

XML 2 SMS

Developers Guide

Vodacom Messaging

www.vodacommessaging.co.za

087 55 00 200

Contents

Version history	3
Introduction	4
Important Notice: URL Encoding	4
Send SMS Command (GET Method)	5
Send SMS Command: Multiple SMS	6
Send SMS Command: EMS	7
Send SMS Command (POST Method)	8
SMS Status and Errors.....	9
Received SMS Command	11
Limits	11
Querying Sent SMS Command.....	12
Limits	13
HTTP Call-Backs	13
Delivery Receipts	13
Incoming SMS	13
Web Service	14
Invocation	14
Web Reference.....	14
Web Service Proxy.....	14
SOAP.....	14
HTTP POST.....	16
SMS Reports.....	17
Internal Links.....	17
Support	17

Version history

Revision number	Revision Date	Author	Changes
1.0	2010-02-08	Jonathan Hudson	Initial document release
1.1	2010-08-25	Jonathan Hudson	Updated Send Method
1.2	2011-11-18	Jonathan Hudson	Added HTTP Call-Backs
1.3	2013-10-03	Jonathan Hudson	Updated Error Codes
1.4	2013-11-27	Jonathan Hudson	Updated Sendlog
1.5	2014-08-19	Jonathan Hudson	Updated HTTP Call-Backs
1.6	2014-10-30	Jonathan Hudson	Updated Error Codes
1.7	2015-04-28	Jonathan Hudson	Updated Syntax & URL
1.8	2015-05-27	Jonathan Hudson	Updated Error Codes / Added SOAP Details

Introduction

Designed for developers, the XML / HTTP interface will allow you to send and receive SMS messages and query current and historical logs using HTTP request and XML response.

This provides the Developer with a comprehensive, customizable and scalable SMS functionality.

The benefit of this service is that there is no interference with existing firewall or security infrastructure, and it can be speedily integrated into your system.

XML 2 SMS now also includes an EMS option, which allows for messages of up to 2,000 characters.

This Help File provides a reference for the XML 2 SMS user. We suggest the new user read through this entire document before using the XML 2 SMS product. Experienced users can also benefit.

Please note this help file is based on the URL <https://xml2sms.gsm.co.za> which along with the necessary parameters can be used to Send / Receive and query SMS from both Admin and Linked User Accounts.

Note that Account Administrators should use the following format, which is faster than the default URL:
<http://msgwX.gsm.co.za/xml/>

Where msgwX = the name of the SMS Gateway you have been allocated.

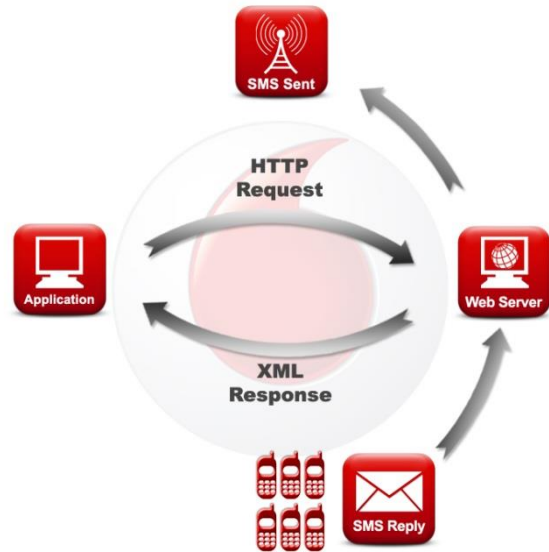
Please Note: The above mentioned URL cannot be used by Linked Users.

Please visit: <https://www.vodacommessaging.co.za/XML2SMS/functions.asp?tab=20> for more info.

Important Notice: URL Encoding

Please note that as the XML2SMS requests are sent over HTTP, certain characters are "reserved characters" and will need to be percent-encoded.

