



# SMPP v3.4 Protocol Implementation guide for GSM / UMTS

Version 1.0

# **SMPP v3.4 Protocol Implementation guide for GSM / UMTS**

## **May 30<sup>th</sup> 2002 Version 1.0**

©2001 SMS Forum

### **COPYRIGHT**

All rights reserved. This document or any part thereof may not, without the prior written consent of SMS Forum, be copied, reprinted or reproduced in any material form including, but without prejudice to the foregoing and not by way of exception photocopying, transcribing or storing in any medium or translating into any language, in any form or by any means, including but not limited to, electronics, mechanical, xerographic, optical, magnetic, digital or other methodology

### **DISCLAIMER**

WHILST THE GREATEST CARE HAS BEEN TAKEN TO ENSURE THE ACCURACY OF THE INFORMATION AND DATA CONTAINED HEREIN, SMS FORUM DOES NOT WARRANT THE ACCURACY OR SUITABILITY OF SAME FOR ANY SPECIFIC USE. SMS FORUM EXPRESSLY DISCLAIMS ALL AND ANY LIABILITY TO ANY PERSON, WHETHER A PURCHASER OR OTHERWISE, IN RESPECT OF ANY CONSEQUENCES OF ANYTHING DONE OR OMITTED TO BE DONE BY ANY SUCH PERSON IN PARTIAL OR TOTAL RELIANCE UPON THE WHOLE OR ANY PART OF THE CONTENTS OF THIS PUBLICATION OR ANY DERIVATIVE THEREOF.

THE INFORMATION CONTAINED HEREIN IS BELIEVED TO BE ACCURATE AND RELIABLE. HOWEVER, SMS FORUM ACCEPTS NO RESPONSIBILITY FOR IT'S USE BY ANY MEANS OR IN ANY WAY WHATSOEVER. SMS FORUM SHALL NOT BE LIABLE FOR ANY EXPENSE, COSTS OR DAMAGE THAT MAY RESULT FROM THE INFORMATION CONTAINED HOWSOEVER ARISING IN THIS DOCUMENT OR ANY DERIVATIVE THEREOF.

THE INFORMATION CONTAINED IN THE WITHIN DOCUMENT AND ANY DERIVATIVE THEREOF IS SUBJECT TO CHANGE WITHOUT NOTICE.

THE SMS FORUM IS COMPANY NUMBER 309113, REGISTERED OFFICE GARDNER HOUSE, WILTON PLACE, DUBLIN 2.

# Table of Contents

1.1 Overview .....	4
1.2 Scope.....	4
1.3 Glossary.....	5
1.4 References .....	5
2.1 Introduction .....	6
2.2 Sending regular text.....	6
2.2.1 Operation and fields to use.....	7
2.2.1.1 Operations .....	7
2.2.1.2 Fields .....	7
2.2.2 Data coding and size .....	8
2.2.2.1 Data coding.....	8
2.2.2.2 Data size .....	9
2.2.3 Mapping.....	9
2.2.3.1 Standard mapping.....	9
2.2.3.2 Extended mapping .....	10
2.2.3.3 Using financial symbols in a short message .....	11
2.2.4 Long messages and Concatenation .....	11
2.2.4.1 User data headers .....	12
2.2.4.2 SMPP parameters.....	13
2.3 Sending binary data, ring tones and operator logos .....	13
2.3.1 Operation and fields to use.....	13
2.3.1.1 Operations .....	13
2.3.1.2 Fields .....	14
2.3.2 Data coding .....	14
2.3.3 User data header.....	15
2.4 Enhanced messaging service.....	16
2.4.1 User data headers.....	16
2.4.2 Concatenation .....	17
2.5 Using priority .....	18
2.6 Requesting notifications.....	20
2.7 Addressing.....	21
2.7.1 Addressing a mobile.....	21
2.7.2 Addressing an ESME .....	21
2.8 Route messages on a mobile / direct displaying .....	22
2.9 Setting indications for voice mail, e-mail and FAX.....	22
2.9.1 Return Call Message.....	23
2.9.2 Data Coding Scheme .....	24
2.9.3 User Data Header, Special SMS Message Indication .....	24
2.10 Up-to-date messaging via message replacement .....	25
2.10.1 Message replacement on the SMSC .....	25
2.10.2 Message replacement at the mobile itself .....	26

# 1 Introduction

## 1.1 Overview

The Short Message Peer To Peer (SMPP) protocol is an open, industry standard protocol designed to provide a flexible data communication interface for transfer of short message data between a Message Centre and a SMS application system. A Bearer Message Centre can be a Short Message Service Centre (SMSC), a GSM Unstructured Supplementary Services Data (USSD) Server, a Broadcast Message Service Centre or another type of Message Centre.

This document provides guidelines for getting an application based on the SMPP protocol version 3.4 specification working in a SMSC/GSM/UMTS network environment.

## 1.2 Scope

This document provides practical information for developers building ESME applications that communicate with a SMSC working in a GSM environment, using SMPP protocol version 3.4. SMPP features common to all network types or not specifically SMSC related will not be discussed.

The information about the subjects discussed in this document is retrieved from related official specification documents, like the SMPP protocol version 3.4 specification, the GSM 23.038/23.040 and UCS2 specifications.

In order to keep the document a practical reference manual, the document will not contain specific in-depth information that is available in those specifications, but will include them as reference.

For maximising the reading efficiency, it is advisable to download the referenced specifications prior to reading this document.

