1.   Introduction

1.1   Purpose

This document specifies a generalized interface between an SMSC and **non-PLMN SMEs**. Typically it specifies the interface used between the SMSC and Paging or VoiceMail systems. The command format defines a Short Message Peer to Peer Protocol (hereafter referred to as SMPP). This protocol may be implemented over a variety of underlying interfaces/communications protocols, namely X.25, or TCP/IP.

Using this interface, an external Short Message Entity such as a Paging or VoiceMail system may bind/ unbind to the SMSC, submit , cancel, replace and query short messages. The SMSC forwards responses and short messages (e.g delivery receipts, pager messages) to the external Short Message Entity.

1.2   Scope

This document is intended for designers and implementers of the interface between an SMSC and SMEs (Short Message Entities).

1.3   References

[1]	SMPP Applications Guide	Version 1.3	Aldiscon Limited
[2]	Technical Realisation of the Short Message Service Point to Point, GSM 03.40	Version: 4.6.0	European Telecommunications Standards Institute. (ETSI)
[3]	SMPP Provisioning Interface Guide	Version 1.1	Aldiscon Limited
[4]	SMPP Provisioning Application Guide	Version 1.1	Aldiscon Limited

1.4   Glossary

ACK	Acknowledgement
AIM	Application Interface Module
API	Application Programming Interface
CDR	Call Detail Record
ESME	External Short Message Entity. Refer to note[1]
MB	Message Bureau - This is typically an operator message bureau.
MSC	Mobile Switching Centre
MS	Mobile Station
NAK	Negative Acknowledgement
SME	Short Message Entity
SMSC	Short Message Service Centre
SMPP	Short Message Peer to Peer Protocol
VC	Virtual Connection. Refer to note [2]
VMA	VoiceMail Alert or Message Waiting Indication (MWI)
VPS	Voice Processing System

Note 1   External Short Message Entity. In the context of this document this refers to such external sources and sinks of short messages as Voice Processing or Message Handling computers. It specifically excludes SMEs which are part of the interface to the PLMN.

Note 2   Virtual Connection. This refers to a virtual circuit in the X.25 implementation.

2. Functional overview

Interworking between the SMSC and ESMEs are categorised as:

- (protocol) messages from ESMEs to the SMSC, and

- (protocol) messages from SMSC to ESMEs.

Figure 2.1 illustrates these categories which are detailed in the following sections.
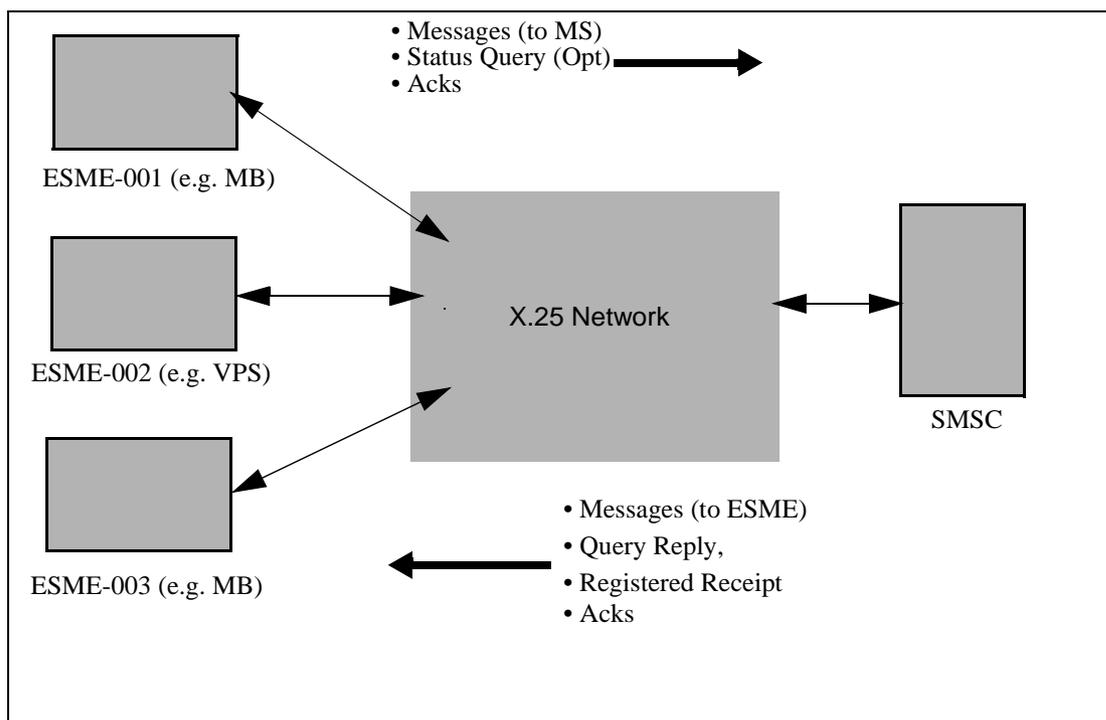

Figure 2-1:   SMSC & ESME Interworking using X.25

2.1 ESMEs to SMSC

Subscribers to a GSM Network may receive short messages from ESMEs. The means whereby these messages are originally generated within or are submitted to the ESME is beyond the scope of this document, but the following are possible examples:

- Calls directly dialled or diverted to a Message Bureau operator and forwarded to the SMSC.

- Messages originated from terminals at a corporate customer's site.

- Voice-Mail Alerts originating for a VPS indicating voice messages at a customer's mailbox.

Messages that are submitted to the SMSC by an ESME are immediately acknowledged. This acknowledgment informs the ESME that the message submitted is a valid message (i.e. fields are set to valid values).

In addition to "Message Submission", an ESME may "Query" the SMSC for the status of previously submitted messages, or cancel delivery of previously submitted messages using the Message ID returned by the SMSC when the particular message was originally submitted.

2.2   SMSC to ESME

The SMSC can deliver short messages to the ESME. A typical example would be the SMSC sending short messages to an MB for onward delivery as pager messages.

In addition the SMSC may use the "deliver short message" mechanism to generate a "Delivery Receipt". (See SMPP Applications guide [1] for details).

2.3   Backward Compatibility.

Where changes have occurred in the Interface Specification between versions, the "interface_version" provided in the "Bind" primitive is used to discriminate between version numbers for backward compatibility.

3.   Interface Specification

The interface between the SMSC and ESME may be based on X.25, or TCP/IP. For details of a particular implementation refer to the SMPP Applications Guide [1].

The interface between the SMSC and the ESMEs regardless of the underlying network type will be a client-server model, in which, the SMSC has the server role and the ESME the client role. In the remainder of this document, "client" is referred to the system that initiates a connection and "server" is referred to the system that services a connection.

Note that this document specifies the interface at the network layer. However, this interface may be implemented over the transport layer. Figure 3.1 provides a perspective on the scope of this document:
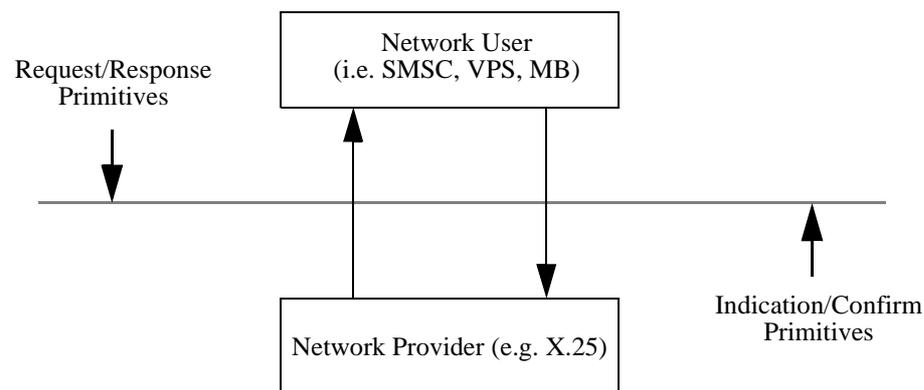

Figure 3-1:   Model of SMSC-ESME Interface

4.   Protocol Messages

All messages sent, either from ESME to SMSC, or SMSC to ESME, will generate immediate responses.

