Oracle® Communications Network Charging and Control

Short Message Peer-to-Peer Protocol (SMPP) Protocol Implementation Conformance Statement

Release 15.2

January 2026

Copyright

Contents

About This Document

Scope

This document describes the extent to which Messaging Manager conforms to the Short Message Peer-to-Peer Protocol Specification.

Audience

This document is intended to be read by Oracle staff. It has been prepared on the assumption that the reader is familiar with Messaging Manager as well as short message peer-to-peer protocols.

Document Conventions

Typographical Conventions

The following terms and typographical conventions are used in the Oracle Communications Network Charging and Control (NCC) documentation.

Formatting Convention	**Type of Information**
Special Bold	Items you must select, such as names of tabs. Names of database tables and fields.
Italics	Name of a document, chapter, topic or other publication. Emphasis within text.
Button	The name of a button to click or a key to press. **Example:** To close the window, either click **Close**, or press **Esc**.
Key+Key	Key combinations for which the user must press and hold down one key and then press another. Example: **Ctrl+P** or **Alt+F4**.
`Monospace`	Examples of code or standard output.
`Monospace Bold`	Text that you must enter.
variable	Used to indicate variables or text that should be replaced with an actual value.
menu option > menu option >	Used to indicate the cascading menu option to be selected. Example: **Operator Functions > Report Functions**
hypertext link	Used to indicate a hypertext link.

Specialized terms and acronyms are defined in the glossary at the end of this guide.

Chapter 1

Messaging Manager and SMPP Document Versions

Overview

Introduction

This chapter defines the version of the Messaging Manager implementation and the SMPP document against which it is compared.

In this chapter

This chapter contains the following topics.

Messaging Manager

Messaging Manager implementation

- Target platform\
  - Platform - SPARC Solaris
  - OS - SunOs 5.9
  - Oracle - 9.2.05
- Build environment
  - Compiler - GNU GCC 3.2.3
  - Binutils - GNU binutils 2.1.4
  - bison - 1.35
  - flex - 2.5.4
- Oracle packages
  - Full installation of:
    - SLEE - 3.2.0
    - HssScIf - 3.4.27
  - Plus the following SLC packages:
    - smsScp - 3.0.0
    - acsScp - 2.4.0
    - beApiScp - 2.2.0.5
    - acsCbScp - 2.2.0.6
  - Plus the following SMS packages:
    - smsSms - 3.0.0
    - acsSms - 2.4.0
    - beApiSms - 2.2.0.5
    - acsCbSms - 2.2.0.6

SMPP

SMPP document

This statement of compliance refers to SMS Forum document entitled:

Short Message Peer-to-Peer
Protocol Specification
Version 5.0

For the purpose of this document, *Short Message Peer-to-Peer Protocol Specification Version 5.0* will be referred to as *The Specification*.

Chapter 2

Compliance Statements For SMPP Sessions (2)

Overview

Introduction

This chapter states the compliance of Messaging Manager with clauses of Section 2 of *The Specification*.

In this chapter

This chapter contains the following topics.

References to The Specification

Convention

As a cross reference, the clause number of *The Specification* is included in brackets at the end of each compliance statement title.

Specification Clauses 2.1 and 2.2

Application Layer Communication (2.1)

For TCP/IP connections, Messaging Manager complies.

For X.25 connections, Messaging Manager does not comply.

Messaging Manager does not support X.25.

Establishing a SMPP Session (2.2)

For TCP/IP connections, Messaging Manager complies.

For X.25 connections, Messaging Manager does not comply.

For SMPP, Messaging Manager can be configured to use any port including IANA standard port 2775.

X.25 is not supported by Messaging Manager.

Session States (2.3)

Open (2.3.1)

For TCP/IP connections, Messaging Manager complies.

For X.25 connections, Messaging Manager does not comply.

Bound_TX (2.3.2)

Messaging Manager complies.

Bound_RX (2.3.3)

Messaging Manager complies.

Bound_TRX (2.3.4)

Messaging Manager complies.

Unbound (2.3.5)

Messaging Manager complies.

Closed (2.3.6)

Messaging Manager complies.

Outbound (2.3.7)

Messaging Manager complies.

Messaging Manager can ignore outbind requests to a port if the port is configured not to receive.

Operation Matrix (2.4)

Relevance

The following compliance statements refer to *Table 2-1 Operation Matrix* of *The Specification*.

alert_notification

Messaging Manager complies.

Messaging Manager does not construct these messages but does relay them.

bind_receiver

Messaging Manager complies.

bind_receiver_ resp

Messaging Manager complies.

bind_transceiver

Messaging Manager complies.

bind_transceiver_resp

Messaging Manager complies.

bind_transmitter

Messaging Manager complies.

bind_transmitter_resp

Messaging Manager complies.

broadcast_sm

Messaging Manager does not comply.

Code exists to decode the incoming broadcast message but ProtocolHandler::stateBound() does not consider this possibility and returns genericNack.

Messaging Manager does not construct this message.

broadcast_sm_ resp

Messaging Manager does not comply.

Code exists to decode the incoming broadcast message but ProtocolHandler::stateBound() does not consider this possibility and returns genericNack.

Messaging Manager does not construct this message.

cancel_ broadcast_sm

Messaging Manager does not comply.

Code exists to decode the incoming broadcast message but ProtocolHandler::stateBound() does not consider this possibility and returns genericNack.

Messaging Manager does not construct this message.

cancel_ broadcast_sm_ resp

Messaging Manager does not comply.

Code exists to decode the incoming broadcast message but ProtocolHandler::stateBound() does not consider this possibility and returns genericNack.

Messaging Manager does not construct this message.

cancel_sm

Messaging Manager complies.

Messaging Manager does not construct these messages but, if in a bound state, sends on received cancel messages. These messages are sent straight to the outgoing ProtocolHandler, bypassing ACS and xmsTrigger.

cancel_sm_resp

Messaging Manager complies.

- Messaging Manager sends received cancel response messages if in a bound state.
- These messages are sent straight to the outgoing ProtocolHandler, bypassing ACS and xmsTrigger.
- Messaging Manager only constructs these messages to reply to the cancel_sm if it is unable to forward the cancel_sm.

data_sm

Messaging Manager complies.

data_sm_resp

Messaging Manager complies.

deliver_sm

Messaging Manager complies.

deliver_sm_resp

Messaging Manager complies.

enquire_link

Messaging Manager complies.

enquire_link_resp

Messaging Manager complies.

generic_nack

Messaging Manager complies.

outbind

Messaging Manager complies.

query_broadcast_sm

Messaging Manager does not comply.

- Code exists to decode the incoming broadcast message, but ProtocolHandler::stateBound() does not consider this possibility and returns a genericNack.
- Messaging Manager does not construct this type of message.

query_broadcast_sm_resp

Messaging Manager does not comply.

- Code exists to decode the incoming broadcast message, but ProtocolHandler::stateBound() does not consider this possibility and returns a genericNack.
- Messaging Manager does not construct this type of message.

query_sm

Messaging Manager complies.

