

**Oracle® Communications**  
**Network Charging and Control**

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

Release 6.0.1

April 2017

# Copyright

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

|  |           |
|--|-----------|
| About This Document .....  | v         |
| Document Conventions .....   | vi        |
| <b>Chapter 1</b>   |           |
| <b>Messaging Manager and SMPP Document Versions .....</b>                | <b>1</b>  |
| Overview .....   | 1         |
| Messaging Manager .....  | 1         |
| SMPP .....   | 2         |
| <b>Chapter 2</b>   |           |
| <b>Compliance Statements For SMPP Sessions (2) .....</b>                 | <b>3</b>  |
| Overview .....   | 3         |
| References to The Specification .....                                    | 3         |
| Specification Clauses 2.1 and 2.2 .....                                  | 3         |
| Session States (2.3) .....   | 4         |
| Operation Matrix (2.4) .....   | 4         |
| PDU Sequencing (2.6) .....   | 8         |
| Session Timers (2.7) .....   | 8         |
| Error Handling (2.8) .....   | 9         |
| Flow Control and Congestion Avoidance (2.9) .....                        | 10        |
| Session Security and Encryption (2.10) .....                             | 10        |
| Forward and Backward Compatibility (2.11) .....                          | 11        |
| <b>Chapter 3</b>   |           |
| <b>Compliance Statements For SMPP Parameter and PDU Format (3) .....</b> | <b>13</b> |
| Overview .....   | 13        |
| Parameter Type Definitions (3.1) .....                                   | 13        |
| General PDU Format (3.2) .....   | 13        |
| <b>Chapter 4</b>   |           |
| <b>Compliance Statements For SMPP PDU Definitions (4) .....</b>          | <b>15</b> |
| Overview .....   | 15        |
| Session Management Operations (4.1) .....                                | 15        |
| Message Submission Operations (4.2) .....                                | 17        |
| Message Delivery Operations (4.3) .....                                  | 23        |
| Message Broadcast Operations (4.4) .....                                 | 25        |
| Ancillary Submission Operations (4.5) .....                              | 26        |
| Ancillary Broadcast Operations (4.6) .....                               | 27        |
| PDU Field Definitions (4.7) .....  | 28        |
| PDU TLV Definitions (4.8) .....  | 35        |
| <b>Glossary of Terms .....</b>   | <b>43</b> |
| <b>Index .....</b>   | <b>47</b> |



# 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  |
|--------------------------------|--|
| <b>Special Bold</b>            | Items you must select, such as names of tabs.<br>Names of database tables and fields.  |
| <i>Italics</i>                 | Name of a document, chapter, topic or other publication.<br>Emphasis within text.  |
| <b>Button</b>                  | The name of a button to click or a key to press.<br><b>Example:</b> To close the window, either click <b>Close</b> , or press <b>Esc</b> . |
| <b>Key+Key</b>                 | Key combinations for which the user must press and hold down one key and then press another.<br>Example: <b>Ctrl+P</b> or <b>Alt+F4</b> .  |
| Monospace                      | Examples of code or standard output.   |
| <b>Monospace Bold</b>          | Text that you must enter.  |
| <i>variable</i>                | 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.<br>Example: <b>Operator Functions &gt; Report Functions</b>                     |
| <a href="#">hypertext link</a> | Used to indicate a hypertext link.   |

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

# 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 ..... | 1 |
| SMPP.....               | 2 |

## 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*.



# 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 .....             | 3  |
| Specification Clauses 2.1 and 2.2 .....           | 3  |
| Session States (2.3) .....                        | 4  |
| Operation Matrix (2.4) .....                      | 4  |
| PDU Sequencing (2.6) .....                        | 8  |
| Session Timers (2.7) .....                        | 8  |
| Error Handling (2.8) .....                        | 9  |
| Flow Control and Congestion Avoidance (2.9) ..... | 10 |
| Session Security and Encryption (2.10) .....      | 10 |
| Forward and Backward Compatibility (2.11) .....   | 11 |

## 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 `messageld` size is made, so it is possible to send a `messageld` greater than 8 octets in size.