As previously mentioned, a message submitted from an ESME to SMSC can generate up to two responses. These are:

•   an application level "resp", and

•   where the message was submitted to the SMSC with the registered delivery flag set, a status report generated after the submitted short message reaches its final state.

Figure 4.1 depicts a possible sequence of these messages (e.g for an X.25 or TCP/IP based implementation).

ESME   SMSC

submit_sm (1)

submit_sm_resp(1)

deliver_sm(1)

deliver_sm_resp (1)

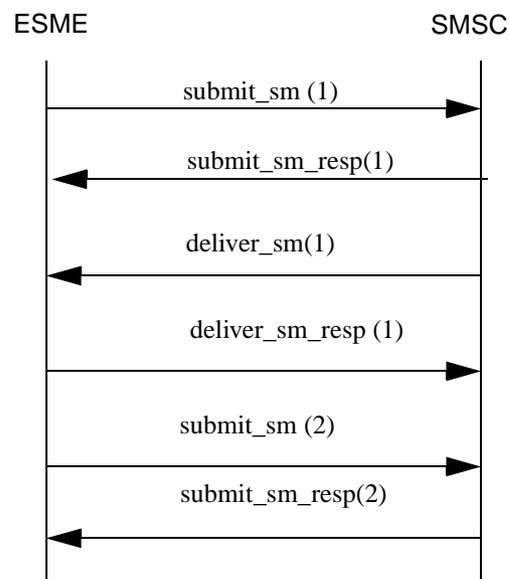submit_sm (2)

submit_sm_resp(2)

Figure 4.1 Sample Message Sequence

For details of ESME/SMSC protocol message sequences refer to the SMPP Applications Guide[1].

5.   Use of Primitives

This section describes an overview of the mechanism for exchange of primitives between the ESME and SMSC. For details for a particular network implementation, such as X.25 or TCP/IP, see the SMPP Applications guide [1].

5.1   Initiation of Communication with SMSC

The ESME establishes communication with the SMSC, by an implementation specific mechanism (see SMPP Applications guide [1]).

Two 'virtual connections' are required. One will be used for messages originating in the ESME system, and the response messages for them. (e.g. submit_sm, query_sm, cancel_sm etc.), while the other will be used for messages originating in the SMSC and their responses (e.g. deliver_sm).

Once a 'virtual connection' has been established, each of the two processes on the ESME should send either a Bind-Transmitter request or a Bind-Receiver request. If a Bind Transmitter request is sent, the process on the SMSC that receives it will receive messages originating in the ESME system. If a Bind Receiver request is sent, the process on the SMSC that receives it will forward messages to the ESME. Responses will invariably be returned on the same 'virtual connection' as the corresponding request messages.
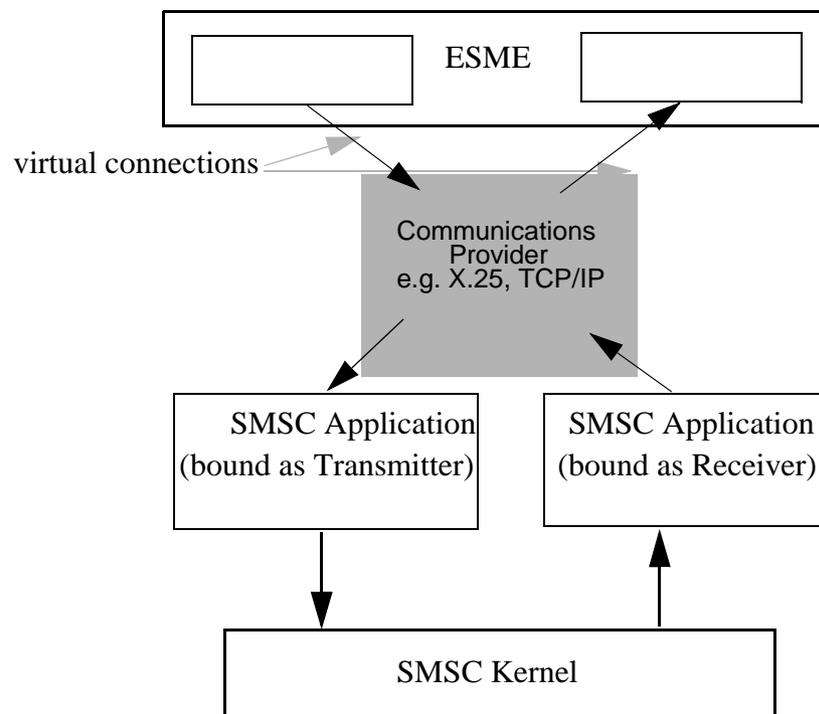
The following diagram illustrates this:

Figure 5-1:   ESME/SMSC Communication

5.2   Steady-State Communication with the SMSC

Once a connection has been established and an authenticated 'bind' request has been acknowledged, further requests/responses can be exchanged. A response will be issued for each request.

5.3   Terminating Communication with the SMSC

If, at any time, either the ESME or the SMSC needs to terminate communications with the other, it should issue an "unbind" request over the appropriate 'virtual connection'. This enables the receiving system to break communications in an orderly fashion. For both 'virtual connections', the unbind request should be acknowledged by the receiving system before the 'virtual connection' is closed.

5.4   Error Handling and Retransmission

On receipt of a message the receiving system will ensure that the message type is valid, and then check, where appropriate, the validity of the fields of the message body. If the message type or the values of the fields are incorrect an error code indicating this will be returned in the response message to the originator. A table of error and status codes can be found in Section 7.1

Should an error be generated by the underlying communication network or the application being used on the host machine it is the responsibility of the sender of the message to retransmit to the destination. The originator should maintain a retry count and when this limit has been reached on a single message attempt the connection should be closed. The ESME should attempt to re-connect. The re-connect method will be the same as the startup protocol.

The Sequence number in the message header should be generated by the ESME. This number should   be incremented monotonically with each new transaction. This field will be preserved by the receiving system and returned in the acknowledgement message. This allows for transaction mapping and the detection of duplicate messages.

5.5   Protocol Message Types

The following message types are supported by the SMPP. The "command id" field of the protocol message is set to specify the particular message.

The detailed formats of these messages are defined in Section [6.].

5.5.1   ESME to SMSC

The following messages are sent from the ESME to the SMSC

Command ID	Description
bind_receiver	This command is issued by the ESME to inform the SMSC that this ESME wishes to act as a Server
bind_transmitter	This command is issued by the ESME to inform the SMSC that this ESME wishes to act as a Client
unbind	This command is issued by the ESME to inform the SMSC that this ESME wishes to terminate its activities.
submit_sm	This command is issued by the ESME to submit a short message to the SMSC for transmission to a specified subscriber.
submit_multi	This command is issued by the ESME to submit a short message to the SMSC for transmission to a specified subscriber or Distribution List or Multiple Recipients.
deliver_sm_resp	This command is issued by the ESME to acknowledge the receipt of a deliver_sm.
query_sm	This command is issued by the ESME to query the status of a previously submitted Short Message.
query_last_msgs	This command is issued by the ESME to query the message ids of a number of messages in the system for a subscribers originating address.
query_msg_details	This command is issued by the ESME to query all aspects of a previously submitted Short Message.
cancel_sm	This command is issued by the ESME to cancel one or more outstanding short messages for a subscriber. The command may specify a particular message or all messages for a particular source and destination.
replace_sm	This command is issued by the ESME to replace an outstanding short message for a subscriber.

