SMS 365 SMPP Technical Specification

Version 3.1.0 – April 2019

TABLE OF CONTENTS

IMPORTANT NOTICE:
The information in this document is subject to change without notice and should not be construed as a commitment by Sinch. Sinch assumes no responsibility for any errors that may appear in this document.

1 ABOUT THIS DOCUMENT

1.1 Purpose

The purpose of this document is to publish how SMS 365, enterprise service, supports the SMPP protocol. The intended audience for this document is Sinch's customers who want to communicate with the Application-to-Person (A2P) messaging systems provided by Sinch using the SMPP protocol.

1.2 Scope

The scope of this document is to show how Sinch has implemented the standard SMPP protocol definition.

2 AUDIENCE

This document is for software developers who want to develop software applications that interact with Sinch's messaging systems using SMPP protocol.

The plaform enables many customers to connect to the A2P or P2P messaging systems using the SMPP protocol. This document details how Sinch has implemented the SMPP protocol and what is required from client systems wanting to interact with Sinch's systems using the SMPP protocol. Currently, version 3.4 of the SMPP protocol is supported. Details in this document include: the SMPP commands that are supported, the fully-supported (and partially-supported) parameters for each supported command, and the error codes returned to the originator in cases where a message cannot be processed successfully.

This document should be used in conjunction with two standards associated with SMS:

- The basic Point-to-Point Short Message Service, as defined in ETSI specification ref [GSM03.40] and 3GPP 23.040.
- The Short Message Peer-to-Peer Protocol (SMPP), as defined in ref [SMPP]. SMPP3.4 version is supported as described below.

These are open standard protocols that together — in their generic form — support a full-featured set of messaging functions. Implementation of these open protocols within the network permits the use of a subset of the available messaging functions by customer applications.

SMS messages can be up to 140 octets (8-bit bytes) in length and can carry information coded in different ways. The plaform is using GSM default alphabet “coding” scheme. This enables a simplified text alphabet to be coded into 7 bits per character. These 7-bit characters are packed, by the SMS-C, into the 140 octets, such that a message using the GSM default alphabet can be up to 160 characters long – see ref [SMPP] for more details. More advanced applications will typically use “freeform” 8-bit data, where the SMS-C makes no assumptions on the coding scheme and allows applications to use the 140 octets as they wish.

Each short message has a number of standard “header” elements — in addition to the 140 octet user data — which dictate the behavior of the network and associated applications when handling the message. The following are examples of standard header elements:

- **Validity-Period** — assuming a message has not already been delivered, specifies how long the message remains valid before the operator SMS-C will delete it.
- **Unique message identifier** — enables the SMS-C/users/applications to track and control individual messages.
- **Protocol-Identifier (PID)** — indicates certain type of telematic, interworking, etc.
- **Data Coding Scheme (DCS)** — indicates how the data is encoded within the message, etc.
- **Source & Destination** address of the message.

SMPP is a binary-based protocol. It requires the external system to initiate a login (or “bind”) sequence, with password-based authentication, before accepting any messaging commands from the external system. Available commands on the plaform’s network:

- Submit one or multiple messages
- Deliver MO
- Delivery notifications — can be operator SMS-C Ack, Handset Ack, or a NACK (in the case of a delivery failure)

Note: To receive messages, the customer system is required to provide an “always connected” termination (i.e., must be permanently “bound” in). This enables messages to be delivered as soon as the network receives them, avoiding message queue build-up within the network. The network will not initiate a connection back to the host if the host is not already connected.

3 CONNECTION PROVISIONING PROCESS

The design is based around use of the Internet, as a basic transport, to access the SMPP Access service.

The customer’s configuration is required to support SMPP, TCP/IP routing over the Internet via a fixed/permanent connection.

Sinch will provide the IP address and port customers must use to connect into the Sinch network. Customers must provide their incoming IP address(es) from which they will connect.
All IP addresses required to be provided by the customer, for connection through their equipment, and must be unique and Internet-registered before access is permitted.

Depending on their location clients may want to connect to data centers in Europe or in the USA.

Description	URL	IP address
Europe, VPN	smpp-pi.sapmobileservices.com	178.248.228.2
Europe, non-VPN	smpp-pp.sapmobileservices.com	178.248.228.139
USA, VPN	smpp-ci.sapmobileservices.com	74.117.12.32
USA, non-VPN	smpp-cp.sapmobileservices.com	74.117.12.160

Table 1: Gateway URLs

4 ACRONYMS AND ABBREVIATIONS

PDU – Protocol Data Unit.
SMPP – Short Message Peer to Peer Protocol
SMSC – Short Message Service Center
ESME – External Short Message Entity
UDH – User Data Header
MT – Mobile Terminated
MO – Mobile Originated
EMS – Enhanced Messaging Service
SM – Short Message

5 PROTOCOL DESCRIPTION

5.1 Handling of PDU bytes

Upon receiving PDUs, some validation checks are performed. The PDU is marked invalid if any of the following conditions are met:

- There is a discrepancy between the actual size of the PDU body and that indicated by the PDU header
- Length of the header section is less than 16 bytes or greater than 65536 bytes
- Command_id associated with the PDU is unknown9

For invalid PDUs, a GENERIC_NACK response is sent back to the originator of the PDU. The following table shows the value of command_status field for each type of error condition:

Error	**Command_status**
command_id is unknown	ESME_RINVCMDID
Error in optional part of the PDU	ESME_RINVOPTPARSTREAM
All other error conditions	ESME_RSYSERR

5.2 Opening and Closing session messages

5.2.1 Bind Request

A client can bind in one of three standard SMPP bind modes — Transmitter, Receiver, and Transceiver. All the fields in the Bind Request are supported as per the SMPP Protocol Specification v3.4 barring the below.

system_type

The system_type field can be set to null. The maximum length of the system type is 20 characters.

Example:

Below is an example of a bind request PDU.

0000002600000002000000000000000174657374310074657374310074657374310034000000

PDU Description

00000026: Command length for Bind Request
00000002: Command id for Bind Request
00000000: Command status
00000001: Sequence Number
746573743100: System id
746573743100: Password
746573743100: System type
34000000: Interface version

5.2.2 Bind Response

On receiving a bind request, the service_id of the system requesting to bind is used to find the account associated with the request. Using the account information, the request is authenticated using basic SMPP rules. If the authentication succeeds, a response is sent back to the client with the command status set to ESME_ROK.

Mandatory Parameters

All mandatory fields in the bind response PDU are set as per the SMPP Protocol Specification v3.4.

Error Conditions

The following table shows the error condition and the value of command_status parameter sent in the response in case a request cannot be processed successfully:

Error Condition	Command_status
Account is inactive	ESME_RBINDFAIL
Invalid Password	ESME_RINVPASWD
Invalid non-null System Type	ESME_RINVSYSTYP
Maximum number of connections reached for a particular binding mode for a given account	ESME_RBINDFAIL

Example:

Below is an example of a bind response PDU.
0000001180000002000000000000000100

PDU Description

00000011: Command length
80000002: Command Id for Bind Response
00000000: Command status
00000001: Sequence Number
00: System id

5.2.3 Unbind Request

The format of Unbind Request confirms to the SMPP Protocol Specification v3.4. All the fields in the request are supported as per the specification.

Example:

Below is an example of an unbind request PDU.
00000010000000060000000000000003

PDU Description

00000010: Command length
00000006: Command Id for Unbind request
00000000: Command status
00000003: Sequence Number

5.2.4 Unbind Response

All fields in the unbind response are supported as per the SMPP Protocol Specification v3.4.

Example:

Below is an example of an Unbind response PDU.
00000010800000060000000000000003

PDU Description

00000010: Command length
80000006: Command Id for Unbind response
00000000: Command status
00000003: Sequence Number

5.3 Submit_SM Request

5.3.1 Mandatory Parameters

A majority of SMPP parameters are supported as per the SMPP Protocol Specification v3.4. Below is more specific information on the handling of some mandatory parameters:

service_type

The service_type is not checked and can be set to null.

source and destination addresses

E164 is the preferred format for the source and destination addresses fields. However, this is not mandated.

source_addr_ton and source_addr_npi

The enterprise messaging system does not check the source_addr_ton and source_addr_npi values.

dest_addr_ton and dest_addr_npi

These fields must be set to International (1/1).

Data Encoding for the Short Message

For text messaging only GSM Message Class Control (GSM 3.38) is supported, only the least significant 4 bits will be considered as the character code. For example if the data_encoding parameter is 0xF1 the F will be stripped off and the data_encoding parameter value will be 0x01.

If certain characters are not supported by the server, they may be converted to other supported characters.

Also, the short message text is not null terminated and this is not configurable.

The default supported character set is the Default GSMS Character Set (DGCS) – this is configured on the server.
Latin1 character set is also supported; please contact your Account representative should you want your account to be configured to support this charset.

schedule_delivery_time

This field is not supported by SMS 365, enterprise service.

validity_period

If set to NULL AP system will use its default value of 48 hours.
To avoid issues with Daylight Saving Time changes and time zones it is recommended to use a relative validity period (as opposed to an absolute validity period).

registered delivery

This field is passed on to the SMSC. If set to 0 no delivery receipt will be sent to the client. If set to 1 the confirmation that the message was delivered to the SMSC (SMSC-ack) will be delivered in a deliver_sm operation and, if supported by the SMSC, the confirmation that the message was delivered to the handset (handset-ack) will be delivered in a deliver_sm operation too.

replace_if_present

replace_if_present flag is not supported.
sm_default_msg_id
This field is not supported.

