SDP Solution

[API Reference (SMS, SMPP 34)]

Ethio telecom

Huawei Technologies Co., Ltd

Contents

1 Overview

About This Chapter

1.1 API Functions

The SDP provides SMS capability application programming interfaces (APIs) for third-party applications (App for short) to connect to it and use its SMS capability to send and receive SMS messages. The App is generally developed by various partners of the SDP.

NOTE

Partners are the enterprises and individuals who sign a contract and cooperate with carriers in utilizing the SDP. Partners include SPs, CPs, Developers, and Enterprises. In this document, partners are mainly the SPs, Developers, and Enterprises who use APIs for secondary development.

Table 1-1 describes functions of SMS capability APIs provided by the SDP.

Table 1-1 Functions of SMS capability APIs

Function	Subfunction	Description	API
Receiving SMS messages	Receiving SMS messages in Notify mode	The App (functioning as the client) invokes the bind_receiver API to create a unidirectional message receiving link with the SDP or invokes the bind_transceiver API to create bidirectional link with the SDP.	- 2.2 bind_receiver - 2.3 bind_transceiver - 2.4

Function	Subfunction	Description	API
		After the link is created, the App (functioning as the client) invokes the enquire_link API to send heartbeat messages to the SDP (functioning as the server) to maintain the link. When the SDP receives a MO SMS message from a user, the SDP (functioning as the client) invokes the deliver_sm API to send the SMS message to the App (functioning as the server).	enquire_link - 2.5 deliver_sm
Sending SMS messages	Sending SMS messages to users and receiving status reports in Notify mode	The App (functioning as the client) invokes the bind_transmitter API to create a unidirectional message sending link with the SDP or invokes the bind_transceiver API to create bidirectional link with the SDP. After the link is created, the App (functioning as the client) invokes the enquire_link API to send heartbeat messages to the SDP (functioning as the server) to maintain the link. The App (functioning as the client) invokes the submit_sm API to send single SMS messages or the submit_multi API to send bulk SMS messages to users through the SDP (functioning as the server). When receiving a status report from the SMSC, the SDP (functioning as the client) then invokes the deliver_sm(report) API to send the status report to the App (functioning as the server).	- 3.2 bind_transmitter - 2.3 bind_transceiver - 2.4 enquire_link - 3.3 submit_sm - 3.4 submit_multi - 3.5 deliver_sm(report)

1.2 Level of Requirement for Parameters

The App must develop APIs based on the level of requirement for each parameter.

Table 1-2 Level of requirement for parameters

Type	Description
Mandatory	A parameter is always mandatory in a request. Parameters with the **Mandatory** requirement are used for access authentication or service processing. If a parameter with the **Mandatory** requirement is left empty in a request, access authentication or service processing fails and the request fails.

Type	Description
Conditional	A parameter is mandatory or optional in specified conditions. Parameters with the **Conditional** requirement are used for access authentication or service processing in specified conditions. If the specified conditions are met but a parameter with the **Conditional** requirement is left empty in a request, access authentication or service processing fails and the request fails.
Optional	A parameter is always optional. Parameters with the **Optional** requirement are not used for service processing.

1.3 API Parameter Types

The App must develop APIs based on the type required for each parameter.

Table 1-3 describes the Short Message Peer to Peer (SMPP) API parameter types.

Table 1-3 API parameter types

Type	Description
Integer	An unsigned value with the defined number of octets. The octets will always be transmitted MSB first (Big Endian). Where reference is made to NULL settings of Integer fields, this implies that the field is zero filled.
C-Octet String	A series of ASCII characters terminated with the NULL character. Reference made to NULL settings of Octet-String fields implies that the field consists of a single NULL character, for example, an octet encoded with value 0x00 (zero). In the case of C-Octet String formats, the maximum field size is shown as a combination of string length and the NULL terminator, for example, an 8-character C-Octet String is encoded in 9 octets when the NULL terminator is included.
C-Octet String (Decimal)	A series of ASCII characters, each character representing a decimal digit (0 - 9) and terminated with the NULL character.
C-Octet String(Hex)	A series of ASCII characters, each character representing a hexadecimal digit (0 - F) and terminated with the NULL character.
Octet String	A series of octets, not necessarily NULL terminated.

Table 1-4 describes the length of parameters in different types.

Table 1-4 Parameter length

Length	Type	Description
4	Integer	Fixed size integer field. In this example the integer is of size 32 bits (4 octets).
Var Max 16	C-Octet String	This string is of variable length from 1-15 ASCII characters, followed by an octet containing the NULL terminator. An empty string is encoded as a single octet containing the NULL character (0x00).
Fixed 1 or 17	C-Octet String	This string has two possible lengths:- 1 octet containing the NULL character or a fixed number of characters terminated with the NULL character (in this example 16 characters plus the NULL character).
Var 0 - 254	Octet String	Variable size octet string field. In this example the size of the octet string field can vary from 0 to 254 octets.

1.4 SMPP PDU Format

The SMPP protocol is used to exchange request and response packet data units (PDUs) between SPs and the SDP on the TCP/IP network. Each SMPP operation involves a request PDU and a response PDU.

A request PDU and a response PDU have the same format (PDU header and PDU body). The PDU header is mandatory and the PDU body is optional. Table 1-5 describes the PDU format.

Table 1-5 SMPP PDU format overview

SMPP PDU				
PDU Header (mandatory)				PDU Body (Optional)
command length	command id	command status	sequence number	*PDU Body*
4 octets	Length = (Command Length value - 4) octets			

The PDU header consists of the **command length**, **command id**, **command status**, and **sequence number** parameters. The PDU body contains mandatory parameters and optional parameters. Table 1-6 describes the PDU format.

Table 1-6 SMPP PDU format description

PDU Header/Body	SMPP PDU Field	Length	Type	Level of Requirement	Description
Header	command_length	4	Integer	Mandatory	Define the total octet length of the SMPP PDU packet including the length field. [Example] 1024
	command_id	4	Integer	Mandatory	The command_id field identifies the particular SMPP PDU, for example, submit_sm, deliver_sm. A unique command identifier is allocated to each SMPP request PDU in the range: 0x00000000 to 0x000001FF . A unique command identifier is also allocated to each SMPP response PDU in the range: 0x80000000 to 0x800001FF. **NOTE** An SMPP response command_id is identical to the corresponding request SMPP command_id, but with bit 31 set. [Example] 0x80000001
	command_status	4	Integer	Mandatory	The command_status field indicates the success or failure of an SMPP request. It is relevant only in the SMPP response PDU and it must contain a NULL value in an SMPP request PDU. [Example] 0x00000000
	sequence_number	4	Integer	Mandatory	This field contains a sequence number which allows SMPP requests and responses to be associated for correlation purposes. The use of sequence numbers for message correlation allows SMPP PDUs to be exchanged asynchronously. Assignment of the sequence_number is the responsibility of the SMPP PDU originator. The sequence_number should be increased monotonically for each submitted SMPP request PDU and must be preserved in the associated SMPP response PDU.

PDU Header/ Body	SMPP PDU Field	Length	Type	Level of Requirement	Description
					The sequence_number may range from: 0x00000001 to 0x7FFFFFFF. [Example] 0x00000001
Body	*Mandatory Parameters*	var.	mixed	Mandatory	A list of mandatory parameters corresponding to that SMPP PDU defined in the command_id field. For details, see the description of each parameter.
	Optional Parameters	var.	mixed	Optional	A list of Optional Parameters corresponding to that SMPP PDU defined in the command_id field and included as required. For details, see the description of each parameter.
	Conditional Parameters	var.	mixed	Conditional	A list of conditional parameters corresponding to that SMPP PDU defined in the command_id field. For details, see the description of each parameter.

NOTE

When Apps invoke SDP APIs, requests cannot contain the following XML characters: & > < ' "

If the preceding characters are really required, you must add escape characters before such characters. Otherwise, SDP APIs will fail to be invoked.

1.5 Mapping Between SMPP PDUs and APIs

Table 1-7 describes the mapping between SMPP PDUs and APIs.

Table 1-7 Mapping between PDUs and APIs

API	Request PDU	Response PDU
bind_transmitter	bind_transmitter	bind_transmitter_resp
bind_receiver	bind_receiver	bind_receiver_resp
bind_transceiver	bind_transceiver	bind_transceiver_resp
enquire_link	enquire_link	enquire_link_resp

API	Request PDU	Response PDU
generic_nack	-	generic_nack
submit_sm	submit_sm	submit_sm_resp
submit_sm_multi	submit_sm_multi	submit_sm_multi_resp
deliver_sm	deliver_sm	deliver_sm_resp

1.6 Optional Parameter Format

Optional Parameters are fields, which may be optionally included in an SMPP message. Optional Parameters must always appear at the end of a message, in the **Optional Parameters** section or **Conditional Parameters** section of the SMPP PDU. However, they may be included in any convenient order within the **Optional Parameters** section or **Conditional Parameters** section of the SMPP PDU and need not be encoded in the order presented in this document.

All SMPP optional parameters have a 16 bit Parameter Tag Identifier.

All optional parameters have the following general TLV (Tag, Length, Value) format. The definition of the Tag, Length and Value for each optional parameter is defined in the Table 1-8.

Table 1-8 SMPP PDU Format Description

Parameter	Size	Type	Description
Tag	2	Integer	The Tag field is used to uniquely identify the particular optional parameter in question. The optional parameter Tag field is always 2 octets in length. [Example] 0x4004
Length	2	Integer	The Length field indicates the length of the Value field in octets. The optional parameter Length field is always 2 octets in length. **NOTE** The length does not include the length of the Tag and Length fields. [Example] 0x0005
Value	variable	variable	The Value field contains the actual data for the optional parameter in question. [Example] 0x826299E01

Table 1-9 show the SMPP supported Optional Parameters or Conditional Parameters and their associated Tag Values.

Table 1-9 Optional or Conditional Parameter Tag values

Tag	Value	Wireless Network Technology
source_port	0x020A	Generic
destination_port	0x020B	Generic
sar_msg_ref_num	0x020C	Generic
sar_total_segments	0x020E	Generic
sar_segment_seqnum	0x020F	Generic
payload_type	0x0019	Generic
message_payload	0x0424	Generic
receipted_message_id	0x001E	Generic
link_id	0x400D	Generic
presentID	0x400E	Generic
ServiceID	0x4004	Generic

NOTE

Generic optional parameters may be applicable to all wireless network technologies i.e., GSM/ iDEN, TDMA and CDMA.

1.7 Error Response Format

This is a generic negative acknowledgement to an SMPP PDU submitted with an invalid message header. A generic_nack response is returned in the following cases:

- Invalid **command_length**

  If the receiving SMPP entity, on decoding an SMPP PDU, detects an invalid **command_length** (either too short or too long), it should assume that the data is corrupt. In such cases a generic_nack PDU must be returned to the message originator.

- Unknown **command_id**

  If an unknown or invalid **command_id** is received, a generic_nack PDU must also be returned to the originator.

Table 1-10 describes the format of the SMPP generic_nack PDU. It comprises the SMPP message header only.

Table 1-10 GENERIC_NACK PDU Format Description

PDU Header/Body	SMPP PDU Field	Size octets	Type	Level of Requirement	Description
Header	command_length	4	Integer	Mandatory	Define the total octet length of the SMPP PDU packet including the length field. [Example] 1024
	command_id	4	Integer	Mandatory	Value corresponding to generic_nack PDU. set it to **0x80000000**. [Example] 0x80000000
	command_status	4	Integer	Mandatory	Error code corresponding to reason for sending the generic_nack. [Example] 0x00000000
	sequence_number	4	Integer	Mandatory	Set to sequence number of original PDU or to NULL if the original PDU cannot be decoded. [Example] 0x00000001

Table 1-11 describes SMPP34 return codes.

