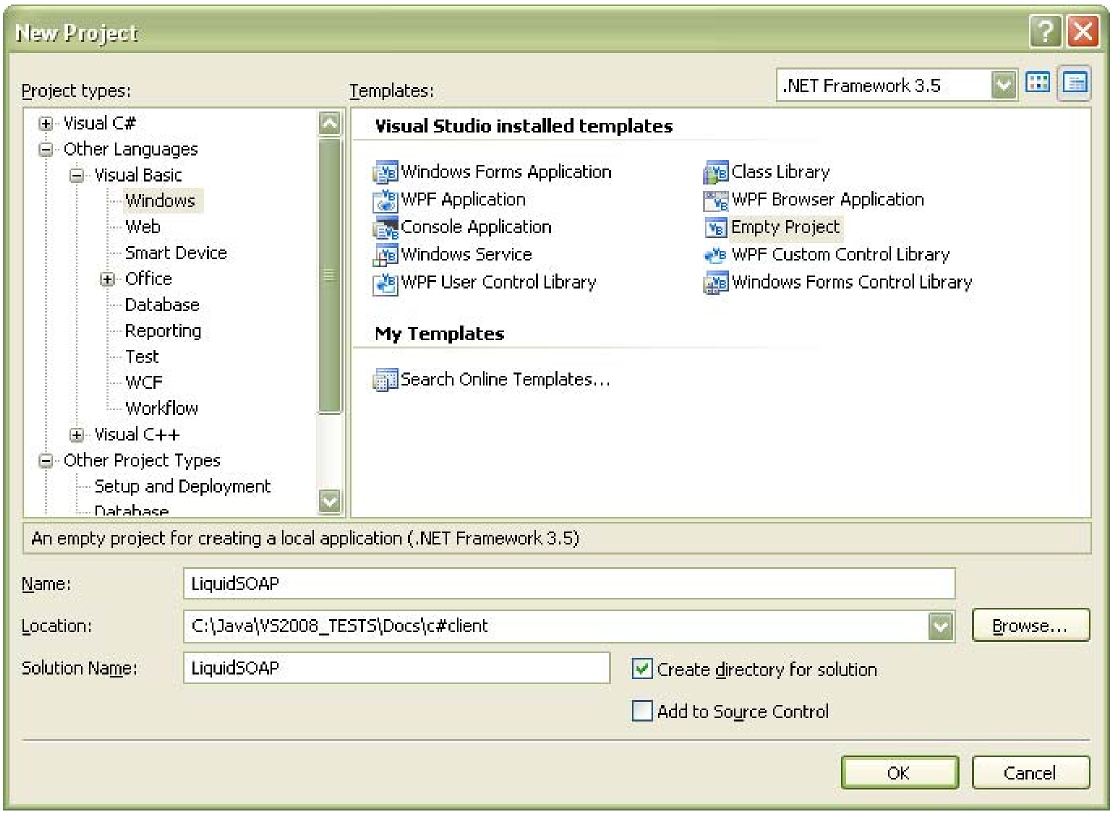
Implementing a SOAP Client in VB using Visual Studio 2008

Create a new project

In Visual studio 2008, select File → New. In the new projects dialog navigate to VB, then under windows select Empty Project as shown below: (figure 1)



An empty project for creating a local application (.NET Framework 3.5)