Table 5-1:   Message Types from ESME to SMSC.

enquire_link	Enquires whether the ESME-SMSC session is functioning, and thereby provides a link confidence-check.
enquire_link_resp	Response to an "enquire link" previously sent by the SMSC.
generic_nak	Generic response to a command for which the message header is invalid.

Table 5-1:　Message Types from ESME to SMSC.

5.5.2   SMSC to ESME

The following messages are sent from the SMSC to the ESME.

Command ID	Description
bind_transmitter_resp	Response to "bind_transmitter". Messages submitted with this command id will contain a status indicating success or failure of the corresponding "bind_transmitter".
bind_receiver_resp	Response to "bind_receiver". Messages submitted with this command id will include a status indicating success or failure of the corresponding "bind_receiver".
unbind_resp	Response to "unbind". Messages submitted with this command id will include a status indicating success or failure of the corresponding "unbind".
submit_sm_resp	Response indicating that a short message has been accepted successfully or not. Messages submitted with this command id will include the status indicating success or failure of the corresponding "submit_sm".
submit_multi_resp	Response indicating that a short message has been accepted successfully or not. Messages submitted with this command id will include the status indicating success or failure of the corresponding "submit_multi".
deliver_sm	This command is issued by the SMSC to submit a short message to the ESME for delivery. It may also be used to return a delivery receipt for a message which had been submitted with the delivery receipt flag set.
query_sm_resp	Response to "query_sm". Messages submitted with this command id will include the status indicating success or failure of the corresponding "query_sm" in addition to data relating to the queried message.
query_last_msgs_resp	Response to "query_last_msgs". Messages submitted with this command id will include the status indicating success or failure of the corresponding "query_last_msgs" in addition to data relating to the queried messages.
query_msg_details_resp	Response to "query_msg_details". Messages submitted with this command id will include the status indicating success or failure of the corresponding "query_msg_details" in addition to data relating to the queried message.

Table 5-2:   Message Types from SMSC to ESME

cancel_sm_resp	Response to "cancel_sm". Messages submitted with this command id will include the status indicating success or failure of the corresponding "cancel_sm".
replace_sm_resp	Response to "replace_sm". Messages submitted with this command id will include the status indicating success or failure of the corresponding "replace_sm".
enquire_link	Enquires whether the SMSC-ESME session is functioning, and thereby provides a link confidence-check.
enquire_link_resp	Response to "enquire_link". Messages submitted with this command id will include the status indicating success or failure of the corresponding "enquire_link".
generic_nak	Generic response to a command for which the message header is invalid.

Table 5-2:   Message Types from SMSC to ESME

6. Message Layouts.

The general format of all protocol messages exchanged between the ESME and the SMSC will consist of a message header followed by a message body.

6.1 Definitions

In the following descriptions the following definitions will be used:

Integer	• a signed value with the defined number of bytes
	The bytes will always be transmitted MSB first
C-Octet String	• a series of ASCII characters terminated with the NUL character.
C-Octet String (Decimal)	• a series of ASCII characters terminated with the NUL character.
	The octet string should represent a sequence of decimal digits
C-Octet String (Hex)	• a series of ASCII characters terminated with the NUL character.
	The octet string should represent a sequence of hexadecimal digits
Octet String	• Series of octets which may/may not be null terminated.
	The octets themselves can contain nulls.

Where reference is made below to NULL settings of Octet-String fields this implies that the field consists of a single NUL character, i.e. an Octet encoded with value zero.

Where reference is made to NULL settings of Integer fields this implies that the field is unused and can be set to 0.

6.2   Message Header Format

Element	Size bytes	Type	Description
Command Length	4	Integer	This field defines the total length of the packet including the length field.
Command ID	4	Integer	The field indicates the type of request to be invoked by this protocol message, e.g. '*submit_sm*', '*query_sm*' etc. A request command identifier will be allocated to each request primitive. The following range is reserved for these purposes: 0h to FFh. A response command identifier will be allocated to each response primitive. The following range is reserved for these purposes: 080000000h to 08000000FF (In general a response command identifier will be identical to the corresponding request command identifier, but with bit 31 set.) For details of the actual IDs see Section 7.2.
Command Status	4	Integer	This field will indicate the success or failure of a request. This field is only relevant in the response message, so in the request message it should contain NULL. A list of error codes is given in Section 7.1.
Sequence No.	4	Integer	A sequence number allowing requests and responses to be associated. Allocation of this reference number is the responsibility of the originator, who should ensure that the number is monotonically increasing for each submitted request. The associated response packet must preserve this field. The range is 01h to 07FFFFFFFh
Optional Message Body	var.	mixed	A list of parameters corresponding to the Command type. These fields are detailed in section 6.3

Table 6-1:   **Message Header Format**

6.2.1   "GENERIC_NAK" Command

This is a generic response to a command for which the message header is invalid.

6.2.1.1   "GENERIC_NAK" Syntax

Apart from setting the header fields, no other parameters are required in the data body.

6.3   Message Body Formats

6.3.1   "BIND" Operation

There are two variations of the Bind Command namely "bind_transmitter" and "bind_receiver". The Command ID setting specifies whether the Bind is the "bind_transmitter" or "bind_receiver" primitive.

The purpose of the Bind operation is to register an instance of an ESME with the SMSC system, and inform the SMSC that the sending SME wishes to use this virtual circuit for commands initiated by the SMSC. To this end the Bind must provide key information within the "message" field of the protocol message.

- The **password** must match the SMSC administration password for the instance of the ESME.

- The **system_id** and **system_type** provide a unique identification of the interface.

  Associated with the interface is a unique default "callback address" which is configured via SMSC administration. The "callback address" is employed as the default source address, in cases where the actual ESME address is not supplied.

  The interface may act as either an ESME in it's own right or as an agent for the transport of messages to or from other ESME's. (See figure 6-1).

  In it's role as agent, the range of ESME addresses served by the interface is specified via a "regular expression" (See Note 2). This may be defined explicitly in the bind request or configured by SMSC administration.

  Note 1: For the bind_transmitter the addr_ton, addr_npi and range of SME addresses (address_range) is not relevant and should be set to NULL.

  Note2: The "regular expression" in this context is a text pattern representing a range of addresses or a specific address. For further detail refer to the SMPP Application Guide[1].
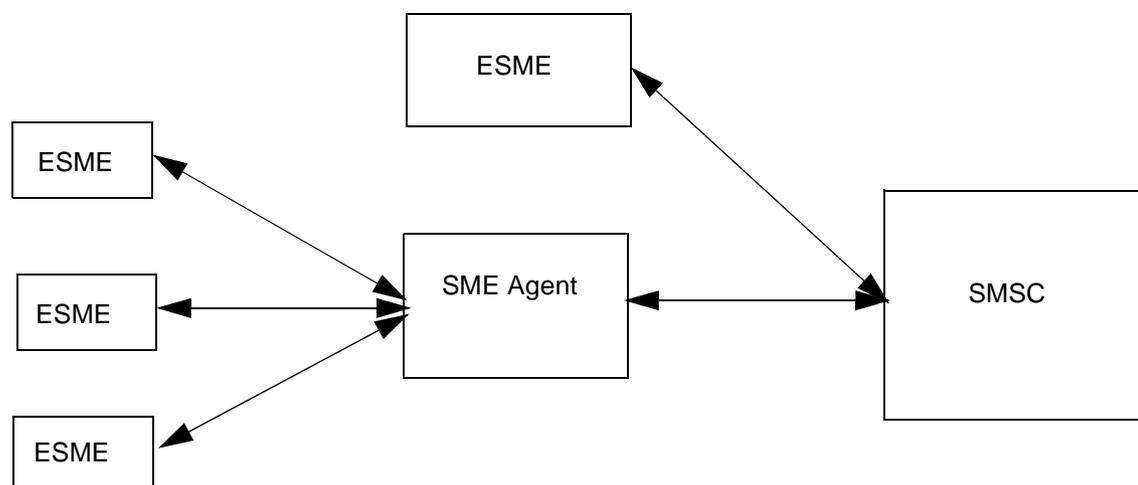

Figure 6-1:   ESME/SME address routing to/from SMSC

6.3.1.1 "BIND_RECEIVER" Syntax

These parameters are included in the "message" field of the protocol message when the "command id" field is "bind_receiver".