- Messaging Manager does not construct these messages.
- Messaging Manager relays received query messages if in a bound state. These messages are sent straight to the outgoing ProtocolHandler, bypassing ACS and smsTrigger.

query_sm_resp

Messaging Manager complies.

- Messaging Manager relays received query messages if in a bound state. These messages are sent straight to the outgoing ProtocolHandler, bypassing ACS and smsTrigger.
- Messaging Manager only constructs these messages to reply to a query_sm if it is unable to forward the query_sm.

replace_sm

Messaging Manager complies.

- Messaging Manager does not construct these messages.
- Messaging Manager relays received replace messages if in a bound state. These messages are sent straight to the outgoing ProtocolHandler, bypassing ACS and smsTrigger.

replace_sm_resp

Messaging Manager complies.

- Messaging Manager relays received replace response messages if in a bound state. These messages are sent straight to the outgoing ProtocolHandler, bypassing ACS and smsTrigger.
- Messaging Manager only constructs these messages to reply to a replace_sm if it is unable to forward the replace_sm.

submit_multi

Messaging Manager complies.

submit_multi _ resp

Messaging Manager complies.

submit_sm

Messaging Manager complies.

submit_sm_resp

Messaging Manager complies.

unbind

Messaging Manager complies.

unbind_resp

Messaging Manager complies.

PDU Sequencing (2.6)

The PDU Sequence Number (2.6.1)

Messaging Manager complies.

Messaging Manager follows the recommended practice of monotonically increasing a sequence number that starts at 1.   The number will clock-over at 2^{31}-1.

Why use Monotonically Increasing Sequence numbers? (2.6.2)

Messaging Manager complies.

Sequence Numbers Across Sessions (2.6.3)

Messaging Manager complies.

Each Messaging Manager SMPP connection maintains its own sequence number.

Synchronous Vs. Asynchronous (2.6.4)

Messaging Manager complies.

Why Asynchronous? (2.6.5)

Messaging Manager complies.

Session Timers (2.7)

Relevance

The following compliance statements refer to *Table 2-2 SMPP Session Timers* of *The Specification*.

Session Init Timer

Messaging Manager does not comply.

The timer value is the configured outgoingTimeout value.   However, when a TCP/IP outbind connection has been established and the ESME is waiting for the outbound message, Messaging Manager obtains a timer value from the idleTimeout configuration option.

Enquire Link Timer

Messaging Manager complies.

The timer value is obtained from the heartbeatTimeout value.

Inactivity Timer

Messaging Manager complies.

- The timer value is obtained from the heartbeatTimeout value.

- The timer value expires in all states but the expiry handling code checks the current state before taking action.

Response Timer

Messaging Manager complies.

The timer value is the configured outgoingTimeout value.

Error Handling (2.8)

Handling Connection Failure (2.8.1)

Messaging Manager complies.

- After a failed SMSC connection attempt, the IP plugin tries every 10 seconds to reconnect.
- If an established connection is lost, reconnection attempts are only made if the connection is to an SMSC.
- Section 2.8.1 recommends retry for outbinds, but this is not performed by Messaging Manager itself.

Operation Failure (2.8.2)

The following six statements refer to the bulleted list in *The Standard*.

The PDU is unrecognised

Messaging Manager complies.

If a command ID cannot be determined, a genericNack is returned with ESME_RINVCMDID set.

The PDU is malformed

Messaging Manager complies.

- If a command ID cannot be determined, a genericNack is returned with ESME_RINVCMDID set.
- If the section length of a PDU is the reason for the command ID not being determined, ESME_RINVCMDLEN is returned in a genericNack.
- See also *command_status, error_status_code (4.7.6)* (on page 29).

Invalid Field Length

Messaging Manager does not comply.

The type of message returned is a genericNack with ESME_RINVCMDID.   This is not a response of the appropriate type.

The PDU data is unexpected and deemed invalid

Messaging Manager complies.

Messaging Manager does not need to consider any message data as invalid.

The PDU is not allowed in the current session state

Messaging Manager complies.

Where the received message does not have an appropriate response message type, genericNack is used.

The ESME or MC is restricting the use of certain PDUs or features

Messaging Manager complies.

Messaging Manager does not restrict the use of certain PDUs.

Flow Control and Congestion Avoidance (2.9)

Compliance statement

Messaging Manager does not comply.

- Messages that are unaltered by Messaging Manager and leave via the originating plugin will pass on any encoded congestion_state TLV correctly. However, all Messaging Manager constructed messages, and messages arriving from other plugins will not add a congestion_state TLV to a response.
- To become fully compliant, the GenericSM class:
    - needs to be extended to include a representation of the congestion_state TLV, and
    - needs to populate the TLV from the GenericSM in the outgoing plugin.

  Some method of determining Messaging Manager's own congestion state (and populating GenericSM with it) would also be desirable.

Session Security and Encryption (2.10)

Leased Lines (2.10.1)

Messaging Manager complies.

The privacy of the network where Messaging Manager is deployed is obviously not determined by Messaging Manager itself.

Secure Transport Layer (2.10.2)

Messaging Manager does not comply.

- The SMPP plugin uses cmn::Socket for its connections. cmn::Socket does not support SSL.
- To become compliant:
    - SSL support needs to be added to the socket class, and
    - configuration for the SSL connection needs to be added to the plugin.

Secure VPN (2.10.3)

Messaging Manager complies.

For this type of encryption, there are no demands placed on either the ESME or MC.

Secure Tunnel (2.10.4)

Messaging Manager complies.

For this type of encryption, there are no demands placed on either the ESME or MC.

Forward and Backward Compatibility (2.11)

General

Messaging Manager complies with clause 2.11.

With bind requests, Messaging Manager sets interface_version to either the ASP's version or the version in Messaging Manager's **eserv.config**, whichever is the smaller. Only 0x34 and 0x50 are considered valid values.

Forward Compatibility (2.11.1)

Messaging Manager complies.

If a message leaves by the plugin that received it, unrecognised TLVs are inserted into the outgoing message.

Backward Compatibility (2.11.2)

Messaging Manager does not comply.

- Messaging Manager does not correctly support connections to an ESME or MC that only supports SMPP version 3.3 or earlier.
- In several places Messaging Manager adds TLVs to messages which may be SMPP version 3.3.
- No check on the messageId size is made, so it is possible to send a messageId greater than 8 octets in size.

Chapter 3

Compliance Statements For SMPP Parameter and PDU Format (3)

Overview

Introduction

This chapter states the compliance of Messaging Manager with clauses of Section 3 of *The Specification*.

In this chapter

This chapter contains the following topics.

Parameter Type Definitions (3.1)

SMPP PDU Parameter Types (Table 3-1)

Messaging Manager complies.

NULL Settings (3.1.1)

Messaging Manager complies.

SMPP Parameter Field Size Notation (3.1.2)

Messaging Manager complies.

General PDU Format (3.2)

SMPP PDU Format (Table 3-4)

Messaging Manager complies.

PDU Format (3.2.1)

Messaging Manager complies.

Command_length (3.2.1.1)

Messaging Manager complies.

Command_id (3.2.1.2)

Messaging Manager complies.

Command_status (3.2.1.3)

Messaging Manager complies.