Table 1-11 SMPP34 return codes

Return codes	Value	Description
ESME_ROK	0x00000000	No Error
ESME_RINVMSGLEN	0x00000001	Message Length is invalid
ESME_RINVCMDLEN	0x00000002	Command Length is invalid
ESME_RINVCMDID	0x00000003	Invalid Command ID
ESME_RINVBNDSTS	x00000004	Incorrect BIND Status for given command
ESME_RALYBND	0x00000005	ESME Already in Bound State
ESME_RINVPRTFLG	0x00000006	Invalid Priority Flag
ESME_RINVREGDLVFLG	0x00000007	Invalid Registered Delivery Flag
ESME_RSYSERR	0x00000008	System Error
Reserved	0x00000009	Reserved

Return codes	Value	Description
ESME_RINVSRCADR	0x0000000A	Invalid Source Address
ESME_RINVDSTADR	0x0000000B	Invalid Dest Addr
ESME_RINVMSGID	0x0000000C	Message ID is invalid
ESME_RBINDFAIL	0x0000000D	0x0000000D Bind Failed
ESME_RINVPASWD	0x0000000E	Invalid Password
ESME_RINVSYSID	0x0000000F	Invalid System ID
Reserved	0x00000010	Reserved
ESME_RCANCELFAIL	0x00000011	Cancel SM Failed
Reserved	0x00000012	Reserved
ESME_RREPLACEFAIL	0x00000013	Replace SM Failed
ESME_RMSGQFUL	0x00000014	Message Queue Full
ESME_RINVSERTYP	0x00000015	Invalid Service Type
Reserved	0x00000016-0x00000032	Reserved
ESME_RINVNUMDESTS	0x00000033	Invalid number of destinations
ESME_RINVDLNAME	0x00000034	Invalid Distribution List name
Reserved	0x00000035-0x0000003F	Reserved
ESME_RINVDESTFLAG	0x00000040	Destination flag is invalid (submit_multi)
Reserved	0x00000041	Reserved
ESME_RINVSUBREP	0x00000042	Invalid ‘submit with replace’ request (i.e. submit_sm with replace_if_present_flag set)
ESME_RINVESMCLASS	0x00000043	Invalid esm_class field data
ESME_RCNTSUBDL	0x00000044	Cannot Submit to Distribution List
ESME_RSUBMITFAIL	0x00000045	submit_sm or submit_multi failed
Reserved	0x00000046-0x00000047	Reserved
ESME_RINVSRCTON	0x00000048	Invalid Source address TON
ESME_RINVSRCNPI	0x00000049	Invalid Source address NPI
ESME_RINVDSTTON	0x00000050	Invalid Destination address TON
ESME_RINVDSTNPI	0x00000051	Invalid Destination address NPI

Return codes	Value	Description
Reserved	0x00000052	Reserved
ESME_RINVSYSTYP	0x00000053	Invalid system_type field
ESME_RINVREPFLAG	0x00000054	Invalid replace_if_present flag
ESME_RINVNUMMSGS	0x00000055	Invalid number of messages
Reserved	0x00000056-0x00000057	Reserved
ESME_RTHROTTLED	0x00000058	Throttling error (ESME has exceeded allowed message limits)
Reserved	0x00000059-0x00000060	Reserved
ESME_RINVSCHED	0x00000061	Invalid Scheduled Delivery Time
ESME_RINVEXPIRY	0x00000062	Invalid message validity period (Expiry time)
ESME_RINVDFTMSGID	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
Reserved	0x00000068 - 0x000000BF	Reserved
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
Reserved	0x000000C5 - 0x000000FD	Reserved
ESME_RUNKNOWNERR	0x000000FF	Unknown Error
Reserved for SMPP extension	0x00000100-0x000003FF	Reserved for SMPP extension

Return codes	Value	Description
Reserved	0x00000500-0xFFFFFFFF	Reserved
Reserved for SMSC vendor specific errors	0x00000400-0x000004FF	Reserved for SMSC vendor specific errors

2 APIs for Receiving SMS Messages

About This Chapter

2.1 Process

The process of the App receiving SMS messages consists of the following main steps:

- Creating a message receiving link: The App creates a message receiving link with the SDP.
- Maintaining the link: After the link is created, the App sends heartbeat messages to the SDP to maintain the link.
- Receiving MO SMS messages: The App receives MO SMS messages from the SDP in real time.

Figure 2-1 shows the process of receiving SMS messages in Notify mode.

Figure 2-1 Process of receiving SMS messages in Notify mode

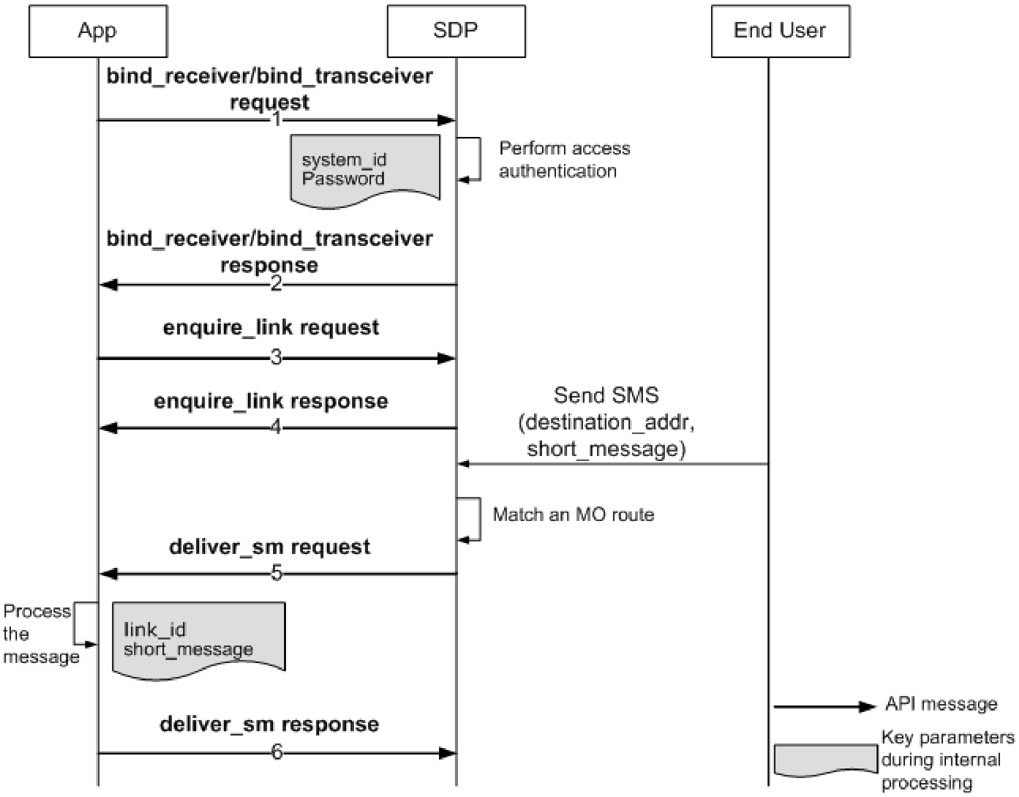
App
SDP
End User
bind_receiver/bind_transceiver request
1
system_id
Password
Perform access authentication
bind_receiver/bind_transceiver response
2
enquire_link request
3
enquire_link response
4
Send SMS (destination_addr, short_message)
Match an MO route
deliver_sm request
5
Process the message
link_id
short_message
deliver_sm response
6
API message
Key parameters during internal processing

Table 2-1 describes the process.

Table 2-1 Description for the process of receiving SMS messages

Step	Description
1-2	• The App sends a request to the SDP to create a link. • The SDP performs authentication and service level agreement (SLA) control based on fields in the request, saves MO routing information in the database, and sends a response to the App.
3–4	• The App sends heartbeat messages to the SDP to maintain the link. • The SDP sends responses to the App.
5-6	• The SDP receives an MO SMS message from a user, matches an MO route based on the access code, and sends a notification of the SMS message to the App. • The App parses the notification and sends a response to the SDP.

2.2 bind_receiver

2.2.1 Function

The App (functioning as the client) invokes the bind_receiver API to establish a network connection with the SDP (functioning as the server) to receive messages. The SDP send a response within 60 seconds by default.

After the network connection is established, the SDP sends MO SMS messages received from users to the App based on the network connection.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP.

2.2.2 Request

The App functions as the client and sends a bind_receiver request message to the SDP to establish a network connection for receiving messages.

Message Header Parameters

Table 2-2 describes parameters in a bind_receiver request message header.

Table 2-2 Parameters in a bind_receiver request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the bind_receiver PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to bind_receiver request. Set it to **0x00000001**. [Example] 0x00000001
command_status	Integer	4	Mandatory	Not used in bind_receiver PDU. Set it to **NULL**. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set to a unique sequence number. The associated bind_receiver_resp PDU will echo the same sequence number. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 2-3 describes parameters in a bind_receiver request message body.

Table 2-3 Parameters in a bind_receiver request message body

Parameter	Type	Length	Level of Requirement	Description
system_id	C-Octet String	Var. max 16	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 0xC9
password	C-Octet String	Var. max 9	Mandatory	Authentication key for the SDP to authenticate partners. To obtain the ID: • An SP can obtain the password from the email notification received after successful registration. • A Developer can log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set the password. • An Enterprise must contact the carrier to obtain the password.
system_type	C-Octet String	Var. max 13	Mandatory	Identifies the type of App requesting to bind as a receiver with the SDP. Set it to **NULL**. [Example] 0x00
interface_version	Integer	1	Mandatory	The version of the SMPP protocol. • 0x00-0x33: SMPP version 3.3 • 0x34: SMPP version 3.4 [Example] 0x34

Parameter	Type	Length	Level of Requirement	Description
addr_ton	Integer	1	Mandatory	Type of Number (TON) for App address(es) served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00
addr_npi	Integer	1	Mandatory	Numbering Plan Indicator (NPI) for App address(es) served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00
address_range	C-Octet String	Var. max 41	Mandatory	A single App address or a range of App addresses served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00

2.2.3 Response

The SDP functions as the server, processes bind_receiver request messages received from the App, and sends bind_receiver response messages to the App.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see 1.7 Error Response Format .

Message Header Parameters

Table 2-4 describes parameters in a response message header.

Table 2-4 Parameters in a bind_receiver request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the bind_receiver_resp PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to bind_receiver_resp. Set it to **0x80000001**. [Example] 0x80000001
command_st	Integer	4	Mandatory	Indicates status (success or error code) of original bind_receiver request.

Parameter	Type	Length	Level of Requirement	Description
atus				[Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set it to the sequence number of original bind_receiver request. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Step 2 **Message Body Parameters**

Table 2-5 describes parameters in a response message body.

Table 2-5 Parameters in a response message body

Parameter	Type	Length	Level of Requirement	Description
system_id	C-Octet String	Var. max 16	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 0xC9

2.2.4 Error Codes

Table 2-6 describes bind_receiver error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 2-6 bind_receiver error codes

Error Code	Description
0x00000001	Message Length is invalid.
0x0000000D	Bind Failed.
0x0000000E	Invalid Password.

2.3 bind_transceiver

2.3.1 Function

The App (functioning as the client) invokes the API to create a bidirectional link with the SDP (functioning as the server). The SDP send a response within 60 seconds by default.

After the link is created, the App sends and receives messages using the link.

- The SDP sends MO SMS messages received from users to the App using the link.
- The App (functioning as the client) sends SMS messages to the SDP (functioning as the server) using the link.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP.

2.3.2 Request

The App functions as the client and sends a bind_transceiver request message to the SDP to establish a network connection for receiving or sending messages.

Message Header Parameters

Table 2-7 describes parameters in a bind_transceiver request message header.

Table 2-7 Parameters in a bind_transceiver request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the bind_transceiver PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to bind_transceiver request set it to **0x00000009**. [Example] 0x00000009
command_st	Integer	4	Mandatory	Not used in bind_transceiver PDU.

Parameter	Type	Length	Level of Requirement	Description
atus				Set it to **NULL**. [Example] 0x00000000
sequence_number	Integer	4	Conditional	sequence number. Set to a unique sequence number. The associated bind_transceiver_resp PDU will echo the same sequence number. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 2-8 describes parameters in a bind_transceiver request message body.

Table 2-8 Parameters in a bind_transceiver request message body

Parameter	Type	Length	Level of Requirement	Description
system_id	C-Octet String	Var. max 16	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 0xC9
password	C-Octet String	Var. max 9	Mandatory	Authentication key for the SDP to authenticate partners. To obtain the ID: • An SP can obtain the password from the email notification received after successful registration. • A Developer can log in to the Developer Portal, choose **Member Center** > **Account** >

Parameter	Type	Length	Level of Requirement	Description
				Registration Information > **Invoke Password**, and set the password. - An Enterprise must contact the carrier to obtain the password.
system_type	C-Octet String	Var. max 13	Mandatory	Identifies the type of App requesting to bind as a receiver with the SDP. Set it to **NULL**. [Example] 0x00
interface_version	Integer	1	Mandatory	The version of the SMPP protocol. - 0x00-0x33: SMPP version 3.3 - 0x34: SMPP version 3.4 [Example] 0x34
addr_ton	Integer	1	Mandatory	Type of Number (TON) for App address(es) served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00
addr_npi	Integer	1	Mandatory	Numbering Plan Indicator (NPI) for App address(es) served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00
address_range	C-Octet String	Var. max 41	Mandatory	A single App address or a range of App addresses served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00

2.3.3 Response

The SDP functions as the server, processes bind_transceiver request messages received from the App, and sends bind_transceiver response messages to the App.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see 1.7 Error Response Format .

Message Header Parameters

Table 2-9 describes parameters in a response message header.

Table 2-9 Parameters in a bind_transceiver request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the bind_transceiver_resp PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to bind_transceiver_resp. Set it to **0x80000009**. [Example] 0x80000009
command_status	Integer	4	Mandatory	Indicates status (success or error code) of original bind_transceiver request. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set it to the sequence number of original bind_transceiver request. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 2-10 describes parameters in a response message body.

Table 2-10 Parameters in a response message body

Parameter	Type	Length	Level of Requirement	Description
system_id	C-Octet String	Var. max 16	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier.

Parameter	Type	Length	Level of Requirement	Description
				[Example] 0xC9

2.3.4 Error Codes

Table 2-11 describes bind_transceiver error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 2-11 bind_transceiver error codes

Error Code	Description
0x00000001	Message Length is invalid.
0x0000000D	Bind Failed.
0x0000000E	Invalid Password.

2.4 enquire_link

2.4.1 Function

After the link is created, the App (functioning as the client) invokes the enquire_link API to send heartbeat messages to the SDP (functioning as the server) to maintain the link.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP.

2.4.2 Request

The App (functioning as the client) invokes the enquire_link API to send heartbeat messages to the SDP (functioning as the server) to maintain the link.