For more information and examples on this, please visit: http://en.wikipedia.org/wiki/URL_Encoding



Send SMS Command (GET Method)

:: Function

Sends an SMS

:: Syntax

send/?number=X&message=Y

:: Parameters

Number:

Cellphone number to send the SMS to. Although it is acceptable to enter this in a national format, it is recommended you enter this with the full international code (i.e.: South Africa Vodacom is 2782xxxxxx) as it enables our SMS Gateway to route it more intelligently, and you will get better performance.

Message:

The message to be sent, in standard RFC1738 encoding. Most web interfaces/toolkits conform to RFC1738.

Returns:

```
<aatsms>
<submitresult action=X key=Y result=Z />
</aatsms>
```

:: XML Parms

Action:

The result of what the gateway did with the message. Valid values are: enqueued (SMS put into outgoing queue) and error (Gateway could not process SMS.)

Key:

If there was no error, this is the key the message was assigned in the sent logs.

Result:

Numeric value, 1 for success or 0 for failure.

:: Sample In

<https://xml2sms.gsm.co.za/send/?username=x&password=y&number=27825551234&message=This+is+a+test>

:: Sample Out

If the script is able to successfully submit the message to the Gateway, this will be the result:

```
<?xml version="1.0" ?>
- <aatsms>
<submitresult action="enqueued" key="3951103" result="1" />
</aatsms>
</span>
```

Where 'key' is the key you can lookup. (Result 1 means it was successfully delivered to the SMS Gateway). Once the Gateway has accepted the SMS it will submit it to the SMSC, and you can then query the message status.

Send SMS Command: Multiple SMS

:: Function

Sends multiple SMS

:: Syntax

send/?**number1**=X&**message1**=Y&**number2**=X&**message2**=Y&**number3**=X&**message3**=Y

:: Parameters

Number:

Cellphone number to send the SMS to. Although it is acceptable to enter this in a national format, it is recommended you enter this with the full international code (i.e.: South Africa Vodacom is 2782xxxxxx) as it enables the Gateway to route it more intelligently, and you will get better performance.

Message:

The message to be sent, in standard RFC1738 encoding. Most web interfaces/toolkits conform to RFC1738.

Returns:

```
<aatsms>
<submitresult action=X key=Y1 result=Z />
<submitresult action=X key=Y2 result=Z />
<submitresult action=X key=Y3 result=Z />
</aatsms>
```

:: XML Parm

Action:

The result of what the gateway did with the message. Valid values are: enqueued (SMS put into outgoing queue) and error (Gateway could not process SMS.)

Key:

If there was no error, this is the key the message was assigned in the sent logs.

Result:

Numeric value, 1 for success or 0 for failure.

:: Sample In

[https://xml2sms.gsm.co.za/send/?username=x&password=y&](https://xml2sms.gsm.co.za/send/?username=x&password=y&number1=27825551234&message1=This+is+a+test&number2=27825551234&message2=This+is+a+test+2)

number1=27825551234&**message1**=This+is+a+test&**number2**=27825551234&**message2**=This+is+a+test+2

:: Sample Out

If the script is able to successfully submit the message to the Gateway, this will be the result:

```
<?xml version="1.0" ?>
- <aatsms>
<submitresult action="enqueued" key="3951103" result="1" />
<submitresult action="enqueued" key="3951104" result="1" />
</aatsms>
</span>
```

Where 'key' is the key you can lookup. (Result 1 means it was successfully delivered to the SMS Gateway). Once the Gateway has accepted the SMS it will submit it to the SMSC, and you can then query the message status.

Send SMS Command: EMS

:: Function

Sends an EMS (Extended Message, up to 2,000 characters, suggested limit is 1,000 characters)

:: Syntax

send/?**number**=X&**message**=Y&**ems**=1

:: Parameters

Number:

Cellphone number to send the SMS to. Although it is acceptable to enter this in a national format, it is recommended you enter this with the full international code (i.e.: South Africa Vodacom is 2782xxxxxx) as it enables the Gateway to route it more intelligently, and you will get better performance.

