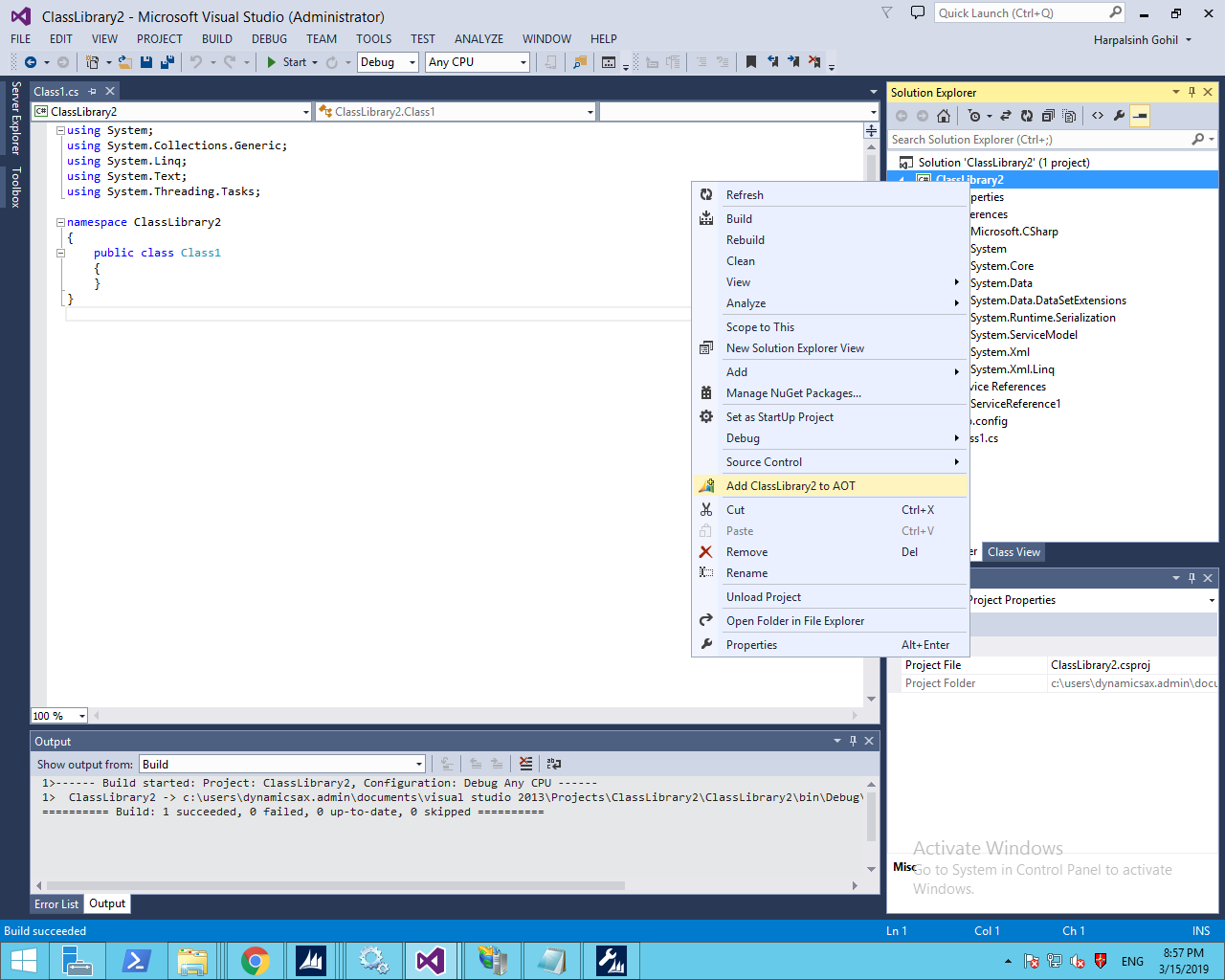
1. Create new project with class library and add your custom web service as a reference.