# 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) ..... | 13 |
| General PDU Format (3.2) .....         | 13 |

## 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.

# 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).....    | 15 |
| Message Submission Operations (4.2) .....   | 17 |
| Message Delivery Operations (4.3) .....     | 23 |
| Message Broadcast Operations (4.4).....     | 25 |
| Ancillary Submission Operations (4.5) ..... | 26 |
| Ancillary Broadcast Operations (4.6).....   | 27 |
| PDU Field Definitions (4.7) .....           | 28 |
| PDU TLV Definitions (4.8) .....             | 35 |

## 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 subclassed 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 <i>MC Delivery Receipt</i> (4.3.5.1) (on page 24), <i>Intermediate Notification</i> (4.3.5.2) (on page 24), <i>SME Delivery Acknowledgement</i> (4.3.5.3) (on page 25), <i>SME Manual/User Acknowledgement</i> (4.3.5.4) (on page 25) and <i>Conversation Abort</i> (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)<br>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 greater 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  |
|-----------------------------------|--|
| <code>message_state</code>        | Stored in <code>GenericSM::deliverySucceeded</code> (as equal to <code>DELIVERED</code> or not). This field is set to false if the TLV is not present. |
| <code>receipted_message_id</code> | Stored in <code>GenericSM::deliveryReceiptId</code> . 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 Acknowledgement (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 Acknowledgement (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.<br>Treated as GenericSMResult::resultSuccess.  |
| ESME_RINVMSGLEN     | Messaging Manager complies.  |
| ESME_RINVCMDLEN     | Messaging Manager complies.  |
| ESME_RINVCMDID      | Messaging Manager complies.  |
| ESME_RINVBNDSTS     | Messaging Manager complies.<br>Receipt is treated as a<br>GenericSMResult::resultTransientFailure.   |
| ESME_RALYBND        | Messaging Manager complies.  |
| ESME_RINVPRFTLG     | Not used.  |
| ESME_RINVREGDLVFLG  | Not used.  |
| ESME_RSYSERR        | Messaging Manager complies. <ul style="list-style-type: none"> <li>• In the message outbound direction, receipt is treated as a GenericSMResult::resultTransient-Failure.</li> <li>• In the message inbound direction, an abort result type (GenericSMResult::resultAbort) will cause this code to be sent.</li> </ul> |
| 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. <ul style="list-style-type: none"> <li>• Messaging Manager is configured with maxConcurrentTransactions. When this is exceeded, MSGQFUL is replied to the sender.</li> <li>• Receipt of a MSGQFUL is treated as an GenericSMResult::resultTransient-Failure</li> </ul>                     |

| 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_RINVSRCNPI       | Not used  |
| ESME_RINVSRCTON       | Not used  |
| ESME_RINVSRCPNPI      | Not used  |
| ESME_RINVSTNPI        | Not used  |
| ESME_RINVSYSTYP       | Not used  |
| ESME_RINVREPFLAG      | Not used  |
| ESME_RINVNUMMSGS      | Not used  |
| ESME_RTHROTTLED       | Messaging Manager complies.<br>The throttling response code can be changed via the <b>eserv.config</b> default smpp parameter 'throttledCommandStatus'.   |
| ESME_RINVSCHED        | Not used  |
| ESME_RINVEXPIRY       | Not used  |
| ESME_RINVDFMSGID      | Not used  |
| ESME_RX_T_APPN        | Treated as a GenericSMResult::resultTransient-Failure.<br>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.<br>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_RINV DSTADDRSUBUNIT | 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::mwig (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, mwig, 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 |
|------------------|-----------|
| vmcsc_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.



# Glossary of Terms

## AAA

Authentication, Authorization, and Accounting. Specified in Diameter RFC 3588.

## ACS

Advanced Control Services configuration platform.

## AMC

The Advanced Message Control component of Messaging Manager. It provides:

- Real time billing interaction
- Customized Service Control
- SMS "service plan" capability
- Enhanced service support

**Note:** Also known as "SMS Director".

## ASP

- Application Service Provider, or
- Application Server Process. An IP based instance of an AS. An ASP implements a SCTP connection between 2 platforms.

## CC

Country Code. Prefix identifying the country for a numeric international address.

## CDMA

Code Division Multiple Access is a method for describing physical radio channels. Data intended for a specific channel is modulated with that channel's code. These are typically pseudo-random in nature, and possess favourable correlation properties to ensure physical channels are not confused with one another.