Message Header Parameters

Table 2-12 describes parameters in a enquire_link request message header.

Table 2-12 Parameters in a enquire_link request message header

Parameter	Type	Length	Level of Requirement	Description
command_l	Integer	4	Mandatory	The overall length of the enquire_link PDU in octets.

Parameter	Type	Length	Level of Requirement	Description
ength				[Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to enquire_link request Set it to **0x00000015**. [Example] 0x00000015
command_status	Integer	4	Mandatory	Not used in bind_transceiver PDU. Set it to **NULL**. [Example] 0x00000000
sequence_number	Integer	4	Conditional	sequence number. Set to a unique sequence number. The associated enquire_link_resp PDU will echo the same sequence number. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

None.

2.4.3 Response

The SDP functions as the server, processes enquire_link request messages received from the App, and sends enquire_link response messages to the App.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see 1.7 Error Response Format .

Message Header Parameters

Table 2-13 describes parameters in a response message header.

Table 2-13 Parameters in a enquire_link request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the enquire_link_resp PDU in octets. [Example] 0x400
comm	Integer	4	Mandat	Value corresponding to enquire_link_resp.

Parameter	Type	Length	Level of Requirement	Description
and_id			ory	Set it to **0x80000015**. [Example] 0x80000015
command_status	Integer	4	Mandatory	Indicates status (success or error code) of original enquire_link request. Set it to **ESME_ROK**. [Example] ESME_ROK
sequence_number	Integer	4	Mandatory	sequence number. Set it to the sequence number of original enquire_link request. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

None.

2.4.4 Error Codes

Table 2-14 describes enquire_link error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 2-14 enquire_link error codes

Error Code	Description
0x00000003	Invalid Command ID

2.5 deliver_sm

2.5.1 Function

The SDP (functioning as the client) invokes the deliver_sm API to send MO SMS messages to the App (functioning as the server).

After the App creates a unidirectional or bidirectional link using the 2.3 bind_transceiver or 3.2 bind_transmitter API, the SDP invokes the deliver_sm API to send SMS messages received from users to the App. If the MO SMS messages fail to be sent, the SDP resends the messages to the App when any of the cached message resending criteria is met. Cached SMS messages can be resent for a maximum of five times. SMS messages can be resent at least 1800 seconds after a sending failure.

Partners must code the App based on the API field requirements so that the App can correctly parse and respond to requests received from the SDP. The App must send a response to the SDP within 30 seconds.

2.5.2 Request

The SDP functions as the client and sends deliver_sm request messages to the App to delivery an MO SMS message.

Message Header Parameters

Table 2-15 describes parameters in a deliver_sm request message header.

Table 2-15 Parameters in a deliver_sm request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the deliver_sm PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to deliver_sm request. Set it to **0x00000005**. [Example] 0x00000005
command_status	Integer	4	Mandatory	Not used in deliver_sm PDU. Set it to **NULL**. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set to a unique sequence number. The associated deliver_sm_resp PDU will echo the same sequence number. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 2-16 describes parameters in a deliver_sm request message body.

Table 2-16 Parameters in a deliver_sm request message body

Parameter	Type	Length	Level of Requirement	Description
servic	C-Octet	Var.	Mandat	The service_type parameter can be used to

Parameter	Type	Length	Level of Requirement	Description
e_type	String	max 13	ory	indicate the SMS Application service associated with the message. Specifying the service_type allows the App to avail of enhanced messaging services such as **replace by service** type to control the teleservice used on the air interface. The options are as follows: • NULL: Default • CMT: Cellular Messaging • CPT: Cellular Paging • VMN: Voice Mail Notification • VMA: Voice Mail Alerting • WAP: Wireless Application Protocol • USSD: Unstructured Supplementary Services Data [Example] 0x00
source_addr_ton	Integer	1	Mandatory	Type of Number for source address. The options are as follows: • 00000000: Unknown • 00000001: International • 00000010: National • 00000011: Network Specific • 00000100: Subscriber Number • 00000101: Alphanumeric • 00000110: Abbreviated • other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
source_addr_npi	Integer	1	Mandatory	Numbering Plan Indicator for source address. The options are as follows: • 00000000: Unknown • 00000001: ISDN (E163/E164) • 00000011: Data (X.121) • 00000100: Telex (F.69) • 00000110: Land Mobile (E.212) • 00001000: National

Parameter	Type	Length	Level of Requirement	Description
				• 00001001: Private • 00001010: ERMES • 00001110: Internet (IP) • 00010010: WAP Client Id (to be defined by WAP Forum) • Other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
source_addr	C-Octet String	Var. max 41	Mandatory	Mobile number of the message sender. The format of the source mobile number is determined by the **source_addr_ton** and **source_addr_npi** parameters. [Example] 0x7D535D16C4E
dest_addr_ton	Integer	1	Mandatory	Type of Number for destination. The options are as follows: • 00000000: Unknown • 00000001: International • 00000010: National • 00000011: Network Specific • 00000100: Subscriber Number • 00000101: Alphanumeric • 00000110: Abbreviated • other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. [Example] 0x00
dest_addr_npi	Integer	1	Mandatory	Numbering Plan Indicator for destination. The options are as follows: • 00000000: Unknown • 00000001: ISDN (E163/E164) • 00000011: Data (X.121) • 00000100: Telex (F.69) • 00000110: Land Mobile (E.212)

Parameter	Type	Length	Level of Requirement	Description
				- 00001000: National - 00001001: Private - 00001010: ERMES - 00001110: Internet (IP) - 00010010: WAP Client Id (to be defined by WAP Forum) - Other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. [Example] 0x00
destination_addr	C-Octet String	Var. max 21	Mandatory	Address of the App which received this message. - In an SP's request, the value is a service access code obtained from carriers before service release. - In an Enterprise's or a Developer's request, the value is an access code allocated by carriers during capability product purchase. The message format is determined by the **destination_addr_ton** and **destination_addr_npi** parameters. [Example] 0x7D7B
esm_class	Integer	1	Mandatory	Indicates Message Type and enhanced network services. Bits 5-2 indicates the Message Type. When the bits 2 to 5 in the **esm_class** field are **0000**, it indicates an MO SMS message. xx0000xx: Default message Type (for example, normal message) [Example] 0x000000
protocol_id	Integer	1	Mandatory	Protocol Identifier. Network specific field. - GSM: Set according to GSM 03.40. **NOTE** For details, see *Technical Realisation of the Short Message Service Point to Point* at http://www.etsi.fr. - ANSI-136 (TDMA): For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For ANSI-136 mobile originated messages, the SMSC should set this value to NULL.

Parameter	Type	Length	Level of Requirement	Description
				- IS-95 (CDMA): For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For IS-95 mobile originated messages, the SMSC should set this value to NULL. [Example] 0x00
priority_flag	Integer	1	Mandatory	Designates the priority level of the message. The options are as follows: - 0x00: Level 0 (lowest) priority - 0x01: Level 1 priority - 0x02: Level 2 priority - 0x03: Level 3 (highest) priority - >0x03: Reserved The value options have different meanings on the GSM, ANSI-136, and IS-95 mobile networks. - GSM: For GSM mobile terminated messages with priority greater than Level 0 are treated as priority when making a delivery attempt (for example, a delivery attempt is made even when MWD is set in the HLR). ◆ 0x00: non-priority ◆ 0x01, 0x02, 0x03: priority - ANSI-136: ◆ 0x00: Bulk ◆ 0x01: Normal ◆ 0x02: Urgent ◆ 0x03: Very Urgent - IS-95: ◆ 0x00: Normal ◆ 0x01: Interactive ◆ 0x02: Urgent ◆ 0x03: Emergency [Example] 0x00
schedule_delivery_time	C-Octet String	1 or 17	Mandatory	This field is unused for **deliver_sm**. It must be set to NULL. [Example] 0x00
validity_peri	C-Octet	1 or 17	Mandat	This field is unused for **deliver_sm**. It must be

Parameter	Type	Length	Level of Requirement	Description
od	String		ory	set to NULL. [Example] 0x00
registered_delivery	Integer	1	Mandatory	Indicates if an App acknowledgement is required. The options are as follows: • xxxx00xx: No recipient SME acknowledgment requested (default). • xxxx01xx: SME Delivery Acknowledgement requested. • xxxx10xx: SME Manual/User Acknowledgment requested. • xxxx11xx: Both Delivery and Manual/User Acknowledgment requested [Example] 0x0A
replace_if_present_flag	Integer	1	Mandatory	Reserved. Must be set it to NULL. [Example] 0x00
data_coding	Integer	1	Mandatory	Defines the encoding scheme of the short message user data. The options are as follows: • 00000000L: SMSC Default Alphabet. • 00000001: IA5(CCITT T.50)/ASCII(ANSI X3.4). • 00000010: Octet unspecified (8-bit binary). • 00000011: Latin 1(ISO-8859-1). • 00000100: Octet unspecified (8-bit binary). • 00000101: JIS(X 0208-1990). • 00000110: Cyrllic(ISO-8859-5). • 00000111: Latin/Hebrew(ISO-8859-8). • 00001000: UCS2(ISO/IEC-10646). • 00001001: Pictogram Encoding. • 00001010: ISO-2022-JP(Music Codes). • 00001011: Reserved. • 00001100: Reserved. • 00001101: Extended Kanji JIS(X 0212-1990). • 00001110: KS C 5601. • 00001111–10111111: Reserved. • 1100xxxx: GSM MWI control. See *Digital*

Parameter	Type	Length	Level of Requirement	Description
				Cellular telecommunications system (Phase 2+) from http://www.etsi.fr for details. - 1101xxxx: GSM MWI control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. - 11101111: Reserved. - 1111xxxx: GSM message class control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. [Example] 0x00
sm_default_msg_id	Integer	1	Mandatory	This field is unused for **deliver_sm**. It must be set to NULL. [Example] 0x00
sm_length	Integer	1	Mandatory	Length in octets of the **short_message** field. It should be set to **0** if the message_payload parameter is being used to send user data larger than 254 octets. - 0x00: No user data in short message field. - 0x01–0xFE: Allowed - 0xFF: Not allowed [Example] 0x11
short_message	Octet String	Var. 0-254	Mandatory	MO SMS message content. When the bits 2 to 5 in the **esm_class** field are **0000**, the **short_message** indicates an MO SMS message content. Up to 254 octets of short message user data. The exact physical limit for **short_message** size may vary according to the underlying network. Apps which need to send messages longer than 254 octets should use the **message_payload** parameter. In this case the **sm_length** field should be set to zero. **NOTE** The short message data should be inserted in either the **short_message** or **message_payload** fields. Both fields must not be used simultaneously. [Example] sms test

Parameter	Type	Length	Level of Requirement	Description
source_port	TLV	N/A	Optional	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications. For details about the **source_port** type, see Table 2-17.
destination_port	TLV	N/A	Optional	Indicates the application port number associated with the destination address of the message. This parameter should be present for WAP applications. For details about the **destination_port** type, see Table 2-18.
sar_msg_ref_num	TLV	N/A	Optional	The reference number for a particular concatenated short message. This parameter shall contain a originator generated reference number so that a segmented short message may be reassembled into a single original message. This allows the parallel transmission of several segmented messages. This reference number shall remain constant for every segment which makes up a particular concatenated short message. When present, the PDU must also contain the **sar_total_segments** and **sar_segment_seqnum** parameters. Otherwise this parameter shall be ignored. For details about the **sar_msg_ref_num** type, see Table 2-19.
sar_total_segments	TLV	N/A	Optional	Indicates the total number of short messages within the concatenated short message. When present, the PDU must also contain the **sar_msg_ref_num** and **sar_segment_seqnum** parameters. Otherwise this parameter shall be ignored. For details about the **sar_total_segments** type, see Table 2-20.
sar_segment_seqnum	TLV	N/A	Optional	Indicates the sequence number of a particular short message fragment within the concatenated short message. When present, the PDU must also contain the **sar_total_segments** and **sar_msg_ref_num** parameters. Otherwise this parameter shall be ignored.

Parameter	Type	Length	Level of Requirement	Description
				For details about the **sar_segment_seqnum** type, see Table 2-21.
payload_type	TLV	N/A	Optional	Defines the type of payload (for example, WDP and WCMP). For details about the **payload_type** type, see Table 2-22.
message_payload	TLV	N/A	Optional	Contains the extended short message user data. Up to 64K octets can be transmitted. 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. For details about the **message_payload** type, see Table 2-23.
receipted_message_id	TLV	N/A	Optional	Unique ID of the MO SMS message received by the SDP. For details about the **receipted_message_id** type, see Table 2-24.

Table 2-17 describes the parameter structure of the **source_port** type.

Table 2-17 Parameter structure of the source_port type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020A**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	All values allowed. [Example] 0x50

Table 2-18 describes the parameter structure of the **destination_port** type.

Table 2-18 Parameter structure of the destination_port type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020B**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	All values allowed. [Example] 0x50

Table 2-19 describes the parameter structure of the **sar_msg_ref_num** type.

Table 2-19 Parameter structure of the sar_msg_ref_num type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020C**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	2	Mandatory	Reference number. [Example] 0x0001

Table 2-20 describes the parameter structure of the **sar_total_segments** type.

