

VADIAN.NET AG

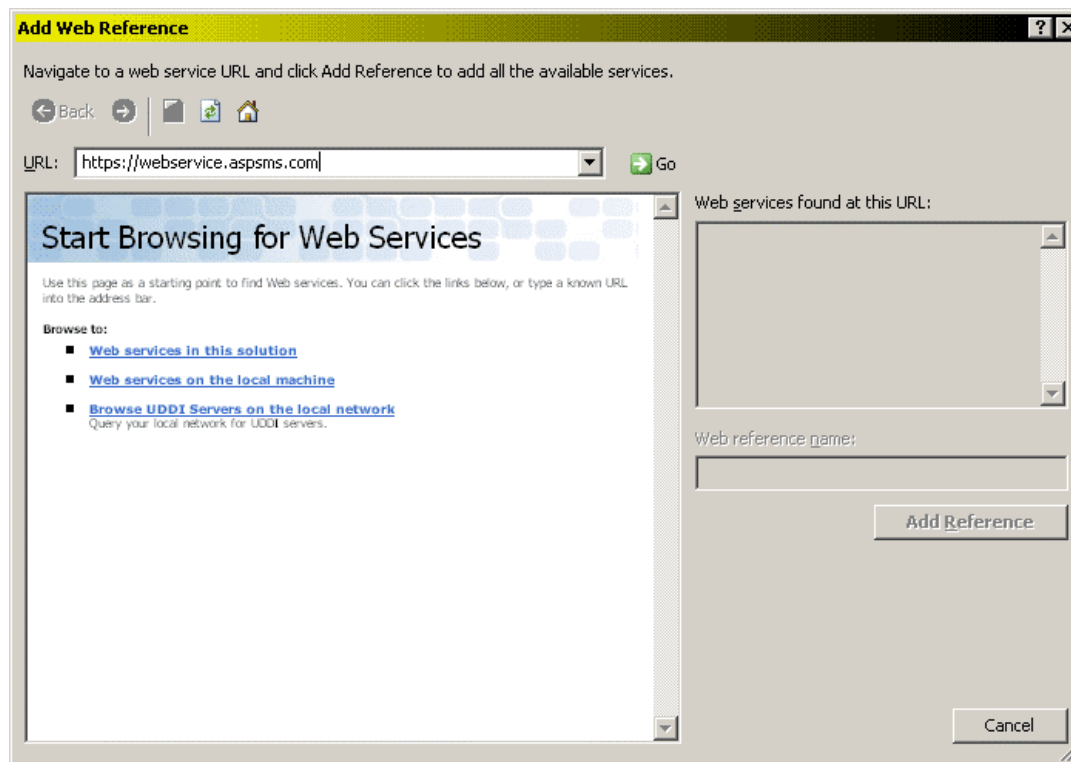
# ASPSMS.COM SOAP WEB SERVICE

USER GUIDE / USAGE NOTES

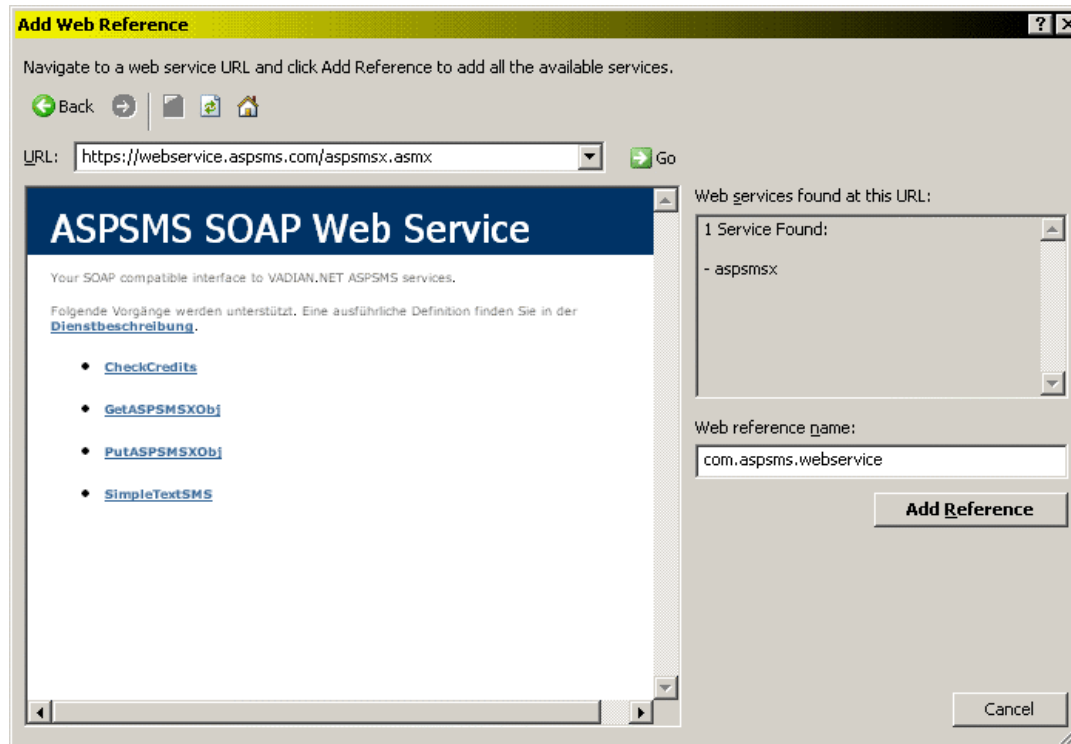
[www.aspsms.com](http://www.aspsms.com)

## Connecting to webservice.aspsms.com (VB.NET 2005):

- 1) In VB.NET 2005 IDE, choose the *Project* Menu, and select *Add Web Reference...* from there.
- 2) In the Add Web Reference dialog, enter the webservice's URL and press enter. The address of the aspsms.com webservice is: <https://webservice.aspsms.com/aspsmsx.asmx>. Notice that you can use the http:// variant as well, if your development environment does not support HTTPS webservices. However, the security benefits of HTTPS will not be available when you are using the http:// variant.



3) Upon successful connection, the Add Web Reference dialog will look like this:



At this point, click the *Add Reference* Button to add the aspsms.com webservice to your project. You are now ready to use the webservice within your code.

## Basic Usage (VB.NET 2005):

```
Private Sub ExecuteOP()  
    ' create web service instance  
    Dim ASPSMSX As com.aspsms.webservice.ASPSMSSOAPWebService = New com.aspsms.webservice.ASPSMSSOAPWebService  
    ' create SMSX object instance and initialize subobjects  
    Dim SMSXObj As New com.aspsms.webservice.cASPSMSX  
    SMSXObj.MessageWaitingIndicationSettings = New com.aspsms.webservice.cMessageWaitingIndicationSettings  