5.3.2 Optional Parameters

The table below shows all the optional parameters in the SMPP Protocol Specification v3.4 and marks those which are supported by SMS 365, enterprise service.

Parameter Name	**Description**	**Ref. in SMPP Specification v3.4**	**Supported**
user_message_reference	ESME assigned message reference number.	5.3.2.17	No
source_port	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications.	5.3.2.20	Yes
source_addr_subunit	The subcomponent in the destination device which created the user data.	5.3.2.2	Yes
destination_port	Indicates the application port number associated with the destination address of the message.	5.3.2.21	Yes
dest_addr_subunit	The subcomponent in the destination device for which the user data is intended.	5.3.2.1	No
sar_msg_ref_num	The reference number for a particular concatenated short message.	5.3.2.22	No
sar_total_segments	Indicates the total number of short messages within the concatenated short message.	5.3.2.24	No
more_messages_to_send	Indicates that there are more messages to follow for the destination SME.	5.3.2.34	Yes
payload_type	defines the type of payload (e.g. WDP, WCMP, etc.).	5.3.2.10	No
message_payload	Contains the extended short message user data.	5.3.2.32	No
privacy_indicator	Indicates the level of privacy associated with the message.	5.3.2.14	No
callback_num	A callback number associated with the short message.	5.3.2.36	No
callback_num_pres_ind	Defines the callback number presentation and screening.	5.3.2.37	No
callback_num_atag	Associates a displayable alphanumeric tag with the callback number.	5.3.2.38	No
source_subaddress	The subaddress of the message originator.	5.3.2.15	No
dest_subaddress	The subaddress of the message destination.	5.3.2.16	No
user_response_code	A user response code.	5.3.2.18	No
display_time	Provides the receiving MS with a display time associated with the message.	5.3.2.26	No
sms_signal	Indicates the alerting mechanism when the message is received by an MS.	5.3.2.40	No

ms_validity	Indicates validity information for this message to the recipient MS.	5.3.2.27	No
ms_msg_wait_facilities	This parameter controls the indication and specifies the message type (of the message associated with the MWI) at the mobile station.	5.3.2.13	No
number_of_messages	Indicates the number of messages stored in a mail box	5.3.2.39	No
alert_on_msg_delivery	Request an MS alert signal be invoked on message delivery.	5.3.2.41	No
language_indicator	Indicates the language of an alphanumeric text message.	5.3.2.19	No
its_reply_type	The MS user's reply method to an SMS delivery message received from the network is indicated and controlled by this parameter.	5.3.2.42	No
its_session_info	Session control information for Interactive Teleservice.	5.3.2.43	No
ussd_service_op	This parameter is used to identify the required USSD Service type when interfacing to a USSD system.	5.3.2.44	No

5.3.3 Examples:

Short message in UCS2
Full PDU:
0000008200000004000000000000008D0003093037393038373837333733000101343437393038373837333733000000000000303030303032303030303030303030305200000008003A757662115011540C57284E008D7757284E008D7757284E008D77002E002E002E002E0020000A0020004D006F00620069006C0065003300360035

Header: <0000008200000004000000000000008D>

[0] command_length: 00000082
[1] command_id: 00000004
[2] command_status: 00000000
[3] sequence_number: 0000008D

Mandatory:
<0003093037393038373837333733000101343437393038373837333733000000000000303030303032303030303030303030305200000008003A757662115011540C57284E008D7757284E008D7757284E008D77002E002E002E002E0020000A0020004D006F00620069006C0065003300360035>

[4] service_type: 00
[5] source_addr_ton: 03
[6] source_addr_npi: 09
[7] source_addr: 303739303837383733373300
[8] dest_addr_ton: 01
[9] dest_addr_npi: 01
[10] destination_addr: 34343739303837383733373300
[11] esm_class: 00
[12] protocol_id: 00
[13] priority_flag: 00
[14] schedule_delivery_time: 00
[15] validity_period: 3030303030323030303030303030305200
[16] registered_delivery: 00

```
[17] replace_if_present_flag:    00
[18] data_coding:                08
[19] sm_default_msg_id:          00
[20] sm_length:                  3A
[21] short_message:
757662115011540C57284E008D7757284E008D7757284E008D77002E002E002E002E0020000A0020004
D006F00620069006C0065003300360035
```

Concatenated text message

```
Part 1:
Ful PDU:
000000E7000000040000000000000091000309303739303837383733373300010134343739303837383733
3733004000000030303030303230303030303030303052000000000009F0500037D020153544152540A0D5
4686973206D657373616765206973207468652070726F7065727479206F66205379626173653336352C20
496E632E204974206D6179206265206C6567616C6C792070726976696C6567656420616E642F6F722063
6F6E666964656E7469616C20616E6420697320696E74656E646564206F6E6C7920666F72207468652075
7365206F662074686520616464726573736565

        Header: <000000E7000000040000000000000091>

[0] command_length:          000000E7
[1] command_id:              00000004
[2] command_status:          00000000
[3] sequence_number:         00000091

Mandatory:
<0003093037393038373837333733000101343437393038373837333733004000000030303030303230303
030303030303052000000000009F0500037D020153544152540A0D54686973206D65737361676520697320
7468652070726F7065727479206F66205379626173653336352C20496E632E204974206D6179206265206
C6567616C6C792070726976696C6567656420616E642F6F7220636F6E666964656E7469616C20616E642
0697320696E74656E646564206F6E6C7920666F722074686520757365206F662074686520616464726573
736565>

[4] service_type:            00
[5] source_addr_ton:         03
[6] source_addr_npi:         09
[7] source_addr:             303739303837383733373300
[8] dest_addr_ton:           01
[9] dest_addr_npi:           01
[10] destination_addr:       34343739303837383733373300
[11] esm_class:              40
[12] protocol_id:            00
[13] priority_flag:          00
[14] schedule_delivery_time:     00
[15] validity_period:        30303030303230303030303030305200
[16] registered_delivery:    00
[17] replace_if_present_flag:    00
[18] data_coding:            00
[19] sm_default_msg_id:      00
[20] sm_length:              9F
[21] short_message:
0500037D020153544152540A0D54686973206D657373616765206973207468652070726F7065727479206
F66205379626173653336352C20496E632E204974206D6179206265206C6567616C6C792070726976696
C6567656420616E642F6F7220636F6E666964656E7469616C20616E6420697320696E74656E646564206
F6E6C7920666F722074686520757365206F662074686520616464726573736565


Part 2:
Full PDU:
```

```
000000E60000000400000000000000090000309303739303837383733373300010134343739303837383733
37330040000000303030303032303030303030303030305200000000009E0500037D02022873292E4E6F206
164647265737365652073686F756C6420666F72776172642C207072696E742C20636F70792C206F72206
F746865727769736520726570726F647563652074686973206D657373616765206E20616E79206D616E
6E65722074686174206F756C6420616C6C6F7720697420746F20626520766965776564206279206E7
920696E646976696475616C2E0A0D454E44

        Header: <000000E6000000040000000000000090>

[0] command_length:           000000E6
[1] command_id:             00000004
[2] command_status:           00000000
[3] sequence_number:           00000090

Mandatory:
<00030930373930383738373337330001013434373930383738373337330040000000303030303032303030
30303030303030305200000000009E0500037D02022873292E4E6F2061646472657373656520736865C64
20666F72776172642C207072696E742C20636F70792C206F72206F7468657277697365207265707
26F647563652074686973206D657373616765206E20616E79206D616E6E65722074686174206F756C64206
16C6C6F7720697420746F20626520766965776564206279206E7920696E646976696475616C2E0A0D4
54E44>

[4] service_type:              00
[5] source_addr_ton:             03
[6] source_addr_npi:             09
[7] source_addr:               3037393038373837333733300
[8] dest_addr_ton:              01
[9] dest_addr_npi:              01
[10] destination_addr:           3434373930383738373337330
[11] esm_class:                40
[12] protocol_id:              00
[13] priority_flag:             00
[14] schedule_delivery_time:     00
[15] validity_period:           303030303032303030303030303030305200
[16] registered_delivery:        00
[17] replace_if_present_flag:    00
[18] data_coding:               00
[19] sm_default_msg_id:         00
[20] sm_length:                9E
[21] short_message:
0500037D02022873292E4E6F206164647265737365652073686F756C6420666F72776172642C207072696
E742C20636F70792C206F72206F74686572776973652072657072F647563652074686973206D65737361
676520696E20616E79206D616E6E65722074686174206F756C6420616C6C6F7720697420746F206265
2076696577656420627920616E7920696E646976696475616C2E0A0D454E44
```

Wap push

Full PDU:

```
00000069000000040000000000000193000500737962617365333635000101343437383235333336343438
0040000000303030303032303030303030303030305200000000400230605040B8423F0300601AE02056A004
5C60C037379626173652E636F6D000103000101

        Header: <00000069000000040000000000000193>

[0] command_length:           00000069
[1] command_id:             00000004
[2] command_status:           00000000
[3] sequence_number:           00000193
```

Mandatory:

```
<000500737962617365333635000101343437383235333336343438004000000030303030303230303030303
03030303030520000000400230605040B8423F0300601AE02056A0045C60C037379626173652E636F6D000
103000101>
```

```
[4] service_type:              00
[5] source_addr_ton:           05
[6] source_addr_npi:           00
[7] source_addr:               737962617365333635 00
[8] dest_addr_ton:             01
[9] dest_addr_npi:             01
[10] destination_addr:         3434373832353333363434 3800
[11] esm_class:                40
[12] protocol_id:              00
[13] priority_flag:            00
[14] schedule_delivery_time:   00
[15] validity_period:          30303030303230303030303030305200
[16] registered_delivery:      00
[17] replace_if_present_flag:  00
[18] data_coding:              04
[19] sm_default_msg_id:        00
[20] sm_length:                23
[21] short_message:
0605040B8423F0300601AE02056A0045C60C037379626173652E636F6D000103000101
```

