

SMS SMPP API User Guide



Version	Date	Reason
1.0	03.10.2014	Initial version

- Overview
- Get Started
 - Sign Up
 - Validate Account
 - Log In
 - Buy SMS Plan
 - Get System ID and Password
 - Add Sender Number
- Protocol Definition
 - Supported SMPP Commands
 - SUBMIT_SM Mandatory Parameters
 - Optional Parameters
 - Error Codes
 - Concatenated Messages
 - Sar Optional Parameters
- Establish an SMPP Session
- Send Text SMS
- Close the Connection
- SMS Service Support
 - Data Coding
 - Message Content Support

Overview

This document describes the M800 SMS Short Message Peer-to-Peer Protocol (SMPP) API for the purpose of integrating SMS into your applications and web systems.

The SMPP protocol is an open, industry standard protocol designed to provide a flexible data communications interface for exchanging SMS messages between a Message Center, such as a Short Message Service Centre (SMSC) and a Short Message Service Proxy (SMSP).

Using the SMPP protocol, an External Short Messaging Entity (ESME), such as a promotional marketing tool, may initiate an application layer connection with an SMSP over a TCP/IP network connection in order to send and receive short messages.

The M800 SMPP API supports a full feature set of messaging functions that allow you to:

- Transmit messages from an ESME to a single destination via the SMSC
- Define the data coding type of the short message

The term SMSP will be used throughout this document to describe any SMPP "server" to which an SMPP "client," known as the ESME, can be connected.

Get Started

Sign Up

1. Go to www.m800.com.
2. Click Login/Sign Up.
3. Complete the form by choosing a username and password. The password must be at least eight characters long and must contain both numbers and letters.
4. Click **Sign Up**.

Validate Account

Once you have chosen your username and password, you will receive a confirmation email. Click the **Active Now** button in the email to complete the process of validating and setting up your M800 account.

Log In

Log in with your username and password (the page defaults to the [M800 Dashboard](#)).

- If you forget your password, click the Forgot Password link to reset your password.
- If you forget your username, contact support by live chat or email support@m800.com.

Buy SMS Plan

1. From the left-hand side of your [M800 Dashboard](#), click **SMS > Home > Buy**.
2. Select the plan you want, and then click **Proceed to Checkout**.
3. Check the box to agree to the M800 Service Terms and Conditions, and click **Confirm**.
4. Enter your payment information. If you choose to pay by credit card, your details will be stored in our system to make it easy for you purchase additional products and numbers in the future.

Get System ID and Password

You must purchase an SMS plan before you can get your system ID and password.

1. From the left-hand side of your [M800 Dashboard](#), click **SMS > API**.
2. The Dashboard automatically displays your system ID and password.

Add Sender Number

A sender number is the number that displays when you send an SMS message. You can add up to five sender numbers to your SMS-enabled M800 account.

You must purchase an SMS plan before you can add a sender number.

You must have at least one verified sender number before using the SMS SMPP API to send messages.

1. From the left-hand side of your [M800 Dashboard](#), click **SMS > Source Addresses**.
2. Click the **Add New Number** button.
3. Select your country, enter a mobile number, and then click **Confirm**.
4. When you receive your verification code, enter the code in the pop-up window and click **Confirm** to begin using your sender number.

Protocol Definition

The M800 SMS interface supports the industry standard SMPP V3.3 and V3.4 protocols. Currently, our SMPP server only supports a subset of the SMPP specifications for V3.3 and V3.4, including messages sent from the ESME (transmitter and transceiver) to the SMSC.

Supported SMPP Commands

- bind
- unbind
- submit_sm
- enquire_link

For non-supported SMPP functions or commands, the M800 SMSP will reject the command/request with the corresponding ESME error code.

The submit_sm SMPP commands include mandatory and optional parameters. For mandatory parameters, SMSC will ignore any invalid value/non-supported parameters and reset to the SMSC default value and then deliver the message.

SUBMIT_SM Mandatory Parameters

Parameter	Value
source_addr_ton	1
source_addr_npi	1
source_addr	This is the sender number , which must be in international address format, starting with the country code. The sender number cannot contain any leading "+" or "00." The sender number must be verified. Otherwise, the submit_sm request will be rejected.
dest_addr_ton	1
dest_addr_npi	1
destination_addr	The destination address must be in international format, starting with the country code, and it shall not contain any leading "+" or "00."
protocol ID	GSM Protocol ID (See GSM 03.40 [2] 9.2.3.9)
data_coding	GSM Data-Coding-Scheme (See GSM 03.40 [2] 9.2.3.10)
sm_length	Length of the message text in bytes.
short_message	Up to 140 octets of data. This is the text that is transmitted to the mobile station. <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">Only 'sm_length' bytes will be used.</div>
Others	NULL

Optional Parameters

SMSC has REJECT, FILTER, and PASS-ON options for each optional parameter:

- REJECT – return submit_sm_resp (NACK)
- FILTER – reset the value set to the SMSC default value and process the delivery

The following table includes a list of optional parameters that we support and their hexadecimally-coded tag fields (without descriptions).