## Connection

Transport level link between two peers, providing for multiple sessions.

## Diameter

A feature rich AAA protocol. Utilises SCTP and TCP transports.

## DTMF

Dual Tone Multi-Frequency - system used by touch tone telephones where one high and one low frequency, or tone, is assigned to each touch tone button on the phone.

## FDA

First Delivery Attempt - the delivery of a short message directly to the SME rather than relaying it through the MC.

## **GSM**

Global System for Mobile communication.

It is a second generation cellular telecommunication system. Unlike first generation systems, GSM is digital and thus introduced greater enhancements such as security, capacity, quality and the ability to support integrated services.

## **IN**

Intelligent Network

## **IP**

1) Internet Protocol

2) Intelligent Peripheral - This is a node in an Intelligent Network containing a Specialized Resource Function (SRF).

## **ISUP**

ISDN User Part - part of the SS7 protocol layer and used in the setting up, management, and release of trunks that carry voice and data between calling and called parties.

## **MC**

Message Centre. Also known as SMSC.

## **Messaging Manager**

The Messaging Manager service and the Short Message Service components of Oracle Communications Network Charging and Control product. Component acronym is MM (formerly MMX).

## **MM**

Messaging Manager. Formerly MMX, see also *XMS* (on page 46) and *Messaging Manager* (on page 44).

## **MS**

Mobile Station

## **MT**

Mobile Terminated

## **MTP**

Message Transfer Part (part of the SS7 protocol stack).

## **NPI**

Number Plan Indicator

## **Octet**

Byte - 8 bits.

## **PC**

Point Code. The Point Code is the address of a switching point.

## **Peer**

Remote machine, which for our purposes is capable of acting as a Diameter agent.

## **SCCP**

Signalling Connection Control Part (part of the SS7 protocol stack).

## **SCTP**

Stream Control Transmission Protocol. A transport-layer protocol analogous to the TCP or User Datagram Protocol (UDP). SCTP provides some similar services as TCP (reliable, in-sequence transport of messages with congestion control) but adds high availability.

## **Session**

Diameter exchange relating to a particular user or subscriber access to a provided service (for example, a telephone call).

## **SLC**

Service Logic Controller (formerly UAS).

## **SLEE**

Service Logic Execution Environment

## **SME**

Short Message Entity - This is an entity which may send or receive short messages. It may be located in a fixed network, a mobile, or an SMSC.

## **SMPP**

Short Message Peer-to-Peer protocol

## **SMS**

Depending on context, can be:

- Service Management System hardware platform
- Short Message Service
- Service Management System platform
- NCC Service Management System application

## **SMSC**

Short Message Service Centre stores and forwards a short message to the indicated destination subscriber number.

## **SRF**

Specialized Resource Function – This is a node on an IN which can connect to both the SSP and the SLC and delivers additional special resources into the call, mostly related to voice data, for example play voice announcements or collect DTMF tones from the user. Can be present on an SSP or an Intelligent Peripheral (IP).

## **SS7**

A Common Channel Signalling system is used in many modern telecoms networks that provides a suite of protocols which enables circuit and non-circuit related information to be routed about and between networks. The main protocols include MTP, SCCP and ISUP.

## **SSL**

Secure Sockets Layer protocol

## **SSN**

Subsystem Number. An integer identifying applications on the SCCP layer.

For values, refer to *3GPP TS 23.003*.

## **SSP**

Service Switching Point

## **TCP**

Transmission Control Protocol. This is a reliable octet streaming protocol used by the majority of applications on the Internet. It provides a connection-oriented, full-duplex, point to point service between hosts.

## **TDMA**

Time Division Multiple Access - a communications technique that uses a common channel for communications among multiple users by allocating each a unique time slot.

## **TLV**

Tag-Length-Value. Optional parameters introduced in the SMPP protocol since version 3.4.

## **WAP**

Wireless Application Protocol. A standard designed to allow the content of the Internet to be viewed on the screen of a mobile device such as mobile phones, personal organisers and pagers. It also overcomes the processing limitation of such devices. The information and services available are stripped down to their basic text format.