Field Name	Size (bytes)	Type	Description
system_id	Var. Max 16	C-Octet String	Identifies the system requesting a bind to the SMSC. This variable length field may have leading spaces.
password	Var. Max 9	C-Octet String	The password is used for security purposes. This is a configurable attribute within the SMSC.
system_type	Var. Max 13	C-Octet String	Identifies the type of system requesting the bind. This may enable SMSC responses which are particular to a given type of ESME. This variable length field may have leading spaces.
interface_version	1	Integer	Identifies the version number (major) of the interface to be implemented.
addr_ton	1	Integer	Type of Number for use in routing Delivery Receipts. (See GSM 03.40 [2] 9.1.2.5) Where not required this should be NULL.
addr_npi	1	Integer	Numbering Plan Identity for use in routing Delivery Receipts. (See GSM 03.40 [2] 9.1.2.5) Where not required this should be NULL.
address_range	Var. Max 41	C-Octet String	Address range for use in routing short messages and Delivery Receipts to an ESME. This variable length field may have leading spaces. Where not required this should be a single NULL byte.

Table 6-2: bind_receiver

6.3.1.2 "BIND_RECEIVER_RESP" Syntax

Apart from setting the header fields, the acknowledge message to a 'bind_receiver" requires only a single parameter.

Field Name	Size (bytes)	Type	Description
system_id	Var. Max 16	C-Octet String	Identifies the SMSC to the ESME requesting the bind.

Table 6-3: bind_receiver_resp

6.3.1.3   "BIND_TRANSMITTER" Syntax

These parameters are included in the "message" field of the protocol message when the "command id" field is "bind_transmitter".

The Message layout is identical to the "bind_receiver" Message Layout except that the **addr_ton, addr_npi** and the **range of SME addresse**s(address_range) are not relevant and should be set to NULL

6.3.1.4   "BIND_TRANSMITTER_RESP" Syntax

The Message layout is identical to the "bind_receiver_resp" Message Layout except that the "command id" field setting specifies "bind_transmitter_resp".

6.3.2   "UNBIND" Operation.

The purpose of the Unbind operation is to deregister an instance of an ESME from the SMSC system.

6.3.2.1   "UNBIND" Syntax

Apart from setting the header fields, no other parameters are required in the data body.

6.3.2.2   "UNBIND_RESP" Syntax

Apart from setting the header fields, no other parameters are required in the data body.

6.3.3 "SUBMIT_SM" Operation.

This command is issued by the ESME to submit a short message to the SMSC for transmission to a specified subscriber.

When a real source address is provided in a registered submit_sm request, the source address can be used as the destination address for a delivery receipt. It can also be used in identifying the message source in a CDR. This source address must fall in the range of addresses associated with the bind command.

Where the originator of messages from the ESME is the ESME itself, or where the ESME does not have a real source address, the source address fields may be defaulted to NULL, and the source address will be taken from the SMSC administration "callback address" for the particular ESME instance.

The submit_sm operation can also be used to replace a short message which has previously been submitted. This is achieved by setting the replace_if_present_flag to 0x01 in the Interface. The first message found in the SMSC whose source and destination match those given in the submit_sm will have it's text replaced by the text in the short_message field of the submit_sm.

6.3.3.1 "SUBMIT_SM" Syntax

These parameters are included in the "message" field of the protocol message when the "command id" field is "***submit_sm***".

Field Name	Size (bytes)	Type	Description
service_type	Var. Max 6	C-Octet String	Indicates the type of service associated with the message. Where not required this should be set to a single NULL byte.
source_addr_ton	1	Integer	Type of number for source. Where not required this should be NULL. (See GSM 03.40 [2] 9.1.2.5)
source_addr_npi	1	Integer	Numbering Plan Indicator for source Where not required this should be NULL. (See GSM 03.40 [2] 9.1.2.5)
source_addr	Var. Max 21	C-Octet String (Decimal)	Address of SME which originated this message. This is the source address of the short message submitted. This variable length field may have leading spaces. Where not required this should be a single NULL byte.
dest_addr_ton	1	Integer	Type of number for destination. Where not required this should be NULL (See GSM 03.40 [2] 9.1.2.5)
dest_addr_npi	1	Integer	Numbering Plan Indicator for destination Where not required this should be NULL. (See GSM 03.40 [2] 9.1.2.5)

Table 6-4:   **submit_sm**

Field Name	Size (bytes)	Type	Description
destination_addr	Var. Max 21	C-Octet String (Decimal)	Destination address of this short message. For mobile terminated messages, this is the SME address of the target subscriber. This variable length field may have leading spaces. Where not required this should be a single NULL byte.
esm_class	1	Integer	Indication of message type. For the submit_sm command this field is unused, and should be set to NULL. For the deliver_sm command however, this field may identify the message as a delivery receipt.
protocol ID	1	Integer	GSM Protocol ID (See GSM 03.40 [2] 9.2.3.9)
priority_flag	1	Integer	Designates the message as priority. Setting priority on a message moves it to the top of the SMSC message queue for that subscriber. 0 = non-priority (default) 1 = priority >1=Reserved
schedule_delivery_time	17	C-Octet String	The absolute date and time at which delivery of this message must be attempted. The format is defined in section 7.5 Where not required this should be a single NULL byte.
validity_period	17	C-Octet String	The expiration time of this message. This is specified as an absolute date and time of expiry. The format is defined in section 7.5 Where not required this should be a single NULL byte.
registered_delivery_flag	1	Integer	Flag indicating if the message is a registered short message and thus if a Delivery Receipt is required upon the message attaining a final state. 0=No receipt required (non-registered delivery). 1=Receipt required (registered delivery) >1=Reserved
replace_if_present_flag	1	Integer	Flag indicating if submitted message should replace an existing message between the specified source and destination. 0=Don't Replace (default) 1=Replace >1=Reserved
data_coding	1	Integer	GSM Data-Coding-Scheme (See GSM 03.40 [2] 9.2.3.10)

Table 6-4:   submit_sm

Field Name	Size (bytes)	Type	Description
sm_default_msg_id	1	Integer	Indicates the default short message to send, by providing an index into the table of Predefined Messages set up by the SMSC administrator. This should be set to NULL if a text message is being sent. Range is 0x01 to 0x64. (See SMPP Applications Guide [1] - Default Short Message).
sm_length	1	Integer	Length of the text of the message in bytes.
short_message	Var. Max 161	Octet String	Up to 160 bytes of data. This is the text that is transmitted to the mobile station. Note that only 'sm_length' bytes will be used.