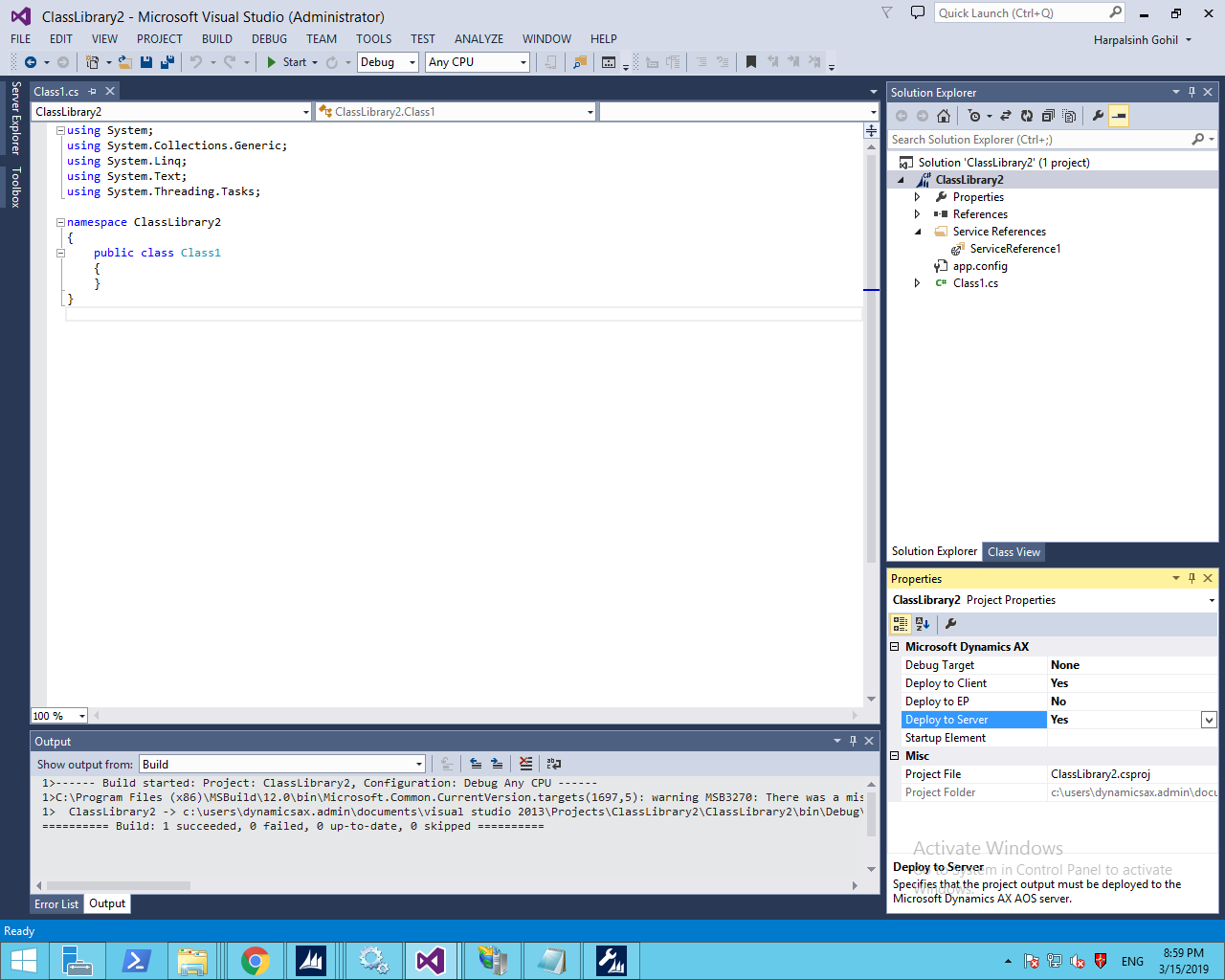


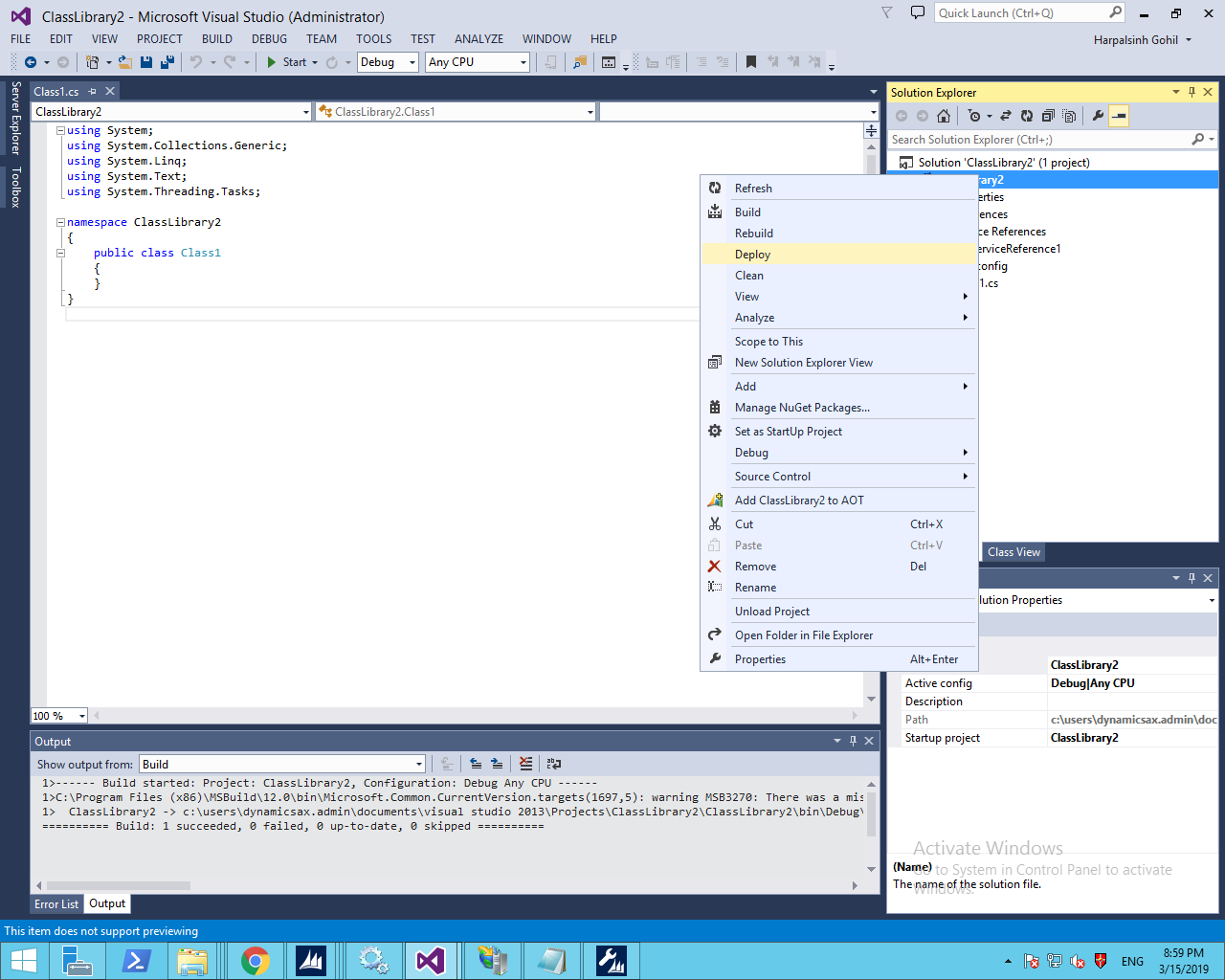