Message:

The message to be sent, in standard RFC1738 encoding. Most web interfaces/toolkits conform to RFC1738.

ems:

Indicates EMS message. If message length is greater than 160 characters, the message will be split and sent as an EMS.

Returns:

```
<aatsms>
<submitresult action=X key=Y1 result=Z />
<submitresult action=X key=Y2 result=Z />
</aatsms>
```

:: XML Parms

Action:

The result of what the gateway did with the message. Valid values are: enqueued (SMS put into outgoing queue) and error (Gateway could not process SMS.)

Key:

If there was no error, this is the key the message was assigned in the sent logs.

Result:

Numeric value, 1 for success or 0 for failure.

:: Sample In

<https://xml2sms.gsm.co.za/send/?username=x&password=y&number1=27825551234&message=This+is+an+EMS+test+and+as+the+SMS+length+is+greater+than+160+characters+it+will+be+split+and+delivered+to+the+handset+as+one+long+message+which+will+be+displayed+on+the+handset+as+one+message&ems=1>

:: Sample Out

If the script is able to successfully submit the message to the Gateway, this will be the result:

```
<?xml version="1.0" ?>
- <aatsms>
<submitresult action="enqueued" key="3951103" result="1" />
<submitresult action="enqueued" key="3951104" result="1" />
</aatsms>
</span>
```

Where 'key' is the key you can lookup. (Result 1 means it was successfully delivered to the SMS Gateway). Once the Gateway has accepted the SMS it will submit it to the SMSC, and you can then query the message status.

Send SMS Command (POST Method)

Important Note: This method can only be used by Administrators

Besides the commonly used GET method of calling our SMS Gateway, it is possible to POST.

:: Sample Form

```
<form action="http://smsgwX.gsm.co.za/xml/send" method="post">

columns:<input name="batchcols" value="number,message" />
<br />

<textarea name="batch" rows="20">
27721234567, Hi there
27821234567, Hi there
27831234567, Hi there
</textarea>

<input name="gwusername" value="myUserName" />
<input name="gwpasword" value="myPassWord" />

<input type="submit" />

</form>
```

Where smsgwX = the name of the SMS Gateway you have been allocated.

:: Inputs

Name	Detail
gwusername	Your account username
gwpasword	Your account password

:: Sample URL being called

http://smsgwX.gsm.co.za/xml/send/?batchcols=number%2Cmessage&batch=27721234567%2C+Hi+there+%0D%0A27821234567%2C+Hi+there+%0D%0A27831234567%2C+Hi+there+%0D%0A&gwusername=myUserName&gwpasword=myPassWord

:: Sample Output

```
<?xml version="1.0"?>
<aatsms>
<submitresult action="enqueued" key="99955262" result="1" number="27721234567" />
<submitresult action="enqueued" key="99955263" result="1" number="27821234567" />
<submitresult action="enqueued" key="99955264" result="1" number="27831234567" />
</aatsms>
```

This method makes it also possible to send larger batches with greater ease.
It is also possible to send single SMS via this method.

SMS Status and Errors

:: Message Status:

The possible values for the "status" property in the send xml are:

Status Code	Definition
1	En Route
0, 2	Delivered
3	Expired
4	Deleted by SMSC
5	Cannot Deliver
6	Accepted for Delivery
7	Unknown Error
8	Rejected by SMSC
10	Abandoned
11	Deleted by operator
12	Waiting to be queued
13	Waiting at SMSC
14	Waiting for Resend
15	Destination out of Range
16	Destination Congested
17	Destination Invalid
18	Refused by destination
19	Destination SIM refused
20	SMSC Error
21	Provider error
22	Destination not on network
23	Invalid Message
24	Network Error
25	No route
26	Congested, will retry
27	Busy, will retry
28	No response, Will retry
29	Rejected, will retry
30	Low signal, will retry
31	Unknown Error

Any other status not listed above will return the status code as well as "Undefined Error" as the description.