### 1.3 Glossary

EMS	Enhanced Messaging Service
ESME	External Short Message Entity
GSM	Global System for Mobile Communications
MS	Mobile Station
SM	Short message
SMPP	Short Message Peer To Peer
SMSC	Short Message Service Centre
UCS	Universal Character Set
UDH	User Data Header
UMTS	Universal Mobile Telecommunications System

### 1.4 References

At the time of writing the latest versions of below stated documents were used, but for most cases other versions of these documents are also applicable as reference for this implementation guide.

Ref.	Document Title	Document Number	Version Number
[SMPP34]	SMPP protocol specifications v3.4	SMPP Forum <a href="http://www.smpp.org">http://www.smpp.org</a>	Issue 1.2
[GSM23038]	Alphabets and language-specific information	3GPP TS 23.038 <a href="http://www.3gpp.org">http://www.3gpp.org</a>	v4.2.0 (release 4)
[GSM23040]	Technical realization of the Short Message Service (SMS)	3GPP TS 23.040 <a href="http://www.3gpp.org">http://www.3gpp.org</a>	v5.0.0 (release 5)
[UCS2]	UCS-2. ISO/IEC 10646 encoding form: Universal Character Set coded in 2 octets	<a href="http://www.unicode.org">http://www.unicode.org</a>	

**Table 1-1: References**

## 2 How To

### 2.1 Introduction

This chapter provides information that can simplify the implementation of an ESME application, which is transferring data via short message to a mobile station in a GSM network.

The discussed items are:

- Sending regular text
- Sending binary data
- Sending ring tones
- Sending operator logos
- Enhanced messaging service
- Using priority
- Requesting notifications
- Addressing a mobile
- Addressing an ESME
- Routing a message on a mobile
- Directly message display
- Setting voice mail, e-mail and FAX indications
- Up-to-date messaging via SMSC message replacement
- Up-to-date messaging via mobile message replacement

Note: an ESME in the context of this document refers to such external sources and sinks of short messages as Voice Processing Systems, WAP Proxy Servers or Message Handling computers. It excludes SMEs, which are located within the Mobile Network, i.e., a mobile station.

### 2.2 Sending regular text

When sending regular text the following items are important to consider:

- Operation and fields to use
- Data coding and size
- Character mapping
- Concatenation

These items will be discussed in the following paragraphs.

## 2.2.1 Operation and fields to use

Two operations and two fields can be used to submit text data to the SMSC. Which operation and fields to use are explained in this paragraph.

### 2.2.1.1 Operations

An ESME application can send regular text by using the *submit\_sm* or *data\_sm* operation. The operation *submit\_sm* is only used by an ESME to submit a short message to the SMSC for onward transmission to a specified SME.

The operation *data\_sm* is used to transfer data between a SMSC and an ESME. The ESME may use this operation to request the SMSC to transfer a message to a mobile station and the SMSC may also use this operation to transfer a mobile station originated message to an ESME. This operation is an alternative to the *submit\_sm* and *deliver\_sm* operations.

### 2.2.1.2 Fields

When using the *submit\_sm* operation the message text data should be inserted in one of the following two fields:

- the *short\_message* field (mandatory field)
- the *message\_payload* field (optional field)

Simultaneous use of both fields is not allowed.

When using the *short\_message* field, the *sm\_length* field indicates the length in octets of the *short\_message* field data and must be set. If the *message\_payload* field is used the *sm\_length* field must be used as well, but will now be set to 0, which indicates the use of *message\_payload* field.

When using the operation *data\_sm* the message text data can only be inserted in the *message\_payload* field and the *sm\_length* field is not present.

For both operations is defined that up to 254 octets can be inserted in the *short\_message* field, while the *message\_payload* field is able to contain up to 64K octets. The GSM standard however defines a maximum of 140 octets for a single short message and thus does not support the transmission of more than these 140 octets per message. Therefore, a receiving SMSC will usually not accept a submit operation which will result in a short message of >140 octets, unless it has implemented an automatic concatenation mechanism, which divides a long message in multiple parts of 140 octets.

NOTE: although the following issue is more related to SMPP PDU encoding, it is important to realise another difference between the *short\_message* field and the *message\_payload* field:

As the *short\_message* field is a mandatory parameter and is defined as an octet string type, it needs a NULL character after the short message text. This is not the case for the *message\_payload* field, as it is an optional parameter that is based on TLVs.

For more information about these subjects see document [SMPP34] paragraphs 3.1, 3.1.1, 3.2.2, 4.4 and 4.7

## 2.2.2 Data coding and size

This paragraph discusses the short message data coding and the influence of this coding on the available data size.

### 2.2.2.1 Data coding

When sending text from ESME to the SMSC, it is important that the coding, for text also seen as the alphabet, of the short message text inside the SMPP PDU Field is supported by the SMSC and defined in such a way that the SMSC is able to interpret it.

It is important as well to realise that at the other side of the SMSC, when sending a short message from a SMSC to a mobile station, coding of the data takes place as well and this coding actually determines what is possible at the ESME side.

GSM defined the following (text) data coding options:

- GSM Default Alphabet (7-bits),
- UCS2 (16-bits)

The GSM Default Alphabet looks like the ASCII table (characters 0-127), with the difference that most of the control characters are not present and are replaced by characters from the LATIN-1 upper table (characters 128-255).