Table 2-20 Parameter structure of the sar_total_segments type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020E**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	This parameter shall contain a value in the range 1 to 255 indicating the total number of fragments within the concatenated short message. The value shall start at 1 and remain constant for

Parameter	Type	Size octets	Level of Requirement	Description
				every short message which makes up the concatenated short message. [Example] 0x05

Table 2-21 describes the parameter structure of the **sar_segment_seqnum** type.

Table 2-21 Parameter structure of the sar_segment_seqnum type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020F**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	This octet shall contain a value in the range 1 to 255 indicating the sequence number of a particular message within the concatenated short message. The value shall start at 1 and increment by one for every message sent within the concatenated short message. [Example] 0x01

Table 2-22 describes the parameter structure of the **payload_type** type.

Table 2-22 Parameter structure of the payload_type type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x0019**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Integer	1	Mandatory	• 0x00: Default.In the case of a WAP application, the default

Parameter	Type	Size octets	Level of Requirement	Description
				higher layer message type is a WDP message. See *Wireless Datagram Protocol Specification* from http://www.wapforum.org for details. • 0x01: WCMP message. Wireless Control Message Protocol formatted data. See *Wireless Datagram Protocol Specification* from http://www.wapforum.org for details. • 0x02–0xFF: Reserved [Example] 0x01

Table 2-23 describes the parameter structure of the **message_payload** type.

Table 2-23 Parameter structure of the message_payload type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x0424**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Integer	1	Mandatory	Short message user data. The maximum size is SMSC and network implementation specific. Contact carriers to obtain the value. [Example] 0x01

Table 2-24 describes the parameter structure of the **receipted_message_id** type.

Table 2-24 Parameter structure of the receipted_message_id type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x001E**.

Parameter	Type	Size octets	Level of Requirement	Description
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	C-Octet String	Var. max 65	Mandatory	Message ID. [Example] 0xB3A73CE2F77

2.5.3 Response

The App functions as the server, processes deliver_sm request messages received from the SDP, and sends response messages to the SDP

This topic provides a success response example and describes parameters in the response. If a request fails, the App sends an error response that contains an error code. For details about error responses, see API Error Responses.

Message Header Parameters

Table 2-25 describes parameters in a response message header.

Table 2-25 Parameters in a deliver_sm request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the deliver_sm_resp PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to deliver_sm_resp. Set it to **0x80000005**. [Example] 0x80000005
command_status	Integer	4	Mandatory	Indicates status (success or error code) of original deliver_sm request. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set it to the sequence number of original deliver_sm request. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 2-26 describes parameters in a response message body.

Table 2-26 Parameters in a response message body

Parameter	Type	Length	Level of Requirement	Description
message_id	C-Octet String	Var. max 65	Mandatory	This field is unused and is set to NULL. [Example] 0x00

2.5.4 Error Codes

The App returns error codes to the SDP when an exception occurs in response to deliver_sm requests. The error codes are provided by the App based on SMPP specifications.

3 APIs for Sending SMS Messages

About This Chapter

3.1 Process

After the App sends an SMS message to a user, the SDP sends a status report to the App in Notify mode. Figure 3-1 shows the process of sending SMS messages and receiving status reports in Notify mode.

Figure 3-1 Process of sending SMS messages and receiving status reports in Notify mode

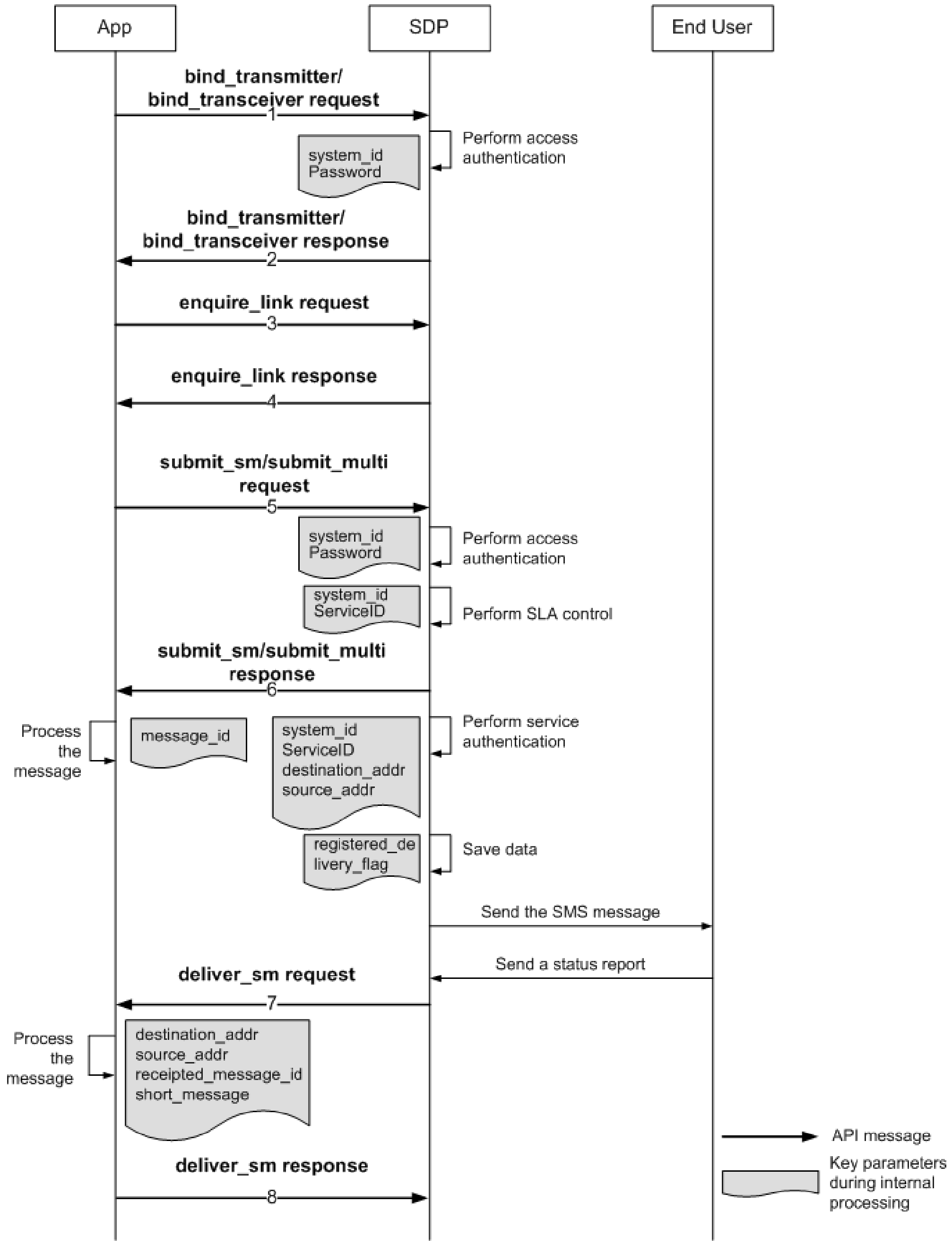

Table 3-1 describes the process.

Table 3-1 Description for the process of sending SMS messages and receiving status reports

Step	Description
1–2	• The App sends a request to the App to create a link. • The App sends a response to the App.
3–4	• The App sends heartbeat messages to the SDP to maintain the link.

Step	Description
	• The SDP sends responses to the App.
5–6	• The App sends a request to the SDP to send an SMS message. • The SDP performs authentication and SLA control based on fields in the request and sends a response to the App. Then the SDP authenticates the service, determines whether the App requires a status report based on the **registered_delivery_flag** field in the request, and saves the mapping between the SMS message and status report.
7–8	• The SDP receives a status report and sends a notification of the status report to the App based on **registered_delivery_flag**. • The App parses the notification and sends a response to the SDP.

3.2 bind_transmitter

3.2.1 Function

The App (functioning as the client) invokes the bind_transmitter API to establish a network connection with the SDP (functioning as the server) to send messages.

After the network connection is established, the App can send MT SMS messages to the SDP based on the network connection.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

3.2.2 Request

The App functions as the client and sends a bind_transmitter request message to the SDP to establish a network connection for receiving messages.

Message Header Parameters

Table 3-2 describes parameters in a bind_transmitter request message header.

Table 3-2 Parameters in a bind_transmitter request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the bind_transmitter PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to bind_transmitter request Set it to **0x00000002**.

Parameter	Type	Length	Level of Requirement	Description
				[Example] 0x00000002
command_status	Integer	4	Mandatory	Not used in bind_transmitter PDU. Set it to **NULL**. [Example] 0x00000000
sequence_number	Integer	4	Conditional	sequence number. Set to a unique sequence number. The associated bind_transmitter_resp PDU will echo the same sequence number. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 3-3 describes parameters in a bind_transmitter request message body.

Table 3-3 Parameters in a bind_transmitter request message body

Parameter	Type	Length	Level of Requirement	Description
system_id	C-Octet String	Var. max 16	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 0xC9
password	C-Octet String	Var. max 9	Mandatory	Authentication key for the SDP to authenticate partners. To obtain the ID: • An SP can obtain the password from the email notification received after

Parameter	Type	Length	Level of Requirement	Description
				successful registration. • A Developer can log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set the password. • An Enterprise must contact the carrier to obtain the password.
system_type	C-Octet String	Var. max 13	Mandatory	Identifies the type of App requesting to bind as a receiver with the SDP. Set it to **NULL**. [Example] 0x00
interface_version	Integer	1	Mandatory	The version of the SMPP protocol. • 0x00-0x33: SMPP version 3.3 • 0x34: SMPP version 3.4 [Example] 0x34
addr_ton	Integer	1	Mandatory	Type of Number (TON) for App address(es) served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00
addr_npi	Integer	1	Mandatory	Numbering Plan Indicator (NPI) for App address(es) served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00
address_range	C-Octet String	Var. max 41	Mandatory	A single App address or a range of App addresses served via this SMPP receiver session. Set it to **NULL**. [Example] 0x00

3.2.3 Response

The SDP functions as the server, processes bind_transmitter request messages received from the App, and sends bind_transmitter response messages to the App.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see 1.7 Error Response Format .

Message Header Parameters

Table 3-4 describes parameters in a response message header.

Table 3-4 Parameters in a bind_transmitter request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the bind_transmitter_resp PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to bind_transmitter_resp. Set it to **0x80000002**. [Example] 0x80000002
command_status	Integer	4	Mandatory	Indicates status (success or error code) of original bind_transmitter request. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set it to the sequence number of original bind_transmitter request. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 3-5 describes parameters in a response message body.

Table 3-5 Parameters in a response message body

Parameter	Type	Length	Level of Requirement	Description
system_id	C-Octet String	Var. max 16	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer

Parameter	Type	Length	Level of Requirement	Description
				Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 0xC9

3.2.4 Error Codes

Table 3-6 describes bind_transmitter error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 3-6 bind_transmitter error codes

Error Code	Description
0x00000001	Message Length is invalid.
0x0000000D	Bind Failed.
0x0000000E	Invalid Password.

3.3 submit_sm

3.3.1 Function

The App (functioning as the client) invokes the submit_sm API to send single SMS messages to the SDP (functioning as the server).

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

3.3.2 Request

The App functions as the client and sends submit_sm request messages to the SDP.

Message Header Parameters

Table 3-7 describes parameters in a submit_sm request message header.

Table 3-7 Parameters in a submit_sm request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the submit_sm PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to submit_sm request. Set it to **0x00000004**. [Example] 0x00000004
command_status	Integer	4	Mandatory	Not used in submit_sm PDU. Set it to **NULL**. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set to a unique sequence number. The associated submit_sm_resp PDU will echo the same sequence number. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 3-8 describes parameters in a submit_sm request message body.

Table 3-8 Parameters in a submit_sm request message body

Parameter	Type	Length	Level of Requirement	Description
service_type	C-Octet String	Var. max 13	Mandatory	The service_type parameter can be used to indicate the SMS Application service associated with the message. Specifying the service_type allows the App to avail of enhanced messaging services such as **replace by service** type to control the teleservice used on the air interface. The options are as follows: • NULL: Default • CMT: Cellular Messaging • CPT: Cellular Paging • VMN: Voice Mail Notification • VMA: Voice Mail Alerting

Parameter	Type	Length	Level of Requirement	Description
				• WAP: Wireless Application Protocol • USSD: Unstructured Supplementary Services Data Set it to NULL for default SMSC settings. [Example] 0x00
source_addr_ton	Integer	1	Mandatory	Type of Number for source address. The options are as follows: • 00000000: Unknown • 00000001: International • 00000010: National • 00000011: Network Specific • 00000100: Subscriber Number • 00000101: Alphanumeric • 00000110: Abbreviated • other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
source_addr_npi	Integer	1	Mandatory	Numbering Plan Indicator for source address. The options are as follows: • 00000000: Unknown • 00000001: ISDN (E163/E164) • 00000011: Data (X.121) • 00000100: Telex (F.69) • 00000110: Land Mobile (E.212) • 00001000: National • 00001001: Private • 00001010: ERMES • 00001110: Internet (IP) • 00010010: WAP Client Id (to be defined by WAP Forum) • Other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format.