Please Note: Receipts may take up to 24 hours for accurate results

:: Errors

If your application is unable to successfully submit the message to the Gateway, an error such as the one below would be returned in the form of XML. It is normally due to syntactical errors like forgetting to append a cellnumber.

```
<?xml version="1.0"?>
<aatsms>
  <submitresult action="not queued - error(errorcode)" error="errorcode" key="0" result="0" number="number" />
</aatsms>
```

The error code will be one of the following:

Code	Definition
150	Invalid username or password.
151	Reading of send / receive logs disabled on Gateway account.
152	Sending SMS disabled on Gateway account.
153	Not enough credits to send SMS.
154	Attempted to submit an invalid number (format or banned).
155	Duplicate message submitted within the last 15 minutes.
156	There is no routing available for the destination number, which is usually due to an invalid number being submitted.
157	Gateway DNS not found.
158	No subscription to XML 2 SMS.
159	Invalid path.
160	Problem communicating with Gateway.
161	Invalid query string.

Important Note: It is vital that you check the response returned from the SMS Gateway and base logic against this if necessary.

Received SMS Command

:: Function

Queries received messages log.

:: Syntax

recvlog/?key=X

recvlog/?watermark=X

recvlog/?startdate=YYYYMMDD&enddate=YYYYMMDD

:: Parameters

key:

Key of the message to look up. Only one message will be returned.

startdate / enddate :

The start and end dates, respectively, for which the logs should be returned. These are in the format YYYYMMDD. All messages occurring within the start and end dates as well as the messages in between are returned.

watermark :

A message key to return all messages since (the message corresponding to the watermark key is also returned); to see what is new in the received message log you will use the highest message-key returned by the previous "watermark" command and increment it by one.

Returns:

```
<message type="received" key=X fromnumber=Y message=Z timereceived=V />
[more <message> tags]
</aatsms>
```

:: XML Parm

Type:

Type of message. Will always be "received" in the received logs.

Key:

Message key.

Fromnumber:

Originating number; in international format.

Message:

Text of the message, in plain ASCII.

Timereceived:

Date and Time the message was received into the database. Records are sorted with the most recent one first.

:: Sample In

<https://xml2sms.gsm.co.za/recvlog/?username=x&password=y&watermark=12345>

:: Sample Out

```
<aatsms>
<message type="received" key="12345" fromnumber="27825551234" message="test message 1" timereceived="23-Oct-
2002 19:30:56" />
</aatsms>
```

Limits

Please be advised we will enforce limits on the number of times you can call the RECVLOG.

For new SMS look ups (i.e. watermark):

- At most once every 30 seconds.

For general historical look ups (i.e. date ranges):

- At most once every 10 minutes, and should only be used once per date range.

Querying Sent SMS Command

:: Function

Queries sent messages log.

:: Syntax

```
sendlog/?key=X  
sendlog/?watermark=X  
sendlog/?startdate=YYYYMMDD&enddate=YYYYMMDD
```

:: Parameters

Key:

Key of the message to look up. Only one message will be returned. The key returned by a “send” command may be passed here to check on the status of the message.

Startdate / Enddate :

The start and end dates, respectively, for which the logs should be returned. These are in the format YYYYMMDD. All messages occurring within the start and end dates as well as the messages in between are returned.

Watermark :

A message key to return all messages since (the message corresponding to the watermark key is also returned); to see what is new in the received message log you will use the highest message-key returned by the previous “watermark” command and increment it by one.

Returns:

```
<message type="sent" key=X tonumber=Y message=Z timesent=T timedelivered=U delivered=A status=B  
statusdescription=C />
```

:: XML Parm

Type:

Type of message. Will always be “sent” in the sent logs.

Key:

Message key.

Tonumber:

The number that the message was sent to; this will be in international format, e.g.: 27825551234.

Message:

Text of the message, in plain ASCII.

Timesent:

Date and time the message was dispatched from the SMS Gateway.

Timedelivered:

Date and time the message was confirmed as delivered. If it has not been delivered or there was an error, this will be “01-Jan-1970 00:00:00”.