## **XMS**

Three letter code used to designate some components and path locations used by the Oracle Communications Network Charging and Control *Messaging Manager* (on page 44) service and the Short Message Service. The published code is *MM* (on page 44) (formerly *MMX*).

# Index

## A

A sample PDU (3.2.2) • 14  
AAA • 43  
About This Document • v  
Absolute Time Format (4.7.23.4) • 33  
ACS • 43  
additional\_status\_info\_text (4.8.4.1) • 35  
addr\_npi, source\_addr\_npi, dest\_addr\_npi,  
  esme\_addr\_npi (4.7.2) • 28  
addr\_ton, source\_addr\_ton, dest\_addr\_ton,  
  esme\_addr\_ton (4.7.1) • 28  
address\_range (4.7.3) • 28  
Alert Notification Operation (4.1.3) • 17  
alert\_notification • 4  
alert\_notification Syntax (4.1.3.1) • 17  
alert\_on\_message\_delivery (4.8.4.2) • 35  
Alphanumeric Format (4.2.6.1.2) • 21  
AMC • 43  
Ancillary Broadcast Operations (4.6) • 27  
Ancillary Submission Operations (4.5) • 26  
Application Layer Communication (2.1) • 3  
ASP • 43  
Audience • v

## B

Backward Compatibility (2.11.2) • 11  
billing\_identification (4.8.4.3) • 35  
Bind Operation (4.1.1) • 15  
bind\_receiver • 4  
bind\_receiver Syntax (4.1.1.3) • 16  
bind\_receiver\_resp • 4  
bind\_receiver\_resp Syntax (4.1.1.4) • 16  
bind\_transceiver • 5  
bind\_transceiver Syntax (4.1.1.5) • 16  
bind\_transceiver\_resp • 5  
bind\_transceiver\_resp Syntax (4.1.1.6) • 16  
bind\_transmitter • 5  
bind\_transmitter Syntax (4.1.1.1) • 15  
bind\_transmitter\_resp • 5  
bind\_transmitter\_resp Syntax (4.1.1.2) • 16  
Bound\_RX (2.3.3) • 4  
Bound\_TRX (2.3.4) • 4  
Bound\_TX (2.3.2) • 4  
Broadcast Area Format types (4.8.4.4.1) • 36  
Broadcast Request Optional TLVs (4.4.2) • 25  
Broadcast Response Optional TLVs (4.4.3) • 25  
broadcast\_channel\_indicator (4.8.4.7) • 36  
broadcast\_content\_type (4.8.4.8) • 36  
broadcast\_content\_type\_info (4.8.4.6) • 36  
broadcast\_frequency\_interval (4.8.4.11) • 36  
broadcast\_message\_class (4.8.4.12) • 36  
broadcast\_service\_group (4.8.4.14) • 37  
broadcast\_area\_identifier,  
  failed\_broadcast\_area\_identifier (4.8.4.4) • 36

broadcast\_area\_success (4.8.4.5) • 36  
broadcast\_end\_time (4.8.4.9) • 36  
broadcast\_error\_status (4.8.4.10) • 36  
broadcast\_rep\_num (4.8.4.13) • 37  
broadcast\_sm • 5  
broadcast\_sm Operation (4.4.1) • 25  
broadcast\_sm Syntax (4.4.1.1) • 25  
broadcast\_sm\_resp • 5  
broadcast\_sm\_resp Syntax (4.4.1.2) • 25

## C

callback\_num (4.8.4.15) • 37  
callback\_num\_atag (4.8.4.16) • 37  
callback\_num\_pres\_ind (4.8.4.17) • 37  
Cancel Broadcast Optional TLVs (4.6.2.2) • 28  
cancel\_broadcast\_sm • 5  
cancel\_broadcast\_sm Operation (4.6.2) • 27  
cancel\_broadcast\_sm Syntax (4.6.2.1) • 28  
cancel\_broadcast\_sm\_resp • 5  
cancel\_broadcast\_sm\_resp Syntax (4.6.2.3) •  
  28  
cancel\_sm • 5  
cancel\_sm Operation (4.5.1) • 26  
cancel\_sm Syntax (4.5.1.1) • 26  
cancel\_sm\_resp • 6  
cancel\_sm\_resp Syntax (4.5.1.2) • 26  
CC • 43  
CDMA • 43  
Closed (2.3.6) • 4  
Command\_id (3.2.1.2) • 14  
command\_id (4.7.5) • 28  
Command\_length (3.2.1.1) • 13  
command\_length (4.7.4) • 28  
Command\_status (3.2.1.3) • 14  
command\_status, error\_status\_code (4.7.6) •  
  9, 29  
Compliance statement • 10  
Compliance Statements For SMPP Parameter  
  and PDU Format (3) • 13  
Compliance Statements For SMPP PDU  
  Definitions (4) • 15  
Compliance Statements For SMPP Sessions  
  (2) • 3  
congestion\_state (4.8.4.18) • 37  
Connection • 43  
Convention • 3  
Conversation Abort (4.3.5.5) • 17, 24, 25  
Copyright • ii