Text message with handset ack requested

Full PDU:

```
000000510000000400000000000019D000500537962617365333635000101343437393038373837333733
00000000003030303030323030303030303030305200010000000B48656C6C6F20576F726C64
```

```
        Header: <000000510000000400000000000019D>

[0] command_length:            00000051
[1] command_id:                00000004
[2] command_status:            00000000
[3] sequence_number:           0000019D
```

Mandatory:

```
<000500537962617365333635000101343437393038373837333733000000000030303030303230303030303
030303030520001000000B48656C6C6F20576F726C64>
```

```
[4] service_type:              00
[5] source_addr_ton:           05
[6] source_addr_npi:           00
[7] source_addr:               537962617365333635 00
[8] dest_addr_ton:             01
[9] dest_addr_npi:             01
[10] destination_addr:         3434373930383738373337 3300
[11] esm_class:                00
[12] protocol_id:              00
[13] priority_flag:            00
[14] schedule_delivery_time:   00
[15] validity_period:          30303030303230303030303030305200
[16] registered_delivery:      01
[17] replace_if_present_flag:  00
[18] data_coding:              00
[19] sm_default_msg_id:        00
[20] sm_length:                0B
[21] short_message:            48656C6C6F20576F726C64
```

[22] opt_params:

5.4 submit_sm Response

All fields in the submit_sm response are supported as per the SMPP Protocol Specification v3.4.

The orderid provided in the *message_id* is in hexadecimal format.

Response codes:

ESME_ROK (0x00000000) ==> No Error

ESME_RINVCMDID (0x00000003) ==> unknown command ID

ESME_RINVOPTPARSTREAM (0x000000C0) ==> invalid optional parameters

ESME_RINVOPTPARAMVAL (0x000000C4) ==> unknown optional parameters

ESME_RINVSRCADR (0x0000000A) ==> invalid source address

ESME_RINVDSTADR (0x0000000B) ==> invalid destination address

ESME_RINVEXPIRY (0x00000062) ==> invalid message validity period

ESME_RSYSERR (0x00000008) ==> for all other cases

5.5 deliver_sm Request

5.5.1 Mandatory Parameters

A majority of SMPP parameters are supported as per the SMPP Protocol Specification v3.4. Below is more specific information on the handling of some mandatory parameters.

What TON and NPI will we set?

source_addr_ton and source_addr_npi

SMS 365, enterprise service, does not provide information about TON and NPI associated with the source address as this is not applicable for an application origination case. Hence when composing a SMPP PDU for deliver_sm, these values are set to 0.

If customers were connected to a server setting different TON/NPI values have to be aware that this server always set the values to 00.

service_type

The service_type is set to null.

Customers who were connected to a server setting service_type to CMT have to be aware the service_type is set to null.

short_message

The short message is not null terminated.

Customers used to receive null terminated short_message have to be aware of the difference.

5.5.2 Optional Parameters

The table below shows all the optional parameters in the SMPP Protocol Specification v3.4 and marks those which are supported by SMS 365, enterprise service.

This list is subject to change, therefore it is expected CP's applications are designed in a way to accept future implementations of optional parameters.

Parameter Name	Description	Ref. in SMPP Specification v3.4	Supported
user_message_reference	A reference assigned by the originating SME to the message. In the case that the deliver_sm is carrying an SMSC delivery receipt, an SME delivery acknowledgement or an SME user acknowledgement (as indicated in the esm_class field), the user_message_reference parameter is set to the message reference of the original message.	5.3.2.17	No
source_port	Indicates the application port number associated with the source address of the message. The parameter should be present for WAP applications.	5.3.2.20	Yes
destination_port	Indicates the application port number associated with the destination address of the message. The parameter should be present for WAP applications.	5.3.2.21	Yes
sar_msg_ref_num	The reference number for a particular concatenated short message.	5.3.2.22	No
sar_total_segments	Indicates the total number of short messages within the concatenated short message.	5.3.2.23	No
sar_segment_seqnum	Indicates the sequence number of a particular short message fragment within the concatenated	5.3.2.24	No

	short message.		
user_response_code	A user response code. The actual response codes are SMS application specific	5.3.2.18	No
privacy_indicator	Indicates a level of privacy associated with the message.	5.3.2.14	No
payload_type	Defines the type of payload (e.g. WDP, WCMP, etc.)	5.3.2.10	No
message_payload	Contains the extended short message user data. Up to 64K octets can be transmitted. **Note:** The short message data should be inserted in either the short_message or message_payload fields. Both fields should not be used simultaneously. The sm_length field should be set to zero if using the message_payload parameter.	5.3.2.32	No
callback_num	A callback number associated with the short message. This parameter can be included a number of times for multiple call back addresses.	5.3.2.36	No
source_subaddress	The subaddress of the message originator.	5.3.2.15	No
dest_subaddress	The subaddress of the message destination.	5.3.2.16	No
language_indicator	Indicates the language of an alphanumeric text message.	5.3.2.19	No
its_session_info	Session control information for Interactive Teleservice.	5.3.2.43	No
Network_error_code	Network Error Code.	May be present for Intermediate Notifications and SMSC Delivery Receipts	No
message_state		5.3.2.35	No
receipted_message_id		5.3.2.12	No

5.5.3 Example:

The below PDU is of a SMS sent from +61404885254 to +61416907006 with text “Dosnew”

0000003E00000005000000000000000200000036313430343838353235340000002B363134313639303730
3036000000000000000000006446F736E6577

Header: <0000003E000000050000000000000002>
[0] command_length: 0000003E
[1] command_id: 00000005
[2] command_status: 00000000
[3] sequence_number: 00000002

Mandatory:
<00000036313430343838353235340000002B36313431363930373030360000000000000000000006446F7
36E6577>

[4] service_type: 00
[5] source_addr_ton: 00
[6] source_addr_npi: 00
[7] source_addr: 363134303438383532353400
[8] dest_addr_ton: 00
[9] dest_addr_npi: 00

```
[10] destination_addr:       2B3631343136393037303036 00
[11] esm_class:             00
[12] protocol_id:          00
[13] priority_flag:       00
[14] schedule_delivery_time:     00
[15] validity_period:       00
[16] registered_delivery:     00
[17] replace_if_present_flag:   00
[18] data_coding:           00
[19] sm_default_msg_id:         00
[20] sm_length:            06
[21] short_message:           446F736E6577
```

5.6 deliver_sm Response

All fields in the deliver_sm response are supported as per the SMPP Protocol Specification v3.4.

If a deliver_sm response indicates that there was an error in delivery, the message is retried after a fixed, configurable time interval. The delivery will be retried until the message is delivered successfully.

5.7 submit_multi Request

*5.7.1 **Mandatory Parameters***

It is possible to send up to 100 MSISDNs per submit_multi request, however this requires specific configuration of the account. Please contact your account representative should you desire to use this feature.

All mandatory SMPP parameters are supported as per the SMPP Protocol Specification v3.4 and paragraph #**Error! Reference source not found.**(**Error! Reference source not found.**).

*5.7.2 **Optional Parameters***

The table below shows the optional parameters which are supported and not supported:

Parameter Name	**Description**	**Ref. in SMPP Specification v3.4**	**Supported**
User_message_reference	ESME assigned message reference number	5.3.2.17	No
source_port	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications	5.3.2.20	Yes
source_addr_subunit	The subcomponent in the destination device which created the user data.	5.3.2.2	No
destination_port	Indicates the application port number associated with the destination address of the message. This parameter should be present for WAP applications	5.3.2.21	Yes
Dest_addr_subunit	dest_addr_subunit TLV The subcomponent in the destination device for which the user data is intended.	5.3.2.1	No
sar_msg_ref_num	The reference number for a particular concatenated short message.	5.3.2.22	No
sar_total_segments	Indicates the total number of short messages within the concatenated short message.	5.3.2.23	No
sar_segment_seqnum	Indicates the sequence number of a particular short message fragment within the concatenated short message.	5.3.2.24	No
payload_type	Defines the type of payload (e.g. WDP, WCMP, etc.)	5.3.2.10	No
message_payload	Contains the extended short message user data. Up to 64K octets can be transmitted. Note: The short message data should be inserted in either the short_message or message_payload fields. Both fields should not be used simultaneously. The sm_length field should be set to zero if using the message_payload parameter.	5.3.2.32	No
privacy_indicator	Indicates the level of privacy associated with the message	5.3.2.14	No
callback_num	A callback number associated with the short message. This parameter can be included a number of times for multiple callback addresses.	5.3.2.36	No
callback_num_pres_ind	callback_num_pres_ind TLV Identifies the presentation and screening associated with the callback number. If this parameter is present and there are multiple instances of the callback_num	5.3.2.37	No

	parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular callback_num_pres_ind which corresponds to a particular callback_num.		
callback_num_atag	Associates a displayable alphanumeric tag with the callback number. If this parameter is present and there are multiple instances of the callback_num parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular callback_num_atag which corresponds to a particular callback_num.	5.3.2.38	No
source_subaddress	The subaddress of the message originator.	5.3.2.15	No
Dest_subaddress	The subaddress of the message destination.	5.3.2.16	No
display_time	Provides the receiving MS based SME with a display time associated with the message.	5.3.2.26	No
Sms_signal	Indicates the alerting mechanism when the message is received by an MS.	5.3.2.40	No
ms_validity	ms_validity TLV Indicates validity information for this message to the recipient MS.	5.3.2.27	No
ms_msg_wait_facilities	This parameter controls the indication and specifies the message type (of the message associated with the MWI) at the mobile station.	5.3.2.13	No
Alert_on_msg_delivery	Requests an MS alert signal be invoked on message delivery.	5.3.2.41	No
Language_indicator	Indicates the language of an alphanumeric text message.	5.3.2.19	No

5.7.3 Error Conditions

In a submit_multi request, if the number of destination addresses is greater than 254, the command_status field in the response is set to error code ESME_RINVNUMDESTS. However, the customer is advised to limit the number of destination addresses to 100.

