

## Summary

This document lists the steps required to create a simple Windows Console Application that can transmit messages to the AIMS Platform Simple Web Service.

The AIMS Platform Simple Web Service accepts various types of data. This example shows how to send and XML formatted request to it. In all cases the web service will respond with a simple string response (see the Responses section).

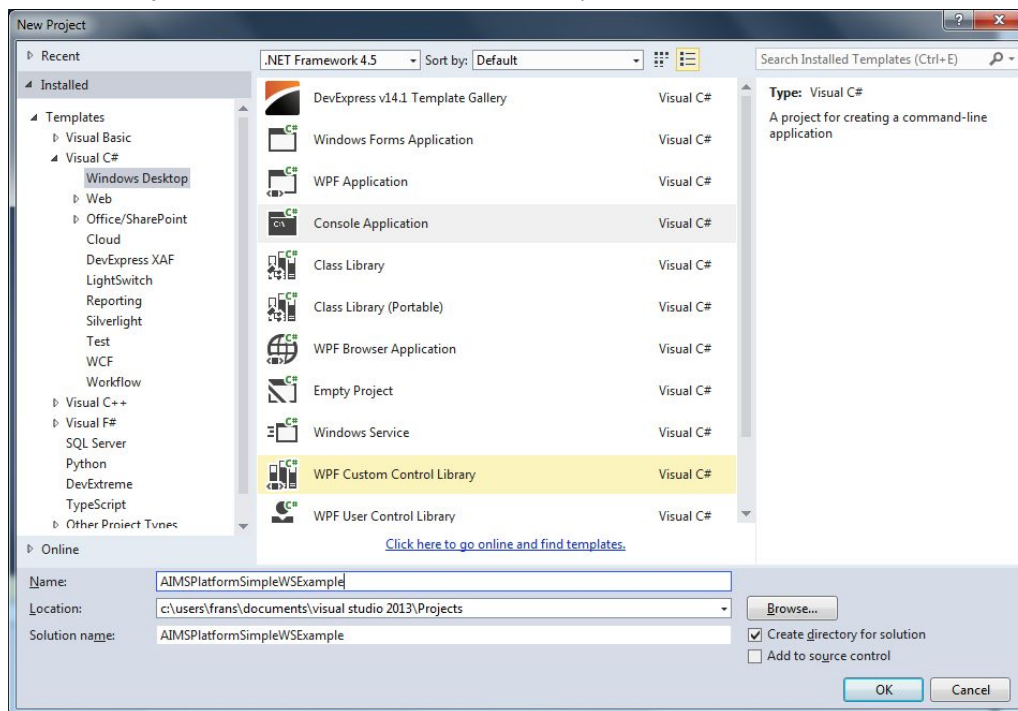
## Step 1 - Create a Windows Console Application

Open Visual Studio and choose to create a new project. The actual interface may vary slightly due to the specific version of Visual Studio you are using. Also remember to choose a .NET Framework appropriate to your environment.

This example creates a Visual C# project.

Choose the Windows Desktop group and the Console Application specifically.

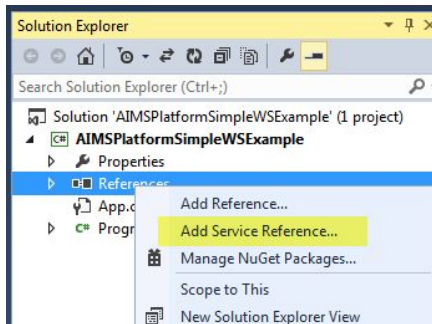
Give the Project and Solution a name that suits your needs.



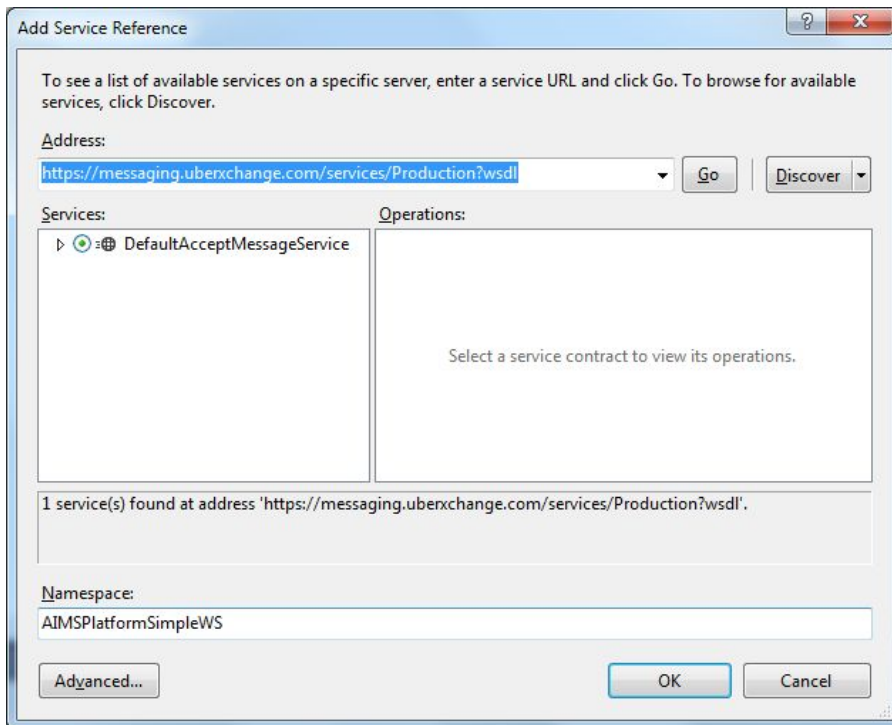
## Step 2 - Add a Service Reference

At this point the solution and project should be created. Dependent on your interface configuration you should have a view as shown below.

