



## SMPP Platform Documentation

Written by Jateen Mistry  
Revised: 29<sup>th</sup> 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.

NOTE: This documentation assumes the reader is familiar with SMPP technology and only contains information on how to use the SMPP in connection with sms2email.

## Introduction to the sms2email SMPP API

Sms2email supports SMPP v3.4. The following contains a list of the PDU's supported by sms2email:

### Client to sms2email

Bind Commands
BIND_TRANSMITTER
BIND_RECEIVER
BIND_TRANSCEIVER

Other Commands
ENQUIRE_LINK
SUBMIT_SM
QUERY_SM
UNBIND
DELIVER_SM_RESP

### Sms2email to Client

Bind Commands
BIND_TRANSMITTER_RESP
BIND_RECEIVER_RESP
BIND_TRANSCEIVER_RESP

Commands
DELIVER_SM
ENQUIRE_LINK
ENQUIRE_LINK_RESP
SUBMIT_SM_RESP
QUERY_SM_RESP
UNBIND_RESP
GENERIC_NACK



## Binding to sms2email

There are 2 options available when binding to sms2email. You can either bind using the transmitter and receiver pair from SMPP v3.3 or bind using the transceiver mode new to SMPP v3.4

When the SMPP account is set up for you, you will be given a username (system\_id), password and a system\_type value to connect with.

### Maximum number of Binds

You are initially restricted to the number of binds you can make to sms2email. Depending on the version of SMPP you are using it will be either 1 transceiver connection or 1 transmitter and 1 receiver connection. In most cases this is sufficient. If, however, you require this configuration changing on your account, please raise a support ticket via our support system (see bottom of page 1). We will then be able to discuss your needs and make the appropriate changes.

### **IMPORTANT:**

You can only connect to sms2email via certain IP addresses that you have registered. If you wish to connect from multiple IP addresses, please raise a support ticket via our support system (see bottom of page 1)

**sms2email.com**

*sms2email is a service provided by (aq)*



## Submitting messages to sms2email

The following table gives a breakdown of the SUBMIT\_SM pdu. It contains details regarding each of the parameters of the pdu and how they should be used when passing messages to the sms2email servers.

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	Use the system_type parameter that is given to you by sms2email
source_addr_ton	See explanation of source_addr_ton and source_addr_npi at end of table
source_addr_npi	See explanation of source_addr_ton and source_addr_npi at end of table
source_addr	The originator to be used for this message. If left blank, the default originator on your account will be used. Either 16 digits (in international format) for a mobile number or 11 alphanumeric characters for text.
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	The destination number in international format (e.g 447911111111).
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.

**NOTE:** Optional parameters are not required by sms2email and are not supported.

### 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

### source\_addr\_ton and source\_addr\_npi parameters

When mimicking a mobile number, international format must be used. In this case both source\_addr\_ton and source\_addr\_npi must be set to 1. In other cases, it is generally acceptable to leave these parameters set to 0 (zero).

## Querying a previously submitted message

Sms2email support the QUERY\_SM pdu. This can be used to determine the state of a message at a time that is suitable to you. The table below gives a breakdown of the QUERY\_SM pdu and how it should be used with sms2email.

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
message_id	The message id that was originally sent back in the SUBMIT_SM_RESP pdu
source_addr_ton	Must be 0 (zero). This is auto-detect mode
source_addr_npi	Must be 0 (zero). This is auto-detect mode
source_addr	Not used. Set to NULL

The QUERY\_SM\_RESP pdu is as defined in the SMPP specification.

## Error response codes

The error responses that can be sent back are as defined in the SMPP specification. The additional error codes specific to sms2email are outlined in the table below.

Error code	Description
0x00000400	You do not have enough credits to send message. Please contact us to credit your account.

## Delivery Responses

If the register\_delivery parameter is set to 1 in the original SUBMIT\_SM pdu then a DELIVER\_SM response will be sent back to you when the message has reached a final state. This could take anywhere between seconds and hours. Some cases could take even longer.

The following table shows a breakdown of the parameters that are sent in a DELIVER\_SM and how to interpret them when received from sms2email.

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	The system_type value that the connection that the message was submitted with is echoed in this parameter
source_addr_ton	Will always be 0 (zero).
source_addr_npi	Will always be 0 (zero).
source_addr	This is the mobile number that the original message was sent to
dest_addr_ton	Will always be 0 (zero).
dest_addr_npi	Will always be 0 (zero).
destination_addr	This is set to the originator of the original message.
esm_class	Always set to 4 (00000100)
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	The short message itself as defined by the SMPP specification.

The short message for a delivery receipt is a string and will look like the following:

**“id:IIIIIIII sub:SSS dlvr:DDD submit date:YYMMDDhhmm done date:YYMMDDhhmm stat:DDDDDD err:E Text: . . . . .”**

The fields in the delivery receipt are explained in the following table:

Field	Size	Type	Description
id	10 (max)	NULL Terminated	id supplied by sms2email in the original SUBMIT_SM pdu
sub	3 (max)	NULL Terminated	Always 000
dlvrd	3	NULL Terminated	Always 000
submit date	10	NULL Terminated	Date + time submitted. See SMPP Specification for the format
done date	10	NULL Terminated	Date + time delivered. See SMPP Specification for the format
stat	7	NULL Terminated	Final status of the message. See table below
err	3	NULL Terminated	Associated error code. See table below
text	20	String without NULL termination	Will always contain the text 'Not available' (without quotes)

The stat parameter contains the status of the message. The values it can take along with associated error codes are described below:

'stat' value	'err' value	Description
DELIVRD	000	The message has been delivered
UNDELIV	101	The message is undeliverable
UNKNOWN	102	The message is in a final invalid state. It is unknown if the message has been delivered.

In the near future we will be adding additional status codes. This will allow us to pass along more detailed information.

As with all SMPP communications, the sms2email servers expect a DELIVER\_SM\_RESP pdu in response to a delivery report.

## Terminating the connection

In order to stop the connection to sms2email you must first issue the UNBIND pdu. Sms2email will send back an UNBIND\_RESP pdu at which point it is safe to terminate the connection.

## 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

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>