Optional Parameter	Value
dest_addr_subunit	0x0005

source_addr_subunit	0x000D
payload_type	0x0019
receipted_message_id	0x001E
ms_msg_wait_facilities	0x0030
privacy_indicator	0x0201
source_subaddress	0x0202
dest_subaddress	0x0203
user_message_reference	0x0204
user_response_code	0x0205
source_port	0x020A
destination_port	0x020B
sar_msg_ref_num	0x020C
language_indicator	0x020D
sar_total_segments	0x020E
sar_segment_seqnum	0x020F
callback_num_pres_ind	0x0302
callback_num_atag	0x0303
number_of_messages	0x0304
callback_num	0x0381
network_error_code	0x0423
message_payload	0x0424
more_messages_to_send	0x0426
message_state	0x0427
ussd_service_op	0x0501
display_time	0x1201
sms_signal	0x1203
ms_validity	0x1204
alert_on_message_delivery	0x130C
its_reply_type	0x1380
its_session_info	0x1383

Error Codes

Error Code	Value	Description
ESME_ROK	0x00000000	No Error
ESME_RINVCMID	0x00000003	Invalid Command ID
ESME_RINVPRTFLG	0x00000006	Invalid Priority Flag

ESME_RINVREGDLVFLG	0x00000007	Invalid Registered Delivery Flag
ESME_RSYSERR	0x00000008	System Error
ESME_RINVSRCADR	0x0000000A	Invalid Sender Number
ESME_RINVSTADR	0x0000000B	Invalid Dest Addr
SME_RMSGQFUL	0x00000014	Message Queue Full
Content filtered	0x00000020	Content filtered
Reserved	0x00000041	Reserved
ESME_RINVESMCLASS	0x00000043	Invalid esm_class field data
ESME_RSUBMITFAIL	0x00000045	submit_sm or submit_multi failed
ESME_RINVSRCTON	0x00000048	Invalid Sender Number TON
ESME_RINVSRCNPI	0x00000049	Invalid Sender Number NPI
ESME_RINVSTTON	0x00000050	Invalid Destination address TON
ESME_RINVSTNPI	0x00000051	Invalid Destination address NPI
ESME_RINVREPFLAG	0x00000054	Invalid replace_if_present flag (Replace_if_present in submit_sm)
ESME_RINVNUMMSGS	0x00000055	Invalid number of messages
ESME_RTHROTTLED	0x00000058	Throttling error (ESME has exceeded allowed message limits)
ESME_RINVSCHEM	0x00000061	Invalid Scheduled Delivery Time
ESME_RINVEXPIRY	0x00000062	Invalid message validity period (Expiry time)
ESME_RINVDFMSGID	0x00000063	Predefined Message Invalid or Not Found
ESME_RX_T_APPN	0x00000064	ESME Receiver Temporary App Error Code
ESME_RX_P_APPN	0x00000065	ESME Receiver Permanent App Error Code
ESME_RX_R_APPN	0x00000066	ESME Receiver Reject Message Error Code
ESME_RINVOPTPARSTREAM	0x000000C0	Error in the optional part of the PDU Body.
ESME_ROPTPARNOTALLWD	0x000000C1	Optional Parameter not allowed
ESME_RINVPARLEN	0x000000C2	Invalid Parameter Length.
ESME_RMISSINGOPTPARAM	0x000000C3	Expected Optional Parameter missing
ESME_RINVOPTPARAMVAL	0x000000C4	Invalid Optional Parameter Value
ESME_RUNKNOWNERR	0x000000FF	Unknown Error
Black listed number	0x0000040E	Fail to pass individual (per SMSC binding) black list
Black listed number	0x00000410	Fail to pass Global SMSC black list
Charging error due to low balance	0x00000400	Error while charging due to low balance

Concatenated Messages

As defined by the [GSM 03.40](#) or [3GPP TS 23.040](#) specification, concatenation is provided by use of a User Data Header (UDH) in each message (submit_sm). A UDH contains and sets the following items:

- The reference group to which the concatenated short message belongs (two versions are available: 8-bits and 16-bits),
- The number of concatenated short messages in that group
- The location of the concatenated short message in the complete sequence.

In order to enable long messages, the UDH must be set in the SMPP parameter esm_class, and special user data header information must be

present in front of the user data.

Following are two examples of UDH in two messages:

```
UDH SM 1 : UDHL=05 IEI(1)=00 IEIDL(1)=03 IED(1)=64 IED(1)=02 IED(1)=01
DATA SM 1 : <text part 1-2>

UDH SM 2 : UDHL=05 IEI(1)=00 IEIDL(1)=03 IED(1)=64 IED(1)=02 IED(1)=02
DATA SM 1 : <text part 2-2>
```

Sar Optional Parameters

The SMPP protocol provides the following optional parameters to enable concatenation. Use of these SMPP parameters in a GSM environment results in the use of the UDHs, but the formatting is left to the SMSC and not to the ESME application, which makes it easier to use concatenation.