Responses will not include a message body if the command status is non-zero.

Sequence_ number (3.2.1.4)

Messaging Manager complies.

Standard Parameters (3.2.1.5)

Messaging Manager complies.

TLV Parameters (3.2.1.6)

Messaging Manager complies.

Note that for mandatory TLVs, Messaging Manager expects the order to be the same as that specified in *The Standard*.

A sample PDU (3.2.2)

Messaging Manager complies.

Chapter 4

Compliance Statements For SMPP PDU Definitions (4)

Overview

Introduction

This chapter states the compliance of Messaging Manager with clauses of Section 4 of *The Specification*.

In this chapter

This chapter contains the following topics.

Session Management Operations (4.1)

General

When specified, no limits are enforced for the parameters described in this section. However limits exist in the Messaging Manager database and the Routing Scheme subsystem which will limit the size of the specific parameter being passed on to the SMPP interface:

- The database and routing scheme limit the password to 50 and 51 characters respectively.
- The database and routingScheme limit the system_id to 15 and 16 characters respectively.

Bind Operation (4.1.1)

Messaging Manager complies.

bind_transmitter Syntax (4.1.1.1)

Messaging Manager complies.

- No limit is placed on the length of a C-Octet string.
- If configured incorrectly, Messaging Manager creates bind operations with passwords longer than nine characters.

bind_transmitter_resp Syntax (4.1.1.2)

Messaging Manager complies.

- The maximum of 16 characters for system_id is not enforced.
- sc_interface_version is not used.

bind_receiver Syntax (4.1.1.3)

Messaging Manager complies.

The maximum of 16 characters for system_id is not enforced.

bind_receiver_ resp Syntax (4.1.1.4)

Messaging Manager complies.

- The maximum of 16 characters for system_id is not enforced.
- sc_interface_version is not used.

bind_transceiver Syntax (4.1.1.5)

Messaging Manager complies.

The maximum number of characters is not enforced for the variable length fields.

bind_transceiver_resp Syntax (4.1.1.6)

Messaging Manager complies.

- The maximum number of characters is not enforced for the variable length fields.
- sc_interface_version is not used.

outbind Syntax (4.1.1.7)

Messaging Manager complies.

The maximum number of characters is not enforced for the variable length fields.

unbind Syntax (4.1.1.8)

Messaging Manager complies.

unbind_resp Syntax (4.1.1.9)

Messaging Manager complies.

Enquire Link Operation (4.1.2)

Messaging Manager complies.

Messaging Manager takes any message type as a valid response.

enquire_link Syntax (4.1.2.1)

Messaging Manager complies.

enquire_link_resp Syntax (4.1.2.2)

Messaging Manager complies.

Alert Notification Operation (4.1.3)

Messaging Manager complies.

Messaging Manager does not construct these messages. It forwards them from the sender to the receiver.

alert_notification Syntax (4.1.3.1)

Messaging Manager complies.

Messaging Manager can decode and encode these messages correctly, but does not create them. In its current form, if Messaging Manager were to create an alert_notification message, the Address size maximum would not be enforced.

Generic NACK Operation (4.1.4)

Messaging Manager complies.

generic_nack Syntax (4.1.4.1)

Messaging Manager complies.

Message Submission Operations (4.2)

submit_sm Syntax (4.2.1.1)

Messaging Manager complies.

- Messaging Manager does not check for exceeding the maximum length of the variable length fields.
- The SMPP submit_sm message is stored as a GenericSM object. The GenericSM class is sub-classed from the GenericMessage parent class with message type set to Submit.
- The mapping from SMPP to GenericSM is described for each parameter as follows.
  - **service_type** is set to null for outgoing messages. For incoming messages service_type is used to set the teleservice and allowAlternateDelivery via the teleserviceRoutingMap.
  - **source_addr_ton, source_addr_npi, source_addr** is stored as the GenericMessage::OriginatingAddress in GenericSM.
  - **dest_addr_ton, dest_addr_npi, dest_addr** is stored as the GenericMessage::DestinationAddress in GenericSM.
  - **esm_class** is not stored as one field in submit_sm, but individual bits are set/read from many fields. The esm_class is stored in multiple fields in the GenericMessage/GenericSM:

SMPP esm_class bits	**GenericSM fields**
0x3c (Message Type: Bits 2-5)	Determines how the message type is set. See *MC Delivery Receipt (4.3.5.1)* (on page 24), *Intermediate Notification (4.3.5.2)* (on page 24), *SME Delivery Acknowledgement (4.3.5.3)* (on page 25), *SME Manual/User Acknowledgement (4.3.5.4)* (on page 25) and *Conversation Abort (4.3.5.5)* (on page 25) for information on delivery receipt handling.
0x40 (GSM Specific: UDHI Bit)	Used to determine the presence of a userDataHeader (GenericSM:: userDataHeaderPresent).

SMPP esm_class bits	GenericSM fields
0x03 (Messaging Mode: Bits 1-0)	GenericSM::singleShot (0x01 => true, all others => false).
0x80 (GSM Specific: Reply Path Bit)	ProvideReplyPath (direct copy) GenericMessage::allowAlternateDelivery (false for non zero).

- **protocol_id** is stored in GenericSM::protocolIdentifier.
- **priority_flag** is stored in GenericMessage::priorityIndicator (0 => PriorityNormal, 1=> PriorityInteractive).
- **schedule_delivery_time** is not stored in GenericSM. Therefore only maintained if the message exits Messaging Manager via the incoming plugin and is not modified.
- **validity_period** qos_time_to_live is ignored by Messaging Manager when creating GenericSM. The validity period is converted to the GenericSM::ValidityPeriod class and stored in GenericSM::validityPeriod.
- **registered_delivery** is stored in GenericSM::statusReportRequest for non-deliver_sm messages. deliver_sm requests with registered_delivery are ignored and not stored in GenericSM::statusReportRequest. Messaging Manager outbound messages have registeredDelivery updated to reflect the statusReportRequest field of GenericSM.

GenericSM::statusReportRequest	Effect on registeredDelivery
xmsRequested, bothRequested	Set bit 0 to 1 and bit 1 to 0.
SmeRequested	Set to 1 if registeredDelivery is 0.
NotRequired	Set to 0.

- **replace_if_present_flag** is not stored in GenericSM.
- **data_coding** is generally stored in GenericSM as GenericSM::desiredAlphabet and also in several other fields for GSM data coding values. DataCodingElement structure is used as an intermediary in the mapping of data_coding to and from GenericSM parameters. The SMPP data_coding value is mapped and stored in one or more of the following parameters of the GenericSM:
  - desiredAlphabet
  - messageClass - only for GSM MC data_coding values.
  - mwg (MessageWaitingGroup) - only for GSM MWI data_coding values.
  - mwi (messageWaitingIndicator) - only for GSM MWI data_coding values.
  - mwt (MessageWaitingType) - only for GSM MWI data_coding values.