Parameter	Type	Length	Level of Requirement	Description
				If not known, set it to **NULL**. [Example] 0x00
source_addr	C-Octet String	Var. max 41	Mandatory	Address of the App which originated this message. • In an SP's request, the value is a service access code obtained from carriers before service release. • In an Enterprise's or a Developer's request, the value is an access code allocated by carriers during capability product purchase. [Example] 0x4E663
dest_addr_ton	Integer	1	Mandatory	Type of Number for destination. The options are as follows: • 00000000: Unknown • 00000001: International • 00000010: National • 00000011: Network Specific • 00000100: Subscriber Number • 00000101: Alphanumeric • 00000110: Abbreviated • other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
dest_addr_npi	Integer	1	Mandatory	Numbering Plan Indicator for destination. The options are as follows: • 00000000: Unknown • 00000001: ISDN (E163/E164) • 00000011: Data (X.121) • 00000100: Telex (F.69) • 00000110: Land Mobile (E.212) • 00001000: National • 00001001: Private • 00001010: ERMES • 00001110: Internet (IP) • 00010010: WAP Client Id (to be

Parameter	Type	Length	Level of Requirement	Description
				defined by WAP Forum) - Other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
destination_addr	C-Octet String	Var. max 21	Mandatory	Mobile number of the message recipient. The format of the destination mobile number is determined by the **dest_addr_ton** and **dest_addr_npi** parameters. [Example] 8612312345678
esm_class	Integer	1	Mandatory	Special message attributes (including the message mode and type) of the SMS message. The esm_class parameter is encoded as follows: - bits 1-0: Messaging Mode ◆ xxxxxx00: Default SMSC Mode (for example, Store and Forward) ◆ xxxxxx01: Datagram mode ◆ xxxxxx10: Forward (for example, Transaction) mode ◆ xxxxxx11: Store and Forward mode(use to select Store and Forward mode if Default SMSC Mode is non Store and Forward) - bits 5-2: Message Type ◆ xx0000xx: Default message Type (for example, normal message) ◆ xx0010xx: Short Message contains App Delivery Acknowledgement ◆ xx0100xx: Short Message contains App Manual/User Acknowledgement - bits 7-6: GSM Network Specific Features ◆ 00xxxxxx: No specific features selected ◆ 01xxxxxx: UDHI Indicator (only relevant for MT short messages)

Parameter	Type	Length	Level of Requirement	Description
				◆ 10xxxxxx: Set Reply Path (only relevant for GSM network) ◆ 11xxxxxx: Set UDHI and Reply Path (only relevant for GSM network) [Example] 0x40
protocol_id	Integer	1	Mandatory	Protocol Identifier. Network specific field. • GSM: Set according to GSM 03.40. **NOTE** For details, see *Technical Realisation of the Short Message Service Point to Point* at http://www.etsi.fr. • ANSI-136 (TDMA): For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For ANSI-136 mobile originated messages, the SMSC should set this value to NULL. • IS-95 (CDMA): For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For IS-95 mobile originated messages, the SMSC should set this value to NULL. [Example] 0x00
priority_flag	Integer	1	Mandatory	Designates the priority level of the message. The options are as follows: • 0x00: Level 0 (lowest) priority • 0x01: Level 1 priority • 0x02: Level 2 priority • 0x03: Level 3 (highest) priority • >0x03: Reserved The value options have different meanings on the GSM, ANSI-136, and IS-95 mobile networks. • GSM: For GSM mobile terminated messages with priority greater than Level 0 are treated as priority when making a delivery attempt (for example, a delivery attempt is made even when MWD is set in the HLR). ◆ 0x00: non-priority ◆ 0x01, 0x02, 0x03: priority

Parameter	Type	Length	Level of Requirement	Description
				• ANSI-136: ◆ 0x00: Bulk ◆ 0x01: Normal ◆ 0x02: Urgent ◆ 0x03: Very Urgent • IS-95: ◆ 0x00: Normal ◆ 0x01: Interactive ◆ 0x02: Urgent ◆ 0x03: Emergency [Example] 0x00
schedule_delivery_time	C-Octet String	1 or 17	Mandatory	Scheduled time at which the message delivery is first attempted. The time can be the absolute time or relative time. If it is the relative time, the SMSC calculates the time based on the system time. Set to NULL for immediate message delivery. [Format] YYMMDDhhmmsstnnp In the format: • t: tenths of second (0-9) • nn: Time difference in quarter hours between local time (as expressed in the first 13 octets) and UTC (Universal Time Constant) time (00-48). • p: UTC wizard ID. ◆ +: Local time is in quarter hours advanced in relation to UTC time. ◆ -: Local time is in quarter hours retarded in relation to UTC time. ◆ R: Local time is relative to the current SMSC time. If the time is the relative time, set **t** and **nn** to **0**. [Example] 0x4CC773455000R
validity_period	C-Octet String	1 or 17	Mandatory	Validity period of this message. The parameter indicates the SMSC expiration time, after which the message should be discarded if not delivered to the destination. The time can be the absolute time or relative time. If it is the relative time, the SMSC calculates the time based

Parameter	Type	Length	Level of Requirement	Description
				on the system time. Set it to NULL to request the SMSC default validity period. [Format] YYMMDDhhmmsstnnp In the format: • t: tenths of second (0-9) • nn: Time difference in quarter hours between local time (as expressed in the first 13 octets) and UTC (Universal Time Constant) time (00-48). • p: UTC wizard ID. ◆ +: Local time is in quarter hours advanced in relation to UTC time. ◆ -: Local time is in quarter hours retarded in relation to UTC time. ◆ R: Local time is relative to the current SMSC time. If the time is the relative time, set **t** and **nn** to **0**. [Example] 0xE1754058D8120+
registered_delivery	Integer	1	Mandatory	Indicator to signify if an SMSC delivery receipt or an App acknowledgement is required. The options are as follows: • xxxxxx00: No SMSC Delivery Receipt requested (default). • xxxxxx01: SMSC Delivery Receipt requested where final delivery outcome is delivery success or failure. • xxxxxx10: SMSC Delivery Receipt requested where the final delivery outcome is delivery failure. [Example] 0x01
replace_if_present_flag	Integer	1	Mandatory	Flag indicating if submitted message should replace an existing message. The SMSC will replace an existing message provided that the source address, destination address and service_type match the same fields in the new message. The options are as follows: • 0x00: Do not replace (default) • 0x01: Replace • 0x02–0xFF: reserved

Parameter	Type	Length	Level of Requirement	Description
				[Example] 0x00
data_coding	Integer	1	Mandatory	Defines the encoding scheme of the short message user data. The options are as follows: • 00000000L: SMSC Default Alphabet. • 00000001: IA5(CCITT T.50)/ASCII(ANSI X3.4). • 00000010: Octet unspecified (8-bit binary). • 00000011: Latin 1(ISO-8859-1). • 00000100: Octet unspecified (8-bit binary). • 00000101: JIS(X 0208-1990). • 00000110: Cyrllic(ISO-8859-5). • 00000111: Latin/Hebrew(ISO-8859-8). • 00001000: UCS2(ISO/IEC-10646). • 00001001: Pictogram Encoding. • 00001010: ISO-2022-JP(Music Codes). • 00001011: Reserved. • 00001100: Reserved. • 00001101: Extended Kanji JIS(X 0212-1990). • 00001110: KS C 5601. • 00001111–10111111: Reserved. • 1100xxxx: GSM MWI control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. • 1101xxxx: GSM MWI control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. • 11101111: Reserved. • 1111xxxx: GSM message class control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. [Example] 0x00

Parameter	Type	Length	Level of Requirement	Description
sm_default_msg_id	Integer	1	Mandatory	Indicates the short message to send from a list of predefined (canned) short messages stored on the SMSC. If not using an SMSC canned message, set to NULL. The options are as follows: • 0x00: Reserved • 0x01–0xFE: Allowed values • 0XFF: Reserved [Example] 0x00
sm_length	Integer	1	Mandatory	Length in octets of the **short_message** field. It should be set to **0** if the message_payload parameter is being used to send user data larger than 254 octets. • 0x00: No user data in short message field. • 0x01–0xFE: Allowed • 0xFF: Not allowed [Example] 0x11
short_message	Octet String	Var. 0-254	Mandatory	Up to 254 octets of short message user data. The exact physical limit for short_message size may vary according to the underlying network. Apps which need to send messages longer than 254 octets should use the **message_payload** parameter. In this case the **sm_length** field should be set to zero. **NOTE** The short message data should be inserted in either the **short_message** or **message_payload** fields. Both fields must not be used simultaneously. [Example] sms test
source_port	TLV	-	Optional	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications. For details about the **source_port** type, see Table 3-9.
destination_port	TLV	-	Optional	Indicates the application port number associated with the destination address of the message. This parameter should be

Parameter	Type	Length	Level of Requirement	Description
				present for WAP applications. For details about the **destination_port** type, see Table 3-10.
sar_msg_ref_num	TLV	-	Optional	The reference number for a particular concatenated short message. This parameter shall contain a originator generated reference number so that a segmented short message may be reassembled into a single original message. This allows the parallel transmission of several segmented messages. This reference number shall remain constant for every segment which makes up a particular concatenated short message. When present, the PDU must also contain the **sar_total_segments** and **sar_segment_seqnum** parameters. Otherwise this parameter shall be ignored. For details about the **sar_msg_ref_num** type, see Table 3-11.
sar_total_segments	TLV	-	Optional	Indicates the total number of short messages within the concatenated short message. When present, the PDU must also contain the **sar_msg_ref_num** and **sar_segment_seqnum** parameters. Otherwise this parameter shall be ignored. For details about the **sar_total_segments** type, see Table 3-12.
sar_segment_seqnum	TLV	-	Optional	Indicates the sequence number of a particular short message fragment within the concatenated short message. When present, the PDU must also contain the **sar_total_segments** and **sar_msg_ref_num** parameters. Otherwise this parameter shall be ignored. For details about the **sar_segment_seqnum** type, see Table 3-13.
payload_type	TLV	-	Optional	Defines the type of payload (for example, WDP and WCMP). For details about the **payload_type** type, see Table 3-14.

Parameter	Type	Length	Level of Requirement	Description
message_payload	TLV	-	Optional	Contains the extended short message user data. Up to 64K octets can be transmitted. 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. For details about the **message_payload** type, see Table 3-15.
serviceId	TLV	-	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in an SP's request, and can be left empty in a Developer's or an Enterprise's request. For details about the **serviceId** type, see Table 3-16.
link_id	TLV	-	Conditional	Service order ID. The ID is automatically generated by the SDP when a user orders a service in the SDP. This parameter is mandatory during on-demand service delivery by SMS message. The SDP sends the value to SPs as follows in different scenarios: ● Invokes the ServiceOnDemand API to send the value when a user orders a service on the SDP portals. ● Invokes the deliver_sm API to send the value when a user orders a service by sending an SMS message. For details about the **link_id** type, see Table 3-17.
presentID	TLV	-	Conditional	Service gift ID. The ID is automatically generated by the SDP when a user sends a service to another user as a gift on the SDP. This parameter is mandatory in an SP's request for sending an SMS message to a

Parameter	Type	Length	Level of Requirement	Description
				gift recipient, and can be left empty in an Enterprise's or a Developer's request. The SDP invokes the assignPresentToUser API to send the value to SPs. For details about the **presentID** type, see Table 3-18.

Table 3-9 describes the parameter structure of the **source_port** type.

Table 3-9 Parameter structure of the source_port type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020A**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	All values allowed. [Example] 0x50

Table 3-10 describes the parameter structure of the **destination_port** type.

Table 3-10 Parameter structure of the destination_port type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020B**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	All values allowed. [Example] 0x50

Table 3-11 describes the parameter structure of the **sar_msg_ref_num** type.

Table 3-11 Parameter structure of the sar_msg_ref_num type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020C**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	2	Mandatory	Reference number. [Example] 0x0001

Table 3-12 describes the parameter structure of the **sar_total_segments** type.

Table 3-12 Parameter structure of the sar_total_segments type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020E**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	This parameter shall contain a value in the range 1 to 255 indicating the total number of fragments within the concatenated short message. The value shall start at 1 and remain constant for every short message which makes up the concatenated short message. [Example] 0x05

Table 3-13 describes the parameter structure of the **sar_segment_seqnum** type.

Table 3-13 Parameter structure of the sar_segment_seqnum type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020F**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets.

Parameter	Type	Size octets	Level of Requirement	Description
				[Example] 0x0050
Value	Integer	1	Mandatory	This octet shall contain a value in the range 1 to 255 indicating the sequence number of a particular message within the concatenated short message. The value shall start at 1 and increment by one for every message sent within the concatenated short message. [Example] 0x01

Table 3-14 describes the parameter structure of the **payload_type** type.

Table 3-14 Parameter structure of the payload_type type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x0019**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Integer	1	Mandatory	• 0x00: Default.In the case of a WAP application, the default higher layer message type is a WDP message. See *Wireless Datagram Protocol Specification* from http://www.wapforum.org for details. • 0x01: WCMP message. Wireless Control Message Protocol formatted data. See *Wireless Datagram Protocol Specification* from http://www.wapforum.org for details. • 0x02–0xFF: Reserved

Parameter	Type	Size octets	Level of Requirement	Description
				[Example] 0x01

Table 3-15 describes the parameter structure of the **message_payload** type.

Table 3-15 Parameter structure of the message_payload type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x0424**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Integer	1	Mandatory	Short message user data. The maximum size is SMSC and network implementation specific. Contact carriers to obtain the value. [Example] 0x01

Table 3-16 describes the parameter structure of the **serviceId** type.

