



Premium SMPP Platform Documentation

Written by Jateen Mistry
Revised: 29th April 2004

SUPPORT:

For support related issues please logon to the support forum at <http://aqcorporate.com/support.php> and raise a new ticket. The support screen loads in a new window so please ensure all pop-up blocking software is **disabled** for the support site.

Contact us

If you have a sales related question please call us on
08707 449227
08707 449228 (fax)

If you have any technical questions regarding SMPP please raise a support ticket.

sms2email.com

sms2email is a service provided by (aq)





Introduction to the Service

The SMPP platform provided by sms2email.com is a high-speed method of sending global messages. This document assumes the reader is familiar with the SMPP specification and has read the sms2email specific SMPP documentation at:
http://aql.com/supportdocuments/smpp_api.pdf

This document will describe how to use SMPP in conjunction with Premium SMS.

Overview of Premium SMS via SMPP

The premium SMPP account is an MT chargeable service. This means that the first part of any premium transaction is receiving an MO (mobile originated) message from the mobile. This is routed to your server via a DELIVER_SM pdu. At this point the end user has NOT been charged. This can be seen as the end user giving you permission to charge them.

If you wish to charge the end user, you are now required to send a message back to them. This will be in the form of a SUBMIT_SM pdu. Once this message has been successfully delivered to the phone, the user will be billed at the amount the short code is contracted for.

sms2email.com

sms2email is a service provided by (aq)



MO Message (DELIVER_SM pdu)

The DELIVER_SM pdu is defined as follows for premium SMS services.

Parameter	Description
command_length	As in SMPP Specification
command_id	As in SMPP Specification
command_status	As in SMPP Specification
sequence_number	As in SMPP Specification
service_type	See below for details
source_addr_ton	Will always be 0 (zero).
source_addr_npi	Will always be 0 (zero).
source_addr	This is the mobile number that this message originated from
dest_addr_ton	Will always be 0 (zero).
dest_addr_npi	Will always be 0 (zero).
destination_addr	This is set to your short code.
esm_class	Always set to 0 (00000000)
protocol_id	Not used, will always be 0 (zero).
priority_flag	Not used, will always be 0 (zero).
schedule_delivery_time	Always set to NULL
validity_period	Always set to NULL
registered_delivery	Always set to 0
replace_if_present_flag	Always set to 0
data_coding	Always set to 0
sm_default_msg_id	Always set to 0
sm_length	Length in octets of the short_message parameter
short_message	This contains the message that the user has sent to your short code

service_type parameter in DELIVER_SM pdu

The service type parameter dictates which mobile operator the premium MO message was received on. When your account is activated, you will be given a list of valid service_type values and which mobile operators they refer to.

This parameter is important, as it is required in the SUBMIT_SM response to this message.

MT Message (SUBMIT_SM pdu)

When you have received a DELIVER_SM pdu from our SMPP server, your response should be a SUBMIT_SM. This is only if you wish to charge the end user.

The SUBMIT_SM pdu is defined as follows for premium SMS services.

Parameter	Description
command_length	As in SMPP Specification
command_id	As in SMPP Specification
command_status	As in SMPP Specification
sequence_number	As in SMPP Specification
service_type	See below for details
source_addr_ton	Set to 0 (zero)
source_addr_npi	Set to 0 (zero)
source_addr	You must set this value to your short code
dest_addr_ton	Must be 0 (zero). This is auto-detect mode
dest_addr_npi	Must be 0 (zero). This is auto-detect mode
destination_addr	You must set this to the source number received in the original DELIVER_SM message
esm_class	See explanation of esm_class and data_coding at end of table
protocol_id	Not supported, use any value in accordance with SMPP Specification
priority_flag	Not supported, use any value in accordance with SMPP Specification
schedule_delivery_time	If left blank the message will be sent immediately. To send message at a different time, set the time as defined in the SMPP specification.
validity_period	Not supported, use any value in accordance with SMPP Specification
registered_delivery	If a delivery report is required set to 1 otherwise set to 0 (zero)
replace_if_present_flag	Not supported, use any value in accordance with SMPP Specification
data_coding	See explanation of esm_class and data_coding at end of table
sm_default_msg_id	Not supported, use any value in accordance with SMPP Specification
sm_length	Length in octets of the short_message parameter
short_message	The short message itself as defined by the SMPP specification.

esm_class and data_coding parameters

The following table describes the values that need to be set for the esm_class and data_coding parameters depending on what type of content you require to be sent.

Message type	esm_class	data_coding
Text	00000011	11110001 or 00000000
Flash	00000011	11110000
UDH	00000011	01000000

service_type parameter

The service_type parameter **must** be the same value as received in the originating DELIVER_SM

Repeat/Multiple Billing

With your sms2email premium account, repeat / multiple billing is an option. The one requirement with this, as with any other premium SMS transaction, is that an MO message must first have been sent to your short code from the mobile phone in question.

Whenever you wish to make a repeat billing request, you must simply reissue the SUBMIT_SM pdu as outlined above. Each SUBMIT_SM you send will charge the end user at the amount your short code is contracted for.

The service_type parameter should be set to the value received from sms2email for the original DELIVER_SM message.

NOTE:

If a repeat billing event is not getting through to the mobile, this could be because we have closed the repeat billing session. This will only happen in exceptional circumstances such as the end user requesting this.

This means that to bill this user in the future, they must send another MO message to your short code.

Errors when using Premium SMPP

When any information regarding a previously submitted message need to be propagated back to your servers, a DELIVER_SM pdu will be used. This DELIVER_SM pdu must not be mixed up with the MO DELIVER_SM pdu.

IMPORTANT:

The esm_class distinguishes an MO message from a delivery report or error report.

The following table dictates the values the esm_class parameter can take.

Message Type	esm_class value
MO message.	0
Delivery report	4
Error Report	4

The information in the deliver/error report in encoded within the short message parameter.

The exact format and the values this can take can be found in the sms2email SMPP platform documentation under the **'Delivery Responses'** Heading.

Example of situations in which you will receive an error report are:

- If you try to send a SUBMIT_SM message via the premium SMPP account to a mobile that has not sent an MO to your short code prior to this.
- The destination was unreachable for any reason
- The destination number has been locked. This means no billing requests will be sent to the destination mobile.

This is not an exhaustive list of possible errors but a small sample of reasons why billing events may fail.



Additional Notes

You will require an SMPP client to send messages via the sms2email SMPP server. The following section may prove useful in this regard.

There is an excellent open source SMPP client for Linux available at <http://www.kannel.org>. This provides functionality above and beyond what is required to send messages via sms2email. It is fairly complex to configure but it is extremely reliable and worth considering.

There is a small SMPP client written in PHP located at:
<http://www.phpclasses.org/browse/package/1373.html>

If you would like to develop your own SMPP client then there is a Java API available to download. This can be found at:
<http://www.logicacmg.com/wirelessnetworks>
<http://opensmpp.logica.com/>
<https://sourceforge.net/projects/smstools/>

An equivalent Perl API is available at:
<http://search.cpan.org/author/SAMPO/Net-SMPP-1.03/SMPP.pm>

SMPP Specification

The SMPP specification for use with sms2email can be found at the following location:

http://aql.com/supportdocuments/smpp_api.pdf

If you require the SMPP v3.4 specification, this can be downloaded from the following site:

<http://www.smpp.org/doc/public/index.html>

If you need further information relating to SMPP, a good starting point is:
<http://www.smpp.org>

sms2email.com

sms2email is a service provided by (aq)