Name: LiquidSOAP

Location: C:\Java\VS2008_TESTS\Docs\c#client

Solution Name: LiquidSOAP

☑ Create directory for solution

□ Add to Source Control

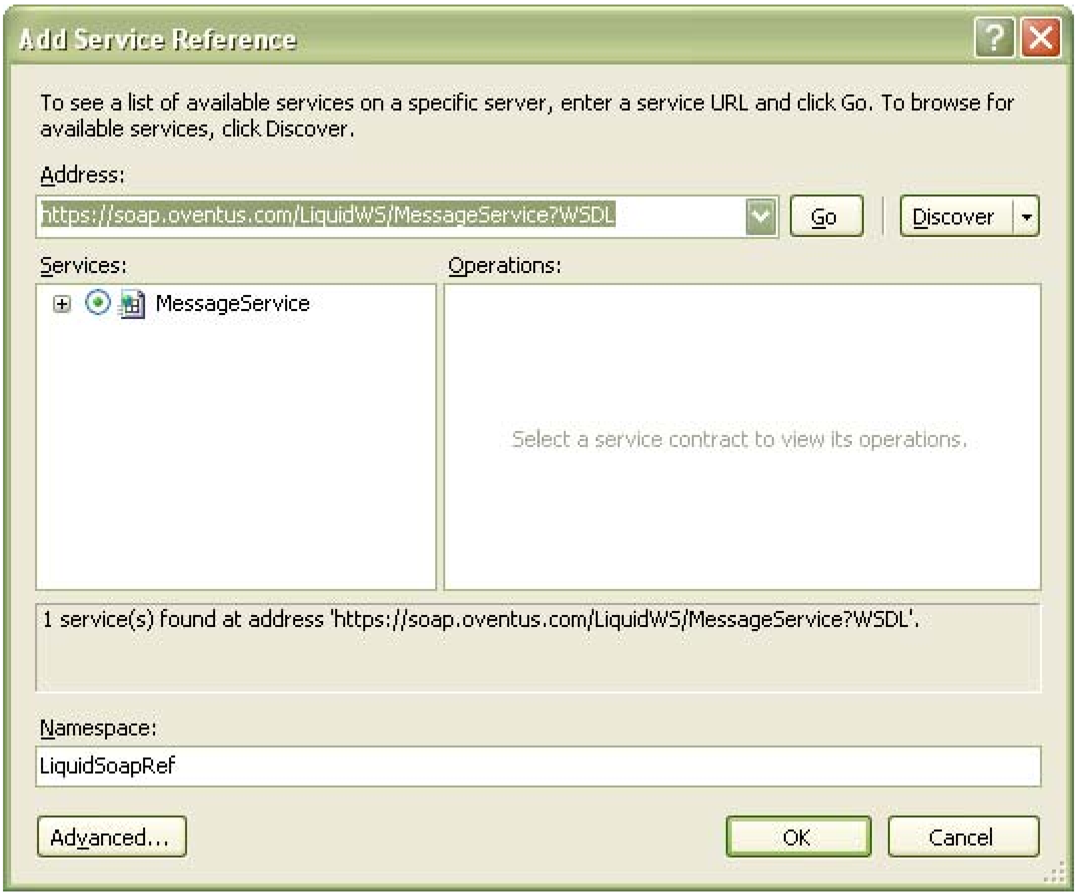
Figure 1

Adding a Service reference

On the solutions explorer, right click on Project name and select Add Service Reference.

On the Service reference dialog, give the following WSDL as the address.(figure 2)

https://soap.oventus.com/LiquidWS/MessageService?WSDL



To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

Address: https://soap.oventus.com/LiquidWS/MessageService?WSDL

1 service(s) found at address 'https://soap.oventus.com/LiquidWS/MessageService?WSDL'.

Namespace: LiquidSoapRef

Figure 2

Click on OK. This will generate the required Stubs .

Implement the clients

Right click on project in solutions explorer and add new item (class) LiquidSOAPImpl.cs

Example- implement login method

```
Private messageServicePortType As MessageServicePortTypeClient = New
MessageServicePortTypeClient

Private p1Header As pageoneHeader = Nothing
Dim loginReq As New loginRequest()

'PLEASE ENTER A VALID USERNAME AND PASSWORD HERE
loginReq.userid = "username"
loginReq.pwd = "password"

Dim loginResponse As New loginResponse()
p1Header = messageServicePortType.login(loginReq, loginResponse)

Dim sessionID As [String] = p1Header.sessionid
Dim statusType As statusType = loginResponse.status
```

Please refer to Appendix A for the full implementation of this class.