The SMPP specification offers many coding options, but not every offered coding is by default implemented at the SMSC. Usually the following are implemented at the SMSC:

- SMSC default alphabet (7 or 8-bits)
- LATIN 1 (8-bits)
- US ASCII (7-bits)
- UCS2 (16-bits)

The figure below shows the available coding options between an ESME, a SMSC and a mobile station:

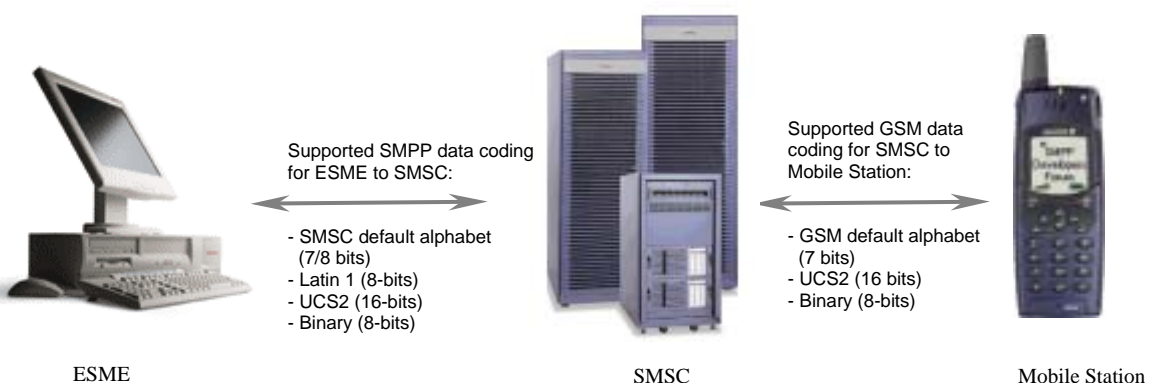


Figure 2-1: Coding options between an ESME, SMSC and a mobile station



When to use which coding?

When LATIN-1 character based text needs to be send the SMPP *data\_coding* parameter can be set to *0x03* for LATIN-1 alphabet. It can also be set to *0x00* for SMSC Default Alphabet if the ESME character coding is compatible with this character set (contact your SMSC vendor).

When the needed character is not available in the LATIN-1 character set the UCS2 data coding can be used. USC2 ensures that all characters present in the USC2 character set, like Japanese, Chinese, Hebrew, Arabic, can be send to and displayed by a mobile that supports UCS2.

It is important to realise that an ESME application needs to format the short message text in the application development coding as indicated by the *data\_coding*. For example when LATIN-1 is chosen and the used ESME platform is not LATIN-1 based, a mapping on the ESME has to take place for the ESME specific platform to LATIN-1.

### 2.2.2.2 Data size

As mentioned before, the maximum number of characters is limited to 140 octets. With 140 octets, 160 7-bit, 140 8-bit and 70 16-bit encoded characters can be used.

This means when the *data\_coding* is set to UCS2, only 70 characters can be send as the characters are 16-bit encoded by GSM. When the *data\_coding* is set to LATIN-1 or SMSC default alphabet, 160 characters can be send, as the characters are mapped to a 7-bit encoding by GSM.

Summary of data coding and characters:

Character set	Data_coding parameter	Character set	Max available characters
SMSC default alphabet	0x00	GSM default	160
LATIN-1	0x03	GSM default	160
UCS2	0x08	UCS2	70

**Table 2-1: Maximum available characters**

### 2.2.3 Mapping

This paragraph discusses the standard mapping, the extended mapping and how to use financial symbols in a short message.

#### 2.2.3.1 Standard mapping

Depending on the chosen ESME data coding the short message text data is send from the SMSC to the mobile in one of the following ways:

- Transparently
- Mapped to the default GSM alphabet

When text is send from ESME to SMSC in USC2 coding the data will be transparently send to the mobile.

When the text is coded for example in LATIN-1 or the SMSC Default Alphabet, usually a mapping will be performed by the SMSC to the GSM Default Alphabet before sending the text to the mobile. As the GSM Default Alphabet is 7-bit coded and uses other codes for

some characters and in some cases does not even provide a certain character, this implies that during the mapping process not every character can be mapped one-to-one.

Examples of a possible mapping (vendor dependent):

Type of character	Result
Control characters	All control characters are converted to inverted question marks with exception of the control characters line feed, form feed and carriage return
The dollar sign \$	Changes from code 0x24 to code 0x02 of the GSM alphabet, but stays the dollar sign \$
The commercial at sign @	Changes from code 0x40 to code 0x00 of the GSM alphabet, but stays the commercial at sign @
The left square bracket [	Changes from 0x5B to 0x3C, a less-than-sign <
The right square bracket ]	Changes from 0x5D to 0x3E, a greater-than-sign >
The left curly bracket {	Changes from 0x7B to 0x28, a left parenthesis (
The right curly bracket }	Changes from 0x7D to 0x29, a right parenthesis )
The backward slash \	Changes from 0x5C to 0x2F, a forward slash /
The circumflex accent ^	Changes from 0x5E to 0x14, a lambda ¶
The grave accent `	Changes from 0x60 to 0x27, an apostrophe ‘
Vertical line	Changes from 0x7C to 0x40, an inverted exclamation mark
The tilde ~	Changes from 0x7E to 0x3D, an equal sign =

**Table 2-2: Mapping**

NOTE: a quick way of finding out the SMSC default alphabet, is to send the @ character (0x00) to a mobile, if the SMSC implemented for example the 7-bit GSM table the same character will be shown at the GSM etc...