## D

data\_coding (4.7.7) • 18, 31  
data\_sm • 6  
data\_sm Operation (4.3.2) • 24  
data\_sm Syntax (4.2.2.1) • 19, 24  
data\_sm\_resp • 6  
data\_sm\_resp Syntax (4.2.2.2) • 19, 24  
Datagram Message Mode (4.2.10.3) • 23

- Default Message Mode (4.2.10.1) • 23
- deliver\_sm • 6
- deliver\_sm Syntax (4.3.1.1) • 23
- deliver\_sm\_resp • 6
- deliver\_sm\_resp Syntax (4.3.1.2) • 24
- delivery\_failure\_reason (4.8.4.19) • 37
- dest\_addr\_subunit (4.8.4.23) • 38
- dest\_addr\_np\_country (4.8.4.20) • 37
- dest\_addr\_np\_information (4.8.4.21) • 37
- dest\_addr\_np\_resolution (4.8.4.22) • 37
- dest\_bearer\_type (4.8.4.24) • 38
- dest\_flag (4.7.9) • 31
- dest\_network\_type (4.8.4.26) • 38
- dest\_network\_id (4.8.4.25) • 38
- dest\_node\_id (4.8.4.27) • 38
- dest\_port (4.8.4.30) • 38
- dest\_subaddress (4.8.4.28) • 38
- dest\_telematics\_id (4.8.4.29) • 38
- destination\_addr (4.7.8) • 31
- Diameter • 43
- display\_time (4.8.4.31) • 38
- dl\_name (4.7.10) • 32
- Document Conventions • vi
- dpf\_result (4.8.4.32) • 38
- DTMF • 43

## E

- Enquire Link Operation (4.1.2) • 16
- Enquire Link Timer • 8
- enquire\_link • 6
- enquire\_link Syntax (4.1.2.1) • 16
- enquire\_link\_resp • 6
- enquire\_link\_resp Syntax (4.1.2.2) • 17
- Error Handling (2.8) • 9
- esm\_class (4.7.12) • 32
- ESME Addresses (4.2.6.3) • 22
- esme\_addr (4.7.11) • 32
- Establishing a SMPP Session (2.2) • 3

## F

- FDA • 43
- final\_date (4.7.23.3) • 33
- Flow Control and Congestion Avoidance (2.9) • 10
- Forward and Backward Compatibility (2.11) • 11
- Forward Compatibility (2.11.1) • 11

## G

- General • 11, 15
- General PDU Format (3.2) • 13
- Generic NACK Operation (4.1.4) • 17
- generic\_nack • 6
- generic\_nack Syntax (4.1.4.1) • 17
- GSM • 44

## H

- Handling Connection Failure (2.8.1) • 9

## I

- IN • 44
- Inactivity Timer • 8
- interface\_version (4.7.13) • 32
- Intermediate Notification (4.3.5.2) • 17, 24
- International and National Format (4.2.6.1.1) • 21
- International Number • 22
- Invalid Field Length • 9
- IP • 44
- ISUP • 44
- its\_reply\_type (4.8.4.33) • 39
- its\_session\_info (4.8.4.34) • 39