5.8 submit_multi Response

All fields in the deliver_sm response are supported as per the SMPP Protocol Specification v3.4.

5.9 generic_nack

All the fields are supported as per the SMPP Protocol Specification v3.4. If an unknown or invalid command Id is received, the generic_nack response PDU is returned to the sender.

5.10 enquire_link

5.10.1 Sending enquire_link Request to a customer

All fields in the enquire_link request are supported as per the SMPP Protocol Specification v3.4.Enquire Link is sent only when receiver of the enquire_link request is bound and enquiry is enabled. An enquire_link request is sent every 60 seconds.

Example:
Below is an example of an Enquire Link Request PDU.
00000010000000150000000000000000

PDU Description
00000010: Command length
00000015: Command Id for Enquire link request

00000000: Command status
00000000: Sequence number

5.10.2 Receiving enquire_link Response from customer

No action is taken on receiving an enquire_link response from the customer other than letting the system knows that the bind is active.

5.10.3 Receiving enquire_link Request from customer

On receiving an enquire_link request, an enquire_link response is sent back. This response follows the SMPP Protocol Specification v3.4. The customer needs to send an enquire link request to the messaging system every 60 seconds. Despite it is recommended that the client sends enquire_links when there is no activity on the bind, a connection is not broken if an enquire link request is not received from the client as the SMPP server will send an enquire_link when there is no activity on the bind.

Example:
Below is an example of an Enquire Link Response PDU.
00000010800000150000000000000002

PDU Description
00000010: Command length
80000015: Command Id for Enquire link response
00000000: Command status
00000002: Sequence number

6 SMS 365, ENTERPRISE SERVICE, VENDOR SPECIFIC OPTIONAL PARAMETERS

The platform supports Vendor Specific optional parameter implemented as per #5.3 of SMPP Protocol Specification v3.4 issue 1.2.

The plaform might introduce new optional parameters in the future, the content provider application shall support any additional parameter and discard the information provided if it is not of any use for their service.

6.1 List of the Parameters

As per SMPP specification, the range for vendor specific TAG identifiers are 0x1400 to 0x3FFF. The platform did chose to start at 0x1600.

The table below provides a summary of the vendor specific parameters available with the platform SMPP API.
The table provides the context in which the parameter is used: either MT, MO or Delivery Report.
Full details are provided for each TAG identifier in the paragraphs below.

TAG ID	Decimal	Name	Context	Value Type
0x1600	5632	Operator ID	MT/MO/DR	C-Octet String (length >= 1)
0x1601	5633	Session ID	MT/MO	C-Octet String (length >= 1)
0x1602	5634	Customer Reference / Subject	MT/DR	C-Octet String (length >= 1)
0x1603	5635	SMSC ID	MT/MO	C-Octet String (length >= 1)
0x1604	5636	Final Status Indication	DR	C-Octet String (length = 1): Y|N
0x1605	5637	Status ID	DR	C-Octet String (length >= 1)
0x1606	5638	Internal Status ID	DR	C-Octet String (length >= 1)
0x1607	5639	Request for final status indication	MT	Integer (1 byte): 0|1
0x1608	5640	Request for OperatorID	MT	Integer (1 byte): 0|1
0x1609	5641	Request for internal status ID	MT	Integer (1 byte): 0|1
0x160A	5642	Request for subject	MT	Integer (1 byte): 0|1
0x160B	5643	Request for status ID	MT	Integer (1 byte): 0|1
0x160C	5644	Request for OrderID	MT	Integer (1 byte): 0|1|2

6.2 Detailed Parameters Information

6.2.1 Operator ID

In the MT context:
OperatorID parameter could be used within the Submit_SM in order to get the destination of the SMS enforced to that specific operator. To do so, the TAGID 0x1600 shall be added to the Submit_SM and the operatorID shall be specified.

If there is no OperatorId specified, the platform will determine the destination operator based on its global numbering plan. In order to specify an OperatorId, please contact your Account representative for a list of connected operators and their corresponding OperatorIds. The network can only accept OperatorIds from this list.

In the MO context:
OperatorID tag could also be added to the MO sent by the platform in order to specify the origin operator of the MO request (to which the origin MSISDN belongs to).

In order to receive Operator ID in MO, the account configuration is required, please contact your account manager to enable this parameter in MO

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x1600	5632	Operator ID	C-Octet String (length >= 1)	Force MT to be delivered thru Operator ID
MO	0x1600	5632	Operator ID	C-Octet String (length >= 1)	Operator ID

In the DR context:
The Destination operator ID shall be returned in the DR as part of the OPERATORID identifier is being sent as an additional parameter to the existing DR.

In order to receive Operator ID in DR, the corresponding request TAG ID (0x1608) must be set in Submit_SM

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x1608	5640	Request for Operator ID	Integer (1 byte): 0|1	Whether to receive operator ID optional parameter in DR
DR	0x1600	5632	Operator ID	C-Octet String (length >= 1)	Operator ID ***(See sample 8.1)***

6.2.2 *Session ID*

SESSION ID is used for session tracking or other purposes. When this field is present, its contents will need to be transferred directly to the SESSION ID field for each MT associated with this MO.

Depending on the service, this field may be used for other information. If so, this will be discussed by the account manager.

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x1601	5633	Session ID	C-Octet String (length >= 1)	Used for session tracking or other purposes
MO	0x1601	5633	Session ID	C-Octet String (length >= 1)	Used for session tracking or other purposes

6.2.3 *Customer Reference: Subject*

In the MT context:
Used to specify a customer reference or a subject matter that could be sent back by the platform in the Delivery Reports.

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x1602	5634	Subject	C-Octet String (length >= 1)	Used to specify a customer reference or a subject matter

In the DR context:
The original Subject set in Submit_SM is returned in the DR as part of the Subject is being sent as an additional parameter to the existing DR.

In order to receive Subject in DR, the corresponding request TAG ID (0x160A) must be set in Submit_SM

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x160A	5642	Request for Subject	Integer (1 byte): 0|1	Whether to receive Subject optional parameter in DR
DR	0x1602	5634	Subject	C-Octet String (length >= 1)	Subject ***(See sample 8.3)***

6.2.4 *Specific routing Identifier: SMSC ID*

In the MT context:
SMSC D parameter could be used within the Submit_SM in order to get the destination of the SMS enforced to that specific SMSC. To do so, the TAGID 0x1603 shall be added to the Submit_SM and the SMSC ID shall be specified.

If there is no SMSC ID specified, the platform will determine the destination SMSC based on its global numbering plan and routing settup. In order to specify an SMSC ID, please contact your Account representative for a list of connected SMSC and their corresponding SMSC IDs. The network can only accept SMSC IDs from this list.

In the MO context:
SMSC D tag could also be added to the MO sent by the platform in order to specify the origin SMSC of the MO request (to which the origin MO was received from).

In order to receive SMSC ID in MO, the account configuration is required, please contact your account manager to enable this parameter in MO

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x1603	5635	SMSC ID	C-Octet String (length >= 1)	Force MT to be delivered thru SMSC ID
MO	0x1603	5635	SMSC ID	C-Octet String (length >= 1)	SMSC ID

6.2.5 *Final Status Indicator*

DR optional parameter that indicates if this DR is the final status.

In order to receive Final Status Indicator in DR, the corresponding request TAG ID (0x1607) must be set in Submit_SM

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x1607	5639	Request for Final Status Indicator	Integer (1 byte): 0|1	Whether to receive Final Status Indicator optional parameter in DR
DR	0x1604	5636	Final Status Indicator	C-Octet String (length >= 1)	Final Status Indicator: Y or N ***(See sample 8.2)***

6.2.6 *Status Identifier: Status ID*

DR optional parameter that provides Status ID.

In order to receive Status ID in DR, the corresponding request TAG ID (0x160B) must be set in Submit_SM

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x160B	5643	Request for Status ID	Integer (1 byte): 0|1	Whether to receive Status ID optional parameter in DR
DR	0x1605	5637	Status ID	C-Octet String (length >= 1)	External Status ID (45XX or DBXX) ***(See sample 8.6)***

6.2.7 *Internal Status Identifier: Internal Status ID*

DR optional parameter that provides Internal Status ID.

In order to receive Internal Status ID in DR, the corresponding request TAG ID (0x1609) must be set in Submit_SM

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x1609	5641	Request for Internal Status ID	Integer (1 byte): 0|1	Whether to receive Internal Status ID optional parameter in DR
DR	0x1606	5638	Internal Status ID	C-Octet String (length >= 1)	Internal Status ID (45XX or DBXX) ***(See sample 8.7)***

6.2.8 *Order ID*

DR optional parameter that provides Order ID.