Delivered:

Integer value. 1 = Message delivered. 0 = Not yet delivered or error. (Refer to Chapter on SMS Status and Errors).

Status:

Integer status code in event of non delivery or error. (Refer to Chapter on SMS Status and Errors)

Statusdescription :

Text description for the reason the message has not yet been / will not be delivered.

:: Sample In and Out

<https://xml2sms.gsm.co.za/sendlog/?username=x&password=y&watermark=12345>

```
<message type="sent" key="12345" tonumber="+27835551234" message="test 1" timesent="19-Nov-2002 23:44:30"
timedelivered="20-Nov-2002 23:44:32" delivered="1" status="0" statusdescription="Delivered" />
```

Please Note: Receipts may take up to 24 hours for accurate results

Important Note: The SENDLOG will display a maximum of 1000 rows.

Please use the HTTP Call-Backs described in the next section as a recommended method for larger volume accounts.

Limits

Please be advised we will enforce limits on the number of times you call the SENDLOG.

Please ensure you set your max limits as follows:

For single key look ups:

- 1 minute after an SMS has been sent.
- Every 10 minutes up to 60 minutes.
- Once an hour thereafter.

For general historical look ups (i.e. date ranges):

- At most once every 10 minutes, and should only be used once per date range.

Again for Admin Accounts we strongly recommend the HTTP Call-back method described below.

HTTP Call-Backs

HTTP Call-Backs allow you to configure a URL in your XML2SMS product in order for us to call you when you have an updated delivery receipt or new incoming message.

Delivery Receipts

Example URL: <http://www.mywebsite.co.za/mypage.asp>

We will then pass the following parameters to your URL:

- FN: (FROMNUMBER) The cell number sent to
- TN: (TONUMBER) Your XML 2 SMS Account from number
- SC: (SUCCESS) 1 or 0
- ST: (STATUS) Integer representing the SMSC Delivery Status
- RF: (REF) Unique Reference number for the SMS
- TS: (TIMESTAMP) Date and Time, in SQL Server format

Incoming SMS

Example URL: <http://www.mywebsite.co.za/mypage.asp>

As above we will then pass the following parameters to your URL:

- FN: (FROMNUMBER) The cell number that sent the SMS
- TN: (TONUMBER) The number the user has replied to (Your XML 2 SMS number)
- MS: (MESSAGE) The actual SMS message you have received
- TS: (TIMESTAMP) Date and Time, in SQL Server format

Note: HTTP Call-Backs can only be used on the Administrator account and not for linked users who may be subscribed to XML 2 SMS

Web Service

Web service URL: <https://soap.gsm.co.za/>

When navigating there, a list of all the supported methods is shown.

WSDL: <https://soap.gsm.co.za/?WSDL>

Simply put, the WSDL (Web service description language) describes the functionality of the web service.

Invocation

There are four ways to invoke the web service:

Web Reference

This can be done in Visual studio! Clients would add a web reference by right clicking the references node for your project in the Solution Explorer, clicking "Add Service Reference", clicking "Advanced...", clicking "Add Web Reference...", inserting the URL in the URL input, clicking "Go" and finally clicking "Add Reference". Upon confirmation, a web reference will be created.

Clients can then declare and instantiate an instance of the web service to use its methods.

Web Service Proxy

For clients that prefer a slightly more manual route, the .NET Framework comes with a utility called `wsdl` that will generate a C# or VB .NET source file that the client can then use in his application. This will, like the web reference method, allow the client to declare and instantiate an instance of the web service. In fact, the web reference method detailed above is the automatic version of this method. The source of the web service itself is not compromised by use of this method - the `wsdl` utility merely creates methods that call the web service on behalf of the developer. To generate the proxy code, the developer must open up the Visual Studio command prompt and execute the following command for a C# source file:

```
wsdl https://soap.gsm.co.za/?wsdl
```