Table 3-16 Parameter structure of the serviceId type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x4004**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Octet String	Var max 21	Mandatory	Service ID. [Example] 0x1FD512A07241

Table 3-17 describes the parameter structure of the **link_id** type.

Table 3-17 Parameter structure of the link_id type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x400D**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Octet String	Var max 21	Mandatory	Service order ID. [Example] 0xB3A73CE2F77

Table 3-18 describes the parameter structure of the **presentID** type.

Table 3-18 Parameter structure of the presentID type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x400E**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Octet String	Var max 15	Mandatory	Service gift ID. [Example] 0x1452C240CF79

3.3.3 Response

The SDP functions as the server, processes submit_sm request messages received from the App, and sends submit_sm response messages to the App.

This topic provides a success response example and describes parameters in the response. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Message Header Parameters

Table 3-19 describes parameters in a response message header.

Table 3-19 Parameters in a submit_sm request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the submit_sm_resp PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to submit_sm_resp. Set it to **0x80000004**. [Example] 0x80000004
command_status	Integer	4	Mandatory	Indicates status (success or error code) of original submit_sm request. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set it to the sequence number of original submit_sm request. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 3-20 describes parameters in a response message body.

Table 3-20 Parameters in a response message body

Parameter	Type	Length	Level of Requirement	Description
message_id	C-Octet String	Var. max 65	Mandatory	Unique ID of the SMS message. The SDP will contain the parameter in the request to send the status report. This parameter specifies the mapping between a status report and an MT SMS message. [Example] 0xDE0C19E529AA592

3.3.4 Error Codes

Table 3-21 describes submit_sm error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 3-21 submit_sm error codes

Error Code	Description
0x0000000A	Invalid Source Address.
0x0000000B	Invalid Dest Addr.
0x00000501	The message has been licence controled.
0x00000538	The sp %1 has not orderd the api %2 in current date.
0x00000539	The sp %1 has not orderd the SCF %2.
0x0000053C	Service ID %1 is not existed!
0x0000053D	The service status is configuring.
0x0000053E	The service status is suspended.
0x0000053F	The service status is pre-deregistered.
0x00000540	The service status is deregistered.
0x00000541	Service %1 is in blacklist!
0x00000542	The API %1 is not existed.
0x00000543	The API status is disabled.
0x00000544	The sp %1 has not ordered the service %2!
0x00000545	The service %1 has not orderd the api %2.
0x00000546	The service %1 has not orderd the SCF %2.
0x00000547	SP level gross control not pass.
0x00000558	SP level request rate control not pass.
0x00000559	System level request rate control not pass.
0x0000055B	Message Length is too long.
0x0000055C	Dest Address size is too large.
0x0000055F	request rate control not pass, sla id is %1.
0x00000560	msgType %1 is invalid.

3.4 submit_multi

3.4.1 Function

The App (functioning as the client) invokes the submit_sm API to send bulk SMS messages to the SDP (functioning as the server).

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

3.4.2 Request

The App functions as the client and sends submit_multi request messages to the SDP.

Message Header Parameters

Table 3-22 describes parameters in a submit_multi request message header.

Table 3-22 Parameters in a submit_multi request message header

Parameter	Type	Size octets	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the submit_multi PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to submit_multi request. Set it to **0x00000021**. [Example] 0x00000021
command_status	Integer	4	Mandatory	Not used in submit_multi PDU. Set it to **NULL**. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set to a unique sequence number. The associated submit_multi_resp PDU will echo the same sequence number. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 3-23 describes parameters in a submit_multi request message body.

Table 3-23 Parameters in a submit_multi request message body

Parameter	Type	Size octets	Level of Requirement	Description
service_type	C-Octet String	Var. max 13	Mandatory	The service_type parameter can be used to indicate the SMS Application service associated with the message. Specifying the service_type allows the App to avail of enhanced messaging

Parameter	Type	Size octets	Level of Requirement	Description
				services such as **“replace by service”** type to control the teleservice used on the air interface. The options are as follows: • NULL: Default • CMT: Cellular Messaging • CPT: Cellular Paging • VMN: Voice Mail Notification • VMA: Voice Mail Alerting • WAP: Wireless Application Protocol • USSD: Unstructured Supplementary Services Data Set it to NULL for default SMSC settings. [Example] 0x00
source_addr_ton	Integer	1	Mandatory	Type of Number for source address. The options are as follows: • 00000000: Unknown • 00000001: International • 00000010: National • 00000011: Network Specific • 00000100: Subscriber Number • 00000101: Alphanumeric • 00000110: Abbreviated • other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
source_addr_npi	Integer	1	Mandatory	Numbering Plan Indicator for source address. The options are as follows: • 00000000: Unknown • 00000001: ISDN (E163/E164) • 00000011: Data (X.121) • 00000100: Telex (F.69) • 00000110: Land Mobile (E.212) • 00001000: National • 00001001: Private

Parameter	Type	Size octets	Level of Requirement	Description
				- 00001010: ERMES - 00001110: Internet (IP) - 00010010: WAP Client Id (to be defined by WAP Forum) - Other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
source_addr	C-Octet String	Var. max 41	Mandatory	Address of the App which originated this message. - In an SP's request, the value is a service access code obtained from carriers before service release. - In an Enterprise's or a Developer's request, the value is an access code allocated by carriers during capability product purchase. [Example] 0x4E663
number_of_dests	Integer	1	Mandatory	Number of destination addresses. Indicates the number of **dest_address** structures that are to follow. A maximum of 254 destination addresses are allowed. Set it to 1 when submitting to one Distribution List. [Example] 0x11
dest_address	dest_address	Var. n[2-24] See Ref.	Mandatory	Contains one or more Distribution List names. The dest_address parameter is of the dest_address type and contains multiple sub-parameters. For details about the dest_address type, see Table 3-24.
esm_class	Integer	1	Mandatory	The esm_class parameter is used to indicate special message attributes associated with the short message. Indicates Message Mode and Message Type. The esm_class parameter is encoded as follows: - bits 1-0: Messaging Mode ◆ xxxxxx00: Default SMSC Mode (e.g. Store and Forward) ◆ xxxxxx01: Datagram mode

Parameter	Type	Size octets	Level of Requirement	Description
				◆ xxxxxx10: Forward (i.e. Transaction) mode ◆ xxxxxx11: Store and Forward mode(use to select Store and Forward mode if Default SMSC Mode is non Store and Forward) • bits 5-2: Message Type ◆ xx0000xx: Default message Type (i.e. normal message) ◆ xx0010xx: Short Message contains App Delivery Acknowledgement ◆ xx0100xx: Short Message contains App Manual/User Acknowledgement • bits 7-6: GSM Network Specific Features ◆ 00xxxxxx: No specific features selected ◆ 01xxxxxx: UDHI Indicator (only relevant for MT short messages) ◆ 10xxxxxx: Set Reply Path (only relevant for GSM network) ◆ 11xxxxxx: Set UDHI and Reply Path (only relevant for GSM network) [Example] 01000000
protocol_id	Integer	1	Mandatory	Protocol Identifier. Network specific field. • GSM: Set according to GSM 03.40. **NOTE** For details, see *Technical Realisation of the Short Message Service Point to Point* at **http://www.etsi.fr**. • ANSI-136 (TDMA): For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For ANSI-136 mobile originated messages, the SMSC should set this value to NULL. • IS-95 (CDMA): For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For IS-95 mobile originated messages, the SMSC should set this value to NULL. [Example] 0x00
priority_flag	Integer	1	Mandatory	Designates the priority level of the message. The options are as follows: • 0x00: Level 0 (lowest) priority

Parameter	Type	Size octets	Level of Requirement	Description
				- 0x01: Level 1 priority - 0x02: Level 2 priority - 0x03: Level 3 (highest) priority - >0x03: Reserved The value options have different meanings on the GSM, ANSI-136, and IS-95 mobile networks. - GSM: For GSM mobile terminated messages with priority greater than Level 0 are treated as priority when making a delivery attempt (i.e. a delivery attempt is made even when MWD is set in the HLR). ◆ 0x00: non-priority ◆ 0x01, 0x02, 0x03: priority - ANSI-136: ◆ 0x00: Bulk ◆ 0x01: Normal ◆ 0x02: Urgent ◆ 0x03: Very Urgent - IS-95: ◆ 0x00: Normal ◆ 0x01: Interactive ◆ 0x02: Urgent ◆ 0x03: Emergency [Example] 0x00
schedule_delivery_time	C-Octet String	1 or 17	Mandatory	This parameter specifies the scheduled time at which the message delivery should be first attempted.It can be specified in either absolute time format or relative time format. If it is the relative time, the SMSC calculates the time based on the system time. Set to NULL for immediate message delivery. [Format] YYMMDDhhmmsstnnp In the format, - t: tenths of second (0-9) - nn: Time difference in quarter hours between local time (as expressed in the first 13 octets) and UTC (Universal Time Constant) time (00-48). - p: UTC wizard ID. ◆ +: Local time is in quarter hours advanced

Parameter	Type	Size octets	Level of Requirement	Description
				in relation to UTC time. ◆ -: Local time is in quarter hours retarded in relation to UTC time. ◆ R: Local time is relative to the current SMSC time. If the time is the relative time, set **t** and **nn** to **0**. [Example] 0x4CC773455000R
validity_period	C-Octet String	1 or 17	Mandatory	The validity period of this message. The parameter indicates the SMSC expiration time, after which the message should be discarded if not delivered to the destination. It can be defined in absolute time format or relative time format. If it is the relative time, the SMSC calculates the time based on the system time. Set it to NULL to request the SMSC default validity period. [Format] YYMMDDhhmmsstnnp In the format, • t: tenths of second (0-9) • nn: Time difference in quarter hours between local time (as expressed in the first 13 octets) and UTC (Universal Time Constant) time (00-48). • p: UTC wizard ID. ◆ +: Local time is in quarter hours advanced in relation to UTC time. ◆ -: Local time is in quarter hours retarded in relation to UTC time. ◆ R: Local time is relative to the current SMSC time. If the time is the relative time, set **t** and **nn** to **0**. [Example] 0x1E754058D8120+
registered_delivery	Integer	1	Mandatory	Indicator to signify if an SMSC delivery receipt or an App acknowledgement is required. The options are as follows: • xxxxxx00: No SMSC Delivery Receipt requested (default). • xxxxxx01: SMSC Delivery Receipt requested where final delivery outcome is delivery success or failure. • xxxxxx10: SMSC Delivery Receipt requested where the final delivery outcome is delivery

Parameter	Type	Size octets	Level of Requirement	Description
				failure. [Example] 0x01
replace_if_present_flag	Integer	1	Mandatory	Flag indicating if submitted message should replace an existing message. The SMSC will replace an existing message provided that the source address, destination address and service_type match the same fields in the new message. The options are as follows: - 0x0: Don't replace (default) - 0x01: Replace - 0x02–0xFF: reserved [Example] 0x00
data_coding	Integer	1	Mandatory	Defines the encoding scheme of the short message user data. The options are as follows: - 00000000L: SMSC Default Alphabet. - 00000001: IA5(CCITT T.50)/ASCII(ANSI X3.4). - 00000010: Octet unspecified (8-bit binary). - 00000011: Latin 1(ISO-8859-1). - 00000100: Octet unspecified (8-bit binary). - 00000101: JIS(X 0208-1990). - 00000110: Cyrllic(ISO-8859-5). - 00000111: Latin/Hebrew(ISO-8859-8). - 00001000: UCS2(ISO/IEC-10646). - 00001001: Pictogram Encoding. - 00001010: ISO-2022-JP(Music Codes). - 00001011: Reserved. - 00001100: Reserved. - 00001101: Extended Kanji JIS(X 0212-1990). - 00001110: KS C 5601. - 00001111–10111111: Reserved. - 1100xxxx: GSM MWI control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. - 1101xxxx: GSM MWI control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details.

Parameter	Type	Size octets	Level of Requirement	Description
				- 11101111: Reserved. - 1111xxxx: GSM message class control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. [Example] 0x00
sm_default_msg_id	Integer	1	Mandatory	Indicates the short message to send from a list of predefined ('canned') short messages stored on the SMSC. If not using an SMSC canned message, set to NULL. The options are as follows: - 0x0: Reserved - 0x01–0xFE: Allowed values - 0xFF: Reserved [Example] 0x00
sm_length	Integer	1	Mandatory	Size octets in octets of the **short_message** field. It should be set to **0** if the message_payload parameter is being used to send user data larger than 254 octets. - 0x00: No user data in short message field. - 0x01–0xFE: Allowed - 0xFF: Not allowed [Example] 0x11
short_message	Octet String	Var. 0-254	Mandatory	Up to 254 octets of short message user data. The exact physical limit for short_message size may vary according to the underlying network. Apps which need to send messages longer than 254 octets should use the **message_payload** parameter. In this case the **sm_length** field should be set to zero. **NOTE** The short message data should be inserted in either the **short_message** or **message_payload** fields. Both fields must not be used simultaneously. [Example] sms test
source_port	TLV	-	Optional	Indicates the application port number associated with the source address of the message. This

Parameter	Type	Size octets	Level of Requirement	Description
				parameter should be present for WAP applications. For details about the **source_port** type, see Table 3-26.
destination_port	TLV	-	Optional	Indicates the application port number associated with the destination address of the message. This parameter should be present for WAP applications. For details about the **destination_port** type, see Table 3-27.
sar_msg_ref_num	TLV	-	Optional	The reference number for a particular concatenated short message. This parameter shall contain a originator generated reference number so that a segmented short message may be reassembled into a single original message. This allows the parallel transmission of several segmented messages.This reference number shall remain constant for every segment which makes up a particular concatenated short message. When present, the PDU must also contain the **sar_total_segments** and **sar_segment_seqnum** parameters. Otherwise this parameter shall be ignored. For details about the **sar_msg_ref_num** type, see Table 3-28.
sar_total_segments	TLV	-	Optional	Indicates the total number of short messages within the concatenated short message. When present, the PDU must also contain the **sar_msg_ref_num** and **sar_segment_seqnum** parameters. Otherwise this parameter shall be ignored. For details about the **sar_total_segments** type, see Table 3-29.
sar_segment_seqnum	TLV	-	Optional	Indicates the sequence number of a particular short message fragment within the concatenated short message. When present, the PDU must also contain the **sar_total_segments** and **sar_msg_ref_num** parameters. Otherwise this parameter shall be ignored. For details about the **sar_segment_seqnum** type, see Table 3-30.