In order to receive Order ID optional parameter in DR, the corresponding request TAG ID (0x160C) must be set in Submit_SM

Context	TAG ID	Decimal	Name	Value Type	Description
MT	0x160C	5644	Request for Order ID	Integer (1 byte): 0|1|2	Whether to receive Order ID optional parameter in DR 0: default, no OrderID and same format as existing 1: request for OrderID in DR and *HEX* format for all OrderIDs 2: request for OrderID in DR and *DEC* format for all OrderIDs
DR	0x001E	30	Order ID	C-Octet String (length >= 1)	Order ID in Hex or Dec format ***(See sample 8.4, 8.5)***

7 DELIVERY NOTIFICATIONS FORMAT

Delivery Notifications are message acknowledgments sent from the network to customers. When delivering messages through our network, customers can request on the SMPP interface to receive the following acknowledgment levels: SMS-C Ack and Handset Ack. For unsuccessful delivery attempts, the platform returns a negative acknowledgment (NACK) outlining the failure reason.

7.1 Request Format

Delivery Notifications are requested as part of the submit_sm command.

7.2 Delivery Notifications Returned on the SMPP Interface

Notification Status	Status Code	Status Description	Delivery Notification Message Description Sent to SMPP client	Retry?
Temporary Positive	450A	SMS-C Ack, waiting for Handset Ack	id:1891273320 sub:001 dlvrd:001 submit date:20090826152631 done date: 20090826152631 stat:ACCEPTD err:000 Text:DLVRD TO SMSC	N
Final Positive	450A	SMS-C Ack, not waiting for Handset Ack	id:1891273320 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:ACCEPTD err:000 Text:DLVRD TO SMSC	N
	451A	Handset Ack	id:1891273321 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:DELIVRD err:000 Text:DLVRD TO MOBILE	N
Temporary Negative	DBF3	SMS 365 is retrying delivery	id:1891273310 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:ENROUTE err:000 Text:DBF3 RETRYING	N
Final Negative	4524	MSISDN is blacklisted	id:1891273311 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:REJECTD err:000 Text:4524 BLACKLISTED	N
	4503	No operator found for current MSISDN	id:1891273311 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat: UNDELIV err:000 Text:4503 OUT OF COVERAGE	N
	DB05	Call barred by operator	id:1891273313 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:REJECTD err:000 Text:DB05 SMSC ERROR	N
	DB52	Failed Message Delivery	id:1891273314 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:REJECTD err:000 Text:DB52 SMSC ERROR	N
	DB76	Subscriber profile does not permit service	id:1891273317 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:REJECTD err:000 Text:DB76 NO SERVICE	N
	DB99	Ported Number	id:1891273319 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:REJECTD err:000 Text:DB99 PORTED NUMBER	N
	DB62	Failed Message Delivery	id:1891273315 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:REJECTD err:000 Text:DB62 SMSC ERROR	Y
	DB65	Validity Period Expired	id:1891273316 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:EXPIRED err:000 Text:DB65 VP EXPIRED	Y
	DB78	Subscriber is temporarily out of credit	id:1891273318 sub:001 dlvrd:001 submit date: 20090826152631 done date: 20090826152631 stat:REJECTD err:000 Text:DB78 NO BALANCE	Y

- **What is a Temporary Positive Notification Status?**

A temporary positive notification status indicates to a customer an interim delivery confirmation, with a final delivery confirmation yet to be returned.

- **What is a Final Positive Notification Status?**

A final positive notification status is returned when the message has been delivered to the operator's SMS-C (when no handset Ack is requested or the operator does not support Handset Ack) or to the handset (when a Handset Ack is requested and supported by the destination operator). In both cases, the status is final and no additional notifications will be sent for that message.

- **What is a Temporary Negative Notification Status?**

A Temporary Negative notification status is returned when the platform is attempting a retry on a message that previously failed to deliver. In such cases, customers should expect a final status to be sent at the conclusion of the retry process, whether positive or negative. Until a final status is received, customers should not be initiating their own retry, as it could potentially result in duplicate messages arriving at the handset.

- **What is a Final Negative Notification?**

A Final Negative notification is returned when the platform could not deliver a particular message and will not attempt a retry. The returned notification code will indicate to the customer whether the message is retriable or not, in which case the customer can initiate their own retry process. There are no additional notifications sent after a final negative status.

- **What is a Retriable Status?**

A retriable status means that the platform will allow the customer to retry delivery of the message, although the platform itself will not be attempting to re-deliver.

7.3 Delivery Notification Format

"id:<10 decimal digit> sub:<3 decimal digits> dlvrd:<3 decimal digits> submit date:<YYYYMMDDhhmmss> done date:< YYYYMMDDhhmmss> stat:<7 characters> err:000 text:<20 characters max; terminated by NULL character>"

- id: 10 decimal digit = unique number which was returned on the submit_sm_resp. If more than 10 digits, number is truncated.
- sub: 3 decimal digits, corresponding to number of submitted message. Always set to 001
- dlvrd: 3 decimal digits, corresponding to number of delivered message. Always set to 001
- submit date: YYYYMMDDhhmmss; operator submission date, in CEST format.
- done date: YYYYMMDDhhmmss; reception date of the message by the SMS-C, or by the enduser depending on the delievry_receipt flag value, in CEST format.
- stat: 7 characters
  - ACCEPTD: Indicates a temporary positive acknowledgment (SMS-C)
  - DELIVRD: Indicates a final positive acknowledgment (SMS-C or Handset Ack).
  - ENROUTE: Indicates the platform is retrying after the initial message failed delivery.
  - UNDELIV: Indicates a delivery failure where no route was found to deliver the message to this MSISDN.
  - EXPIRED: Indicates a delivery failure due to validity period expiration. Messages rejected due to validity period expiration are retriable.
  - REJECTD: Indicates a delivery failure. See the table above for the retriable status of each REJECTD notification code.
- err: 3 digits. Always set to 000
  - Text: 20 characters max; terminated by NULL character. See table above for message description returned with each delivery notification.

8 APPENDIX A: OPTIONAL PARAMETERS SAMPLES

8.1 Client request for OperatorID in DR

```
-- SubmitSM for MT received
PDU HEX dump:
0000003E0000000400000000000006EB00000033333833330000002B3636313233343536373800000000000000010000000874
65737420736D731608000101

PDU decoded : (submit: (pdu: 62 4 0 1771) (addr: 0 0 33833)  (addr: 0 0 +6612345678)  (esm_class: 0)
(rd: 1) (data_coding: 0) (sm: msg: test sms)  (opt: (byte: (tlv: 5640) 1)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] => 7465737420736D73
            [clear] => test sms

    [optional_param] =>
            [value] => 1608000101
            [clear] =>
                    [unknown_0x1608] =>
                            [tag] => 0x1608
                            [size] => 0001
                            [value] => 01


-- SubmitSM response sent
PDU HEX dump: 0000001980000004000000000000006EB334242433039314500

PDU decoded : (submit_resp: (pdu: 0 80000004 0 1771) 3BBC091E )


-- DeliverSM for SMSC ACK sent
PDU HEX dump:
000000AE0000000500000000000000020000002B36363132333435363738000000333338333300040000000000000000007569
643A31303032313738383436207375623A30303120646C7672643A303031207375626D697420646174653A32303136303331
303039353033342064F6E6520646174653A32303136303331303039353033342073746174A44454C49565244206572723A
30303020546578743A534D5343204F4B1600000437393800

PDU decoded : (deliver: (pdu: 0 5 0 2) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178846 sub:001 dlvrd:001 submit
date:20160310095034 done date:20160310095034 stat:DELIVRD err:000 Text:SMSC OK)  (opt: (str: (tlv:
5632) 798)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A31303032313738383436207375623A30303120646C7672643A303031207375626D697420646174653A323031363033
31303039353033342064F6E6520646174653A32303136303331303039353033342073746174A44454C4956524420657272
3A30303020546578743A534D5343204F4B
            [clear] => id:1002178846 sub:001 dlvrd:001 submit date:20160310095034 done
date:20160310095034 stat:DELIVRD err:000 Text:SMSC OK

    [optional_param] =>
            [value] => 1600000437393800
            [clear] =>
                    [unknown_0x1600] =>
                            [tag] => 0x1600
                            [size] => 0004
                            [value] => 37393800


-- DeliverSM for Mobile ACK sent
PDU HEX dump:
000000B00000000500000000000000030000002B36363132333435363738000000333338333300040000000000000000007769
643A31303032313738383436207375623A30303120646C7672643A303031207375626D697420646174653A32303136303331
303039353033342064F6E6520646174653A32303136303331303039353033372073746174A44454C49565244206572723A
30303020546578743A4D4F42494C45204F4B1600000437393800

PDU decoded: (deliver: (pdu: 0 5 0 3) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178846 sub:001 dlvrd:001 submit
date:20160310095034 done date:20160310095037 stat:DELIVRD err:000 Text:MOBILE OK)  (opt: (str: (tlv:
5632) 798)  ) )

SMPP Decoder output extract:
    [short_message] =>
```

```
            [value] =>
69643A313030323137383834362073756223A30303120646C7672643A303031207375626D697420646174653A323031363033
3130303935303333420646F6E6520646174653A3230313630333130303935303337207374617443A44454C4956524420657272
3A30303020546578743A4D4F42494C45204F4B
            [clear] => id:1002178846 sub:001 dlvrd:001 submit date:20160310095034 done
date:20160310095037 stat:DELIVRD err:000 Text:MOBILE OK

    [optional_param] =>
            [value] => 1600000437393800
            [clear] =>
                    [unknown_0x1600] =>
                            [tag] => 0x1600
                            [size] => 0004
                            [value] => 37393800
```