Table 6-4:   submit_sm

6.3.3.2   "SUBMIT_SM_RESP" Syntax

These parameters are included within the "message" field of the protocol message when the "message type" field is "*submit_sm_resp*".

Field	Size (bytes)	Type	Description
Message ID	Var. Max 9	C-Octet String (Hex)	This field contains the message ID internal to the SMSC. It may be used at a later stage to query the status of a message, to replace a message, or match the original message to a corresponding delivery receipt (deliver_sm) message. If absent this field must contain a single NULL byte. The SMSC will return a value for this field.

Table 6-5:   submit_sm_resp

6.3.4   "SUBMIT_MULTI" Operation

The SUBMIT_MULTI primitive is used to submit messages to an SME Address, a Distribution List and Multiple Recipients. The Command Id of this primitive is "submit_multi". The message field of this body is

Field Name	Size (bytes)	Type	Description
service_type	Var. Max 6	C-Octet String	Indicates the type of service associated with the message. Where not required this should be set to NULL.
source ton	1	Integer	Type of number for source. Where not required this should be NULL.
source npi	1	Integer	Numbering Plan Indicator for source. Where not required this should be NULL.
source address	Var. Max 21	C-Octet String	Address of SME which originated this message. Where not required this should be NULL.
number_of_dests	1	Integer	Number of destination addresses - indicates the number of dest_address structures that are to follow. **NOTE:** This is set to 1 when submitting to one SME Address OR when submitting to one Distribution List.
dest_address	Var. Max 24	C-Octet String	Contains a list of SME addresses or/and Distribution List names. ref table 6.7 dest_address.
dest_address....			
esm_class	1	Integer	Indication of Message Type. For the submit_multi command this field is unused and should be set to NULL.
protocol_Id	1	Integer	GSM Protocol Id.
priority_flag	1	Integer	0 = nonpriority 1 = priority >1= reserved
schedule_delivery_time	17	C-Octet String	The absolute date and time at which delivery of this message must be attempted. The format is defined in section 7.5. Where not required this should be a single NULL.
validity_period	17	C-Octet String	The expiration time of this message. This is specified as an absolute date and time of expiry. The format is defined in section 7.5. Where not required this should be a single NULL.

Table 6-6: submit_multi

Field Name	Size (bytes)	Type	Description
registered_delivery_flag	1	Integer	Flag indicating if the message is a registered short message and thus if a Delivery Receipt is required upon the message attaining a final state.
replace_if_present_flag	1	Integer	Flag indicating if submitted message should replace an existing message between the specified source and destination. Where not required this should be NULL. **NOTE:** This is not supported for submission of short messages to Distribution List(s) and multiple recipients - only supported for single message submission. A single NULL should be supplied in these situations.
data_coding	1	Integer	GSM Data-Coding-Scheme.
sm_default_msg_id	1	Integer	Indicates the default short message to send.
sm_length	1	Integer	Length of the text of the message in bytes.
short_message	Var. Max 161	Octet String	Up to 160 bytes of data. Note that only 'sm_length' bytes will be used.

Table 6-6: submit_multi

Field Name	Size (bytes)	Type	Description
dest_flag	1	Integer	Flag which will identify whether destination address is a DL name or an SME address. 1 - SME Address 2 - Distribution List Name
SME Address OR Distribution List Name	Var. Max 23	Integer	Depending on dest_flag this could be an SME Address or a Distribution List Name. ref Table 6-8 DL Name ref Table 6-10 SME Address.

Table 6-7: dest_address

Field Name	Size (bytes)	Type	Description
dl_name	Var. Max 21	C-Octet String	Name of distribution list.

Table 6-8: DL Name

6.3.4.1 "SUBMIT_MULTI_RESP" Syntax

These parameters are included within the "message" field of the protocol message when the "message type" field is "*submit_multi_resp*".

Field	Size (bytes)	Type	Description
Message ID	Var. Max 9	C-Octet String (Hex)	This field contains the message ID internal to the SMSC. It may be used at a later stage to query the status of a message, to replace a message, or match the original message to a corresponding delivery receipt (deliver_sm) message. If absent this field must contain a single NULL byte. The SMSC will return a value for this field.
No_UnSuccess	1	Integer	The number of SME addresses that were unsuccessfully submitted to the system database.
UnSuccess_SMEs	Var. Max 4600	C-Octet String	The SME addresses to which submission was unsuccessful (Table 6-10 SME_Address).

Table 6-9:   submit_multi_resp

Field	Size (bytes)	Type	Description
SME ton	1	Integer	Type of number for SME.
SME npi	1	Integer	Numbering Plan Indicator for SME.
SME address	Var. Max 21	C-Octet String	Address of SME which originated this message.
Error status	4	Integer	This field will indicate the success or failure of the submit multi request to this SME address. A list of exception codes is given in section 7.1.

Table 6-10: SME_Address

6.3.5   "DELIVER_SM" Operation

This is issued by the SMSC. Using this command, the SMSC may submit a short message to the ESME for delivery. It is also used to return a delivery receipt for a message which had been submitted with the delivery receipt flag set.

The values for destination address will depend on whether the ESME is the final destination of the short message, or merely routes the message to its final recipient (e.g. paging messages).

One should note that delivery receipts are returned to the originating SME using this command. In this instance of a deliver_sm command, the esm_class field will identify the message as a delivery receipt, and the required data relating to the original short message will be given in the message text field. (See SMPP Applications Guide [1] - Delivery Receipts).

6.3.5.1   "DELIVER_SM" Syntax

The parameters included within the "message" field of the protocol message when the "command id" field is "deliver_sm", are the same as for "submit_sm".

6.3.5.2   "DELIVER_SM_RESP" Syntax

The parameters included within the "message" field of the protocol message when the "command id" field is "***deliver_sm_resp***", are the same as for "***submit_sm_resp***".

6.3.6 QUERY

Three different types of Query of short messages are supported by the SMPP application. An ESME can query the status of a message sent to a single SME Address, the status of a message submitted to a single Distribution List and can query the status of a message sent to multiple recipients.

6.3.6.1 "QUERY_SM" Operation

This Command is issued by the ESME to query the status of a previously submitted short message.

Where a message to be replaced was originally submitted with an individually identified SME source address, the originator address in the query_sm command must match. Where the original source address was defaulted to NULL, (i.e. the originator of messages from the ESME is the ESME itself, or the ESME does not have a real source address) then the originator address in the query_sm command should also be NULL, and the source address will be taken from the SMSC administration "callback address" for the particular ESME instance.

6.3.6.2 "QUERY_SM" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "*query_sm*".

Field	Size (bytes)	Type	Description
original_message_id	Var. Max 9	C-Octet String (Hex)	Message ID of the message whose state is to be queried. This must be the Message ID allocated to the original short message when submitted to the SMSC by the submit_sm command, and returned in the submit_sm_resp message by the SMSC. This variable length field may have leading spaces.
originating_ton	1	Integer	Type of Number of originator This is used for verification purposes, and must match that supplied in the corresponding '*submit_sm*' request (See GSM 03.40 [2] 9.1.2.5)
originating_npi	1	Integer	Numbering Plan Identity of originator This is used for verification purposes, and must match that supplied in the corresponding '*submit_sm*' request (See GSM 03.40 [2] 9.1.2.5)
originating_addr	Var. Max 21	C-Octet String (Decimal)	Address of originator This is used for verification purposes, and must match that supplied in the corresponding '*submit_sm*' request

Table 6-11:   query_sm

6.3.6.3   "QUERY_SM_RESP" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "*query_sm_response*".