### 2.2.3.2 Extended mapping

The examples of Table 2-2 make clear that by default not every character can be used. To overcome this, the GSM Default Alphabet also contains an extension table, which can be used to display most of the characters mentioned above. The escape character of the default alphabet table, 0x1B, provides access to the extension table and with that the characters. This way the characters: |, ^, {,}, [,~,],\ and the even the Euro sign € can be displayed on the mobile.

If the mobile does not support the escape facility it should display a space. In the event that a MS receives a code where a symbol is not represented, the MS displays the character shown in the main default 7-bit alphabet. This applies also to the Euro sign €. If it is not supported by the mobile, it will display the regular e symbol.

### 2.2.3.3 Using financial symbols in a short message

By default the LATIN-1 and the GSM-alphabet allow the displaying of the following financial signs: Dollar \$, Pound £, Yen ¥. The LATIN-1 field data needed to display these characters in the same sequence on a mobile would be:

`0x24 0xA3 0xA5` (converted to GSM: `0x02 0x01 0x03`)

What about the Euro sign €? It is not present in the GSM default alphabet, but it is supported via the earlier mentioned GSM alphabet escape character method. For the Euro sign to be included in the SMPP PDU it must be converted by the ESME to `0x1B 0x65`.

As the Euro sign is on position A4 in ISO LATIN-9 some implementations may use this code to display the Euro sign. When an Euro sign is entered in a Windows application, the Euro sign will be represented by `0x80` and a conversion has to take place to `0x1B 0x65`.

NOTE: Making use of the escape character implies that two character positions are used for displaying one character (a message containing only Euro signs will have the capacity of 80 characters).

Summarised:

ESME Code	ESME Character	GSM Code	GSM Character
0x24	Dollar sign	0x02	Dollar sign
0xA3	Pound sign	0x01	Pound sign
0xA5	Yen sign	0x03	Yen sign
0x1B65	Escape character + Small letter E	0x1B65 extended alphabet	Euro currency sign

**Table 2-3: Financial symbols**

### 2.2.4 Long messages and Concatenation

Applications that require more than 160 characters of information transfer can use the *message\_payload* parameter which can hold up to a maximum of 64K. However, the maximum size supported is SMSC dependent and the application should take this into account.

Another available method for sending more than 160 characters of information is the GSM concatenation mechanism. With concatenation it is possible to send several separate short messages to a mobile station and to display them as one message.

Concatenation is realised at the mobile; multiple separated received messages are stringed together and displayed as if it was one single short message. The mobile is able to do this, as all involved send messages contain an additional information fields that provides information needed for stringing them together. Up to a maximum of 255 short messages can be concatenated this way.

Concatenation is provided via two methods:

- GSM User data headers
- SMPP parameters

### 2.2.4.1 User data headers

As defined by the GSM specifications, concatenation is provided by use of User Data Headers in each involved message.

A User Data Header (UDH) contains and sets the following items:

- The reference group the concatenated short message belongs to (two versions are available: 8-bits and 16-bits),
- The number of concatenated short messages in that group
- The location of the concatenated short message in the complete sequence.

These items enable the receiving application to put the concatenated short messages back together in the right order and determine whether all the short messages have been received.

In order to enable long messages the UDHI must be set in the SMPP parameter *esm\_class* and special user data header information must be present in front of the user data.

Settings for making use of user data header:

Esm_class PDU data value	User data header indication
Bit 6=1	Set
Bit 6=0	Not set

**Table 2-4: Indicating use of User data header**

Example of UDH of two message:

```
UDH SM 1      : UDHL=05 IEI(1)=00 IEIDL(1)=03 IED(1)=64 IED(1)=02 IED(1)=01
DATA SM 1     : <text part 1-2>
UDH SM 2      : UDHL=05 IEI(1)=00 IEIDL(1)=03 IED(1)=64 IED(1)=02 IED(1)=02
DATA SM 1     : <text part 2-2>
```

The user data header is part of the user data. This means the amount of user data is decreased. The amount of user data left is dependent of the used data coding:

Data coding*	Max number of characters
8-bit data	134
7-bit data	153
16-bit data	67

**Table 2-5: Maximum number of characters when using data headers**

\* Uncompressed

### 2.2.4.2 SMPP parameters

The SMPP protocol provides the following optional parameters to enable concatenation. Use of these SMPP parameters in a GSM environment result in the use of the User Data Headers as described in the previous paragraph as well, but the formatting is left to the SMSC and not to the ESME application, which makes it easier to use concatenation.

The following parameters need to be set for each short message:

Type of parameter	Description
Sar_msg_ref_num	Originator generated reference number allowing the parallel transmission of several segmented messages.
Sar_total_segments	The total number of fragments within the concatenated short message.
Sar_segements_seqnum	The sequence number of a particular message within the concatenated short message.

**Table 2-6: SMPP parameters for concatenation**

Related documents:

[SMPP34] paragraph 5.2.12 / 5.2.19/ 5.3.2.22 / 5.3.2.23 / 5.3.2.24

[GSM23040] paragraph 9.2.3.24.1

[GSM23038] paragraph 6.2.1

## 2.3 Sending binary data, ring tones and operator logos

When sending binary data like ring tones, operator logo's the following items are important to consider:

- Operation and fields to use
- Data coding
- User data header
- Concatenation

These items will be discussed in the following paragraphs.

### 2.3.1 Operation and fields to use

Two operations and two fields can be used to submit binary data to the SMSC. Which operation and fields to use are explained in this paragraph.

#### 2.3.1.1 Operations

An ESME application can send binary data by using the *submit\_sm* or *data\_sm* operation. The operation *submit\_sm* is only used by an ESME to submit a short message to the SMSC for onward transmission to a specified SME.