Appendix A

```
Imports System
Imports System.Collections.Generic
Imports System.Text
Imports LiquidSOAP.LiquidSoapRef



Namespace LiquidSOAP


    Class LiquidSOAPImpl
        Private messageServicePortType As
MessageServicePortTypeClient = New MessageServicePortTypeClient

        Private p1Header As pageoneHeader = Nothing

        Public Function getSessionID() As pageoneHeader
            Try
                If p1Header Is Nothing Then
                    Dim loginReq As New loginRequest()

                    'PLEASE ENTER A VALID USERNAME AND PASSWORD HERE
                    loginReq.userid = "username"
                    loginReq.pwd = "password"

                    Dim loginResponse As New loginResponse()
                    p1Header = messageServicePortType.login(loginReq,
loginResponse)

                    Dim sessionID As [String] = p1Header.sessionid

                    Dim statusType As statusType =
loginResponse.status
                    Console.WriteLine(" Status Description is >>: " &
statusType.description)
                    Console.WriteLine(" Session ID is >>: " &
sessionID)
                End If
                Return p1Header
            Catch ex As Exception
                Console.WriteLine(vbLf & vbLf & vbLf & vbLf & "Please
check if you have entered a valid username and password in
execute.cs")
                Return Nothing

            End Try
        End Function

        Public Sub logOff()
            Dim logOffReq As New logoffRequest()
            Dim logOffResp As logoffResponse =
messageServicePortType.logoff(Me.getSessionID(), logOffReq)
            Console.WriteLine("Logoff result " & logOffResp.result)
            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub ping()
```

```
            Dim pingReq As New pingRequest()
            Dim pingResp As pingResponse =
messageServicePortType.ping(Me.getSessionID(), pingReq)
            Console.WriteLine("Ping result " & pingResp.description &
" value " & pingResp.Value)
            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub isActive()
            Dim isActiveReq As New isActiveRequest()
            isActiveReq.userid = "username"
            Dim isActiveResp As isActiveResponse =
messageServicePortType.isActive(isActiveReq)

            Console.WriteLine("Is Active result " &
isActiveResp.description & " value " & isActiveResp.Value)
            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub getReceipt()
            Dim getReceiptReq As New getReceiptRequest()
            Dim receiptType As receiptType =
messageServicePortType.getReceipt(Me.getSessionID(), getReceiptReq)

            Console.WriteLine("Destination " &
receiptType.destination)
            Console.WriteLine("resultCode " & receiptType.resultCode)
            Console.WriteLine("source " & receiptType.source)
            Console.WriteLine("sentAt " &
receiptType.sentAt.ToString())
            Console.WriteLine(vbLf & vbLf & "press enter to end")
            Console.Read()
        End Sub


        Public Sub getReceipts()
            Dim getReceiptsReq As New getReceiptsRequest()
            Dim receiptType As receiptType() =
messageServicePortType.getReceipts(Me.getSessionID(), getReceiptsReq)

            For i As Integer = 0 To receiptType.Length - 1
                Console.WriteLine("Destination " &
receiptType(i).destination)
                Console.WriteLine("resultCode " &
receiptType(i).resultCode)
                Console.WriteLine("source " & receiptType(i).source)
                Console.WriteLine("sentAt " &
receiptType(i).sentAt.ToString())
                Console.WriteLine("=====")
            Next

            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub getInbound()
```

```
            Dim getInboundReq As New getInboundRequest()
            Dim inboundType As inboundType =
messageServicePortType.getInbound(Me.getSessionID(), getInboundReq)

            Console.WriteLine(" destination " &
inboundType.destination)
            Console.WriteLine(" message " & inboundType.message)
            Console.WriteLine(" messageID " & inboundType.messageID)
            Console.WriteLine(" source " & inboundType.source)
            Console.WriteLine(" timeReceived " &
inboundType.timeReceived.ToString())

            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub getInbounds()
            Dim getInboundsReq As New getInboundsRequest()

            Dim inboundType As inboundType() =
messageServicePortType.getInbounds(Me.getSessionID(), getInboundsReq)
            For i As Integer = 0 To inboundType.Length - 1
                Console.WriteLine("destination " &
inboundType(i).destination)
                Console.WriteLine("message " &
inboundType(i).message)
                Console.WriteLine("messageID" &
inboundType(i).messageID)
                Console.WriteLine("source " & inboundType(i).source)
                Console.WriteLine("time received " &
inboundType(i).timeReceived.ToString())
                Console.WriteLine("====")
            Next

            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub getMsisdn()
            Dim getMsisdnReq As New getMsisdnRequest()
            Dim msisdns As String() =
messageServicePortType.getMsisdn(Me.getSessionID(), getMsisdnReq)
            For i As Integer = 0 To msisdns.Length - 1