8.2 Client request for Final Status Indicator in DR

```
-- SubmitSM for MT received
PDU HEX dump:
0000003E00000004000000000000071C00000033333833330000002B3636313233343536373800000000000000010000000874
65737420736D731607000101

PDU decoded : (submit: (pdu: 62 4 0 1820) (addr: 0 0 33833)  (addr: 0 0 +6612345678)  (esm_class: 0)
(rd: 1) (data_coding: 0) (sm: msg: test sms) (opt: (byte: (tlv: 5639) 1)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] => 7465737420736D73
            [clear] => test sms

    [optional_param] =>
            [value] => 1607000101
            [clear] =>
                    [unknown_0x1607] =>
                            [tag] => 0x1607
                            [size] => 0001
                            [value] => 01


-- SubmitSM response sent

PDU HEX dump: 0000001980000004000000000000071C334242433039323500

PDU decoded : (submit_resp: (pdu: 0 80000004 0 1820) 3BBC0925 )


-- DeliverSM for SMSC ACK sent
PDU HEX dump:
000000AC00000005000000000000000090000002B36363132333435363738000000333338333300040000000000000007569
643A313030323137383835332073756223A30303120646C7672643A303031207375626D697420646174653A32303136303331
303130333331322064F6E6520646174653A32303136303331303130333331322073746174A44454C495652442065727233A
3030302054657874 3A534D5343204F4B160400024E00

PDU decoded : (deliver: (pdu: 0 5 0 9) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178853 sub:001 dlvrd:001 submit
date:20160310103312 done date:20160310103312 stat:DELIVRD err:000 Text:SMSC OK)  (opt: (str: (tlv:
5636) N)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A313030323137383835332073756223A30303120646C7672643A303031207375626D697420646174653A323031363033
3130313033333132 20646F6E6520646174653A3230313630333130313033333132207374617443A44454C4956524420657272
3A30303020546578743A534D5343204F4B
            [clear] => id:1002178853 sub:001 dlvrd:001 submit date:20160310103312 done
date:20160310103312 stat:DELIVRD err:000 Text:SMSC OK

    [optional_param] =>
            [value] => 160400024E00
            [clear] =>
                    [unknown_0x1604] =>
                            [tag] => 0x1604
                            [size] => 0002
                            [value] => 4E00


-- DeliverSM for Mobile ACK sent

PDU HEX dump:
000000AE0000000500000000000000000A0000002B363631323334353637380000003333383333000400000000000000007769
643A3130303231373838353332073756 23A30303120646C7672643A3030312073756 26D697420646174653A32303136303331
303130333331322064 6F6E6520646174653A32303136303331303133333135207374617 43A44454C4956524420657272 3A
3030302054657874 3A4D4F42494C45204F4B160400025900

PDU decoded : (deliver: (pdu: 0 5 0 10) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178853 sub:001 dlvrd:001 submit
date:20160310103312 done date:20160310103315 stat:DELIVRD err:000 Text:MOBILE OK)  (opt: (str: (tlv:
5636) Y)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A313030323137383835332073756 23A30303120646C7672643A3030312073756 26D697420646174653A323031363033
3130313033333132 20646F6E6520646174653A32303136303331303133333135207374617 43A44454C4956524420657272
3A3030302054657874 3A4D4F42494C45204F4B
            [clear] => id:1002178853 sub:001 dlvrd:001 submit date:20160310103312 done
date:20160310103315 stat:DELIVRD err:000 Text:MOBILE OK

    [optional_param] =>
            [value] => 160400025900
            [clear] =>
                    [unknown_0x1604] =>
                            [tag] => 0x1604
                            [size] => 0002
                            [value] => 5900
```

8.3 Client request for Subject in DR

```
-- SubmitSM for MT received

PDU HEX dump:
0000004A000000040000000000000AC90000003333383333 0000002B36363132333435363738000000000000000100000008 74
65737420736D73160A000101160200085465737469 6E6700

PDU decoded : (submit: (pdu: 74 4 0 2761) (addr: 0 0 33833)  (addr: 0 0 +6612345678)  (esm_class: 0)
(rd: 1) (data_coding: 0) (sm: msg: test sms)  (opt: (byte: (tlv: 5642) 1)  (str: (tlv: 5634)
Testing)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] => 7465737420736D73
            [clear] => test sms

    [optional_param] =>
            [value] => 160A000101160200085465737469 6E6700
            [clear] =>
                    [unknown_0x160A] =>
                            [tag] => 0x160A
                            [size] => 0001
                            [value] => 01

                    [unknown_0x1602] =>
                            [tag] => 0x1602
                            [size] => 0008
                            [value] => 5465737469 6E6700


-- SubmitSM response sent

PDU HEX dump: 0000001980000004000000000000AC93342424330393239 00

PDU decoded : (submit_resp: (pdu: 0 80000004 0 2761) 3BBC0929 )
```

```
-- DeliverSM for SMSC ACK sent

PDU HEX dump:
000000B20000000500000000000000110000002B363631323334353637380000003333383333000400000000000000007569
643A3130303231373838353720737562A30303120646C7672643A303031207375626D697420646174653A32303136303331
313032313033372064F6E6520646174653A3230313630333131303231303337207374617 43A44454C49565244206572723A
30303020546578743A534D5343204F4B1602000854657374696E6700

PDU decoded : (deliver: (pdu: 0 5 0 17) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178857 sub:001 dlvrd:001 submit
date:20160311021037 done date:20160311021037 stat:DELIVRD err:000 Text:SMSC OK)  (opt: (str: (tlv:
5634) Testing)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A3130303231373838353720737562A30303120646C7672643A303031207375626D697420646174653A323031363033
3131303231303337206 46F6E6520646174653A3230313630333131303231303337207374617 43A44454C4956524420657272
3A30303020546578743A534D5343204F4B
            [clear] => id:1002178857 sub:001 dlvrd:001 submit date:20160311021037 done
date:20160311021037 stat:DELIVRD err:000 Text:SMSC OK

    [optional_param] =>
            [value] => 1602000854657374696E6700
            [clear] =>
                    [unknown_0x1602] =>
                            [tag] => 0x1602
                            [size] => 0008
                            [value] => 54657374696E6700


-- DeliverSM for Mobile ACK sent

PDU HEX dump:
000000B40000000500000000000000120000002B363631323334353637380000003333383333000400000000000000007769
643A3130303231373838353720737562A30303120646C7672643A303031207375626D697420646174653A32303136303331
313032313033372064F6E6520646174653A32303136303331313032313034302073746174 3A44454C49565244206572723A
30303020546578743A4D4F42494C45204F4B1602000854657374696E6700

PDU decoded : (deliver: (pdu: 0 5 0 18) (addr: 0 0 +6612345678) (addr: 0 0 33833) (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178857 sub:001 dlvrd:001 submit
date:20160311021037 done date:20160311021040 stat:DELIVRD err:000 Text:MOBILE OK) (opt: (str: (tlv:
5634) Testing) ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A3130303231373838353720737562A30303120646C7672643A303031207375626D697420646174653A323031363033
31313032313033372064F6E6520646174653A3230313630333131303231303430207374617 43A44454C4956524420657272
3A30303020546578743A4D4F42494C45204F4B
            [clear] => id:1002178857 sub:001 dlvrd:001 submit date:20160311021037 done
date:20160311021040 stat:DELIVRD err:000 Text:MOBILE OK

    [optional_param] =>
            [value] => 1602000854657374696E6700
            [clear] =>
                    [unknown_0x1602] =>
                            [tag] => 0x1602
                            [size] => 0008
                            [value] => 54657374696E6700
```

8.4 Client request for HEX Order ID in DR

```
-- SubmitSM for MT received

PDU HEX dump:
0000003E00000004000000000000AD100000033333833330000002B36363132333435363738000000000000010000000874
65737420736D73160C000101
```

```
PDU decoded : (submit: (pdu: 62 4 0 2769) (addr: 0 0 33833)  (addr: 0 0 +6612345678)  (esm_class: 0)
(rd: 1) (data_coding: 0) (sm: msg: test sms)  (opt: (byte: (tlv: 5644) 1)   ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] => 7465737420736D73
            [clear] => test sms

    [optional_param] =>
            [value] => 160C000101
            [clear] =>
                    [unknown_0x160C] =>
                            [tag] => 0x160C
                            [size] => 0001
                            [value] => 01


-- SubmitSM response sent

PDU HEX dump: 00000019800000040000000000000AD1334242433039324100

PDU decoded : (submit_resp: (pdu: 0 80000004 0 2769) 3BBC092A )


-- DeliverSM for SMSC ACK sent

PDU HEX dump:
000000B10000000500000000000000130000002B36363132333435363738000000333338333300040000000000000000007369
643A3342424330393241207375623A30303120646C7672643A303031207375626D697420646174653A323031363033313130
323137323320646F6E6520646174653A323031363033313130323137323320737461743A44454C49565244206572723A3030
3020546578743A534D5343204F4B001E0009334242433039324100

PDU decoded : (deliver: (pdu: 0 5 0 19) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:3BBC092A sub:001 dlvrd:001 submit
date:20160311021723 done date:20160311021723 stat:DELIVRD err:000 Text:SMSC OK)  (opt: (str: (tlv:
30) 3BBC092A)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A3342424330393241207375623A30303120646C7672643A303031207375626D697420646174653A3230313630333131
30323137323320646F6E6520646174653A323031363033313130323137323320737461743A44454C49565244206572723A30
303020546578743A534D5343204F4B
            [clear] => id:3BBC092A sub:001 dlvrd:001 submit date:20160311021723 done
date:20160311021723 stat:DELIVRD err:000 Text:SMSC OK

    [optional_param] =>
            [value] => 001E0009334242433039324100
            [clear] =>
                    [receipted_message_id] =>
                            [tag] => 0x001E
                            [size] => 0009
                            [value] => 334242433039324100
                            [clear] => 3BBC092A


-- DeliverSM for Mobile ACK sent

PDU HEX dump:
000000B30000000500000000000000140000002B36363132333435363738000000333338333300040000000000000000007569
643A3342424330393241207375623A30303120646C7672643A303031207375626D697420646174653A323031363033313130
323137323320646F6E6520646174653A323031363033313130323137323620737461743A44454C49565244206572723A3030
3020546578743A4D4F42494C45204F4B001E0009334242433039324100

PDU decoded : (deliver: (pdu: 0 5 0 20) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:3BBC092A sub:001 dlvrd:001 submit
date:20160311021723 done date:20160311021726 stat:DELIVRD err:000 Text:MOBILE OK)  (opt: (str: (tlv:
30) 3BBC092A)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A3342424330393241207375623A30303120646C7672643A303031207375626D697420646174653A3230313630333131
30323137323320646F6E6520646174653A323031363033313130323137323620737461743A44454C49565244206572723A30
303020546578743A4D4F42494C45204F4B
            [clear] => id:3BBC092A sub:001 dlvrd:001 submit date:20160311021723 done
date:20160311021726 stat:DELIVRD err:000 Text:MOBILE OK
```

```
    [optional_param] =>
            [value] => 001E0009334242433039324100
            [clear] =>
                    [receipted_message_id] =>
                            [tag] => 0x001E
                            [size] => 0009
                            [value] => 334242433039324100
                            [clear] => 3BBC092A
```