Parameter	Type	Size octets	Level of Requirement	Description
payload_type	TLV	-	Optional	Defines the type of payload (e.g. WDP, WCMP, etc.). For details about the **payload_type** type, see Table 3-31.
message_payload	TLV	-	Optional	Contains the extended short message user data. Up to 64K octets can be transmitted. 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. For details about the **message_payload** type, see Table 3-32.
serviceId	TLV	-	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in an SP's request, and can be left empty in a Developer's or an Enterprise's request. For details about the **serviceId** type, see Table 3-33.
link_id	TLV	-	Conditional	Service order ID. The ID is automatically generated by the SDP when a user orders a service in the SDP. This parameter is mandatory during on-demand service delivery by SMS message. The SDP sends the value to SPs as follows in different scenarios: ● Invokes the ServiceOnDemand API to send the value when a user orders a service on the SDP portals. ● Invokes the deliver_sm API to send the value when a user orders a service by sending an SMS message. For details about the **link_id** type, see Table 3-34.
presentID	TLV	-	Conditional	Service gift ID. The ID is automatically generated by the SDP when a user sends a service to another user as a

Parameter	Type	Size octets	Level of Requirement	Description
				gift on the SDP. This parameter is mandatory in an SP's request for sending an SMS message to a gift recipient, and can be left empty in an Enterprise's or a Developer's request. The SDP invokes the assignPresentToUser API to send the value to SPs. For details about the **presentID** type, see Table 3-35.

Table 3-24 describes the parameter structure of the **dest_address** type.

Table 3-24 Parameter structure of the dest_address type

Parameter	Type	Size octets	Level of Requirement	Description
dest_flag	Integer	1	Mandatory	Flag which will identify whether destination address is a Distribution List name or SME address. • 0x01: SME Address • 0x02: Distribution List Name Set it to **1**. [Example] 0x01
Address	SME_dest_address	-	Mandatory	The Address parameter is of the SME_dest_address type and contains multiple sub-parameters. For details about the SME_dest_address type, see Table 3-25.

Table 3-25 describes the parameter structure of the **SME_dest_address** type.

Table 3-25 Parameter structure of the SME_dest_address type

Parameter	Type	Size octets	Level of Requirement	Description
dest_addr_ton	Integer	1	Mandatory	Type of Number for destination. The options are as follows: • 00000000: Unknown

Parameter	Type	Size octets	Level of Requirement	Description
				• 00000001: International • 00000010: National • 00000011: Network Specific • 00000100: Subscriber Number • 00000101: Alphanumeric • 00000110: Abbreviated • other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
dest_addr_npi	Integer	1	Mandatory	Numbering Plan Indicator for destination. The options are as follows: • 00000000: Unknown • 00000001: ISDN (E163/E164) • 00000011: Data (X.121) • 00000100: Telex (F.69) • 00000110: Land Mobile (E.212) • 00001000: National • 00001001: Private • 00001010: ERMES • 00001110: Internet (IP) • 00010010: WAP Client Id (to be defined by WAP Forum) • Other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
destination_addr	C-Octet String	Var. max 21	Mandatory	Mobile number of the message recipient. The format of the destination mobile number is determined by the **dest_addr_ton** and **dest_addr_npi** parameters. [Example] 0x7D535D16C4E

Table 3-26 describes the parameter structure of the **source_port** type.

Table 3-26 Parameter structure of the source_port type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020A**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	All values allowed. [Example] 0x50

Table 3-27 describes the parameter structure of the **destination_port** type.

Table 3-27 Parameter structure of the destination_port type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020B**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	All values allowed. [Example] 0x50

Table 3-28 describes the parameter structure of the **sar_msg_ref_num** type.

Table 3-28 Parameter structure of the sar_msg_ref_num type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020C**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	2	Mandatory	Reference number. [Example] 0x0001

Table 3-29 describes the parameter structure of the **sar_total_segments** type.

Table 3-29 Parameter structure of the sar_total_segments type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020E**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	This parameter shall contain a value in the range 1 to 255 indicating the total number of fragments within the concatenated short message. The value shall start at 1 and remain constant for every short message which makes up the concatenated short message. [Example] 0x05

Table 3-30 describes the parameter structure of the **sar_segment_seqnum** type.

Table 3-30 Parameter structure of the sar_segment_seqnum type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020F**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	This octet shall contain a value in the range 1 to 255 indicating the sequence number of a particular message within the concatenated short message. The value shall start at 1 and increment by one for every message sent within the concatenated short message. [Example] 0x01

Table 3-31 describes the parameter structure of the **payload_type** type.

Table 3-31 Parameter structure of the payload_type type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x0019**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Integer	1	Mandatory	• 0x00: Default.In the case of a WAP application, the default higher layer message type is a WDP message. See *Wireless Datagram Protocol Specification* from http://www.wapforum.org for details. • 0x01: WCMP message. Wireless Control Message Protocol formatted data. See *Wireless Datagram Protocol Specification* from http://www.wapforum.org for details. • 0x02–0xFF: Reserved [Example] 0x01

Table 3-32 describes the parameter structure of the **message_payload** type.

Table 3-32 Parameter structure of the message_payload type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x0424**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Integer	1	Mandatory	Short message user data. The maximum size is SMSC and network implementation specific. Contact carriers to obtain

Parameter	Type	Size octets	Level of Requirement	Description
				the value. [Example] 0x01

Table 3-33 describes the parameter structure of the **serviceId** type.

Table 3-33 Parameter structure of the serviceId type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x4004**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Octet String	Var max 21	Mandatory	Service ID. [Example] 0x1FD512A07241

Table 3-34 describes the parameter structure of the **link_id** type.

Table 3-34 Parameter structure of the link_id type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x400D**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Octet String	Var max 21	Mandatory	Service order ID. [Example] 0xB3A73CE2F77

Table 3-35 describes the parameter structure of the **presentID** type.

Table 3-35 Parameter structure of the presentID type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x400E**.

Parameter	Type	Size octets	Level of Requirement	Description
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Octet String	Var max 15	Mandatory	Service gift ID. [Example] 0x1452C240CF79

3.4.3 Response

The SDP functions as the server, processes submit_multi request messages received from the App, and sends submit_multi response messages to the App.

This topic provides a success response example and describes parameters in the response. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Message Header Parameters

Table 3-36 describes parameters in a response message header.

Table 3-36 Parameters in a submit_multi request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the submit_multi_resp PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to submit_multi_resp. Set it to **0x80000021**. [Example] 0x80000021
command_status	Integer	4	Mandatory	Indicates status (success or error code) of original submit_multi request. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set it to the sequence number of original submit_multi request. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 3-37 describes parameters in a response message body.

Table 3-37 Parameters in a response message body

Parameter	Type	Length	Level of Requirement	Description
message_id	C-Octet String	Var. max 65	Mandatory	Unique ID of the SMS message. The SDP will contain the parameter in the request to send the status report. This parameter specifies the mapping between a status report and an MT SMS message. [Example] 0xDE0C19E529AA592

3.4.4 Error Codes

Table 3-38 describes submit_multi error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 3-38 submit_multi error codes

Error Code	Description
0x0000000A	Invalid Source Address.
0x0000000B	Invalid Dest Addr.
0x00000048	Invalid Source address TON.
0x00000049	Invalid Source address NPI.
0x00000050	Invalid Destination address TON.
0x00000051	Invalid Destination address NPI.
0x00000501	The message has been licence controled.
0x00000538	The sp %1 has not orderd the api %2 in current date.
0x00000539	The sp %1 has not orderd the SCF %2.
0x0000053C	Service ID %1 is not existed!
0x0000053D	The service status is configuring.
0x0000053E	The service status is suspended.
0x0000053F	The service status is pre-deregistered.
0x00000540	The service status is deregistered.
0x00000541	Service %1 is in blacklist!

Error Code	Description
0x00000542	The API %1 is not existed.
0x00000543	The API status is disabled.
0x00000544	The sp %1 has not ordered the service %2!
0x00000545	The service %1 has not orderd the api %2.
0x00000546	The service %1 has not orderd the SCF %2.
0x00000547	SP level gross control not pass.
0x00000558	SP level request rate control not pass.
0x00000559	System level request rate control not pass.
0x0000055B	Message Length is too long.
0x0000055C	Dest Address size is too large.
0x0000055F	request rate control not pass, sla id is %1.
0x00000560	msgType %1 is invalid.

3.5 deliver_sm(report)

3.5.1 Function

The SDP (functioning as the client) invokes the API to send status reports to the App (functioning as the server).

When the App uses the submit_sm or submit_multi API to send SMS messages to users and the SDP successfully sends the request to the SMSC, the SMSC sends a status report to the SDP. The SDP uses the deliver_sm(report) API to report the status report to the App. If the status report fails to be reported, the SDP does not resend it.

Partners must code the App based on the API field requirements so that the App can correctly parse and respond to requests received from the SDP. The App must send a response to the SDP within 30 seconds.

3.5.2 Request

The SDP functions as the client and sends deliver_sm(report) request messages to the App to delivery an MO SMS message delivery status report.

Message Header Parameters

Table 3-39 describes parameters in a deliver_sm(report) request message header.

Table 3-39 Parameters in a deliver_sm(report) request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the deliver_sm PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to deliver_sm request. Set it to **0x00000005**. [Example] 0x00000005
command_status	Integer	4	Mandatory	Not used in deliver_sm PDU. Set it to **NULL**. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	Sequence number. Set to a unique sequence number. The associated deliver_sm_resp PDU will echo the same sequence number. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 3-40 describes parameters in a deliver_sm request message body.

Table 3-40 Parameters in a deliver_sm request message body

Parameter	Type	Length	Level of Requirement	Description
service_type	C-Octet String	Var. max 13	Mandatory	The service_type parameter can be used to indicate the SMS Application service associated with the message. Specifying the service_type allows the App to avail of enhanced messaging services such as **replace by service** type to control the teleservice used on the air interface. The options are as follows: • NULL: Default • CMT: Cellular Messaging • CPT: Cellular Paging • VMN: Voice Mail Notification • VMA: Voice Mail Alerting • WAP: Wireless Application Protocol

Parameter	Type	Length	Level of Requirement	Description
				• USSD: Unstructured Supplementary Services Data [Example] 0x00
source_addr_ton	Integer	1	Mandatory	Type of Number for source address. The options are as follows: • 00000000: Unknown • 00000001: International • 00000010: National • 00000011: Network Specific • 00000100: Subscriber Number • 00000101: Alphanumeric • 00000110: Abbreviated • other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
source_addr_npi	Integer	1	Mandatory	Numbering Plan Indicator for source address. The options are as follows: • 00000000: Unknown • 00000001: ISDN (E163/E164) • 00000011: Data (X.121) • 00000100: Telex (F.69) • 00000110: Land Mobile (E.212) • 00001000: National • 00001001: Private • 00001010: ERMES • 00001110: Internet (IP) • 00010010: WAP Client Id (to be defined by WAP Forum) • Other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. If not known, set it to **NULL**. [Example] 0x00
source	C-Octet	Var.	Mandator	Mobile number of the message sender.