                Console.WriteLine("Msiddn >> " & msisdns(i))
            Next
            Console.WriteLine(vbLf & vbLf & "press enter to end")
            Console.Read()
        End Sub


        Public Sub sendMessage()
            Dim addrArr As String() = {"mobile number here"}
            Dim sendMsgType As New sendMessageType()
            sendMsgType.message = "Hello"
            sendMsgType.destinationAddress = addrArr
            Dim sendMsgRespType As sendMessageResponseType =
messageServicePortType.sendMessage(Me.getSessionID(), sendMsgType)
            Console.WriteLine("SendMessage Response status " &
sendMsgRespType.status.description)
```

```
            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub sendMultiMessage()
            Dim sendMultiReqType As sendMultiMessageType() = New
sendMultiMessageType(1) {}

            Dim sendType1 As New sendMultiMessageType()

            Dim addrArr1 As String() = {"mobile number here"}
            sendType1.destinationAddress = addrArr1
            sendType1.flashSMS = False
            sendType1.message = "first message"
            sendType1.requestID = 1


            Dim sendType2 As New sendMultiMessageType()
            Dim addrArr2 As String() = {"mobile number here"}
            sendType2.destinationAddress = addrArr2
            sendType2.flashSMS = True
            sendType2.message = "second message"
            sendType2.requestID = 2

            sendMultiReqType(0) = sendType1
            sendMultiReqType(1) = sendType2


            Dim multiResp As
sendMultiMessageResponseMessagePacketResponse() =
messageServicePortType.sendMultiMessage(Me.getSessionID(),
sendMultiReqType)

            For i As Integer = 0 To multiResp.Length - 1
                Console.WriteLine("description " &
multiResp(i).status.description)
                Console.WriteLine("value " &
multiResp(i).status.Value)
                Console.WriteLine("transactionID " &
multiResp(i).transactionID)
                Console.WriteLine("requestID" &
multiResp(i).requestID)
            Next

            Console.WriteLine(vbLf & vbLf & "press enter to end")


            Console.Read()
        End Sub

        Public Sub pushWapRequest()
            Dim pushWapType As New pushWapType()
            'pushWapType.sourceAddress ="";

            Dim addrArr As String() = {"mobile number here"}
            pushWapType.destinationAddress = addrArr
            pushWapType.title = "Test"
            pushWapType.url = "http://www.google.com"
            'pushWapType.expireAt
            Dim sendMessageResponseType As sendMessageResponseType =
messageServicePortType.pushWap(Me.getSessionID(), pushWapType)

            Console.WriteLine(" description " &
sendMessageResponseType.status.description)
            Console.WriteLine(" Value " &
sendMessageResponseType.status.Value)
            Console.WriteLine(" transactionID " &
sendMessageResponseType.transactionID)

            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub pushMultiWapRequest()
            Dim pushMltiWapType As pushMultiWapType() = New
pushMultiWapType(1) {}

            Dim pushMultiType1 As New pushMultiWapType()

            Dim addrArr1 As String() = {"mobile number here"}
            pushMultiType1.destinationAddress = addrArr1
            pushMultiType1.title = "Test"
            pushMultiType1.url = "http://www.google.com"
            pushMultiType1.requestID = 1
            pushMultiType1.requestIDSpecified = True

            'pushMultiType1.expireAt

            Dim pushMultiType2 As New pushMultiWapType()
            Dim addrArr2 As String() = {"07799542451"}
            pushMultiType2.destinationAddress = addrArr2
            pushMultiType2.title = "Test"
            pushMultiType2.url = "http://www.google.com"
            pushMultiType2.requestID = 2
            pushMultiType2.requestIDSpecified = True
            'pushMultiType2.expireAt

            pushMltiWapType(0) = pushMultiType1
            pushMltiWapType(1) = pushMultiType2

            Dim resp As pushMultiWapResponseMessagePacketResponse() =
messageServicePortType.pushMultiWap(Me.getSessionID(),
pushMltiWapType)

            For i As Integer = 0 To resp.Length - 1
                Console.WriteLine("requestID " & resp(i).requestID)
                Console.WriteLine("transactionID " &
resp(i).transactionID)
                Console.WriteLine("description " &
resp(i).status.description)
                Console.WriteLine("value " & resp(i).status.Value)
            Next

            Console.WriteLine(vbLf & vbLf & "press enter to end")

            Console.Read()
        End Sub

        Public Sub getCredits()
```

```
            Try
                Dim getCreditsReq As New getCreditsRequest()
                Dim getCreditsResp As getCreditsResponse =
messageServicePortType.getCredits(Me.getSessionID(), getCreditsReq)
                Console.WriteLine("Credits Remaining " &
getCreditsResp.creditsRemaining)
                Console.WriteLine(vbLf & vbLf & "press enter to end")
                Console.Read()
            Catch ex As Exception
                Console.WriteLine(vbLf & vbLf & "Exception caught
press enter to end" & vbLf)
                Console.Read()

            End Try
        End Sub


    End Class
End Namespace
```