    SMSXObj.NotificationInquirySettings = New com.aspsms.webservice.cNotificationInquirySettings  
    SMSXObj.VCard = New com.aspsms.webservice.cVCard  
    SMSXObj.WAPPushSettings = New com.aspsms.webservice.cWAPPushSettings  
    ' configure request  
    With SMSXObj  
        ' apply user specific settings  
        .UserKey = ASPSMSUserKey           ' example: "12345"  
        .Password = ASPSMSPassword        ' example: "abcde"  
        .Originator = Originator          ' example: "mysite.com"  
        .Recipients = RecipientsAndTRNList* ' example: "0791233221;0795554321"  
        ' set operation  
        .Operation = DesiredOperation     ' example: OP_TextSMS  
    End With  
    ' call PutASPSMSXObj method and retrieve result  
    Dim OpResult As com.aspsms.webservice.cResult  
    OpResult = ASPSMSX.PutASPSMSXObj(SMSXObj)  
    ' display errorcode/description and additional data  
    Debug.Print("ErrorCode: " & OpResult.ErrorCode)  
    Debug.Print("ErrorDescription: " & OpResult.ErrorDescription)  
    Debug.Print("Additional Data:" & OpResult.AdditionalData)  
    ' dispose service instance  
    ASPSMSX.Dispose()  
End Sub
```

\*see Usage Notes at the end of this document

## Supported Operations:

### **OP\_BinaryData**

Send Binary Data. Data must be supplied in HEX-encoded string format to the *MessageData* property.

### **OP\_CheckOriginatorAuthorization**

Verifies if a specific numeric originator is authorized for your account.