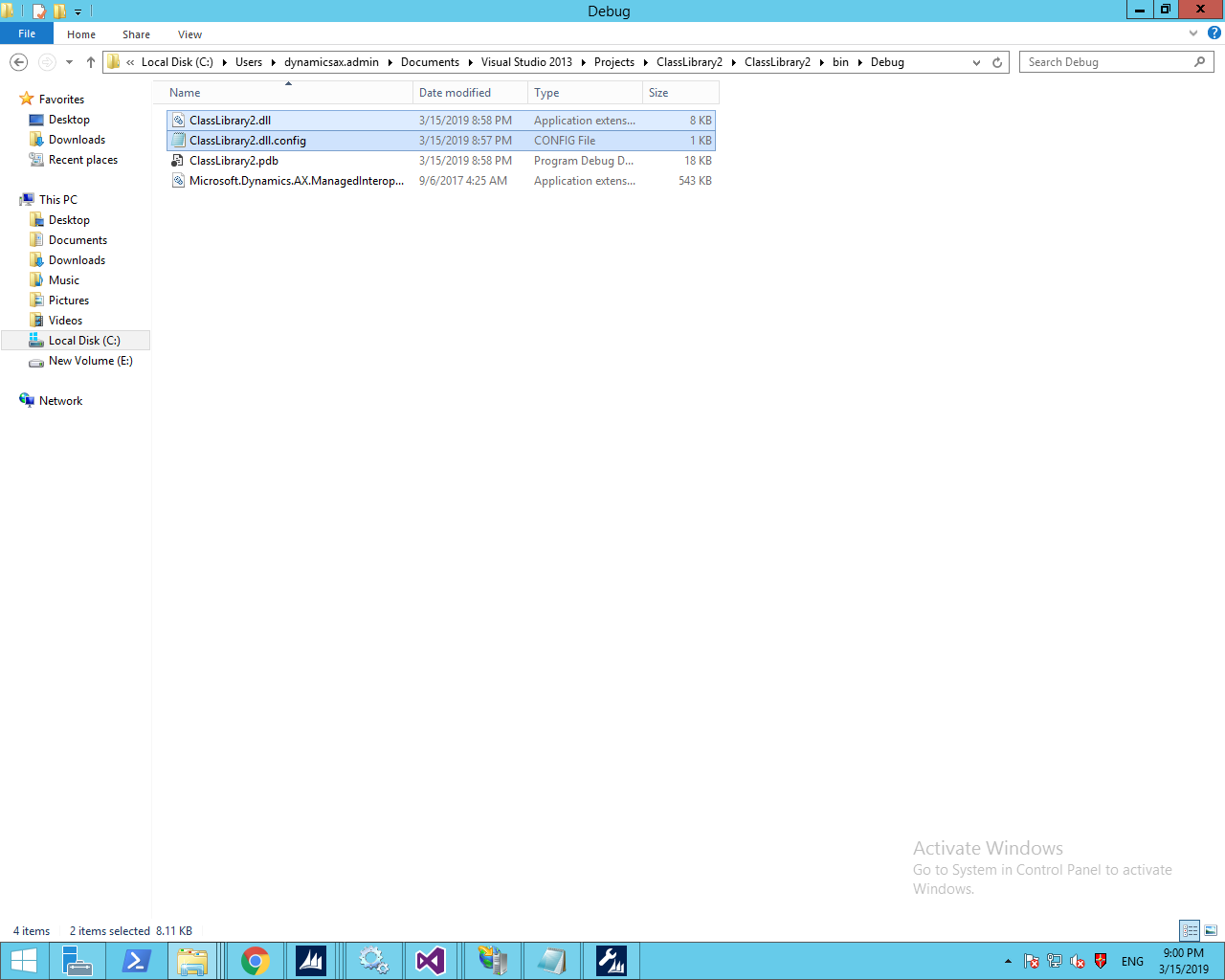






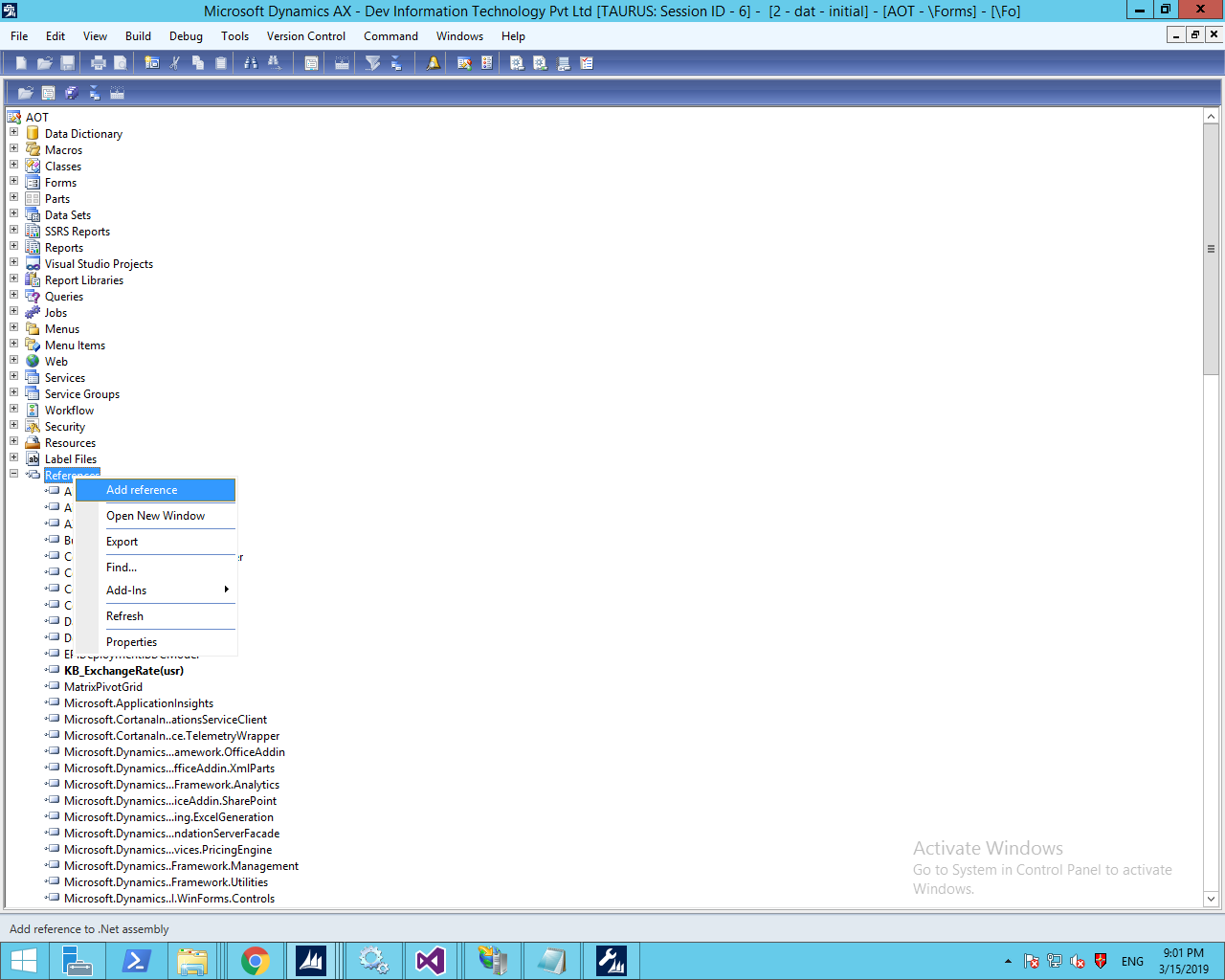


1. Copy the DLL and DLL.config files from the below mentioned path ,as per the below screen shot in AX client bin folder (C:\Program Files (x86)\Microsoft Dynamics AX\60\Client\Bin)