Parameter	Type	Length	Level of Requirement	Description
_addr	String	max 41	y	The format of the source mobile number is determined by the **source_addr_ton** and **source_addr_npi** parameters. [Example] 0x7D535D16C4E
dest_addr_ton	Integer	1	Mandatory	Type of Number for destination. The options are as follows: • 00000000: Unknown • 00000001: International • 00000010: National • 00000011: Network Specific • 00000100: Subscriber Number • 00000101: Alphanumeric • 00000110: Abbreviated • other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. [Example] 0x00
dest_addr_npi	Integer	1	Mandatory	Numbering Plan Indicator for destination. The options are as follows: • 00000000: Unknown • 00000001: ISDN (E163/E164) • 00000011: Data (X.121) • 00000100: Telex (F.69) • 00000110: Land Mobile (E.212) • 00001000: National • 00001001: Private • 00001010: ERMES • 00001110: Internet (IP) • 00010010: WAP Client Id (to be defined by WAP Forum) • Other: Reserved **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. [Example] 0x00
destination_addr	C-Octet String	Var. max 21	Mandatory	Address of the App which received this message. • In an SP's request, the value is a service

Parameter	Type	Length	Level of Requirement	Description
				access code obtained from carriers before service release. - In an Enterprise's or a Developer's request, the value is an access code allocated by carriers during capability product purchase. The message format is determined by the **destination_addr_ton** and **destination_addr_npi** parameters. [Example] 0x4E663
esm_class	Integer	1	Mandatory	Indicates Message Type and enhanced network services. Bits 5-2 indicates the Message Type. When the bits 2 to 5 in the **esm_class** field are **0010**, it indicates a status report. xx0010xx: Short Message contains App delivery acknowledgement [Example] 0x001000
protocol_id	Integer	1	Mandatory	Protocol Identifier. Network specific field. - GSM: Set according to GSM 03.40. **NOTE** For details, see *Technical Realisation of the Short Message Service Point to Point* at http://www.etsi.fr. - ANSI-136 (TDMA): For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For ANSI-136 mobile originated messages, the SMSC should set this value to NULL. - IS-95 (CDMA): For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For IS-95 mobile originated messages, the SMSC should set this value to NULL. [Example] 0x00
priority_flag	Integer	1	Mandatory	Designates the priority level of the message. The options are as follows: - 0x00: Level 0 (lowest) priority - 0x01: Level 1 priority - 0x02: Level 2 priority - 0x03: Level 3 (highest) priority - >0x03: Reserved The value options have different meanings on the GSM, ANSI-136, and IS-95 mobile

Parameter	Type	Length	Level of Requirement	Description
				networks. • GSM: For GSM mobile terminated messages with priority greater than Level 0 are treated as priority when making a delivery attempt (for example, a delivery attempt is made even when MWD is set in the HLR). ◆ 0x00: non-priority ◆ 0x01, 0x02, 0x03: priority • ANSI-136: ◆ 0x00: Bulk ◆ 0x01: Normal ◆ 0x02: Urgent ◆ 0x03: Very Urgent • IS-95: ◆ 0x00: Normal ◆ 0x01: Interactive ◆ 0x02: Urgent ◆ 0x03: Emergency [Example] 0x00
schedule_delivery_time	C-Octet String	1 or 17	Mandatory	This field is unused for **deliver_sm**. It must be set to NULL. [Example] 0x00
validity_period	C-Octet String	1 or 17	Mandatory	This field is unused for **deliver_sm**. It must be set to NULL. [Example] 0x00
registered_delivery	Integer	1	Mandatory	Indicates if an App acknowledgement is required. The options are as follows: • xxxx00xx: No recipient SME acknowledgment requested (default). • xxxx01xx: SME Delivery Acknowledgement requested. • xxxx10xx: SME Manual/User Acknowledgment requested. • xxxx11xx: Both Delivery and Manual/User Acknowledgment requested [Example] 0x0A
replac	Integer	1	Mandator	Reserved.

Parameter	Type	Length	Level of Requirement	Description
e_if_present_flag			y	Must be set it to NULL. [Example] 0x00
data_coding	Integer	1	Mandatory	Defines the encoding scheme of the short message user data. The options are as follows: • 00000000: SMSC Default Alphabet. • 00000001: IA5(CCITT T.50)/ASCII(ANSI X3.4). • 00000010: Octet unspecified (8-bit binary). • 00000011: Latin 1(ISO-8859-1). • 00000100: Octet unspecified (8-bit binary). • 00000101: JIS(X 0208-1990). • 00000110: Cyrllic(ISO-8859-5). • 00000111: Latin/Hebrew(ISO-8859-8). • 00001000: UCS2(ISO/IEC-10646). • 00001001: Pictogram Encoding. • 00001010: ISO-2022-JP(Music Codes). • 00001011: Reserved. • 00001100: Reserved. • 00001101: Extended Kanji JIS(X 0212-1990). • 00001110: KS C 5601. • 00001111–10111111: Reserved. • 1100xxxx: GSM MWI control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. • 1101xxxx: GSM MWI control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. • 11101111: Reserved. • 1111xxxx: GSM message class control. See *Digital Cellular telecommunications system (Phase 2+)* from http://www.etsi.fr for details. **NOTE** Parameter values in the value range are in binary format. The actual parameter value must be in hexadecimal format. [Example] 0x00

Parameter	Type	Length	Level of Requirement	Description
sm_default_msg_id	Integer	1	Mandatory	This field is unused for **deliver_sm**. It must be set to NULL. [Example] 0x00
sm_length	Integer	1	Mandatory	Length in octets of the **short_message** field. It should be set to **0** if the message_payload parameter is being used to send user data larger than 254 octets. • 0x00: No user data in short message field. • 0x01–0xFE: Allowed • 0xFF: Not allowed [Example] 0x11
short_message	Octet String	Var. 0-254	Mandatory	Status report content. When the bits 2 to 5 in the **esm_class** field are **0010**, the **short_message** indicates a status report content. The typical format of a status report is **id:IIIIIIIIII sub:SSS dlvrd: DDD submit date: YYMMDDhhmm done date: YYMMDDhhmm stat:DDDDDDD err: E Text:...**. For details, see Table 3-49.
source_port	TLV	N/A	Optional	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications. For details about the **source_port** type, see Table 3-41.
destination_port	TLV	N/A	Optional	Indicates the application port number associated with the destination address of the message. This parameter should be present for WAP applications. For details about the **destination_port** type, see Table 3-42.
sar_msg_ref_num	TLV	N/A	Optional	The reference number for a particular concatenated short message. This parameter shall contain a originator generated reference number so that a segmented short message may be reassembled into a single original message. This allows the parallel transmission of several segmented messages. This reference number shall remain constant for every segment which makes up a particular concatenated short

Parameter	Type	Length	Level of Requirement	Description
				message. When present, the PDU must also contain the **sar_total_segments** and **sar_segment_seqnum** parameters. Otherwise this parameter shall be ignored. For details about the **sar_msg_ref_num** type, see Table 3-43.
sar_total_segments	TLV	N/A	Optional	Indicates the total number of short messages within the concatenated short message. When present, the PDU must also contain the **sar_msg_ref_num** and **sar_segment_seqnum** parameters. Otherwise this parameter shall be ignored. For details about the **sar_total_segments** type, see Table 3-44.
sar_segment_seqnum	TLV	N/A	Optional	Indicates the sequence number of a particular short message fragment within the concatenated short message. When present, the PDU must also contain the **sar_total_segments** and **sar_msg_ref_num** parameters. Otherwise this parameter shall be ignored. For details about the **sar_segment_seqnum** type, see Table 3-45.
payload_type	TLV	N/A	Optional	Defines the type of payload (for example, WDP and WCMP). For details about the **payload_type** type, see Table 3-46.
message_payload	TLV	N/A	Optional	Contains the extended short message user data. Up to 64K octets can be transmitted. 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. For details about the **message_payload** type, see Table 3-47.
receipted_message_id	TLV	N/A	Optional	Unique ID of the message sent by the SDP. Unique ID of the message delivered by the App. The value is the same as that of **message_id** in the submit_sm_resp or submit_multi_resp message sent by the SDP. This parameter specifies the mapping between

Parameter	Type	Length	Level of Requirement	Description
				a status report and an MT SMS message. For details about the **receipted_message_id** type, see Table 3-48.

Table 3-41 describes the parameter structure of the **source_port** type.

Table 3-41 Parameter structure of the source_port type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020A**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	All values allowed. [Example] 0x50

Table 3-42 describes the parameter structure of the **destination_port** type.

Table 3-42 Parameter structure of the destination_port type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020B**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	All values allowed. [Example] 0x50

Table 3-43 describes the parameter structure of the **sar_msg_ref_num** type.

Table 3-43 Parameter structure of the sar_msg_ref_num type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020C**.
Size	Integer	2	Mandatory	Size octets of value part in

Parameter	Type	Size octets	Level of Requirement	Description
octets				octets. [Example] 0x0050
Value	Integer	2	Mandatory	Reference number. [Example] 0x0001

Table 3-44 describes the parameter structure of the **sar_total_segments** type.

Table 3-44 Parameter structure of the sar_total_segments type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020E**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	This parameter shall contain a value in the range 1 to 255 indicating the total number of fragments within the concatenated short message. The value shall start at 1 and remain constant for every short message which makes up the concatenated short message. [Example] 0x05

Table 3-45 describes the parameter structure of the **sar_segment_seqnum** type.

Table 3-45 Parameter structure of the sar_segment_seqnum type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x020F**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0050
Value	Integer	1	Mandatory	This octet shall contain a value in the range 1 to 255

Parameter	Type	Size octets	Level of Requirement	Description
				indicating the sequence number of a particular message within the concatenated short message. The value shall start at 1 and increment by one for every message sent within the concatenated short message. [Example] 0x01

Table 3-46 describes the parameter structure of the **payload_type** type.

Table 3-46 Parameter structure of the payload_type type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x0019**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Integer	1	Mandatory	• 0x00: Default.In the case of a WAP application, the default higher layer message type is a WDP message. See *Wireless Datagram Protocol Specification* from http://www.wapforum.org for details. • 0x01: WCMP message. Wireless Control Message Protocol formatted data. See *Wireless Datagram Protocol Specification* from http://www.wapforum.org for details. • 0x02–0xFF: Reserved [Example] 0x01

Table 3-47 describes the parameter structure of the **message_payload** type.

Table 3-47 Parameter structure of the message_payload type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x0424**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	Integer	1	Mandatory	Short message user data. The maximum size is SMSC and network implementation specific. Contact carriers to obtain the value. [Example] 0x01

Table 3-48 describes the parameter structure of the **receipted_message_id** type.

Table 3-48 Parameter structure of the receipted_message_id type

Parameter	Type	Size octets	Level of Requirement	Description
Tag	Integer	2	Mandatory	Set it to **0x001E**.
Size octets	Integer	2	Mandatory	Size octets of value part in octets. [Example] 0x0001
Value	C-Octet String	Var. max 65	Mandatory	Message ID. [Example] 0xB3A73CE2F77

Table 3-49 describes the parameter structure of the short_message type.

Table 3-49 Parameter structure of the short_message type

Parameter	Type	Size octets	Level of Requirement	Description
id	C-Octet String (Decimal)	10	Mandatory	Unique ID of the MT SMS message corresponding to the status report. The value is the same as that of **message_id** in the response sent by the SDP after the App invokes the submit_sm or submit_multi API to send SMS messages.

Parameter	Type	Size octets	Level of Requirement	Description
				[Example] 0x499602D3
sub	C-Octet String (Decimal)	3	Mandatory	Number of short messages originally submitted by App. The value is padded with leading zeros if necessary. [Example] 010
dlvrd	C-Octet String (Decimal)	3	Mandatory	Number of SMS messages that the App successfully sends to the SDP. The value is padded with leading zeros if necessary. [Example] 010
Submit date	C-Octet Fixed Lenth String	10	Mandatory	The time and date at which the short message was submitted. In the case of a message which has been replaced, this is the date that the original message was replaced. [Format] YYMMDDhhmm [Example] 0x4DF991B8
done date	C-Octet Fixed Lenth String	10	Mandatory	The time and date at which the short message reached it's final state. [Format] YYMMDDhhmm [Example] 0x4DF991B9
stat	C-Octet Fixed Lenth String	7	Mandatory	Status of the MT message. Enumerated values of stat: ● DELIVRD: Message is delivered to destination. ● EXPIRED: Message validity period has expired. ● DELETED: Message has been deleted. ● UNDELIV: Message is undeliverable. ● ACCEPTD: Message is in accepted state. ● UNKNOWN: Message is in invalid state. ● REJECTD: Message is in a rejected state. [Example] DELIVRD

Parameter	Type	Size octets	Level of Requirement	Description
err	C-Octet Fixed Lenth String	3	Mandatory	Error code.
text	C-Octet Fixed Lenth String	20	Mandatory	First 20 bytes of the original MT SMS message corresponding to the status report. [Example] sub This is a test!!

3.5.3 Response

The App functions as the server, processes deliver_sm(report) request messages received from the SDP, and sends response messages to the SDP

This topic provides a success response example and describes parameters in the response. If a request fails, the App sends an error response that contains an error code. For details about error responses, see API Error Responses.

Message Header Parameters

Table 3-50 describes parameters in a response message header.

Table 3-50 Parameters in a deliver_sm(report) request message header

Parameter	Type	Length	Level of Requirement	Description
command_length	Integer	4	Mandatory	The overall length of the deliver_sm_resp PDU in octets. [Example] 0x400
command_id	Integer	4	Mandatory	Value corresponding to deliver_sm_resp. Set it to **0x80000005**. [Example] 0x80000005
command_status	Integer	4	Mandatory	Indicates status (success or error code) of original deliver_sm request. [Example] 0x00000000
sequence_number	Integer	4	Mandatory	sequence number. Set it to the sequence number of original deliver_sm request. [Range] 0x00000001-0x7FFFFFFF [Example] 0x00000001

Message Body Parameters

Table 3-51 describes parameters in a response message body.

Table 3-51 Parameters in a response message body

Parameter	Type	Length	Level of Requirement	Description
message_id	C-Octet String	Var. max 65	Mandatory	This field is unused and is set to NULL. [Example] 0x00

3.5.4 Error Codes

The App returns error codes to the SDP when an exception occurs in response to deliver_sm(report) request messages. The error codes are provided by the App based on SMPP specifications.