Sar Optional Parameters	Description
Sar_msg_ref_num	Originator generated reference number allowing the parallel transmission of several segmented messages
Sar_total_segments	The total number of fragments within the concatenated short message
Sar_segements_seqnum	The sequence number of a particular message within the concatenated short message

Establish an SMPP Session

Establish a network connection for the ESME with the M800 SMSP, and then issue an SMPP Bind request to open an SMPP session.

- OPEN (Connected and Bind Pending)

An ESME has established a network connection to the SMSC but has not yet issued a Bind request. Log in to the [M800 Dashboard](#) to view the access details.

Throughput : 5 sms / sec

Connections : 2 connections max (may be one TX and one RX, or two TRX)

- BOUND_TX

A connected ESME has requested to bind as an ESME Transmitter (by issuing a *bind_transmitter* or *bind_transceiver* PDU) and has received a response from the SMSC authorizing its bind request. Your SMPP credentials can be found in the M800 Dashboard's API Setting when you log in to the M800 portal.

The correspondence with SMPP parameters are shown in the table below:

BIND_TRANSMITTER Syntax	SMPP Dashboard Configuration
<i>system_id</i>	System ID
<i>password</i>	[password generated by M800]
<i>system_type</i>	NOT required
<i>interface_version</i>	33 or 34 (required for TRX)
<i>addr_ton</i>	NULL
<i>addr_npi</i>	NULL
<i>address_range</i>	NULL

Send Text SMS

SUBMIT_SM is used by an ESME to submit a short message to the SMSP for onward transmission to a specified subscriber. The submit_sm PDU does not support the transaction message mode.

An ESME application can send regular text by using the submit_sm request. When using the submit_sm operation, the message text data should be inserted in one of the following two fields:

- the *short_message* field (mandatory field)
- the *message_payload* field (optional field)

Simultaneous use of both fields is not allowed.

When using the short_message field, the sm_length field indicates the length in octets of the short_message field data and must be set. If the message_payload field is used, the sm_length field must be used, as well, but it will be set to 0 to indicate the use of the message_payload field. Up to 254 octets can be inserted in the short_message field, while the message_payload field is able to contain up to 64K octets.

The GSM standard, however, defines a maximum of 140 octets for a single short message and therefore does not support the transmission of more than 140 octets per message. Therefore, a receiving SMSC will usually not accept a submit operation that results in a short message of >140 octets, unless it has implemented an automatic concatenation mechanism, which divides a long message into multiple parts of 140 octets.

Close the Connection

Use the unbind operation to de-register an instance from the SMSP and inform the SMSC that the ESME no longer wants to use the network connection for the submission or delivery of messages. The unbind operation may be viewed as an SMSC log-off request to close the current SMPP session.

Connections that are kept open longer than the server-configured timeout (currently 40 +/- 5 seconds) without activity are automatically closed with an SMSP-initiated unbind operation. To shorten the connection establishment time, keep the connection persistent. For persistent connections, the ESME must maintain the active link ensuring traffic (eg. A heartbeat enquire_link) before this interval has expired.

Since the server cannot send an ENQUIRE_LINK request to the bound connections, we recommend that developers send the ENQUIRE_LINK request by either the ESME or SMSC in order to provide a confidence check of the communication path between the ESME and SMSC. Upon receipt of the request, the receiving party should respond with an enquire_link_resp, thus verifying that the application level connection between the SMSC and the ESME is functioning. The ESME may respond by sending any valid SMPP primitive.

SMS Service Support

Data Coding

When sending text from an ESME to the SMSC, the coding of the short message text inside the SMPP PDU field must be supported by the SMSC and defined in such a way that the SMSC is able to interpret it. When sending a short message from an SMSC to a mobile station, coding of the data determines what is possible at the ESME side. GSM has defined the following text data coding options:

- GSM Default Alphabet (7-bits),
- UCS-2 (2-byte UCS)

The GSM default alphabet looks like the ASCII table (characters 0-127) with the difference that most of the control characters are not present and are replaced by characters from the LATIN-1 upper table (characters 128-255). The SMPP specification offers many coding options, but not every offered coding is by default implemented at the SMSC. The following options are usually implemented at the SMSC:

- SMSC default alphabet (7 or 8-bits)
- LATIN-1 (8-bits)
- US-ASCII (7-bits)
- UCS-2 (2-byte UCS)

Message Content Support

The M800 SMS interface supports the following message encoding:

- GSM default alphabet
- ASCII alphabet
- Binary data such as STK and EMS
- Unicode (Unicode encoding is used to support languages with non-ASCII character sets except Big5)