Generating VB .NET source is similar:

```
wsdl /l:VB https://soap.gsm.co.za/?wsdl
```

SOAP

Version 1.1

The following is a sample SOAP 1.1 request and response for the `SendSMS()` method. Check the webservice itself for the use of other methods. The placeholders (delimited with square brackets) need to be replaced with actual values.

```
POST /xml2sms.aspx HTTP/1.1
Host: soap.gsm.co.za
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "https://soap.gsm.co.za/SendSMS"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

```

<SendSMS xmlns="https://soap.gsm.co.za/">
  <username>string</username>
  <password>string</password>
  <number>string</number>
  <message>string</message>
</SendSMS>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendSMSResponse xmlns="https://soap.gsm.co.za/">
      <SendSMSResult>xml</SendSMSResult>
    </SendSMSResponse>
  </soap:Body>
</soap:Envelope>

```

Version 1.2

The following is a sample SOAP 1.2 request and response for the SendSMS() method. Check the webservice itself for the use of other methods. The placeholders (delimited with square brackets) need to be replaced with actual values.

```

POST /xml2sms.asmx HTTP/1.1
Host: soap.gsm.co.za
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendSMS xmlns="https://soap.gsm.co.za/">
      <username>string</username>
      <password>string</password>
      <number>string</number>
      <message>string</message>
    </SendSMS>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendSMSResponse xmlns="https://soap.gsm.co.za/">
      <SendSMSResult>xml</SendSMSResult>
    </SendSMSResponse>
  </soap12:Body>
</soap12:Envelope>

```

HTTP POST

The following is a sample HTTP POST request and response for the SendSMS() method. Check the webservice itself for the use of other methods. The placeholders (delimited with square brackets) need to be replaced with actual values.

```
POST /SendSMS HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: [length]

username=[string]&password=[string]&number=[string]&message=[string]
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: [length]

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="https://soap.gsm.co.za/">[string]</string>
```


SMS Reports

Complete SMS Reports are available on the web site under username and password protection. Account Administrators can view all SMS sent off the Account, while Linked Users can only view their own.

An easy to use search facility is available to draw out Reports on Sent or Received SMS. Reports can be drawn by Date, User, Type of SMS (Single, Batch or Received), Service used to send or receive the SMS, SMS message and Cellnumber sent to.

SMS Reports

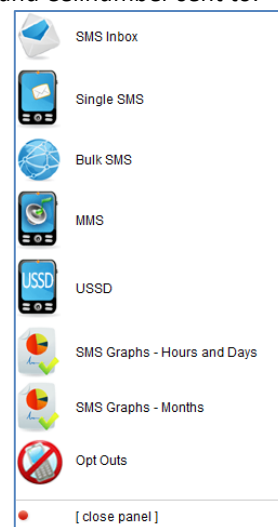
Report on: Bulk SMS From: 31 March 2011 00:00 To: 31 March 2011 23:59 Run Report

Cell: Select User: -- All Products --

Report Detail

SMS Reports offer the following details:

- Product used to send the SMS
- SMS recipient (cellnumber)
- Delivery status
- Time sent
- SMS message



SMS Reports can be drawn based on date sent, user sent from, type of SMS, product used, message and cellnumber sent to. Report Tools then allow you to print results, or have them e-mailed to you in Excel format.

	C	27824112004	S	11:43 on Thu 9 Oct 2008	D	11:43 on Thu 9 Oct 2008
1	Hi Team, Please do not forget the meeting at 3pm today!					
	XML2SMS SMS Status: Delivered (Confirmed)					

Delivered (Confirmed)
SMS has been successfully delivered.

Internal Links

Vodacom Messaging XML 2 SMS page - <https://www.vodacomessaging.co.za/xml2sms.asp>

Support

We've tried to put all the information we could in this Guide, but we are sure you'll have some questions.

If you would like to chat to us, please call us on **087 55 00 200** and we'll help you out as best we can.

Please also visit our website at www.vodacomessaging.co.za where you may log a support query.