The operation *data\_sm* is used to transfer data between a SMSC and an ESME. The ESME may use this operation to request the SMSC to transfer a message to a mobile station and

the SMSC may also use this operation to transfer an mobile station originated message to an ESME. This operation is an alternative to the *submit\_sm* and *deliver\_sm* operation.

### 2.3.1.2 Fields

Sending binary data can be done with the SMPP *submit\_sm* or *data\_sm* operation.

When using the *submit\_sm* operation the binary data should be inserted in one of the following two fields:

- the *short\_message* field (mandatory)
- the *message\_payload* field (optional)

Simultaneous use of both fields is not allowed.

When using the *short\_message* field, the *sm\_length* field indicates the length in octets of the *short\_message* field data and must be set. If the *message\_payload* field is used the *sm\_length* field must be used as well, but will now be set to 0, indicating the use of *message\_payload* field.

When using the operation *data\_sm* the binary data can only be inserted in the *message\_payload* field and the *sm\_length* field is not present.

For both operations is defined that up to 254 octets can be inserted in the *short\_message* field, while the *message\_payload* field is able to contain up to 64K octets. The GSM standard however defines a maximum of 140 octets for a single short message and thus does not support the transmission of more than these 140 octets per message.

Therefore, a receiving SMSC will usually not accept a submit operation which will result in a short message of >140 octets, unless it has implemented an automatic concatenation mechanism, which divides a long message in multiple parts of 140 octets.

### 2.3.2 Data coding

An ESME uses a binary coding to send specific non-text data to a mobile, such as setting a ring tone or operator logo. This is done by setting the *data\_coding* parameter to the value *0x04*.

### 2.3.3 User data header

The binary user data may contain just the plain data itself but it can also contain a header in addition to the user data. This header is called the User Data Header (UDH).

The user data header allows a user to include data in a special format, for example in the case of ring tones, operator logo's etc., and to enable data concatenation (dividing data over several short messages).

It is also used for Enhanced Messaging. For Enhanced Messaging see the topic about the Enhanced Messaging Service in section 2.4.

The *esm\_class* parameter is set to indicate that the binary data includes a user data header.

Settings to indicate the use of a user data header:

Esm_class PDU data value	User data header indication
Bit 6=1	Set
Bit 6=0	Not set

**Table 2-7: Indicating use of User data header**

A user data header is constructed by information elements that are defined by the GSM specification. Each element has its own function, for example the User Data Header information element "Application port addressing scheme" is used to route short messages to one of multiple applications in the mobile via a predefined port.

Sending data to such a specific port is often used for setting ring-tones and operator logo's. The ports to address are mobile vendor dependent. Specific information for the mobile that is used can be obtained from the mobile phone vendor.

The following example shows the user data header and a part of the binary data for sending a ring tone to a mobile via two short messages:

```
UDH SM 1      : UDHL=0B IEI(1)=05 IEIDL(1)=04 IED(1)=15810000 IEI(2)=00 IEIDL(2)=03 IED(2)=010201
Data SM 1     : 024A3A5DA5D185B1A58... (data partly shown)
UDH SM 2      : UDHL=0B IEI(1)=05 IEIDL(1)=04 IED(1)=15810000 IEI(2)=00 IEIDL(2)=03 IED(2)=010202
Data SM 2     : B49C0920930AB12718A... (data partly shown)
```

Whereby:

- UDHL = User Date Header Length
- IEI = Information Element Identifier
- IEIDL = Information Element Identifier Data Length
- IED = Information Element Data

Explanation of UDH:

Value	Meaning
0B	Indicating the total UDH length
05 04 15810000	A 16 bit application port addressing schema is used with destination port 1581 and originator port 0000
00 03 01 02 01	Indicates the use of concatenated messaging that is build by two messages and this is the first message.

**Table 2-8: Explanation of UDH**

As mentioned before, the user data header is part of the user data itself, meaning that the actual available user data will be less than the mentioned maximum. This is an important issue.

For example, in the short messages discussed before, each user data header used 12 octets of space, meaning that 128 (140-12) octets are left for the user data itself.

Related documents:

[SMPP34] paragraph 5.2.12 / 5.2.19

[GSM23040] paragraph 9.2.3.24

## 2.4 Enhanced messaging service

The Enhanced Messaging Service (EMS) is the important interim step on the path that mobile messaging follows to evolve from mobile plain text messaging to mobile multimedia messaging.

EMS is based upon the standard SMS, but with formatting added to the text. The formatting may permit the message to contain objects like animations, pictures, melodies, formatted text, vCard and vCalendar. Objects may be mixed together into one message.

The Enhanced Messaging is builds on two standard mechanisms of GSM SMS messaging:

- User data headers
- Message concatenation

### 2.4.1 User data headers

User data headers enable to include binary data in a normal SM prior to the text message itself. The binary data contains the enhanced messaging data.

A so-called information element in the user data Header identifies each animation, picture, melody, formatted text, vCard and vCalendar object within the SM. The information element will contain an octet that identifies the absolute position of the object within and from the beginning of the SM data. In case of formatting text, an additional octet will give the number of characters for which the formatting applies.

This binary data is in the message field, which means that the 140 octets are reduced.



## 2.4.2 Concatenation

The concatenation mechanism permits messages longer than 140 octets. Up to 255 messages with a size of 140 octets each can be concatenated to one message up to about 38K octets.