  See *data_coding (4.7.7)* (on page 31) for additional details.
- **sm_default_msg_id** is not stored in GenericSM.
- **sm_length** is not stored in GenericSM. Instead the field is generated on outgoing messages from the userData.
- **short_message** The message_payload TLV is read by Messaging Manager with priority over the short_message field. If no message_payload is present, the short_message field is stored in GenericSM::userData. Outgoing messages have short_message set to the userData (with no message_payload TLV present) if less than 255 in size, otherwise it is set in the message_payload TLV.
- **message_submission TLVs** are considered in section *Message Submission Request TLVs (4.2.4)* (on page 19).

submit_sm_resp Syntax (4.2.1.2)

Messaging Manager complies.

- Messaging Manager does not check for exceeding the maximum length of the variable length fields.
- Messaging Manager stores the message as a GenericSMResult.
- **message_id** is stored as GenericSMResult::deliverReceiptId.

- **message_submission response TLVs** are considered in section *Message Submission Response TLVs (4.2.5)* (on page 21).

data_sm Syntax (4.2.2.1)

Messaging Manager complies.

- Messaging Manager does not check for exceeding the maximum length of the variable length fields. The fields are used in the same way as submit_sm to construct an Messaging Manager GenericSM.
- The GenericMessage message type is set to MT_Submit if the message comes from an SME, or MT_Deliver if the message comes from an SMSC.

data_sm_resp Syntax (4.2.2.2)

Messaging Manager complies.

- Messaging Manager does not check for exceeding the maximum length of the variable length fields.
- The fields are used in the same way as submit_sm_resp to construct an Messaging Manager GenericSMResult.

submit_multi Syntax (4.2.3.1)

Messaging Manager complies.

- submit_multi messages are processed internally in Messaging Manager by creating   a GenericSM for each terminating address. The handling of each field is thus the same as for submit_sm, with the exception of destinationAddress.
- The GenericMessage message type is set to Submit.
- Distribution Lists are recognised but not supported.   ESME_RCNTSUBDL, "Cannot Submit to Distribution List", is returned.
- Although individual GenericSM components may be modified by Messaging Manager, these changes are not incorporated into the submit_multi forwarded to the SMSC, as any choice would be arbitrary. The forwarded submit_multi is derived from the originating message.

submit_multi_ resp Syntax (4.2.3.2)

Messaging Manager complies.

Messaging Manager does not check for exceeding the maximum length of the variable length fields. The fields are used in the same way as submit_sm_resp to construct a Messaging Manager GenericSMResult.   The GenericSMResult has no knowledge of the unsuccess_sme structure, as it deals with only a single message.   The SMPP plugin does, however, create the unsuccess_sme from the individual submit_sm_resps.

Message Submission Request TLVs (4.2.4)

Messaging Manager does not comply.

The following table sets out the way Messaging Manager manages each of the TLVs listed in Table 4-20 of *The Specification*.   A TLV stated as being ignored by Messaging Manager can still be passed on if the message is unchanged and the message uses the same outgoing and incoming plugin.

TLV Name	Messaging Manager treatment
alert_on_msg_delivery	Ignored
billing_identification	Ignored

TLV Name	Messaging Manager treatment
callback_num	Ignored
callback_num_atag	Ignored
callback_num_pres_ind	Ignored
dest_addr_np_country	Ignored
dest_addr_np_information	Ignored
dest_addr_np_resolution	Ignored
dest_addr_subunit	Stored in GenericSM::messageClass. Present in outgoing messages if value is not GenericSM::MessageClassNone.
dest_bearer_type	Ignored
dest_network_id	Ignored
dest_network_type	Stored in GenericMessage::messageProtocol.
dest_node_id	Ignored
dest_subaddress	Ignored
dest_telematics_id	Ignored
dest_port	Ignored
display_time	Ignored
its_reply_type	Ignored
its_session_info	Ignored
language_indicator	Ignored
message_payload	Used to create the GenericSM::userData. Present on outgoing messages when userData > 255 characters.
more_messages_to_send	Ignored
ms_msg_wait_facilities	Stored in GenericSM::mwt (MessageWaitingType) and GenericSM::mwi (MessageWaitingIndicator).
ms_validity	Stored in GenericSM::mwg (MessageWaitingGroup). Only value 0 (Store Indefinitely) will be correctly saved. Other values will be treated as GenericSM::MessageWaitingGroupDiscard. Outgoing messages will contain values 0 (Store Indefinitely) or 3 (Display Only) only.
number_of_messages	Ignored
payload_type	Ignored
privacy_indicator	Ignored
qos_time_to_live	Ignored
sar_msg_ref_num	Stored in GenericSM::segmentReference. Outgoing messages will either have this reference in the userDataHeader or this TLV (depending on if the message was modified by MMX).
sar_segment_seqnum	Stored in GenericSM::segmentNumber. Outgoing messages will either have this reference in the userDataHeader or this TLV (depending on if the message was modified by Messaging Manager).
sar_total_segments	Stored in GenericSM::segmentCount. Outgoing messages will either have this reference in the userDataHeader or this TLV (depending on if the message was modified by Messaging Manager).
set_dpf	Ignored
sms_signal	Ignored

TLV Name	Messaging Manager treatment
source_addr_subunit	Ignored
source_bearer_type	Ignored
source_network_id	Used to set the GenericSM::sourceLocationInformation, which is triggered to ACS as the location number.
source_network_type	Ignored
source_node_id	Ignored
source_port	Ignored
source_subaddress	Ignored
source_telematics_id	Ignored
user_message_reference	Stored in GenericSM::messageReference. Present in outgoing messages if value grater than zero.
user_response_code	Ignored
ussd_service_op	Ignored

Message Submission Response TLVs (4.2.5)

Messaging Manager does not comply.

A TLV stated as being ignored by Messaging Manager can still be passed on if the message is unchanged and the message uses the same outgoing and incoming plugin.

The following table sets out the way Messaging Manager manages each of the TLVs listed in Table 4-21 of *The Specification*.

TLV Name	Messaging Manager treatment
additional_status_info_text	Ignored
delivery_failure_reason	Ignored
dpf_result	Ignored
network_error_code	Ignored

Source and Destination Addressing (4.2.6)

Messaging Manager does not comply.

- Messaging Manager does not consider that the source_addr may be NULL. The originating address is populated by the source fields, regardless of their values.
- Messaging Manager complies partially in that it understands the TON, NPI and address fields of a mobile number.

International and National Format (4.2.6.1.1)

Messaging Manager complies.

Alphanumeric Format (4.2.6.1.2)

Messaging Manager does not comply.

- The AMC part of Messaging Manager does not comply because ACS is triggered by BcdDigits which cannot handle alphabetical characters.

- Messaging Manager complies if a message does not trigger a call-plan.

NPI (4.2.6.2)

Messaging Manager complies.

Messaging Manager also handles an NPI of 13 to represent PC:SSN. This value should only be used for SCCP level addresses.

ESME Addresses (4.2.6.3)

Compliance statements are made under the following headings:

- Service Short Code
- International Number, and
- NULL Address.

These headings correspond to the bullet list in clause 4.2.6.3 of *The Specification*.

Service Short Code

Messaging Manager complies.

International Number

Messaging Manager complies.

NULL Address

Messaging Manager does not comply.