## L

- language\_indicator (4.8.4.35) • 39
- Leased Lines (2.10.1) • 10

## M

- MC • 44
- MC Delivery Receipt (4.3.5.1) • 24
- Message Broadcast Operations (4.4) • 25
- Message Delivery Operations (4.3) • 23
- Message Delivery Request TLVs (4.3.3) • 24
- Message Delivery Response TLVs (4.3.4) • 24
- Message Length (4.2.8) • 22
- Message Modes (4.2.10) • 23
- Message Replace operation in submit\_sm (4.2.7) • 22
- Message Replacement TLVs (4.5.3.3) • 27
- Message Replacement with broadcast\_sm (4.4.4) • 26
- Message Submission Operations (4.2) • 17
- Message Submission Request TLVs (4.2.4) • 18, 19, 24
- Message Submission Response TLVs (4.2.5) • 19, 21, 24
- message\_id (4.7.14) • 32
- message\_payload (4.8.4.36) • 39
- message\_state (4.7.15) • 32
- message\_state (4.8.4.37) • 39
- Messaging Manager • 1, 44, 46
- Messaging Manager and SMPP Document Versions • 1
- Messaging Manager implementation • 1
- MM • 44, 46
- more\_messages\_to\_send (4.8.4.38) • 39
- MS • 44
- ms\_availability\_status (4.8.4.39) • 39
- ms\_msg\_wait\_facilities (4.8.4.40) • 39
- ms\_validity (4.8.4.41) • 39
- MT • 44
- MTP • 44

## N

network\_error\_code (4.8.4.42) • 40  
no\_unsuccess (4.7.16) • 32  
NPI • 44  
NPI (4.2.6.2) • 22  
NULL Address • 22  
NULL Settings (3.1.1) • 13  
number\_of\_messages (4.8.4.43) • 40  
number\_of\_dests (4.7.17) • 32

## O

Octet • 44  
Open (2.3.1) • 4  
Operation Failure (2.8.2) • 9  
Operation Matrix (2.4) • 4  
outbind • 6  
outbind Syntax (4.1.1.7) • 16  
Outbound (2.3.7) • 4  
Overview • 1, 3, 13, 15

## P

Parameter Type Definitions (3.1) • 13  
password (4.7.18) • 32  
payload\_type (4.8.4.44) • 40  
PC • 45  
PDU Field Definitions (4.7) • 28  
PDU Format (3.2.1) • 13  
PDU Sequencing (2.6) • 8  
PDU TLV Definitions (4.8) • 35  
Peer • 45  
Position of TLVs in SMPP messages (4.8) • 35  
Pre-defined (4.2.9.3) • 23  
priority\_flag (4.7.19) • 32  
privacy\_indicator (4.8.4.45) • 40  
protocol\_id (4.7.20) • 33

## Q

qos\_time\_to\_live (4.8.4.46) • 40  
Query Broadcast Request Optional TLVs  
(4.6.1.2) • 27  
Query Broadcast Response Optional TLVs  
(4.6.1.4) • 27  
query\_broadcast\_sm • 6  
query\_broadcast\_sm Operation (4.6.1) • 27  
query\_broadcast\_sm Syntax (4.6.1.1) • 27  
query\_broadcast\_sm\_resp • 6  
query\_broadcast\_sm\_resp Syntax (4.6.1.3) • 27  
query\_sm • 7  
query\_sm Operation (4.5.2) • 26  
query\_sm Syntax (4.5.2.1) • 26  
query\_sm\_resp • 7  
query\_sm\_resp Syntax (4.5.2.2) • 26

## R

receipted\_message\_id (4.8.4.47) • 40  
References to The Specification • 3

Registered (4.2.9.1) • 22  
registered\_delivery (4.7.21) • 33  
Relative Time Format (4.7.23.5) • 34  
Relevance • 4, 8  
replace\_if\_present\_flag (4.7.22) • 33  
replace\_sm • 7  
replace\_sm Operation (4.5.3) • 26  
replace\_sm Syntax (4.5.3.1) • 26, 27  
replace\_sm\_resp • 7  
replace\_sm\_resp Syntax (4.5.3.2) • 27  
Response Timer • 9