EMS information elements of the same type may occur more than once in a single message or one segment of a concatenated message.

In order to enable enhanced messaging the UDHI must be set in the SMPP parameter *esm\_class* and special user data header information must be present in front of the user data.

Settings for making use of user data header:

<b>Esm_class PDU data value</b>	<b>User data header indication</b>
Bit 6=1	Set
Bit 6=0	Not set

**Table 2-9: Indicating use of User data header**

### Example 1: Formatting the appears of text

UDHI = set

UDH SM1 : UDHL=05, IEI=0A, IEDL=03, IED1=0F, IED2=12, IED3=10

Data SM1 : This is a text with bold option on following with normal text.

The result on an EMS mobile station would be:

This is a text **with bold option on** following with normal text.

### Example 2: Predefined sounds in the message

UDHI = set

UDH SM1 : UDHL=08, IEI=0B, IEDL=02, IED1=09,<sound5>, IEI=0B, IEDL=2, IED1=1C,<sound7>

Data SM1 : This is a message with two different sounds

The result would be: the sound number 5 shall be played after the 9<sup>th</sup> received character ("a") and sound number 7 shall be played after the 28<sup>th</sup> received character ("e").

Although a user data header is involved, the short message text is normal text, which implies the *data\_coding* parameter must be set to the values used for text.

Related documents:

[SMPP34] paragraph 5.3.2.32

[GSM23040] paragraph 3.10 / 9.2.3.24.1 / 9.2.3.24.10.1

## 2.5 Using priority

The item priority can have two meanings:

- Delivery priority
- Content priority

GSM and SMPP support both delivery priority, but do not provide a form of content priority. The latter has to be implemented by the ESME itself (for example via EMS).

What does delivery priority means in GSM? Delivery priority for GSM implies that message delivery will be attempted even if a mobile is in the following conditions:

- Temporarily absent
- No free message memory capacity

“Temporarily absent” means that the mobile was unreachable during the last attempt to deliver a short message, while the mobile was not registered as “switched-off”. This situation can arise when for example the mobile was temporarily out of reach or when another short message was being delivered at the same time.

No free message memory capacity can arise when a mobile user does not delete any received message and the maximum storage space for short messages has been reached.

For non-priority messages delivery will be attempted if the mobile has not been identified as temporarily absent.

The SMPP specification has four defined priority levels, but for GSM only two levels of priority are defined: non priority and priority. The SMPP *priority\_flag* parameter allows the originating SME to assign a priority to a message.

The SMPP/GSM mapping is as follows:

SMPP Priority Level	GSM Priority Level	Priority_flag PDU value
0	Non-priority	0x00
1	Priority	0x01

**Table 2-10: SMPP/GSM priority level mapping**

Related documents:

[SMPP34] paragraph 5.2.14

[GSM23040] paragraph 3.2.5/ 3.2.6

## 2.6 Requesting notifications

When the ESME has submitted a short message to a mobile it can request for a notification in order to be informed about the outcome of the sent mobile originated short message.

Generally two types of notifications can be distinguished:

- SMSC delivery receipts
- SME originated acknowledgements

SME originated acknowledgements means that the acknowledgement is send from the receiving mobile itself. This form of notification is not provided by GSM.

SMSC delivery receipt is supported and means the SMSC itself informs the EMSE about the outcome of a message delivery. The status of the message can be:

- Successfully delivered to the SME
- The SMSC was not able to forward the message to the SME. The reason can be an error of permanent or temporary nature. Permanent errors can be e.g. validity period expired, unknown mobile subscriber. Errors of temporary nature can be e.g. SMSC-SME connection down, SME temporarily unavailable.

The SMPP *registered\_delivery* parameter is used to request the SMSC delivery receipt and set the following notification options:

Notification type	Registered_delivery setting bit 4,3,2,1,0
No SMSC Delivery Receipt and intermediate notification requested	0 x x 0 0
SMSC Delivery Receipt requested where final delivery outcome is delivery success or failure and no intermediate* notification requested	0 x x 0 1
SMSC Delivery Receipt requested where the final delivery outcome is delivery failure and no intermediate* notification requested	0 x x 1 0
No SMSC Delivery Receipt requested, but requested intermediate* notification.	1 x x 0 0
SMSC Delivery Receipt requested where final delivery outcome is delivery success or failure and Intermediate* notification requested	1 x x 0 1
SMSC Delivery Receipt requested where the final delivery outcome is delivery failure and intermediate* notification requested	1 x x 1 0

**Table 2-11: SMPP registered delivery settings**

\* Intermediate is also known as buffered

Related documents:

[SMPP34] paragraph 5.2.17

[GSM23040] paragraph

## 2.7 Addressing

The ESME and the mobile are both addressed when sending short messages from an ESME to a mobile station and visa versa.

With addressing two GSM parameters are involved:

- Numbering Plan Indicator (NPI)
- Type of Number (TON)

The following paragraphs will discuss the addressing for the ESME and the mobile.

### 2.7.1 Addressing a mobile

For short messages addressed to a mobile the Numbering Plan Indicator must be set to *ISDN*, also know as *E.163/E.164* or *telephony*. This parameter can be set via the SMPP *dest\_addr\_npi* parameter and should in this case have the value *0x01*.

It is preferable to use the International format for the Type of Number. This parameter can be set via the *dest\_addr\_ton* parameter and should have the value *0x01*.