8.5 Client request for DEC Order ID in DR

```
-- SubmitSM for MT received

PDU HEX dump:
0000003E0000000400000000000000AD60000003333383333000002B36363132333435363738000000000000010000000874
65737420736D73160C000102

PDU decoded : (submit: (pdu: 62 4 0 2774) (addr: 0 0 33833)  (addr: 0 0 +6612345678)  (esm_class: 0)
(rd: 1) (data_coding: 0) (sm: msg: test sms)  (opt: (byte: (tlv: 5644) 2)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] => 7465737420736D73
            [clear] => test sms

    [optional_param] =>
            [value] => 160C000102
            [clear] =>
                    [unknown_0x160C] =>
                            [tag] => 0x160C
                            [size] => 0001
                            [value] => 02


-- SubmitSM response sent

PDU HEX dump: 0000001B800000040000000000000AD6313030323137383835390 0

PDU decoded : (submit_resp: (pdu: 0 80000004 0 2774) 1002178859 )

-- DeliverSM for SMSC ACK sent

PDU HEX dump:
000000B500000005000000000000000150000002B363631323334353637380000003333383333000400000000000000007569
643A313030323137383835392073756223A30303120646C7672643A303031207375626D697420646174653A32303136303331
313032323132382064F6E6520646174653A3230313630333131303232313238207374617461743A44454C495652442065727 23A
30303020546578743A534D5343204F4B001E000B31303032313738383539 00

PDU decoded : (deliver: (pdu: 0 5 0 21) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178859 sub:001 dlvrd:001 submit
date:20160311022128 done date:20160311022128 stat:DELIVRD err:000 Text:SMSC OK)  (opt: (str: (tlv:
30) 1002178859)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A313030323137383835392073756223A30303120646C7672643A303031207375626D697420646174653A323031363033
313130323231323820646F6E6520646174653A323031363033313130323231323820737461743A44454C495652442065727 2
3A30303020546578743A534D5343204F4B
            [clear] => id:1002178859 sub:001 dlvrd:001 submit date:20160311022128 done
date:20160311022128 stat:DELIVRD err:000 Text:SMSC OK

    [optional_param] =>
            [value] => 001E000B3130303231373838353900
            [clear] =>
                    [receipted_message_id] =>
                            [tag] => 0x001E
                            [size] => 000B
                            [value] => 3130303231373838353900
                            [clear] => 1002178859
```

```

-- DeliverSM for Mobile ACK sent

PDU HEX dump:
000000B70000000500000000000000160000002B363631323334353637380000003333383333000400000000000000007769
643A31303032313738383539207375623A30303120646C7672643A303031207375626D697420646174653A32303136303331
3130323231323820646F6E6520646174653A32303136303331313032323133313120737461743A44454C49565244206572723A
30303020546578743A4D4F42494C45204F4B001E000B3130303231373838353900

PDU decoded: (deliver: (pdu: 0 5 0 22) (addr: 0 0 +6612345678) (addr: 0 0 33833) (esm_class: 4) (rd:
0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178859 sub:001 dlvrd:001 submit
date:20160311022128 done date:20160311022131 stat:DELIVRD err:000 Text:MOBILE OK) (opt: (str: (tlv:
30) 1002178859) ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A31303032313738383539207375623A30303120646C7672643A303031207375626D697420646174653A323031363033
313130323231323820646F6E6520646174653A32303136303331313032323133313120737461743A44454C4956524420657272
3A30303020546578743A4D4F42494C45204F4B
            [clear] => id:1002178859 sub:001 dlvrd:001 submit date:20160311022128 done
date:20160311022131 stat:DELIVRD err:000 Text:MOBILE OK

    [optional_param] =>
            [value] => 001E000B3130303231373838353900
            [clear] =>
                    [receipted_message_id] =>
                            [tag] => 0x001E
                            [size] => 000B
                            [value] => 3130303231373838353900
                            [clear] => 1002178859
```

8.6 Client request for Status ID in DR

```
-- SubmitSM for MT received

PDU HEX dump:
0000003E0000000400000000000000ADD00000033333833330000002B3636313233343536373800000000000000010000000874
65737420736D73160B000101

PDU decoded : (submit: (pdu: 62 4 0 2781) (addr: 0 0 33833)  (addr: 0 0 +6612345678)  (esm_class: 0)
(rd: 1) (data_coding: 0) (sm: msg: test sms)  (opt: (byte: (tlv: 5643) 1)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] => 7465737420736D73
            [clear] => test sms

    [optional_param] =>
            [value] => 160B000101
            [clear] =>
                    [unknown_0x160B] =>
                            [tag] => 0x160B
                            [size] => 0001
                            [value] => 01


-- SubmitSM response sent

PDU HEX dump: 0000001980000004000000000000ADD334242433039324300

PDU decoded : (submit_resp: (pdu: 0 80000004 0 2781) 3BBC092C )


-- DeliverSM for SMSC ACK sent

PDU HEX dump:
000000AF0000000500000000000000170000002B363631323334353637380000003333383333000400000000000000007569
643A31303032313738383630207375623A30303120646C7672643A303031207375626D697420646174653A32303136303331
3130323236343820646F6E6520646174653A32303136303331313032323634382073746174A44454C49565244206572723A
30303020546578743A534D5343204F4B160500053435304100

PDU decoded : (deliver: (pdu: 0 5 0 23) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178860 sub:001 dlvrd:001 submit
date:20160311022648 done date:20160311022648 stat:DELIVRD err:000 Text:SMSC OK)  (opt: (str: (tlv:
5637) 450A)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A3130303231373838363020737562A30303120646C7672643A303031207375626D697420646174653A323031363033
31313032323634382064 6F6E6520646174653A32303136303331313032323634382073746174A44454C4956524420657272
3A3030302054657874 3A534D5343204F4B
            [clear] => id:1002178860 sub:001 dlvrd:001 submit date:20160311022648 done
date:20160311022648 stat:DELIVRD err:000 Text:SMSC OK

    [optional_param] =>
            [value] => 1605000534353 04100
            [clear] =>
                    [unknown_0x1605] =>
                            [tag] => 0x1605
                            [size] => 0005
                            [value] => 3435304100


-- DeliverSM for Mobile ACK sent

PDU HEX dump:
000000B10000000500000000000000180000002B363631323334353637380000003333383333000400000000000000007769
643A3130303231373838363020737562 3A30303120646C7672643A303031207375626D697420646174653A32303136303331
31303232363438206 46F6E6520646174653A32303136303331313032323635312073746174 3A44454C4956524420657272 3A
30303020546578743A4D4F42494C45204F4B1605000534353 14100

PDU decoded : (deliver: (pdu: 0 5 0 24) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178860 sub:001 dlvrd:001 submit
date:20160311022648 done date:20160311022651 stat:DELIVRD err:000 Text:MOBILE OK)  (opt: (str: (tlv:
5637) 451A)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A31303032313738383630207375623A30303120646C7672643A303031207375626D697420646174653A323031363033
31313032323634382064 6F6E6520646174653A32303136303331313032323635312073746174 3A44454C4956524420657272
3A30303020546578743A4D4F42494C45204F4B
            [clear] => id:1002178860 sub:001 dlvrd:001 submit date:20160311022648 done
date:20160311022651 stat:DELIVRD err:000 Text:MOBILE OK

    [optional_param] =>
            [value] => 160500053435314100
            [clear] =>
                    [unknown_0x1605] =>
                            [tag] => 0x1605
                            [size] => 0005
                            [value] => 3435314100
```

