Simple Solutions Making a Big Difference

# The Developers Guide to SMS Integration



**Document Version: 1.5**

| |
|---|
| *Updated on 05 July 2004 (AT)*<br>Added more information regarding Notification formats and rules.<br>Added C# example code |
| *Updated on 22 September 2003 (AT)*<br>Added <messages> tag to XML send message format |
| *Updated on 13th May 2003 (AT):*<br>Removed validity tag. Updated sender. email tag description |
| *Updated on 24th February 2002(AT) :*<br>Formatting |
| *Updated on 14th August 2002(AT):*<br>Formatting |

## How it works:

The following is a complete guide to enable you to connect to the Boom-SMS System (Business to business / consumer Mobile Phone Text messaging system)

The end product will result in you configuring your software to utilise Boom-SMS for all SMS text messaging.

## What Next:

Ensure you have read the 'Terms and Conditions' below. Then go to www.boom-SMS.co.uk and register.

Once registered you will have 10 free test SMS allocated to your account. You will receive an email confirming your account details.

Should you require more test SMS contact info@boom-sms.co.uk

You will then need to integrate your software to use the Boom-SMS system. (Technical details below)

Once integration is complete and successfully tested, send out updates of your software throughout your company & or Clients c/w a sort paragraph as to how the system works.

Your company will then need to credit the Boom-SMS system using the 'Wordplay Secure Payment Server'

To do this the simply go to www.boom-sms.co.uk / login / 'Account Top-up Service' and choose the required amount of SMS.

Your Clients will need to 'Register' their own account then use the 'Account Top-up Service' in the same way. Once they have entered their Boom-SMS registration details into the software and credited their account, they will be able to start sending SMS immediately.

For further details email: info@boom-SMS.co.uk and we will be more than happy to assist.

***'The Team'***

***Boom-SMS***

**Terms and Conditions**

It is at the sole discretion of the developer if he or she wishes to utilise this service and whether to present it as your own work.

Any conflicts of interest with your contract of employment/engagement will need to be dealt with by you and Boom-SMS cannot have any legal liability.

Users of this service may contact Boom-SMS, however any faults due to incorrect integration to our system will be referred back to you. We highly recommend all developers extensively test the system prior to full implementation.

Whilst every care has been taken to ensure our system will work in unison to any software it is added to, no responsibility can be taken by Boom-SMS for any disruption to that software.

Boom-SMS may change all or part of these Terms and Conditions from time to time in its sole discretion. Boom-SMS will notify its developers of any such change by email and any use of the Website or Services after such notice by such a developer will be deemed to be an acceptance of any such change. Boom-SMS will also update the Terms and Conditions contained on the Website to reflect any such change.

Boom-SMS shall use reasonable endeavours to keep the Website and Services free from viruses and corrupt files but does not warrant that neither the Website nor the Services are free from infection by viruses or anything else with contaminating or destructive properties.

All images, graphics, text and other materials on the Website are protected by copyrights owned or licensed to Boom-SMS. All trademarks, company names, software and logos on the Website, registered or unregistered are the property of Boom-SMS or have been licensed to Boom-SMS for use on the Website.

Termination: Boom-SMS may terminate your account at any time by giving you no less than 14 days written notice.

Boom-SMS may terminate your account immediately if you commit a breach of any of these Terms and Conditions.

You may terminate your account at any time by notifying Boom-SMS in writing and deleting or destroying all details provided to you. Any notice required to be given under these Terms and Conditions will be in writing and in English and must be sent by you to Boom-SMS.

English law will apply to the Terms and Conditions and any disputes will be settled in the English Courts.

We hope, like many others, that you will gain much credibility by utilising our services.

Integrating to Boom-SMS means you have read, understood and agree to the above Terms and Conditions.

# Boom-SMS Developers Guide Technical Details

## *Creating an SMS message to one or more mobile phones:*

**Target:** Third Party Application Developers

**Third Party Application Requirements:** Ability to send an Email Message with an attachment **or** send a HTTP
Post.

This document describes how to integrate Boom-SMS messaging into a 3rd party application.