When using this setting the ESM application must provide the subscriber number in the format *<country code><area code><subscriber number>*, for example 31612345678, which addresses a mobile in the Netherlands. Via this addressing method it is ensured that a subscriber is uniquely identified. The international format will also be accepted when the message is destined to a recipient in the same country.

If it is sure the application will only process national addresses it is of course possible to use the national format as well. For some applications, like e.g. web page based ones, the Type of Number has to be set to unknown. This way, prefixes can be included and the SMSC will analyse the number itself.

### 2.7.2 Addressing an ESME

To enable that a receiving mobile station can respond easily to an ESME originated short message (usually identified with a short number, e.g. 3333), the source address settings need to be set. These settings provide the recipient with the originator number, together with its settings. This way the mobile user doesn't need to be concerned about the to be used number when replying.

The parameters can be set via the SMPP *source\_addr* parameter, the SMPP *source\_addr\_npi* parameter and the SMPP *source\_addr\_ton* parameter.

For addressing an ESME, the Numbering Plan Indicator and the Type of Number can be set theoretically to any value. But if a subscriber replies to an ESME originated short message not containing a source address and the subscriber must enter the ESME address manually, the Numbering Plan Indicator value of this address is set automatically by the mobile station to the value *0x01*, which is *ISDN*, also known as *E.163/E.164* or *telephony*. The Type of Number value is set to the value *0x00*, which is *UNKNOWN*. It is advisable to use the same values for the ESME source values, this way it is ensured the SMSC will always be able to handle it.

It is also possible that an ESME source address needs to be displayed at the mobile as an alphanumeric string. To achieve this the Numbering Plan Indicator value must be set to value *0x00*, which is UNKNOWN. The Type of Number value must be set to *0x05*, which sets the alphanumeric mode.

Related documents:

[SMPP34] paragraph 5.2.5 / 5.2.6

[GSM23040] paragraph 9.1.2.5

## 2.8 Route messages on a mobile / direct displaying

The *dest\_addr\_subunit* parameter is used to route messages on a mobile station, for example to a smart card in the mobile station or to an external device connected to the mobile station. The parameter is also used for direct displaying of a short message without storing it. It also determines the way the mobile will acknowledge message reception to the SMSC.

The *dest\_addr\_subunit* parameter settings:

Behaviour	SMPP setting	GSM DCS equivalent
Unknown (default)	0x00	no message class
MS Display	0x01	Message class 0
Mobile Equipment (mobile memory)	0x02	Message class 1
Smart Card 1 (SIM memory if a SIM exists in the MS)	0x03	Message class 2
External Unit 1	0x04	Message class 3

**Table 2-12: GSM DCS / SMPP mapping**

In GSM message routing and direct displaying of a message on a mobile are controlled by the GSM data coding scheme parameter. This could imply that these functionalities are controlled by the SMPP *data\_coding* parameter as well. But the SMPP *data\_coding* parameter will evolve to specify character code settings only, thus the recommended way to specify GSM message class control is by specifying the relevant setting in the described optional parameter *dest\_addr\_subunit*.

Related documents:

[SMPP34] paragraph 5.2.19 / 5.3.2.1

[GSM23038] chapter 4

## 2.9 Setting indications for voice mail, e-mail and FAX

A GSM mobile can show special icons on a MS screen. The icons can be accompanied by a text indicating that messages are waiting for the subscriber. This can be very useful for example in the case of a unified messaging application that wants to indicate to a mobile subscriber that he or she has a voice-mail, an e-mail, etc. waiting.

There are three levels of "Message Waiting" indication provided within GSM:

- The first level is to set the Protocol Identifier to *Return Call message*, which indicates that a message is waiting and relies on the text of the message to supply the detail.

- The second level uses the Data Coding Scheme with or without Return Call Message (see [GSM23038]) to indicate the type of message waiting and whether there are some messages or no messages.
- The third level uses the User Data Header and provides the maximum detail level for analysis by the mobile, i.e. an indication of the number and type of messages waiting. This third level is provided for future flexibility, as it cannot immediately be used without compatibility problems with the earliest phase mobiles.

Each level will be discussed in the following paragraphs.

### 2.9.1 Return Call Message

A Return Call Message indicates the mobile station to inform the user that a call (e.g. a telephone call) can be established to the specified originator address. The message content (if present) gives displayable information (e.g. the number of waiting voice messages). The message is handled in the same way as messages of the Replace Short Message Types, which means the Return Call Message is controlled via the SMPP *protocol\_id* parameter.

Setting for the *protocol\_id* parameter:

Replace type	Protocol_id PDU data value hex / binary
Return call message	0x5F / 01011111

**Table 2-13: Replace type setting**

For more information about short message replacement see paragraph 0.

### 2.9.2 Data Coding Scheme

Showing special icons on a mobile station screen is in the GSM specification controlled by the Data Coding Scheme (DCS). The equivalent parameter in SMPP is the *ms\_msg\_wait\_facilities* parameter.

Settings for the *ms\_msg\_wait\_facilities*:

Showned indication icon type	<i>ms_msg_wait_facilities</i> PDU data value in hex / binary
Non	0x00-0x7F / 0xxxxxxx
Voicemail Message Waiting	0x80 / 10000000
Fax Message Waiting	0x81 / 10000001
Electronic Mail Message Waiting	0x82 / 10000010
Other Message Waiting*	0x82 / 10000011

**Table 2-14: Settings for indication icon types**

\* The GSM specification advises not to use this setting, a mobile may not display any notification indication

When using the GSM DCS parameter it is possible to define that the mobile should not store the SM text. This is not possible with the *ms\_msg\_wait\_facilities* parameter.

