

SMPP protocol

Introduction

SMPP is an industry standard, reliable method of sending a high number of messages. Our documentation leads you through the process of integrating with our SMPP service.

Interface

Developing and implementing your SMPP based application can be difficult to do without an industry standard test platform. We offer a fully compliant SMPP test profile as part of our wholesale service. The test profile allows your developers to simulate a real production environment (including delivery receipting) without incurring the costs of sending messages.

Introduction to the aql SMPP service

aql supports SMPP v3.4. The following contains a list of the PDUs supported by aql:

Client to aql

- BIND_TRANSMITTER
- BIND_RECEIVER
- BIND_TRANSCEIVER
- ENQUIRE_LNK
- SUBMIT_SM
- QUERY_SM
- UNBIND
- DELIVER_SM_RESP

aql to client

- BIND_TRANSMITTER_RESP
- BIND_RECEIVER_RESP
- BIND_TRANSCEIVER_RESP
- DELIVER_SM
- ENQUIRE_LINK
- ENQUIRE_LINK_RESP
- SUBMIT_SM_RESP
- QUERY_SM_RESP
- UNBIND_RESP
- GENERIC_NACK



Binding to aql

There are two options available when binding to aql. 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 aql. 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 aql 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.

Submitting messages to aql

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 aql 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 aql
source_addr_ton	See explanation of source_addr_ton and source_addr_npi at end of this section
source_addr_npi	See explanation of source_addr_ton and source_addr_npi at end of this section
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	The destination number in international format (e.g. 447911111111)
esm_class	As in SMPP specification
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	As in SMPP specification
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 aql and are not supported.

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

aql 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 aql.

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 aql 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.
0x00000401	Number of concurrent connections exceeds the limit for 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 aql.

Parameter	Description
command_length	As in SMPP specification
command_id	As in SMPP specification
command_status	As in SMPP specification
command_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:IIIIIIIII sub:SSS dlvrD:DDD submit date:YMMDDhhmm done date:YMMDDhhmm stat:DDDDDDD err:E Text:"

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

Field	Type	Required	Length	Description
ID	Blank	Yes	10(max)	id supplied by aql in the original SUBMIT_SM PDU
sub	Blank	Yes	3(max)	Always 000
dlvrD	Blank	Yes	3	Always 000
submit date	Blank	Yes	10	Date + time submitted. See SMPP Specification for the format
done date	Blank	Yes	10	Date + time submitted. See SMPP Specification for the format
stat	Blank	Yes	7	Final status of the message. See table below
err	Blank	Yes	3	Associated error code. See table below
text	Alphanumeric	Yes	20	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:

Status 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 aql servers expect a DELIVER_SM_RESP PDU in response to a delivery report.

Terminating the connection

In order to stop the connection to aql you must first issue the UNBIND PDU. aql 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 aql 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 aql. 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>