## *Step 1.*

Build a string in XML Format as follows:

```
<messages>
  <sms.message>
    <account.id></account.id>
    <password></password>
    <username></username>
    <mobile.to></mobile.to>
    <message></message>
    <notify></notify>
    <sender.email></sender.email>
  </sms.message>
</messages>
```

XML Data Field Definitions:

| | |
|---|---|
| **account.id** | this is the end user's ***Username*** - obtained on registering on the Boom SMS website. |
| **Password** | this is the end user's ***Password*** - obtained on registering on the Boom SMS website |
| **Username**. | *Optional*. *Max 20 Characters*. ***Personal Identifier*** - If your application stores the username of the individual originating the SMS<br>message you can populate this field with that name. This allows more detailed reporting of who is sending messages from a multi user application |
| **mobile.to** | The ***destination Mobile Number***. May be multiple (see example 1.) |
| **message** | *Max 160 Characters*. ***The SMS message***. |

| | |
|---|---|
| **notify** | 1 = Yes 0 = No |
| **sender.email**. | *Optional*. This address is used to send notifications and other error/status messages back to the originator. If you send via email<br>this address will override the email address contained in the email.<br>If you send by HTTP and you do not supply this then notification messages will be sent to the email of the account holder address |

### Example 1 – Single message to multiple recipients:

```
<messages>
  <sms.message>
    <account.id>enduser</account.id>
    <password>letmein</password>
    <username>JOHN</username>
    <mobile.to>07891234567</mobile.to>
    <mobile.to>07891987654</mobile.to>
    <message>Happy Birthday!!! Love Sophie</message>
    <notify>1</notify>
    <sender.email>someone@your-domain.com</sender.email>
  </sms.message>
</messages>
```

### Example 2 – Multiple messages to multiple recipients:

```
<messages>
  <sms.message>
    <account.id>enduser</account.id>
    <password>letmein</password>
    <username>JOHN</username>
    <mobile.to>07891234567</mobile.to>
    <mobile.to>07891987654</mobile.to>
    <message>Happy Birthday!!! Love Sophie</message>
    <notify>1</notify>
    <sender.email>someone@your-domain.com</sender.email>
    </sms.message>
    <sms.message>
    <account.id>enduser</account.id>
    <password>apassword</password>
    <username>JOHN</username>
    <mobile.to>+44789123456</mobile.to>
    <mobile.to>+44734567890</mobile.to>
    <message>See you later</message>
    <notify>1</notify>
    <sender.email>someone@your-domain.com</sender.email>
  </sms.message>
</messages>
```

## *Step 2.*

### If using email as your transport

-you may want to encrypt the attachment. This is ***optional.***
*(*To obtain the DLL go to Downloads page at www.boom-sms.co.uk*)*
The DLL contains a function, which will encrypt a string and save it, ready for you to
use.
EncryptSMSTextToFile( textIn : PChar; FileName : PChar ) : Integer; stdcall;
Parameters:

| | |
|---|---|
| TextIn | Pointer to a null terminated string.<br>Pass your XML formatted SMS message here. |
| FileName | Pointer to a null terminated string<br>Pass the full path to a file with an XMLS extension, e.g. C:\temp\smsmessage.xmls |
| Return Value | 0 for success, any non-zero value indicates Failure. |

If the DLL call is successful you will have an encrypted file <FileName>
*If you do not want to use encryption, simply save your sms message string to file.*

### If using HTTP as you transport

*- you can ignore this step.*

## *Step 3.*

***If using email***
Attach the file to an email and send it to: txt@boom-sms.co.uk
***If using HTTP Post***
Post the data to the following URL:
http://www.boom-sms.co.uk/cgi-bin/sendsms.pl

# Message Notification

## *Return Address Rules*

When a sender requests notification of the result of an SMS send, the following rules are used.

1. If the account manager has set up a Notification URL then all notification are posted via HTTP to that URL.

If no notification URL is configured the notification is sent via Email according to the following rules:

1. If the sms was delivered to boom via the email, then the return email address of the sender is used for the notification, otherwise
2. If the sms is delivered to boom via http then if the message contains an email address in the *<sender. email>* xml tag then that email address is used. If not then the account contact email is used for the notification email.

## *Delivery Status will be one of three values:*

1. success
2. failure
3. pending

If a **pending** message is sent, it will usually be followed by either a success or a failure, provided an update is received from the mobile network within 48 hours. *If no update is available to us from the mobile network then no further notification will be sent out.*

## *Format of Notification Message*

### HTTP Notification

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NotificationRecords>
  <Notification>
    <MessageSent>05/07/2004 15:31:31</MessageSent>
    <From> YOUR_ACCOUNT </From>
    <To>+44712345678</To>
    <Operator></Operator>
    <TransactionID>08D0BBEFDC70AC04</TransactionID>
    <DeliveredDate>:2004-07-05T15:34:36</DeliveredDate>
    <DelivereryStatus> success </DelivereryStatus>
  </Notification>
</NotificationRecords>
```

### Email Notification

Example:

```
Message Sent: 05/07/2004 15:31:31
From: YOUR_ACCOUNT
To: +44712345678
Operator:
TransactionID: 08D0BBEFDC70AC04
Delivered Date: 2004-07-05T15:34:36
Deliverery Status: success
```

Delivery Status wil be one of the following values

## Two Way Response Message

If you have set up 2-way messaging your response will be in the following format:

You can elect to have this response sent to an email address or delivered by HTTP POST to a URL that you supply, or both.