Field	Size (bytes)	Type	Description
original_message_id	Var. Max 9	C-Octet String (Hex)	Message ID of the message whose state is being queried. This must be the Message ID allocated to the original short message when submitted to the SMSC by the submit_sm command, and returned in the submit_sm_resp message by the SMSC. This variable length field may have leading spaces.
final_date	Var. Max 17	C-Octet String	Date and time when the submitted message reached the final state. For messages which have not yet reached a final state this field will contain a single NULL byte. The date format is detailed in Section 7.5.
message_status	1	Integer	Specifies the status of the SM. See section 7.4
Error_code	1	Integer	Where appropriate this holds a GSM error code or an SMSC error code defining the reason for failure of message delivery. (See GSM 03.40 [2] 3.3) (Refer also to section 7.3)

Table 6-12:   query_sm_resp

6.3.6.4   "QUERY_LAST_MSGS" Operation

This operation allows an ESME to query the most recent messages that are in the system for that originating source address. The messages found in the system with the specific originating source address will be returned to the ESME along with some message details. The maximum number of messages that can be queried is 100.

NOTE:

•   If the number of messages specified is greater than 100 then the latest 100 messages will be returned for that source address.

•   If the total number of messages specified is not found in the database for that source address then the total number of messages found will be returned.

6.3.6.5   "QUERY_LAST_MSGS" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "*query_last_msgs*".

Field	Size (bytes)	Type	Description
source_addr_ton	1	Integer	Type of number for source.
source_addr_npi	1	Integer	Numbering Plan Indicator for source.
source_addr	Var. Max 21	C-Octet String	Address of SME which originated this message.
Num_Messages	1	Integer	Number of messages to be queried.

Table 6-13: query_last_msgs

6.3.6.6   "QUERY_LAST_MSGS_RESP" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "*query_last_msgs_resp*".

Field Name	Size (bytes)	Type	Description
number_msgs	1	Integer	Number of messages found in the database for the address specified - indicates total number of message details to follow.
message_details	Var. Max 9	C-Octet String	Message Id for each message in the database ref. table 6-15 message_details.
message_details ...			

Table 6-14: query_last_msgs_resp

Field Name	Size (bytes)	Type	Description
msgid	Var. Max 9	C-Octet String (hex)	The message id allocated to the message by the SMSC when originally submitted.

Table 6-15: message_details

6.3.6.7   "QUERY_MSG_DETAILS" Operation

This operation is used to return all the details of a specific message stored in the database for a particular message id.

6.3.6.8   "QUERY_MSG_DETAILS" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "*query_msg_details*".

Field Name	Size (bytes)	Type	Description
original_message_id	Var. Max 9	C-Octet String (Hex)	Msgid given to the message by the SMSC when the message is originally submitted.
source_addr_ton	1	Integer	Type of number for source.
source_addr_npi	1	Integer	Numbering Plan Indicator for source.
source_addr	Var. Max 21	C-Octet String	Address of SME which originated this message.
sm_length	1	Integer	Number of bytes of message text required.

Table 6-16: query_msg_details

6.3.6.9   "QUERY_MSG_DETAILS_RESP" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "*query_msg_details*".

Field Name	Size (bytes)	Type	Description
service_type	Var. Max 6	C-Octet String	Indicates the type of service associated with the message. Where not required this should be set to NULL.
source_addr_ton	1	Integer	Type of number for source.
source_addr_npi	1	Integer	Numbering Plan Indicator for source.
source_addr	Var. Max 21	C-Octet String	Address of SME which originated this message.
number_of_dests	1	Integer	Number of destinations addresses.

Table 6-17: query_msg_details_resp

Field Name	Size (bytes)	Type	Description
dest_address	Var. Max 24	C-Octet String	Contains an SME Address(es) or DL Name. ref table 6-18 dest_address.
dest_address ...			
protocol_Id	1	Integer	GSM Protocol Id.
priority_flag	1	Integer	0 = nonpriority 1 = priority >1= reserved
schedule_delivery_time	17	C-Octet String	The absolute date and time at which delivery of this message must be attempted.
validity_period	17	C-Octet String	The expiration time of this message. This is specified as an absolute date and time of expiry.
registered_delivery_flag	1	Integer	Flag indicating if the message is a registered short message and thus if a Delivery Receipt is required upon the message attaining a final state.
data_coding	1	Integer	GSM Data-Coding-Scheme.
sm_length	1	Integer	Length of the text of the message in bytes.
short_message	Var. Max 161	Octet String	Up to 160 bytes of data. Note that only 'sm_length' bytes will be used.
msgid	Var. Max 9	Integer	The message id allocated to the message by the SMSC when originally submitted.
final_date	Var. Max 17	C-Octet String	The time and date at which the short message reached it's final state.
message_status	1	Integer	Specifies the status of the short message.
Error_code	1	Integer	Where appropriate this holds the GSM error code or an SMSC error code for the attempted delivery of the message. Refer to section 7.3.

Table 6-17: query_msg_details_resp

Field Name	Size (bytes)	Type	Description
dest_flag	1	Integer	Flag which will identify whether address is a DL name or an SME Address. 1 - SME Address 2 - DL Name.
SME Address OR DL Name	Var. Max 23	Integer	Depending on dest_flag this could be an SME Address or Distribution List Name. Ref. Table 6-19 SME Address. Ref. Table 6-20 DL Name.

Table 6-18: dest_address

Field Name	Size (bytes)	Type	Description
dest_ton	1	Integer	Type of number for destination.
dest_npi	1	Integer	Numbering Plan Indicator for destination.
dest_addr	Var. Max 21	C-Octet String	Destination Address of short message.

Table 6-19: SME Address

Field Name	Size (bytes)	Type	Description
dl_name	Var. Max 21	C-Octet String	Name of distribution list.

Table 6-20: DL Name

6.3.7   "CANCEL_SM" Operation

 This command is issued by the ESME to cancel one or more outstanding short messages. The command may specify a particular message, or all messages for a particular source and destination.

- If the message ID is set to the ID of a previously submitted message, then provided the source and destination addresses supplied in the interface match, that message will be cancelled.

- If the message ID is null all outstanding undelivered messages with the source and destination addresses given in the interface will be cancelled for the particular interface of the AIM. If the source address is set to NULL in the interface the source address will be taken from the SMSC administration "callback address" for the particular ESME instance.

- A typical use of the command is to cancel outstanding undelivered VoiceMail Alert messages for a subscriber whose mailbox has just been directly accessed by the subscriber. The response (cancel_sm_resp) will indicate whether the message(s) had already been sent

6.3.7.1   "CANCEL_SM" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "*cancel_sm*".

Field Name	Size (bytes)	Type	Description
service type	Var. Max 6	C-Octet String	Indicates the type of service associated with the message. Where not required this should be NULL.
original_message_id	Var. Max 9	C-Octet String (Hex)	Message ID of the message to be cancelled. This must be the Message ID allocated to the original short message when submitted to the SMSC by the submit_sm command, and returned in the submit_sm_resp message by the SMSC. This variable length field may have leading spaces.
source_addr_ton	1	Integer	Type of Number of originator. This is used for verification purposes, and must match that supplied in the corresponding '*submit_sm*' request. Where not required this should be NULL. (See GSM 03.40 [2] 9.1.2.5)
source_addr_npi	1	Integer	Numbering Plan Identity of originator. This is used for verification purposes, and must match that supplied in the corresponding 'submit_sm' request. Where not required this should be NULL. (See GSM 03.40 [2] 9.1.2.5)

Table 6-21:   cancel_sm

Field Name	Size (bytes)	Type	Description
source_addr	Var. Max 21	C-Octet String (Decimal)	Source address of message(s) to be cancelled. This is used for verification purposes, and must match that supplied in the corresponding 'submit_sm' request This variable length field may have leading spaces.
dest_addr_ton	1	Integer	Type of number for destination. (See GSM 03.40 [2] 9.1.2.5)
dest_addr_npi	1	Integer	Numbering Plan Indicator for destination (See GSM 03.40 [2] 9.1.2.5)
destination_addr	Var. Max 21	C-Octet String (Decimal)	Destination address of message(s) to be cancelled. This is used for verification purposes, and must match that supplied in the corresponding 'submit_sm' request This variable length field may have leading spaces. Where not required this should be a single NULL byte.