## S

sar\_msg\_ref\_num (4.8.4.48) • 40  
sar\_segment\_seqnum (4.8.4.49) • 40  
sar\_total\_segments (4.8.4.50) • 40  
sc\_interface\_version (4.8.4.51) • 40  
SCCP • 45  
Scheduled (4.2.9.2) • 23  
scheduled\_delivery\_time (4.7.23.1) • 33  
Scope • v  
SCTP • 45  
Secure Transport Layer (2.10.2) • 10  
Secure Tunnel (2.10.4) • 10  
Secure VPN (2.10.3) • 10  
Sequence Numbers Across Sessions (2.6.3) • 8  
Sequence\_number (3.2.1.4) • 14  
sequence\_number (4.7.24) • 34  
Service Short Code • 22  
service\_type (4.7.25) • 34  
Session • 45  
Session Init Timer • 8  
Session Management Operations (4.1) • 15  
Session Security and Encryption (2.10) • 10  
Session States (2.3) • 4  
Session Timers (2.7) • 8  
set\_dpf (4.8.4.52) • 40  
short\_message (4.7.26) • 34  
SLC • 45  
SLEE • 45  
sm\_default\_msg\_id (4.7.27) • 34  
sm\_length (4.7.28) • 34  
SME • 45  
SME Delivery Acknowledgement (4.3.5.3) • 17,  
24, 25  
SME Manual/User Acknowledgement (4.3.5.4)  
• 17, 24, 25  
SMPP • 2, 45  
SMPP document • 2  
SMPP Parameter Field Size Notation (3.1.2) •  
13  
SMPP PDU Format (Table 3-4) • 13  
SMPP PDU Parameter Types (Table 3-1) • 13  
SMS • 45  
sms\_signal (4.8.4.53) • 41  
SMSC • 45  
Source and Destination Addressing (4.2.6) • 21

- source\_subaddress (4.8.4.60) • 41
- source\_telematics\_id (4.8.4.61) • 41
- source\_addr (4.7.29) • 34
- source\_addr\_subunit (4.8.4.54) • 41
- source\_bearer\_type (4.8.4.55) • 41
- source\_network\_id (4.8.4.56) • 41
- source\_network\_type (4.8.4.57) • 41
- source\_node\_id (4.8.4.58) • 41
- source\_port (4.8.4.59) • 41
- Specification Clauses 2.1 and 2.2 • 3
- SRF • 46
- SS7 • 46
- SSL • 46
- SSN • 46
- SSP • 46
- Standard Parameters (3.2.1.5) • 14
- Store and Forward Message Mode (4.2.10.2) • 23
- submit\_multi • 7
- submit\_multi\_resp • 7
- submit\_multi Syntax (4.2.3.1) • 19
- submit\_multi\_resp Syntax (4.2.3.2) • 19
- submit\_sm • 7
- submit\_sm Syntax (4.2.1.1) • 17, 22, 24, 31, 32, 33
- submit\_sm\_resp • 7
- submit\_sm\_resp Syntax (4.2.1.2) • 18, 24
- Synchronous Vs. Asynchronous (2.6.4) • 8
- system\_id (4.7.30) • 34
- system\_type (4.7.31) • 34

## T

- TCP • 46
- TDMA • 46
- The ESME or MC is restricting the use of certain PDUs or features • 10
- The PDU data is unexpected and deemed invalid • 9
- The PDU is malformed • 9
- The PDU is not allowed in the current session state • 9
- The PDU is unrecognised • 9
- The PDU Sequence Number (2.6.1) • 8
- TLV • 46
- TLV Definitions (4.8.4) • 35
- TLV Length (4.8.2) • 35
- TLV Parameters (3.2.1.6) • 14
- TLV Tag (4.8.1) • 35
- TLV Value (4.8.3) • 35
- Transaction Message Mode (4.2.10.4) • 23
- Typographical Conventions • vi

## U

- unbind • 7
- unbind Syntax (4.1.1.8) • 16
- unbind\_resp • 8
- unbind\_resp Syntax (4.1.1.9) • 16

- Unbound (2.3.5) • 4
- UNIX Regular Expressions (4.7.3.1) • 28
- user\_message\_reference (4.8.4.62) • 41
- user\_response\_code (4.8.4.63) • 42
- ussd\_service\_op (4.8.4.64) • 42

## V

- validity\_period (4.7.23.2) • 33

## W

- WAP • 46
- Why Asynchronous? (2.6.5) • 8
- Why use Monotonically Increasing Sequence numbers? (2.6.2) • 8

## X

- XMS • 44, 46