Messaging Manager cannot substitute a default source address into the GenericSM. A non-NULL address is required for a delivery receipt to be sent. This could be implemented in the future with a simple change to the originating plugin.

Message Replace operation in submit_sm (4.2.7)

Messaging Manager does not comply.

The replace_if_present flag is ignored by Messaging Manager, and not placed in outgoing messages (unless the message is unaltered and goes out the incoming plugin). The service_type field is also not preserved by Messaging Manager. replace_sm messages are forwarded on, so this is the only way to send a message replace through Messaging Manager.

Message Length (4.2.8)

Messaging Manager complies.

Messaging Manager can handle up to 255 characters in short_message. Messages that are too long will be placed in the message_payload TLV. Messaging Manager does not consider the possibility of the MC only having space for 140 octets (that is, the 255 limit is hard-coded).

Registered (4.2.9.1)

Messaging Manager does not comply.

See *submit_sm Syntax (4.2.1.1)* (on page 17).

Scheduled (4.2.9.2)

Messaging Manager does not comply.

The scheduled_delivery_value is ignored in constructing a GenericSM. However, Messaging Manager is capable of detecting the presence of the scheduled_delivery_value and such messages will be FDA-barred internally so that these messages, where appropriate, will be sent to an alternative Message Centre for proper handling at the scheduled delivery time.

Pre-defined (4.2.9.3)

Messaging Manager does not comply.

The sm_default_msg_id is not stored in the GenericSM class. Messaging Manager does not examine this value on incoming messages either.

Message Modes (4.2.10)

Messaging Manager does not comply.

The esm_class value is not directly stored in GenericSM. See the following four compliance statements for more detail.

Default Message Mode (4.2.10.1)

Messaging Manager complies.

If the incoming message has bits 0 and 1 set to zero, so will the outgoing message.

Store and Forward Message Mode (4.2.10.2)

Messaging Manager does not comply.

This part of the esm_class is reconstructed as 00 (default message mode) if the outgoing message is changed and is not singleShot.

Datagram Message Mode (4.2.10.3)

Messaging Manager complies.

The singleShot variable in GenericSM correctly captures this behaviour. Note that the delivery receipt may still be requested via the registered_delivery field.

Transaction Message Mode (4.2.10.4)

Messaging Manager does not comply.

Messaging Manager does not set bit 1 to 1 for an altered message. Note that an SMPP Transaction mode message which came into Messaging Manager has allowAlternateDelivery set to false, so we will go out the same plugin, and the outgoing message will be Transaction mode, provided the singleShot nature was not changed by Messaging Manager.

Message Delivery Operations (4.3)

deliver_sm Syntax (4.3.1.1)

Messaging Manager complies.

- A deliver_sm is handled similarly to a submit_sm. See *submit_sm Syntax (4.2.1.1)* (on page 17) for detailed handling of each message tag.
- genericMessage's message type is set to Deliver unless the deliver_sm contains a delivery receipt. See *MC Delivery Receipt (4.3.5.1)* on this page, *Intermediate Notification (4.3.5.2)* on this page, *SME Delivery Acknowledgement (4.3.5.3)* (on page 25), *SME Manual/User Acknowledgement (4.3.5.4)* (on page 25) and *Conversation Abort (4.3.5.5)* (on page 25).

deliver_sm_resp Syntax (4.3.1.2)

Messaging Manager complies.

- A deliver_sm is handled similarly to a submit_sm. The main difference is that the GenericMessage::messageType is changed to MT_Notify for a Status Report.
- See *submit_sm_resp Syntax (4.2.1.2)* (on page 18) for a detailed description of the handling of each message tag.

data_sm Operation (4.3.2)

Messaging Manager complies.

See *data_sm Syntax (4.2.2.1)* (on page 19) and *data_sm_resp Syntax (4.2.2.2)* (on page 19) for more information.

Message Delivery Request TLVs (4.3.3)

TLVs not covered in *Message Submission Request TLVs (4.2.4)* (on page 19) and *Message Submission Response TLVs (4.2.5)* (on page 21) are listed in the following table.

TLV Name	Messaging Manager treatment
message_state	Stored in GenericSM::deliverySucceeded (as equal to DELIVERED or not). This field is set to false if the TLV is not present.
receipted_message_id	Stored in GenericSM::deliveryReceiptId. This field is set to blank if the TLV is not present.

Message Delivery Response TLVs (4.3.4)

For compliance statements see *Message Submission Response TLVs (4.2.5)* (on page 21).

MC Delivery Receipt (4.3.5.1)

Messaging Manager complies.

- message_state and receipted_message_id are observed, but the network_error_id field is ignored by Messaging Manager.
- The GenericMessage's message type is set to Notify and message contents set to Delivery Receipt.

Intermediate Notification (4.3.5.2)

Messaging Manager complies.

- If the MC passes Messaging Manager one of these messages, it will pass it on (setting allowAlternateDelivery to false). Of the fields listed as important, only network_error_id is ignored by Messaging Manager.
- The GenericMessage's message type is set to Notify and message contents set to Delivery Receipt.

SME Delivery Acknowledge-ment (4.3.5.3)

Messaging Manager complies.

- If the MC passes Messaging Manager one of these messages, it will pass it on (setting allowAlternateDelivery to false).
- The GenericMessage's message type is set to Notify and message contents set to Delivery Receipt.

SME Manual/User Acknowledge-ment (4.3.5.4)

Messaging Manager complies.

If the MC passes Messaging Manager one of these messages, it will pass it on (setting allowAlternateDelivery to false).

The GenericMessage's message type is set to Notify and message contents set to Delivery Receipt.

Conversation Abort (4.3.5.5)

Messaging Manager complies.

If the MC passes Messaging Manager one of these messages, Messaging Manager will pass it on.

Message Broadcast Operations (4.4)

broadcast_sm Operation (4.4.1)

Messaging Manager does not comply.

Messaging Manager responds to broadcast_sm messages with a genericNack.

broadcast_sm Syntax (4.4.1.1)

Messaging Manager does not comply.

- Messaging Manager does not attempt to construct a GenericMessage.   Messaging Manager does not attempt to handle individual fields or interpreted them.
- Messaging Manager can construct a fully compliant internal representation of a broadcast_sm, but it cannot translate this object to GenericMessage.

broadcast_sm_ resp Syntax (4.4.1.2)

Messaging Manager does not comply.

See *broadcast_sm Operation (4.4.1)*, above.

Broadcast Request Optional TLVs (4.4.2)

Messaging Manager does not comply.

See *broadcast_sm Operation (4.4.1)*, above.

Broadcast Response Optional TLVs (4.4.3)

Messaging Manager does not comply.

See *broadcast_sm Operation (4.4.1)*, above.

Message Replacement with broadcast_sm (4.4.4)

Messaging Manager does not comply.

See *broadcast_sm Operation (4.4.1)*, above.

Ancillary Submission Operations (4.5)

cancel_sm Operation (4.5.1)

Messaging Manager complies.

cancel_sm operations are sent straight to the outgoing protocolHandler, never entering xmsTrigger or ACS. Thus, the message passed on is an exact copy of the incoming message.

cancel_sm Syntax (4.5.1.1)

Messaging Manager complies.