Table 6-21:　cancel_sm

6.3.7.2　　　　"CANCEL_SM_RESP" Syntax

Apart from setting the header fields, no other parameters are required in the data body.

6.3.8   "REPLACE_SM" Operation

This command is issued by the ESME to replace an outstanding short message for a subscriber.

The message_id is set to the ID of a previously submitted message.Where a message to be replaced was originally submitted with an individually identified SME source address, the originator address in the replace_sm command must match. Where the original source address was defaulted to NULL, (i.e. the originator of messages from the ESME is the ESME itself, or the ESME does not have a real source address)   then the originator address in the replace_sm command should also be NULL, and the source address will be taken from the SMSC administration "callback address" for the particular ESME instance.

6.3.8.1   "REPLACE_SM" Syntax

These parameters are included within the "message" field of the protocol message when the "command id" field is "*replace_sm*".

Field Name	Size (bytes)	Type	Description
original_message_id	Var. Max 9	C-Octet String (Hex)	Message ID of the message to be replaced. This must be the Message ID allocated to the original short message when submitted to the SMSC by the submit_sm command, and returned in the submit_sm_resp message by the SMSC. This variable length field may have leading spaces.
orig_addr_ton	1	Integer	Type of Number of originator. This is used for verification purposes, and must match that supplied in the corresponding '*submit_sm*' request. Where not required this should be NULL.
orig_addr_npi	1	Integer	Numbering Plan Identity of originator. This is used for verification purposes, and must match that supplied in the corresponding 'submit_sm' request Where not required this should be NULL
originating_addr	Var. Max 21	ASCII	Originating address of the short message to be replaced. This is used for verification purposes, and must match that supplied in the corresponding 'submit_sm' request This variable length field may have leading spaces.
schedule_delivery_time	17	C-Octet String	The absolute date and time at which delivery of this message must be attempted. Where not specified the original scheduled delivery time, if specified, will apply. The format is defined in section 7.5 Where not required this should be a single NULL byte.

Table 6-22:   replace_sm

Field Name	Size (bytes)	Type	Description
validity_period	17	C-Octet String	The expiration time of this message. This is specified as an absolute date and time of expiry. Where not specified the original expiration time, if specified, will apply. The format is defined in section 7.5 Where not required this should be a single NULL byte.
registered_delivery_flag	1	Integer	Flag indicating if the message is a registered short message and thus if a Delivery Receipt is required upon the message attaining a final state. (See SMPP Applications Guide [1] - Delivery Receipts) 0=No receipt required (non-registered delivery). 1=Receipt required (registered delivery) >1=Reserved
sm_default_msg_id	1	Integer	Indicates the default short message to send, by providing an index into the table of predefined messages set up by the SMSC administrator. This should be set to NULL if a text message is being sent. Range is 0x01 to 0x64. (See SMPP Applications Guide [1] - Default Short Message).
sm_length	1	Integer	Length of the text of the message in bytes.
short_message	Var. Max 161	Octet String	Up to 160 bytes of data. This is the text that is transmitted to the mobile station. This text, if specified will be used to replace the existing text for the originally submitted SM. (See SMPP Applications Guide [1] - Default Short Message).

Table 6-22: replace_sm

6.3.8.2 "REPLACE_SM_RESP" Syntax

Apart from setting the header fields, no other parameters are required in the data body.

6.3.9   "ENQUIRE_LINK" Operation

This message is used to provide a confidence-check of the communication path between ESME and the SMSC. On receipt of this request the SMSC will simply respond with an ***enquire_link_resp***, thus verifying that the application level connection between the SMSC and the ESME is functioning. The ESME can respond by sending any valid SMPP primitive.

6.3.9.1   "ENQUIRE_LINK" Syntax

Apart from setting the header fields, no other parameters are required in the data body.

6.3.9.2   "ENQUIRE_LINK_RESP" Syntax

Apart from setting the header fields, no other parameters are required in the data body.

6.3.10   "PARAM_RETRIEVE" Operation

This operation is used to retrieve the value for a configurable parameter.

6.3.10.1   "PARAM_RETRIEVE" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "**param_retrieve**".

Field	Size (bytes)	Type	Description
param_name	32	C-Octet String	Name of configurable parameter.

Table 6-23: param_retrieve

6.3.10.2   "PARAM_RETRIEVE_RESP" Syntax

These parameters are included within the "message" field of the protocol message when the message type is "**param_retrieve_resp**".

Field	Size (bytes)	Type	Description
param_value	101	C-Octet String	Value retrieved from the SMSC database for a particular configurable parameter.

Table 6-24: param_retrieve_resp

7.   System Definitions

The following sections define the various system codes for Command-ID's and Error Codes.

Note: For ease of maintenance a 'C' include file is available which defines the actual values for these definitions.

7.1   Error Codes

The following are a list of error codes that can be returned in the status field of a message.

Error Code	Description
ESME_ROK	Ok - Message Acceptable
ESME_RINVMSGLEN	Invalid Message Length
ESME_RINVCMDLEN	Invalid Command Length
ESME_RINVCMDID	Invalid Command ID
ESME_RINVBNDSTS	Invalid bind status
ESME_RALYBND	Bind attempted when already bound
ESME_RINVPRTFLG	Invalid priority flag
ESME_RINVREGDLVFLG	Invalid registered-delivery flag
ESME_RSYSERR	SMSC system error
ESME_RINVPAR	Invalid parameter
ESME_RINVSRCADR	Invalid source address
ESME_RINVDSTADR	Invalid destination address
ESME_RINVMSGID	Invalid message-id
ESME_RINVPASWD	Invalid password
ESME_RINVPASWDLEN	Invalid password length
ESME_RINVSYSIDSRV	Invalid System-ID
ESME_RCNTCANMSG	Cannot cancel a message
ESME_RINVDATFMT	Invalid date format
ESME_RCNTREPMSG	Cannot replace a message
ESME_RMSGQFUL	Too many messages in queue, at present
ESME_RSERNOTSUP	Service Type not supported
ESME_RINVREPADDR	Address Mismatch in Replacement attempt
ESME_RINVNUMDESTS	Invalid number of destination addresses

Table 7-1:   Error Codes

Error Code	Description
ESME_RINVDESTFLAG	Invalid Destination Flag Option
ESME_RINVSUBREP	Invalid value for submit with replace option
ESME_RINVADRLEN	Invalid length for address field
ESME_RINVESMCLASS	Invalid value for esm_class field
ESME_RCNTSUBDL	Cannot submit to a distribution list
ESME_RCNTSUBMULRECP	Cannot submit to multi-recipients
ESME_RCNTRETRPARAM	Cannot retrieve configurable parameter
ESME_EINVPARAMLEN	Invalid length for incoming ESME configurable parameter
ESME_RINVDISTNAMELEN	Invalid length for distribution list name
ESME_RINVSRCADDRLEN	Invalid length for source address
ESME_RINVDSTADDRLEN	Invalid length for destination address
ESME_RINVSRCTON	Invalid type of number for source
ESME_RINVSRCNPI	Invalid numbering plan indicator for source
ESME_RINVDSTTON	Invalid type of number for destination
ESME_RINVDSTNPI	Invalid numbering plan indicator for destination
ESME_RINVESMTYPE	Invalid esm type
ESME_RINVTEXTLEN	Invalid length for short message text
ESME_RINVREPFLAG	Invalid submit with replace flag option
ESME_RINVNUMMSGS	Invalid number of messages specified for query_last_msgs primitive
ESME_RINVSYSTYP	Invalid system type
ESME_RLIMITEXCEED	Overall transaction limit exceeded for AIM session
ESME_RTXNOTALLOWD	Transaction not allowed
ESME_RTHROTTLEXCD	Submit rate exceeded
ESME_RTXEXCEED	Transaction limit exceeded for that primitive
ESME_RINVSCHED	Invalid Schedule Date
ESME_RINVEXPIRY	Invalid Validity Date
ESME_RUNKNOWNERR	Unknown Error

Table 7-1:   Error Codes

7.2 Command I.D. Values

The following is a list of the command ids and their associated values.