### **OP\_GroupLogo**

Sends a Group-Logo.

### **OP\_InquireDeliveryNotifications**

Inquires the status of sent messages. The Result is returned in the *AdditionalData* Property of the *cResult* object returned by the *PutASPSMSXObj* Method of the webservice.

### **OP\_Logo**

Sends an Operator-Logo.

### **OP\_MessageWaitingIndication**

Sends a Message Waiting Indication Message. Note: This feature is reserved for special interest groups and not available publicly. Please contact [www.aspsms.com](http://www.aspsms.com) for further assistance.

### **OP\_PictureMessage**

Sends a Picture-Message.

### **OP\_RandomLogo**

Sends a Random-OperatorLogo out of a collection from several hundred logos.

### **OP\_Ringtone**

Sends a Ringtone.

**OP\_SendOriginatorUnlockCode**

The ASPSMS System creates a random OriginatorUnlockCode and sends it to the specified originator to be unlocked.

**OP\_TextSMS**

Sends a Text-Message. If using messages with overlength, each part will be reduced from 160 to 153 chars, because additional information is required for concatenation handling. The message text may be up to 9 \* 153 chars in length.

**OP\_UnlockOriginator**

Verifies an entered UnlockCode previously sent to a specific Originator. If the UnlockCode matches, the Originator is unlocked.

**OP\_Vcard**

Sends a VCard

## **Properties:**

### **Userkey**

Your ASPSMS.COM userkey.

### **Password**

Your ASPSMS.COM password.

### **Originator**

The visible Originator of the SMS. Can be alphanumeric or numeric. Numeric Originators must be unlocked for use with ASPSMS.COM first (see OriginatorUnlockCode).

### **OriginatorUnlockCode**

The unlock code to use when unlocking an originator.

### **DeferredDeliveryTime**

Specifies the time when the message should be transmitted to the mobile network in the format "ddmmyyyhhmmss". If you reside outside the GMT+1 timezone, you should also set a value for the TimeZone property.

### **LifeTime**

If an SMS can not be delivered instantly, it remains within the GSM network for a given time. During this timeframe, the network tries to deliver the SMS periodically. The LifeTime property represents this validity period of an SMS in Minutes. Please note that the LifeTime can not be shorter than 3 minutes and can not exceed 1440 minutes (24h). If not specified, the LifeTime is 24h by default.

### **MessageData**

Message that will be sent. Can be supplied in plain ASCII text format, or as a hex-encoded binary string.

### **FlashingSMS**

If set to "True", a text-message will displayed directly in the display of the recipient.

**BlinkingSMS**

If set to "True", the text-message or parts of it will blink, when the message is displayed on the mobilephone. If used, the total number of used characters is limited to 69 characters. Note: At present, this feature is supported exclusively on Nokia phones Nokia 3210, 3310, 5110, 6110, 6150, 8210, 8810, 8850. With the command<BLINK>, used in the messagetext, it's possible to control the appearance of the blinking the text. This feature works in conjunction with the methods "SendTextSMS", "SendVCard" and "SendMessageWaitingIndication".

**ReplaceMessage**

If set to a value ranging from 1 to 7, this allows to overwrite existing messages on a handset with the same ReplaceMessage value.

**MCC**

Mobile-Country-Code, should be specified when sending operator-logos.

**MNC**

Mobile-Network-Code, should be specified when sending operator-logos.

**Xser**

Additional information for binary messages. Only used when sending directly to the SMSC gateway. Check UCP specification for additional information.

**URLBufferedMessageNotification**

URL that will be connected when a message is not delivered instantly and is buffered. The value of the submitted TransactionReferenceNumber will be the argument of the URL.

**URLDeliveryNotification**

URL that will be connected when a message is delivered instantly. The value of the submitted TransactionReferenceNumber will be the argument of the URL.

**URLNonDeliveryNotification**

URL that will be connected when a message is not delivered. The value of the submitted TransactionReferenceNumber will be the argument of the URL.



**TimeZone**

Specifies the Timezone where the component is used. It's only necessary to set this property when the property DeferredDeliveryTime is used and the component is used outside the timezone GMT +1.

**Timeout**

Specifies the amount of seconds after which the ASPSMS webservice will give up trying to perform a pending operation.