The incoming message is copied to the outgoing message, so Messaging Manager is compliant, assuming that the message originator is compliant.

cancel_sm_resp Syntax (4.5.1.2)

Messaging Manager complies.

See *cancel_sm Syntax (4.5.1.1)* (on page 26).

query_sm Operation (4.5.2)

Messaging Manager complies.

query_sm operations are sent straight to the outgoing protocolHandler, never entering xmsTrigger or ACS. Thus, the message passed on is an exact copy of the incoming message.

query_sm Syntax (4.5.2.1)

Messaging Manager complies.

The incoming message is copied to the outgoing message, so Messaging Manager is compliant, assuming that the message originator is compliant.

query_sm_resp Syntax (4.5.2.2)

Messaging Manager complies.

See *query_sm Syntax (4.5.2.1)* (on page 26).

replace_sm Operation (4.5.3)

Messaging Manager complies.

replace_sm operations are sent straight to the outgoing protocolHandler, never entering xmsTrigger or ACS. Thus, the message passed on is an exact copy of the incoming message.

replace_sm Syntax (4.5.3.1)

Messaging Manager complies.

The incoming message is copied to the outgoing message, so Messaging Manager is compliant, assuming that the message originator is compliant.

replace_sm_resp Syntax (4.5.3.2)

Messaging Manager complies.

See *replace_sm Syntax (4.5.3.1)* (on page 26).

Message Replacement TLVs (4.5.3.3)

Messaging Manager complies.

See *replace_sm Syntax (4.5.3.1)* (on page 26).

Ancillary Broadcast Operations (4.6)

query_broadcast_sm Operation (4.6.1)

Messaging Manager does not comply.

Messaging Manager responds to query_broadcast_sm messages with a genericNack.

query_broadcast_sm Syntax (4.6.1.1)

Messaging Manager does not comply.

- Messaging Manager does not attempt to construct a GenericMessage. Messaging Manager does not attempt to handle individual fields nor does it interpret them.
- Messaging Manager can construct a fully compliant internal representation of a query_broadcast_sm, but it cannot translate the representation to a GenericMessage.

Query Broadcast Request Optional TLVs (4.6.1.2)

Messaging Manager does not comply.

See *query_broadcast_sm Operation (4.6.1)* in this topic.

query_broadcast_sm_resp Syntax (4.6.1.3)

Messaging Manager does not comply.

See *query_broadcast_sm Operation (4.6.1)* in this topic.

Query Broadcast Response Optional TLVs (4.6.1.4)

Messaging Manager does not comply.

See *query_broadcast_sm Operation (4.6.1)* in this topic.

cancel_ broadcast_sm Operation (4.6.2)

Messaging Manager does not comply.

- Messaging Manager does not attempt to construct a GenericMessage. Messaging Manager does not attempt to handle individual fields nor does it interpret them.

- Messaging Manager can construct a fully compliant internal representation of a cancel_broadcast_sm, but it cannot translate the representation to a GenericMessage.

cancel_ broadcast_sm Syntax (4.6.2.1)

Messaging Manager does not comply.

See *cancel_broadcast_sm Operation (4.6.2)* in this topic.

Cancel Broadcast Optional TLVs (4.6.2.2)

Messaging Manager does not comply.

See *cancel_broadcast_sm Operation (4.6.2)* in this topic.

cancel_ broadcast_sm_ resp Syntax (4.6.2.3)

Messaging Manager does not comply.

See *cancel_broadcast_sm Operation (4.6.2)* in this topic.

PDU Field Definitions (4.7)

addr_ton, source_addr_ton, dest_addr_ton, esme_addr_ton (4.7.1)

Messaging Manager complies.

addr_npi, source_addr_npi, dest_addr_npi, esme_addr_npi (4.7.2)

Messaging Manager does not comply.

The values for Internet (IP) and WAP Client Id are not considered. Messaging Manager also has an extra value of 13 for point codes.

address_range (4.7.3)

Messaging Manager does not comply.

address_range is always NULL in messages constructed by Messaging Manager. The value of the field is ignored in interpreting messages received by Messaging Manager.

UNIX Regular Expressions (4.7.3.1)

Messaging Manager does not comply.

Not relevant as the address_range is never used by Messaging Manager. See *address_range (4.7.3)* (on page 28).

command_length (4.7.4)

Messaging Manager complies.

command_id (4.7.5)

Messaging Manager complies.

command_status, error_status_ code (4.7.6)

Messaging Manager complies.

- All values are correct.
- Errors not explicitly mentioned as transientFailures are treated as permanentFailures.
- If a message is throttled by Messaging Manager, a status code determined by the throttledCommandStatus configuration parameter will be returned. This defaults to ESME_RTHROTTLED.
- "Not used" implies that incoming messages are not checked for the associated error.
- For the Reject action, Messaging Manager can be configured to return any SMPP error_code. To do this Messaging Manager uses a configured mapping from ACS CS1ReleaseCause. If configured by the user, all cause codes are treated as "Compliant in outbound direction".

Default "Not Used" cause code values are shown in the following table.

Command status name	Usage compliance
ESME_ROK	Messaging Manager complies. Treated as GenericSMResult::resultSuccess.
ESME_RINVMSGLEN	Messaging Manager complies.
ESME_RINVCMDLEN	Messaging Manager complies.
ESME_RINVCMDID	Messaging Manager complies.
ESME_RINVBNDSTS	Messaging Manager complies. Receipt is treated as a GenericSMResult::resultTransientFailure.
ESME_RALYBND	Messaging Manager complies.
ESME_RINVPRTFLG	Not used.
ESME_RINVREGDLVFLG	Not used.
ESME_RSYSERR	Messaging Manager complies. • In the message outbound direction, receipt is treated as a GenericSMResult::resultTransient-Failure. • In the message inbound direction, an abort result type (GenericSMResult::resultAbort) will cause this code to be sent.
ESME_RINVSRCADR	Not used
ESME_RINVDSTADR	Not used
ESME_RINVMSGID	Not used
ESME_RBINDFAIL	Messaging Manager complies.
ESME_RINVPASWD	Not used
ESME_RINVSYSID	Not used
ESME_RCANCELFAIL	Messaging Manager complies.
ESME_RREPLACEFAIL	Messaging Manager complies.
ESME_RMSGQFUL	Messaging Manager complies. • Messaging Manager is configured with maxConcurrentTransactions. When this is exceeded, MSGQFUL is replied to the sender. • Receipt of a MSGQFUL is treated as an GenericSMResult::resultTransient-Failure