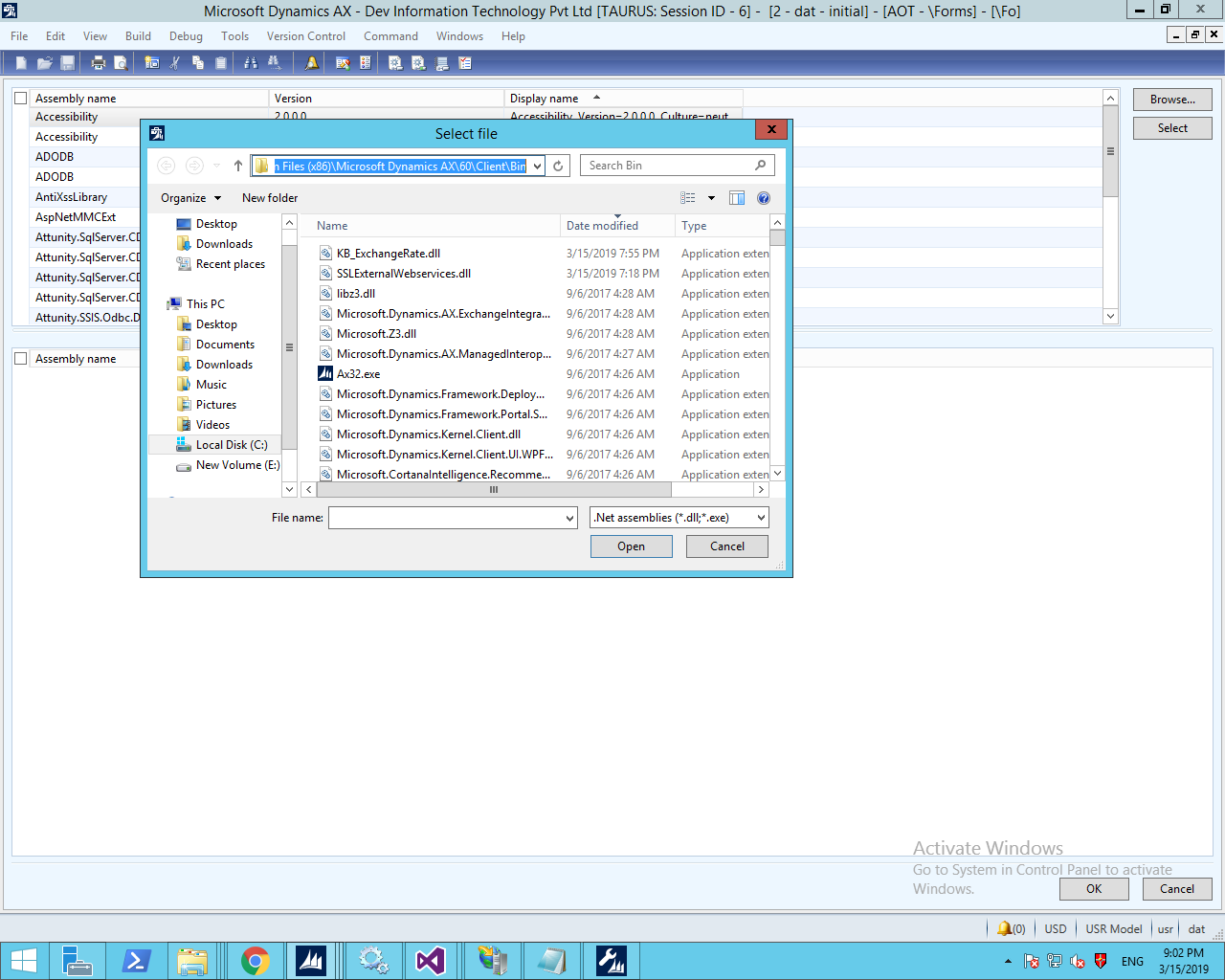


AX client path : C:\Program Files (x86)\Microsoft Dynamics AX\60\Client\Bin

1. Click on the “Add Reference”



1. Go to the bin folder where you copied the DLL files and select the file and click on the “OK” button.



1. Restart service
2. Create Form/Class/job and write the below code to call this web service :

ClassLibrary2.Servicereference1.CurrencyRateConversionSoapClient KowabungaService;

System.Exception ex;

CLRObject type;

**real** decimalval;

**str** **20** t1;

**new** InteropPermission(InteropKind::ClrInterop).assert();

decimalval = **0**;

**try**

{

type = CLRInterop::getType(ClassLibrary2.Servicereference1.CurrencyRateConversionSoapClient');

KowabungaService = AifUtil::createServiceClient(type);

t1 = KowabungaService.CurrencyRateConvert(strFromCurrency, strToCurrency, dtDate);

decimalval = **any2real**(t1);

CodeAccessPermission::revertAssert();

}

**catch**(Exception::CLRError)

{

ex = CLRInterop::getLastException();

info(CLRInterop::getAnyTypeForObject(ex.ToString()));

}

**return** decimalval;

<http://10.99.28.16:8030/CurrencyRateConversion.asmx/CurrencyRateConvert>

1. Without host web service on IIS .you will get the below error message in AX. So it is compulsory to host the webservice in IIS.

**Error** : System.Reflection.TargetInvocationException: Exception has been thrown by the target of an invocation. ---> System.ServiceModel.Security.MessageSecurityException: The HTTP request was forbidden with client authentication scheme 'Anonymous'. ---> System.Net.WebException: The remote server returned an error: (403) Forbidden.

at System.Net.HttpWebRequest.GetResponse()

**Resolution** : Host the web service on IIS.