### 2.9.3 User Data Header, Special SMS Message Indication

Via the Special SMS Message Indication, the User Data Header can send the following information to a mobile station:

- Message Indication type
- Message Count
- Storage

The message indication types are the same as the ones that could be set via the data coding scheme:

- Voice Message Waiting
- Fax Message Waiting
- Electronic Mail Message Waiting
- Other Message Waiting

The message count can have a value in the range 0 to 255 indicating the number of messages indicated by the message indication type. The value 255 indicates 255 or greater. If the number of messages is non-zero the specified indicator will be shown. The indicator will be removed if the number of messages is zero. The message can be stored or discarded after the updating indication.

Example for indicating four voice mails (without message store) and two Fax Mails (with message store) :

UDHI=set

UDH SM1 : UDHL=08, IEI=01, IEDL=02, IED1=00, IED2=04, IEI=01, IEDL=02, IEID=81, IEID=02



Text may be included by the SMS Service Centre for backward compatibility with the earliest Phase mobiles. The Data Coding Scheme may also be used to convey this information in parallel for backward compatibility with "middle" Phase mobiles (which support the use of Data Coding Scheme for Message Waiting Indication).

Related documents:

[SMPP34] paragraph 5.3.2.13

[GSM23038] chapter 4

[GSM23040] paragraph 9.2.3.24.2

## 2.10 Up-to-date messaging via message replacement

It is important to ensure that the user of a mobile station will not be bothered by out-dated information provided when using services like unified messaging, voice mail, e-mail etc. and services that provide actual information about e.g. stock exchange, share prices and traffic information.

Out-dated short messages occur when for example multiple short messages with the same type of information (e.g. updated traffic-jam information) were sent to the same mobile, while it was switched off. As soon as the user switches on the mobile, those messages are received, while only the most recent one is of interest (e.g. traffic-jam information of some hours ago has no meaning).

Also, when a user receives a SMS share-price information, he or she receives at regular intervals short messages at the mobile stating the share price of a particular share. When the user was not able to check the short messages, he or she did not see for example the latest three updates. The first two messages contain out-dated information and do not need to be present anymore (unless the user is specifically interested in historical data).

The examples above could imply the need for a short message replace mechanism. A replace mechanism can be implemented in two ways, via:

- Message replacement on the SMSC
- Message replacement at the mobile itself

These mechanisms will be discussed in the following paragraphs.

The message replacement mechanism on the SMSC is in SMPP not directly GSM specific, but is for clarity discussed in this document as well.

### 2.10.1 Message replacement on the SMSC

For the replacement of a short message two operations can be used: the *replace\_sm* and the *submit\_sm* operation.

The *replace\_sm* command is not GSM specific and issued by the ESME to replace a previously submitted short message on the SMSC that is still pending delivery. Replacement of short messages on the mobile via this command is not possible. If the short message is not present, a new short message will not be send. The matching mechanism is based on the *message\_id* and source address parameters of the original message.

By using the *submit\_sm* command it is possible to replace a short message on the SMSC and on the mobile as well. Replacing a message on the SMSC is achieved via the *replace\_if\_present\_flag* and is not GSM specific. If the *replace\_if\_present\_flag* is set the SMSC is requested to replace a previously submitted message on the SMSC that is still pending for

delivery. If the short is not present, a new short message will be submitted. The replace function can be activated in the *submit\_sm* PDU by setting the *replace\_if\_present\_flag* to 1 (one). The matching mechanism is based on the source address, destination address and service type parameters of the original message.

Replacing a message on the mobile is GSM specific and achieved by setting one of the replace short message types via the *protocol\_id* (see paragraph 2.10.2 of the document)

Combinations of this flag and the GSM PID are not possible.

Related documents:

[SMPP34] paragraphs 4.4 / 4.10 / 5.2.13 / 5.2.18

[GSM23040] paragraph 9.2.3.9

### 2.10.2 Message replacement at the mobile itself

As said before, for some services it can be useful to replace short messages that were previous received on the mobile. For example for share prices, traffic information etc. This way the information is always up-to-date and a full buffer for mobile short message is prevented.

The GSM specification has defined seven replace short message types for mobile terminated messages. On receipt of a short message from the SMSC, the mobile station checks if a Replace Short Message Type code is present. If such a code is present, then the mobile station checks the associated SMSC address and originating address and replace any existing stored message having the same Replace Short Message Type code, SMSC address and originating address with the new short message and other parameter values. If there is no message to be replaced, the MS stores the message in the normal way.

Message replacement is controlled via the *protocol\_id* parameter

Settings for replacing messages on the mobile:

Replace type	protocol_id PDU data value hex / binary
1	0x41 / 01000001
2	0x42 / 01000010
3	0x43 / 01000011
4	0x44 / 01000100
5	0x45 / 01000101
6	0x46 / 01000110
7	0x47 / 01000111
Return call	0x5F / 01011111

**Table 2-15: Replace types**

The presence of seven replace types allows having seven different information services that can replace their own messages. For example, the traffic info service uses replace type 1, the stock exchange info service uses replace type 2, etc...

A Return Call Message indicates to the mobile station to inform the user that a call (e.g. a telephone call) can be established to the address specified within the originator address. The message content (if present) gives displayable information (e.g. the number of waiting voice messages). The message is handled in the same way as all other messages of the Replace Short Message Types and will replace messages as well.

Related documents:

[SMPP34] paragraph 5.2.13

[GSM23040] paragraph 9.2.3.9