**NotificationInquirySettings.Format**

Specifies the format of the returned result of the method InquireDeliveryNotifications. See Format Notes below.

**NotificationInquirySettings.Separator**

Specifies a user defined separator which will be part of the result of the method InquireDeliveryNotifications. Default value is the Tab char (ASCII code 9).

**Vcard.Name**

Specifies the Name that will show up in a phonebook entry.

**Vcard.PhoneNumber**

Specifies the Phonenummer that will used in a phonebook entry.

**MessageWaitingIndicationSettings.VoiceIndication**

Specifies if a mobilephone will be notified about pending voicemessages.

**MessageWaitingIndicationSettings.CountVoiceMessages**

Specifies the number of pending voicemessages. Use the value "0" to delete the indicationsymbol on a mobilephone.

**MessageWaitingIndicationSettings.FaxIndication**

Specifies if a mobilephone will be notified about pending faxmessages.

**MessageWaitingIndicationSettings.CountFaxMessages**

Specifies the number of pending faxmessages. Use the value "0" to delete the indicationsymbol on a mobilephone.

**MessageWaitingIndicationSettings.EmailIndication**

Specifies if a mobilephone will be notified about pending Emailmessages.

**MessageWaitingIndicationSettings.CountEMailMessages**

Specifies the number of pending EMailmessages. Use the value "0" to delete the indicationsymbol on a mobilephone.

**MessageWaitingIndicationSettings.OtherIndication**

Specifies if a mobilephone will be notified about other pending messages.

**MessageWaitingIndicationSettings.CountOtherMessages**

Specifies the number of other pending messages. Use the value "0" to delete the indicationsymbol on a mobilephone.

**MessageWaitingIndicationSettings.StoreMessage**

Specifies, if a sent Message Waiting Indication Message can be stored in the mobilephone or not.

## Check credits (VB.NET 2005)

```
Private Sub CheckAcctCredits()  
    ' create webservice instance  
    Dim ASPSMSX As com.aspsms.webservice.ASPSMSSOAPWebService = New com.aspsms.webservice.ASPSMSSOAPWebService  
    Dim Result As Single  
    ' invoke CheckCredits method  
    Result = ASPSMSX.CheckCredits(ASPSMSUserKey, ASPSMSPassword)  
    ' display result  
    If Result > 0 Then  
        ' credits available  
        MsgBox("You have " & Result & " credits left on your ASPSMS.COM account", MsgBoxStyle.Information)  
    Else  
        ' invalid credentials or no credits left  
        MsgBox("Account doesn't exist, password is invalid, or no credits left.", MsgBoxStyle.Exclamation)  
    End If  
    ' destroy webservice instance  
    ASPSMSX.Dispose()  
End Sub
```

## Shortcut: Send Text SMS (VB.NET 2005)

```
Private Sub SendMessage()  
    ' create webservice instance  
    Dim ASPSMSX As com.aspsms.webservice.ASPSMSSOAPWebService = New com.aspsms.webservice.ASPSMSSOAPWebService  
    ' invoke SimpleTextSMS method  
    If ASPSMSX.SimpleTextSMS(ASPSMSUserKey, ASPSMSPassword, Recipient, Originator, MessageText) Then  
        ' returned true; call was successful  
        MsgBox("SMS successfully sent.", MsgBoxStyle.Information)  
    Else  
        ' returned false; some error occurred  
        MsgBox("The SMS could not be sent. Make sure the supplied credentials are valid.", MsgBoxStyle.Exclamation)  
    End If  
    ' destroy webservice instance  
    ASPSMSX.Dispose()  
End Sub
```

## Usage Notes on Recipients and TRNsForInquiry Lists:

For interoperability reasons, these lists are supplied in a simple string format. The delimiter character is ";" for list entries, and "," (both used without quotes!) for fields of a particular entry. See examples below:

### Recipients List Format (String)

*SubscriberNumber[,TransactionReferenceNumber];...*

Example A (3 Subscribers, 3 TRNs):

0041791234567,1234;0041791234568,1235;0049791234569,1236

Example B (2 Subscribers, 1 TRN):

004179123456;004179123457,1234

### TRNsForInquiry List Format (String)

*TRN;...*

Example A (3 TRNs):

1234;1235;1236