Choose to add a Service Reference.



Enter the URL (<https://messaging.uberexchange.com/services/Production?wsdl>) and click Go, then name it appropriately in the Namespace edit box and hit OK.



## Step 3 - Use the Service Reference in your code

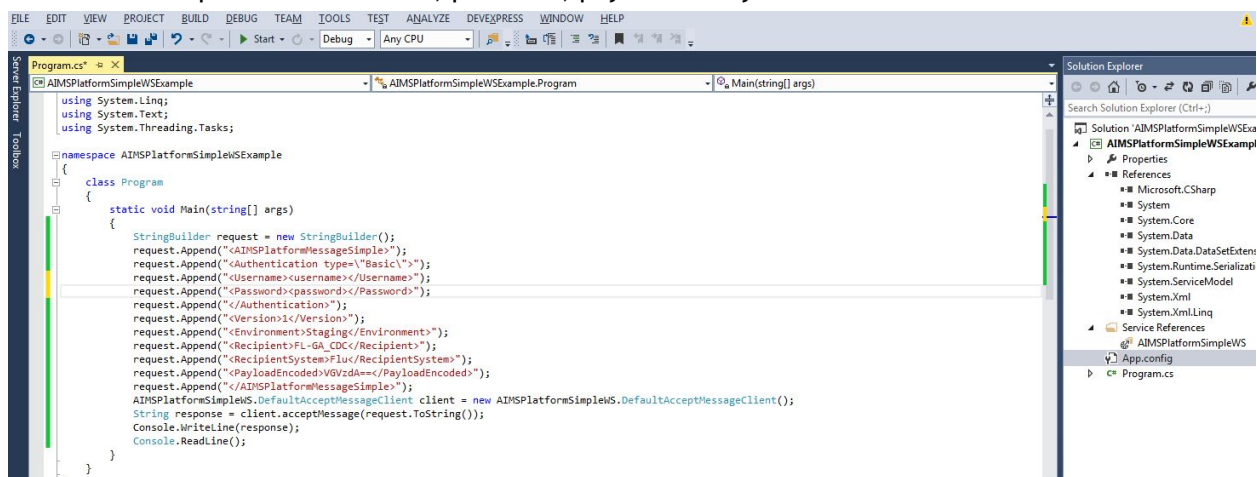
Enter the following code in your Main function:

```

StringBuilder request = new StringBuilder();
request.Append("<AIMSPlatformMessageSimple>");
request.Append("<Authentication type=\"Basic\">");
request.Append("<Username>username</Username>");
request.Append("<Password>password</Password>");
request.Append("</Authentication>");
request.Append("<Version>1</Version>");
request.Append("<Environment>Staging</Environment>");
request.Append("<Recipient>FL-GA_CDC</Recipient>");
request.Append("<RecipientSystem>Flu</RecipientSystem>");
String encodedPayload = System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetBytes("Hello World"));
request.Append("<PayloadEncoded>" + encodedPayload + "</PayloadEncoded>");
request.Append("</AIMSPlatformMessageSimple>");
AIMSPlatformSimpleWS.DefaultAcceptMessageClient client = new
AIMSPlatformSimpleWS.DefaultAcceptMessageClient();
String response = client.acceptMessage(request.ToString());
Console.WriteLine(response);
Console.ReadLine();

```

Remember to replace the username, password, payload with your actual information.



Go ahead and run the application. It should create a console window and successfully connect to the AIMS Platform Simple Web Service.

## Possible Responses

The AIMS Platform Simple Web Service will respond with one of two types of responses:

1. **QUEUED** - The message you sent was accepted and has been queued for the intended recipient.
2. **ERROR-<error codes>** - The message processing encountered some kind of error. The message could not be queued and will not be delivered to any recipient.
  - a. A sample error response is: `ERROR-InvalidAuthentication|InvalidVersion`.
  - b. The error codes will be separated by a pipe character (“|”). You will receive all the applies.
  - c. The possible error codes are:
    - i. **InvalidMessage** - The message could not be parsed, probably because it is not in a format that is expected.
    - ii. **InvalidAuthentication** - The authentication parameters you provided did not match what we have on file.
    - iii. **InvalidEnvironment** - You did not specify Production, Staging, or Development as the environment choice, or the environment has not been enabled for your use yet.
    - iv. **InvalidRecipient** - You did not specify a recipient that you allowed to send to.
    - v. **InvalidRecipientSystem** - You did not specify a recipient system that you are allowed to send to.
    - vi. **InvalidMessageVersion** - The Simple Web Service Message you sent did not have a version number that is understood by the system. You are currently instructed to use “1” as the value for the Version tag.
    - vii. **NotAuthorized** - You are not authorized to send a message. This may be due to an expired subscription, exceeding the agreed upon TOS, or other similar options.