8.7 Client request for Internal Status ID in DR

```
-- SubmitSM for MT received

PDU HEX dump:
0000003E000000040000000000000AE200000033333833330000002B36363132333435363738000000000000010000000874
65737420736D731609000101

PDU decoded : (submit: (pdu: 62 4 0 2786) (addr: 0 0 33833)  (addr: 0 0 +6612345678)  (esm_class: 0)
(rd: 1) (data_coding: 0) (sm: msg: test sms)  (opt: (byte: (tlv: 5641) 1)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] => 7465737420736D73
            [clear] => test sms

    [optional_param] =>
            [value] => 1609000101
            [clear] =>
                    [unknown_0x1609] =>
                            [tag] => 0x1609
                            [size] => 0001
                            [value] => 01


-- SubmitSM response sent

PDU HEX dump: 00000019800000040000000000000AE2334242433039324400

PDU decoded : (submit_resp: (pdu: 0 80000004 0 2786) 3BBC092D )


-- DeliverSM for SMSC ACK sent

PDU HEX dump:
000000AF0000000500000000000000190000002B36363132333435363738000000333338333300040000000000000000007569
643A31303032313738383631207375623A30303120646C7672643A303031207375626D697420646174653A32303136303331
3130323330353420646F6E6520646174653A3230313630333131303233303534207374617443A44454C4956524420657272 3A
30303020546578743A534D5343204F4B160600054442343900

PDU decoded : (deliver: (pdu: 0 5 0 25) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178861 sub:001 dlvrd:001 submit
date:20160311023054 done date:20160311023054 stat:DELIVRD err:000 Text:SMSC OK)  (opt: (str: (tlv:
5638) DB49)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A31303032313738383631207375623A30303120646C7672643A303031207375626D697420646174653A323031363033
3131303233303534 20646F6E6520646174653A32303136303331313032333035342073746174743A44454C4956524420657272
3A30303020546578743A534D5343204F4B
            [clear] => id:1002178861 sub:001 dlvrd:001 submit date:20160311023054 done
date:20160311023054 stat:DELIVRD err:000 Text:SMSC OK

    [optional_param] =>
            [value] => 160600054442343900
            [clear] =>
                    [unknown_0x1606] =>
                            [tag] => 0x1606
                            [size] => 0005
                            [value] => 4442343900


-- DeliverSM for Mobile ACK sent

PDU HEX dump:
000000B10000000500000000000000001A0000002B36363132333435363738000000333338333300040000000000000000007769
643A31303032313738383631207375623A30303120646C7672643A303031207375626D697420646174653A32303136303331
3130323330353420646F6E6520646174653A3230313630333131303233303537207374617443A44454C4956524420657272 3A
30303020546578743A4D4F42494C45204F4B160600054442353000

PDU decoded : (deliver: (pdu: 0 5 0 26) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178861 sub:001 dlvrd:001 submit
date:20160311023054 done date:20160311023057 stat:DELIVRD err:000 Text:MOBILE OK)  (opt: (str: (tlv:
5638) DB50)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A31303032313738383631207375623A30303120646C7672643A303031207375626D697420646174653A323031363033
31313032333035342 0646F6E6520646174653A32303136303331313032333035372073746174743A44454C4956524420657272
3A30303020546578743A4D4F42494C45204F4B
            [clear] => id:1002178861 sub:001 dlvrd:001 submit date:20160311023054 done
date:20160311023057 stat:DELIVRD err:000 Text:MOBILE OK

    [optional_param] =>
            [value] => 160600054442353000
            [clear] =>
                    [unknown_0x1606] =>
                            [tag] => 0x1606
                            [size] => 0005
                            [value] => 4442353000
```

8.8 Client request for multiple optional parameters in DR

```
-- SubmitSM for MT received

PDU HEX dump:
00000052000000040000000000000AE80000003333383333000000002B36363132333435363738000000000000000100000008 74
65737420736D73160C0001021609000101160800010116 0B0001011607000101

PDU decoded : (submit: (pdu: 82 4 0 2792) (addr: 0 0 33833)  (addr: 0 0 +6612345678)  (esm_class: 0)
(rd: 1) (data_coding: 0) (sm: msg: test sms)  (opt: (byte: (tlv: 5644) 2)  (byte: (tlv: 5641) 1)
(byte: (tlv: 5640) 1)  (byte: (tlv: 5643) 1)  (byte: (tlv: 5639) 1)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] => 7465737420736D73
            [clear] => test sms

    [optional_param] =>
            [value] => 160C000102160900010116080001011 60B0001011607000101
            [clear] =>
                    [unknown_0x160C] =>
                            [tag] => 0x160C
                            [size] => 0001
                            [value] => 02

                    [unknown_0x1609] =>
                            [tag] => 0x1609
                            [size] => 0001
                            [value] => 01

                    [unknown_0x1608] =>
                            [tag] => 0x1608
                            [size] => 0001
                            [value] => 01

                    [unknown_0x160B] =>
                            [tag] => 0x160B
                            [size] => 0001
                            [value] => 01

                    [unknown_0x1607] =>
                            [tag] => 0x1607
                            [size] => 0001
                            [value] => 01



-- SubmitSM response sent

PDU HEX dump: 0000001B800000040000000000000AE83130303231373838363200

PDU decoded : (submit_resp: (pdu: 0 80000004 0 2792) 1002178862 )

-- DeliverSM for SMSC ACK sent

PDU HEX dump:
000000D500000005000000000000001B0000002B36363132333435363738000000333338333300040000000000000000007569
643A313030323137383836322073756223A303031206 46C7672643A303031207375626D697420646174653A32303136303331
313032333632372064 6F6E6520646174653A32303136303331313032333632372073746174 3A44454C4956524420657272 3A
30303020546578743A534D5343204F4B16050005343530410016040002 4E00160600054442343900160000043739380001E
000B31303032313738383632 00

PDU decoded : (deliver: (pdu: 0 5 0 27) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178862 sub:001 dlvrd:001 submit
date:20160311023627 done date:20160311023627 stat:DELIVRD err:000 Text:SMSC OK)  (opt: (str: (tlv:
5637) 450A)  (str: (tlv: 5636) N)  (str: (tlv: 5638) DB49)  (str: (tlv: 5632) 798)  (str: (tlv: 30)
1002178862)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A31303032313738383632207375623A30303120646C7672643A303031207375626D697420646174653A323031363033
313130323336323720646F6E6520646174653A32303136303331313032333632372073746174 3A44454C49565244206572 72
3A30303020546578743A534D5343204F4B
            [clear] => id:1002178862 sub:001 dlvrd:001 submit date:20160311023627 done
date:20160311023627 stat:DELIVRD err:000 Text:SMSC OK
```

```
    [optional_param] =>
            [value] =>
160500053435304100160400024E00160600054442343900160000043739380001E000B3130303231373838363200
            [clear] =>
                    [unknown_0x1605] =>
                            [tag] => 0x1605
                            [size] => 0005
                            [value] => 3435304100

                    [unknown_0x1604] =>
                            [tag] => 0x1604
                            [size] => 0002
                            [value] => 4E00

                    [unknown_0x1606] =>
                            [tag] => 0x1606
                            [size] => 0005
                            [value] => 4442343900

                    [unknown_0x1600] =>
                            [tag] => 0x1600
                            [size] => 0004
                            [value] => 37393800

                    [receipted_message_id] =>
                            [tag] => 0x001E
                            [size] => 000B
                            [value] => 3130303231373838363200
                            [clear] => 1002178862


-- DeliverSM for Mobile ACK sent

PDU HEX dump:
000000D70000000500000000000001C0000002B36363132333435363738000000333338333300040000000000000007769
643A31303032313738383632207375623A30303120646C7672643A303031207375626D697420646174653A32303136303331
3130323336323720646F6E6520646174653A323031363033313130323336333020737461743A44454C49565244206572723A
30303020546578743A4D4F42494C45204F4B1605000534353141001604000259001606000544423530001600000437393800
001E000B3130303231373838363200

PDU decoded : (deliver: (pdu: 0 5 0 28) (addr: 0 0 +6612345678)  (addr: 0 0 33833)  (esm_class: 4)
(rd: 0) (data_coding: 0) (sm: enc: ISO-8859-1 msg: id:1002178862 sub:001 dlvrd:001 submit
date:20160311023627 done date:20160311023630 stat:DELIVRD err:000 Text:MOBILE OK)  (opt: (str: (tlv:
5637) 451A)  (str: (tlv: 5636) Y)  (str: (tlv: 5638) DB50)  (str: (tlv: 5632) 798)  (str: (tlv: 30)
1002178862)  ) )

SMPP Decoder output extract:
    [short_message] =>
            [value] =>
69643A31303032313738383632207375623A30303120646C7672643A303031207375626D697420646174653A323031363033
313130323336323720646F6E6520646174653A323031363033313130323336333020737461743A44454C4956524420657272
3A30303020546578743A4D4F42494C45204F4B
            [clear] => id:1002178862 sub:001 dlvrd:001 submit date:20160311023627 done
date:20160311023630 stat:DELIVRD err:000 Text:MOBILE OK

    [optional_param] =>
            [value] =>
160500053435314100160400025900160600054442353000160000043739380001E000B3130303231373838363200
            [clear] =>
                    [unknown_0x1605] =>
                            [tag] => 0x1605
                            [size] => 0005
                            [value] => 3435314100

                    [unknown_0x1604] =>
                            [tag] => 0x1604
                            [size] => 0002
                            [value] => 5900

                    [unknown_0x1606] =>
                            [tag] => 0x1606
                            [size] => 0005
                            [value] => 4442353000

                    [unknown_0x1600] =>
                            [tag] => 0x1600
```

```
                        [size] => 0004
                        [value] => 37393800

                [receipted_message_id] =>
                        [tag] => 0x001E
                        [size] => 000B
                        [value] => 3130303231373838363200
                        [clear] => 1002178862
```