Command status name	Usage compliance
ESME_RINVSERTYP	Not used
ESME_RINVNUMDESTS	Not used
ESME_RINVDLNAME	Not used
ESME_RINVDESTFLAG	Not used
ESME_RINVSUBREP	Not used
ESME_RINVESMCLASS	Not used
ESME_RCNTSUBDL	Messaging Manager complies.
ESME_RSUBMITFAIL	Not used
ESME_RINVSRCTON	Not used
ESME_RINVSRCNPI	Not used
ESME_RINVDSTTON	Not used
ESME_RINVDSTNPI	Not used
ESME_RINVSYSTYP	Not used
ESME_RINVREPFLAG	Not used
ESME_RINVNUMMSGS	Not used
ESME_RTHROTTLED	Messaging Manager complies. The throttling response code can be changed via the **eserv.config** default smpp parameter 'throttledCommandStatus'.
ESME_RINVSCHED	Not used
ESME_RINVEXPIRY	Not used
ESME_RINVDFTMSGID	Not used
ESME_RX_T_APPN	Treated as a GenericSMResult::resultTransient-Failure. TransientFailures are mapped to this value. Also used for duplicate sequence numbers and failure to send a message to transaction or to construct a transaction object.
ESME_RX_P_APPN	PermanentFailures are mapped to this.
ESME_RX_R_APPN	Treated as a GenericSMResult::resultTransientFailure.
ESME_RQUERYFAIL	Messaging Manager complies.
ESME_RINVTLVSTREAM	Not used
ESME_RTLVNOTALLWD	Not used
ESME_RINVTLVLEN	Not used
ESME_RMISSINGTLV	Messaging Manager complies.
ESME_RINVTLVVAL	Not used
ESME_RDELIVERYFAILURE	Not used
ESME_RUNKNOWNERR	Messaging Manager complies. Receipt is treated as a GenericSMResult::resultTransient-Failure.
ESME_RSERTYPUNAUTH	Receipt is treated as a GenericSMResult::resultTransient-Failure.
ESME_RPROHIBITED	Not used.
ESME_RSERTYPUNAVAIL	Not used
ESME_RSERTYPDENIED	Not used
ESME_RINVDCS	Not used

Command status name	Usage compliance
ESME_RINVSRCADDRSUBUNIT	Not used
ESME_RINVDSTADDRSUBUNIT	Not used
ESME_RINVBCASTFREQINT	Not used
ESME_RINVBCASTALIAS_NAME	Not used
ESME_RINVBCASTAREAFMT	Not used
ESME_RINVNUMBCAST_AREAS	Not used
ESME_RINVBCASTCNTTYPE	Not used
ESME_RINVBCASTMSGCLASS	Not used
ESME_RBCASTFAIL	Not used
ESME_RBCASTQUERYFAIL	Not used
ESME_RBCASTCANCELFAIL	Not used
ESME_RINVBCAST_REP	Not used
ESME_RINVBCASTSRVGRP	Not used
ESME_RINVBCASTCHANIND	Not used

data_coding (4.7.7)

Messaging Manager complies.

See *submit_sm Syntax (4.2.1.1)* (on page 17).

- Generally, data_coding is stored in GenericSM::desiredAlphabet, but depending on its value (for GSM MWI and GSM MC values), data_coding may also be stored in:
  - GenericSM::mwi (message waiting indicator)
  - GenericSM::mwg (message waiting group)
  - GenericSM::mwt (message waiting type)
  - GenericSM::messageClass
- Except for GenericSM::desiredAlphabet, the presence of SMPP's optional parameters such as
  - TLV ms_validity,
  - TLV ms_msg_wait_facilities, and
  - TLV dest_addr_subunit

  will override the GenericSM mwi, mwg, mwt and/or messageClass parameters described above.
- In Messaging Manager, data_coding is mapped to and from a dataCodingElement structure. Messaging Manager uses inboundDataCodingMap and outboundDataCodingMap of the SMPP Plugin. The dataCodingElement:
  - is used to populate the GenericSM parameters described above for the inbound case, and
  - is populated from the GenericSM parameters described above for the outbound case.

destination_addr (4.7.8)

Messaging Manager complies.

dest_flag (4.7.9)

Messaging Manager complies.

Messaging Manager does not support distribution lists themselves.

dl_name (4.7.10)

Messaging Manager does not comply.

esme_addr (4.7.11)

Messaging Manager complies.

esm_class (4.7.12)

Messaging Manager does not comply.

- Set Reply Path Bit (Bit 7) is stored in GenericSM::provideReplyPath.
- UDHI Bit (Bit 6) is recognised but not stored in GenericSM. It is used to stop alternate delivery of concatenated messages when no UDHI is present. Messaging Manager will not use esm_class to carry segmentation information if it is carried in TLVs.
- Bit 4 (Conversation Abort and manual/user ack) is ignored by Messaging Manager.
- See *submit_sm Syntax (4.2.1.1)* (on page 17).

interface_version (4.7.13)

Messaging Manager complies.

message_id (4.7.14)

MMC complies.

message_state (4.7.15)

Messaging Manager complies.

Query messages are not interpreted by Messaging Manager, simply passed on, so Messaging Manager does not react to or alter this field.

no_unsuccess (4.7.16)

Messaging Manager complies.

number_of_dests (4.7.17)

Messaging Manager complies.

password (4.7.18)

Messaging Manager complies.

priority_flag (4.7.19)

Messaging Manager complies.

- Stored in GenericMessage:: priorityIndicator.

- Messaging Manager uses the IS-95/ANSI-41 compliant priority mapping. The mapping from these values to ANSI-136 is described below (converting from left to right).

ANSI-136	ANSI-41	ANSI-136
Bulk	Normal	Normal
Normal	Interactive	Urgent
Urgent	Urgent	Urgent
Very urgent	Emergency	Very urgent

protocol_id (4.7.20)

Messaging Manager complies.

Value is stored in GenericSM::protocolIdentifier.

registered_ delivery (4.7.21)

Messaging Manager does not comply.

Bits 0 and 1 are altered. All other bits are left alone. Outgoing messages have the same pattern. Messaging Manager does not compliantly set bit 1. See *submit_sm Syntax (4.2.1.1)* (on page 17).

replace_if_ present_flag (4.7.22)

Messaging Manager does not comply.

Value is not stored in GenericSM, so is essentially ignored by Messaging Manager, unless the message is copied to the outgoing plugin unaltered.

scheduled_ delivery_time (4.7.23.1)

Messaging Manager does not comply.

scheduled_delivery_time is not stored in GenericSM and is ignored by Messaging Manager, unless scheduled_delivery_time is copied to the outgoing plugin unaltered. However, for cases where FDA may be relevant for the message, Messaging Manager recognises the presence of scheduled_delivery_time, bypasses FDA and passes scheduled_delivery_time to the SMSC for proper handling at the correct scheduled delivery time.

validity_period (4.7.23.2)

Messaging Manager complies.

final_date (4.7.23.3)

Messaging Manager complies.

Only used for queries not deciphered by Messaging Manager.

Absolute Time Format (4.7.23.4)

Messaging Manager does not comply.

- Tens of seconds are ignored by Messaging Manager.

- Messaging Manager stores absolute time internally as seconds since midnight UTC on 1 January 1970.

Relative Time Format (4.7.23.5)

Messaging Manager complies.

Messaging Manager stores relative time internally as a number of seconds to offset.

sequence_ number (4.7.24)

Messaging Manager complies.

service_type (4.7.25)

Messaging Manager does not comply.