```
<?xml version="1.0"?>
<deliveries>
<delivery uuid="FF36A020-E73F-11D6-B3D7-C520D0C030D1">
<from>+4479662223456</from>
<to>+447781234567</to>
<message><![CDATA[Message sent to GSN here]]></message>
</delivery>
</deliveries>
```

## Other Integration Notes:

Once registered your application will need to store the Username and Password, unless you want the user to re-enter these details each time they send a message.

Contact Information:
**info@boom-sms.co.uk**

## Example ASP script using VBSCRIPT; Sending an SMS from your Webserver

```
<%@LANGUAGE="VBSCRIPT"%>
<%
'Example of how to POST data to the Boom-SMS Server
Dim objSrvHTTP
Set objSrvHTTP = Server.CreateObject ("MSXML2.ServerXMLHTTP")
objSrvHTTP.open "POST","http://www.boom-sms.co.uk/cgi-
bin/sendsms.pl",false
'Create the XML
Dim objXMLDocument
Set objXMLDocument = Server.CreateObject ("MSXML2.DOMDocument")
objXMLDocument.async= false
objXMLDocument.resolveExternals = false
Dim myXMLSMS
' … myXMLSMS – variable containing valid SMS message(s), built
' elsewhere…code not shown…
'Read in your XML...
objXMLDocument.loadXML myXMLSMS
'Now POST the data to the Boom-SMS Server
objSrvHTTP.send objXMLDocument
'Check the response (if any - currently the Server does not provide any
'validation/response at this stage of the process)
Response.ContentType = "text/html"
Response.Write "RESPONSE:<br>"
Response.Write (objSrvHTTP.responseText)
%>
<html>
<head>
<title>Send SMS</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF" text="#000000">
OK
</body></html>
```

## Example: ASP.net script using C#; Sending an SMS from your Webserver

```
//This code demonstrates sending an SMS message using asp .NET (Language C#)

const String AccountID = "AccountID"; //From your own storage
```

```
                const String AccountPassword = "password"; //From your own
storage
                const String theUSER = "WEBSITE";//From your own storage...
            const String senderEMAIL = "you@yourdomain.com";//From your own
storage...

                //Use myXML to construct the XML document...
                StringBuilder myXML = new StringBuilder("<?xml version=\"1.0\"
?>");
                myXML.Append("<sms.message>");

                myXML.Append("<account.id>");
                myXML.Append(AccountID);
                myXML.Append("</account.id>");

                myXML.Append("<password>");
                myXML.Append(AccountPassword);
                myXML.Append("</password>");

                myXML.Append("<username>");
                myXML.Append(theUSER);
                myXML.Append("</username>");

                myXML.Append("<sender.email>");
                myXML.Append(senderEMAIL);
                myXML.Append("</sender.email>");

                myXML.Append("<mobile.to>");
                myXML.Append(SMSToFLD.Text);
                myXML.Append("</mobile.to>");

                myXML.Append("<message>");
                myXML.Append(SMSMessageFLD.Text);
                myXML.Append("</message>");

                myXML.Append("<notify>");
                myXML.Append("0");  //1 to request notification, otherwise 0
                myXML.Append("</notify>");

                myXML.Append("</sms.message>");

                //OK - you have the document, now post it to the Boom URL...
                Uri myUri =new Uri("http://www.boom-sms.co.uk/cgi-
bin/sendsms.pl");
                WebRequest     myWR = WebRequest.Create(myUri);
                myWR.Method = "POST";
                myWR.ContentType="application/x-www-form-urlencoded";
                myWR.ContentLength = myXML.Length;
                Stream newStream=myWR.GetRequestStream();
                ASCIIEncoding encoding=new ASCIIEncoding();
                byte[]  byte1=encoding.GetBytes(myXML.ToString());
                newStream.Write(byte1,0,byte1.Length);
                // Close the Stream object.
                newStream.Close();

                //Message has been sent to the boom server
```