Command ID Code	Command ID	Description
ESME_BNDRCV	*bind_receiver*	Bind to SMSC Kernel as a receiver
ESME_BNDRCV_RESP	*bind_receiver_resp*	Response to bind_receiver
ESME_BNDTRN	*bind_transmitter*	Bind to SMSC Kernel as transmitter
ESME_BNDTRN_RESP	*bind_transmitrer_resp*	Response to *bind_transmitter*
ESME_UBD	*unbind*	Unbind from SMSC Kernel
ESME_UBD_RESP	*unbind_resp*	Response to *unbind*
ESME_SUB_SM	*submit_sm*	Submit a short-message
ESME_SUB_SM_RESP	*submit_sm_resp*	Response to *submit_sm*
ESME_SUB_MULTI	*submit_multi*	Submit a short message to an SME Address a Distribution list Multiple Recipients
ESME_SUB_MULTI_RESP	*submit_multi_resp*	Respose to *submit_multi*.
SMSC_DELIVER_SM	*deliver_sm*	Submit a short-message to ESME
SMSC_DELIVER_SM_RESP	*deliver_sm_resp*	Response to *deliver_sm*
ESME_QUERY_SM	*query_sm*	Query status of a short-message
ESME_QUERY_SM_RESP	*query_sm_resp*	Response to *query_sm*
ESME_QUERY_LAST_MSGS	*query_last_msgs*	Query status of a certain number of messages in the database with a specific source address.
ESME_QUERY_LAST_MSGS_RESP	*query_last_msgs_resp*	Response to **query_last_msgs**.
ESME_QUERY_MSG_DETAILS	*query_msg_details*	Query the details of a particular message.
ESME_QUERY_MSG_DETAILS_RESP	*query_msg_details_resp*	Response to **query_msg_details**.
ESME_CANCEL_SM	*cancel_sm*	Cancel a short message(s)
ESME_CANCEL_SM_RESP	*cancel_sm_resp*	Response to *cancel_sm*
ESME_REPLACE_SM	*replace_sm*	Replace a short message
ESME_REPLACE_SM_RESP	*replace_sm_resp*	Response to *replace_sm*
ESME_QRYLINK	*enquire_link*	Link confidence check
ESME_QRYLINK_RESP	*enquire_link_resp*	Response to *enquire_link*

Table 7-2:   Command ID Values

Command ID Code	Command ID	Description
ESME_PARAM_RETRIEVE	*param_retrieve*	Retrieve value for configurable parameter
ESME_PARAM_RETRIEVE_RESP	*param_retrieve_resp*	Response to **param_retrieve**
ESME_NACK	*nack*	Negative Acknowledgement

Table 7-2: Command ID Values

7.3　Error Codes

Where the message is submitted to the SMSC with the registered delivery flag set, a status report is generated after the submitted short message reaches it final state.The following is a list of error codes and their associated descriptions that can be returned in the delivery receipt, query_sm and query_msg_details primitives

7.3.1　GSM Error Codes

The following is a list of the GSM error codes (See GSM 03.40 [2] 3.3) and their associated descriptions.

Error Code	Description
ERROR_NONE	No error code given
P_UNKNOWN	unknown subscriber
P_PROVISION	Not Provisioned
T_BARRED	Call barred
T_SUPPORT	Facility not supported
T_ABSENT	Absent Subscriber
T_MSSUPPORT	SMS not supported by MS
T_MSERROR	Error in MS receiving message
P_CUG	CUG reject
T_ABSENT_DETACHED	Absent Subscriber Detached
T_SYSTEM	System Failure
T_MEMCAP	Memory capacity exceeded
T_SUPPORT_ROAMING	Subscriber roamed into different PLMN

Table 7-3:　GSM Error Codes

7.3.2　SMSC Error Codes

The following is a list of possible SMSC error codes.

Error Code	Description
L_RECEIVER_FULL	Reciever is blocked
L_NO_RECEIVER	No receiver bound to SMSC
L_NONE	No error code given

Table 1-4:

7.4   Message States

The following is a list of the states that a short message may achieve.

Message State	Description
EN_ROUTE	Message is enroute
DELIVERED	Message in delivered state
EXPIRED	Messsage validity period has expired.
DELETED	Message has been deleted.
UNDELIVERABLE	Message is undeliverable
ACCEPTED	Message is in accepted state
INVALID	Message is in invalid state

Table 1-5:   Message States

7.5   Time Format

Time and Date fields are represented in a format similar to that specified in GSM 03.40 [2] 9.2.3.11.
In this interface all time/date related fields will be in ASCII with the following format:
"YYMMDDhhmmsstnnp" where

'YY'	last two digits of the year (00-99)
'MM'	month (01-12)
'DD'	day (01-31)
'hh'	hour (00-23)
'mm'	minute (00-59)
'ss'	second (00-59)
't'	tenths of second (0-9)
'nn'	Time difference in quarter hours between local time (as expressed in the first 13 bytes) and UTC (Universal Time Constant) time (00-48).
'p' - "+"	Local time is nn quarter hours advanced in relation to UTC time.
"-"	Local time is nn quarter hours retarded in relation to UTC time.

Note: Where responses are reported by the SMSC the local time of the SMSC will be given, and the format will be "YYMMDDhhmmss", with the same definitions as above.

Change Log

Version(Old->New):	3.0 -> 3.1	Author: S.H.
Source of Change	**Reason**	**Date**
Sonia Fitzpatrick	New functionality added.	30/11/95
Location	**Description**	**Ripple Effect**
Table 7-2 section 7.2 section 7.3 section 6.3.9 section 6.1	New Primitives Added: ESME_QUERY_ALL_MSGS, ESME_QUERY_MSG_DETAILS, ESME_SUBMIT_MULTI, ESME_PARAM_RETRIEVE, Added C-Octet Fixed Length String Modified Enquire Link Primitive Added new Command Id Values Added new Error Codes. Added new event log names.	SMPP Application Guide [1]

Change No. 1

	3.1 -> 3.2	S.H.
Source of Change	**Reason**	**Date**
Formal Riview	Review of document	11/12/95
Location	**Description**	**Ripple Effect**
Throughout document	Review comments added to document	SMPP Application Guide [1]

Change No. 2

3.2 -> 3.3		S.H.
Source of Change	**Reason**	**Date**
Sonia Fitzpatrick	SMPP Provisioning Documents updated - delete member uses member description.	14/01/96
Location	**Description**	**Ripple Effect**
IS-SMS-SMINSP-0055	Updated SMPP Provisioning Documents.	SMPP Application Guide [1] SMPP Provisioning Interface Specification [3] SMPP Provisioning Application Guide [4]

Change No. 3