- For outgoing messages that are altered or generated, service_type is set to null.
- For incoming messages:
  - service_type is used to set the teleservice and allowAlternateDelivery via the teleserviceRoutingMap.
  - Implicit association of a function from a service type such as "replace if present" is not supported.

short_message (4.7.26)

Messaging Manager complies.

sm_default_msg_id (4.7.27)

Messaging Manager does not comply.

sm_default_msg_id is not stored in GenericSM. Messaging Manager ignores sm_default_msg_id unless it is copied, unmodified, to the outgoing plugin.

sm_length (4.7.28)

Messaging Manager complies.

This value is not stored internally in Messaging Manager, but calculated from the current message length. The value is correctly set to 0 if a message_payload TLV is being used.

source_addr (4.7.29)

Messaging Manager complies.

Stored in the originatingAddress field of GenericMessage. A value of NULL is not supported by Messaging Manager.

system_id (4.7.30)

Messaging Manager complies.

system_type (4.7.31)

Messaging Manager complies.

Set from the configuration option "systemType".

PDU TLV Definitions (4.8)

Position of TLVs in SMPP messages (4.8)

Messaging Manager complies.

TLV Tag (4.8.1)

Messaging Manager complies.

Messaging Manager uses some TLV tag values internally, mainly for holding values from the EmiProtocolHandler.   These TLV values are listed in the following table.

Tag Name	Tag Value
vmsc_address	0x3680
num_septets	0x3681
tdma_priority	0x3682
message_modified	0x3683

TLV Length (4.8.2)

Messaging Manager complies.

TLV Value (4.8.3)

Messaging Manager complies.

TLV Definitions (4.8.4)

Messaging Manager does not comply.

Where a TLV definition is stated as "ignored" by Messaging Manager, the TLV will only be compliant if the message exits via the originating plugin. In this case unaltered TLVs will be preserved into the outgoing message.

additional_ status_info_text (4.8.4.1)

Ignored

alert_on_ message_delivery (4.8.4.2)

Ignored

billing_ identification (4.8.4.3)

Ignored

broadcast_area_ identifier, failed_broadcast_area_identifier (4.8.4.4)

Ignored

Broadcast Area Format types (4.8.4.4.1)

Ignored

broadcast_area_ success (4.8.4.5)

Ignored

broadcast_ content_type_info (4.8.4.6)

Ignored

broadcast_ channel_indicator (4.8.4.7)

Ignored

broadcast_ content_type (4.8.4.8)

Ignored

broadcast_end_ time (4.8.4.9)

Ignored

broadcast_error_status (4.8.4.10)

Ignored

broadcast_ frequency_ interval (4.8.4.11)

Ignored

broadcast_ message_class (4.8.4.12)

Ignored

broadcast_rep_ num (4.8.4.13)

Ignored

broadcast_ service_group (4.8.4.14)

Ignored

callback_num (4.8.4.15)

Ignored

callback_num_ atag (4.8.4.16)

Ignored

callback_num_ pres_ind (4.8.4.17)

Ignored

congestion_state (4.8.4.18)

Ignored

delivery_failure_ reason (4.8.4.19)

Ignored

dest_addr_np_ country (4.8.4.20)

Ignored

dest_addr_np_ information (4.8.4.21)

Ignored

dest_addr_np_ resolution (4.8.4.22)

Ignored

dest_addr_ subunit (4.8.4.23)

Messaging Manager complies.

Stored in GenericSM::messageClass.

dest_bearer_type (4.8.4.24)

Ignored

dest_network_id (4.8.4.25)

Ignored

dest_network_ type (4.8.4.26)

Messaging Manager does not comply.

Stored in GenericMessage::messageProtocol. Only the following values will be stored:

- 0x02 - ANSI-136/TDMA
- 0x03 - IS-95/CDMA
- Other values treated as GenericSM::UNKNOWN message protocol.

dest_node_id (4.8.4.27)

Ignored

dest_subaddress (4.8.4.28)

Ignored

dest_telematics_ id (4.8.4.29)

Ignored

dest_port (4.8.4.30)

Ignored

display_time (4.8.4.31)

Ignored

dpf_result (4.8.4.32)

Ignored

its_reply_type (4.8.4.33)

Ignored

its_session_info (4.8.4.34)

Ignored

language_indicator (4.8.4.35)

Ignored

message_payload (4.8.4.36)

Messaging Manager complies.

Only used if message_size exceeds 255 characters.

message_state (4.8.4.37)

Messaging Manager does not comply.

Stored (as a bool) in GenericSM::deliverySucceeded. This field is set to true if the state is DELIVERED, and false for all other values or if the TLV is not present. Outgoing messages originating from other protocols will have a value of UNKNOWN.

For delivery receipts, the message_state may be set to DELIVERED or UNDELIVERED depending on the value of GenericSM::deliverySucceeded.

more_messages_to_send (4.8.4.38)

Ignored

ms_availability_ status (4.8.4.39)

Ignored

ms_msg_wait_ facilities (4.8.4.40)

Messaging Manager complies.

Stored in GenericSM::mwi (MessageWaitingIndicator) and GenericSM::mwt (MessageWaitingType).

ms_validity (4.8.4.41)

Messaging Manager does not comply.

- Stored in GenericSM:mwg (messageWaitingGroup).
- Messaging Manager complies for value 0 (Store Indefinitely).
- Value 3 (Display Only) used only for outbound messages.

network_error_ code (4.8.4.42)

Ignored

number_of_ messages (4.8.4.43)

Ignored

payload_type (4.8.4.44)

Ignored

privacy_indicator (4.8.4.45)

Ignored

qos_time_to_live (4.8.4.46)

Ignored

receipted_ message_id (4.8.4.47)

Messaging Manager complies.

Stored in GenericSM::deliveryReceiptId.   Set to blank if not present.

sar_msg_ref_num (4.8.4.48)

Messaging Manager complies.

Stored in GenericSM::segmentReference.

sar_segment_ seqnum (4.8.4.49)

Messaging Manager complies.

Stored in GenericSM::segmentNumber.

sar_total_ segments (4.8.4.50)

Messaging Manager complies.

Stored in GenericSM::segmentCount.

sc_interface_ version (4.8.4.51)

Ignored

set_dpf (4.8.4.52)

Ignored

sms_signal (4.8.4.53)

Ignored

source_addr_subunit (4.8.4.54)

Ignored

source_bearer_ type (4.8.4.55)

Ignored

source_network_ id (4.8.4.56)

Messaging Manager complies.

- Stored in GenericSM::sourceLocationInformation, and triggered to ACS as location number.
- Only compatible with ESME Operator encoding.

source_network_ type (4.8.4.57)

Ignored

source_node_id (4.8.4.58)

Ignored

source_port (4.8.4.59)

Ignored

source_ subaddress (4.8.4.60)

Ignored

source_ telematics_id (4.8.4.61)

Ignored

user_message_ reference (4.8.4.62)

Messaging Manager complies.

Stored in GenericSM::messageReference.   Placed in outgoing messages if messageReference is greater than zero.

user_response_ code (4.8.4.63)

Ignored

ussd_service_ op (4.8.4.64)

Ignored.