



Technical Bulletin

SMPP+ USSD Services

This document contains proprietary and confidential material of Comverse Inc. Any unauthorized reproduction, use, or disclosure of this material, or any part thereof, is strictly prohibited. This document is solely for the use of Comverse Inc. employees and authorized Comverse Inc. customers.

REVISION HISTORY

Rev	Date	Author	Title
0.1	8/12/05	Shahar B.	SMPP+ draft
1.0	15/12/05	Shahar B.	Add flows in 2.2.1, 2.2.2 Add Annex 4.3 – message/parameter mappings Update according to review comments
1.1	15/12/05	Shahar B.	Add BIND_TRANSCEIVER PDU definition Extend Annex 2 to include parameter detailed definitions Modify ussd_session_id allocation algorithm (see submit_sm, submit_sm_resp PDUs description) Add reference to GSM default alphabet encoding translation to SMPP SMSC default alphabet value (Annex 3).
1.2	27/12/05	Shahar B.	Update timers used in USSD GW in 3.3 Add overview of USSD session termination procedures in 2. Clarify that USSD session termination possibilities in 3.7.1 and 3.8.1.
1.3	5/1/06	Shahar B.	Correct mistake in <i>ussd_service_op</i> type definition (from C-ocetet String to Octet String) + add encoding note example (4.2.1) Add <i>language_indicator</i> to submit_sm and deliver_sm optional part. Add Annex 4 – mapping of <i>language_indicator</i> to USSD MAP and vice versa. Update Annex 3 mappings to indicate use of <i>language_indicator</i>
1.4	12/1/06	Shahar B.	Add USSD GW vendor specific error codes (4.1)
1.5	16/1/06	Shahar B.	Change table format for flows + add TC-Primitive to MAP messages (2.1,2.2) Fix headings in 4.3.2.4, fix broken link in 4.2 Add notes for <i>data_coding</i> translation in 4.4
1.6	18/1/06	Shahar B.	Change SMPP error codes not to conflict with SMSC vendor specific error codes + complete the GSM errors mapping (4.1) Add note to 4.2.1 to explain the “last message indication” values can be used only in application originated USSD sessions. Add NOTE2 to message mappings in 4.3 to point to <i>language_indicator</i> mapping.
1.7	22/1/06	Shahar B.	Add SMPP+ message dumps as examples (4.5) Copy note about last message indication usage to 3.7.1 – <i>ussd_service_op</i> parameter description. Update maximum <i>short_message</i> length to 200 octets (3.7.1, 3.8.1) Extend range of <i>ussd_session_id</i> to allow 0 (4.2.2)
1.8	26/1/06	Shahar B.	Add session initiation failure flow in MT USSD (2.2.2)
1.9	29/1/06	Shahar B.	Add optional TLV (<i>ussd_session_closed</i>) in responses (4.2.3, 3.7.2, 3.8.2). Update error ending flow in 2.2.2
1.10	29/1/06	Shahar B.	Change implementation of correction in revision 1.9 – instead of a <i>ussd_session_closed</i> TLV an error (ESME_RUSSDSESSIONTERMABN) is used. Removed 4.2.3, 3.7.2, 3.8.2 & corrected flow in 2.2.2.

1.11	31/1/06	Shahar B.	Create 2 new mapping sections in Annex 4 (4.4.2 ton mappings, 4.4.3 npi mappings)
1.12	9/3/06	Irena S. / Shahar B.	Update usage of dest/source address in submit_sm (3.7.1) Update usage of dest/source address in deliver_sm (3.8.1) Remove support of message_payload in both submit_sm and deliver_sm. Add details to mapping tables in 4.3.2.3, 4.3.2.4
1.13	2/4/06	Ella P. / Shahar B.	Add new optional TLVs to deliver_sm and vendor specific definitions (3.8.1, 4.2) + correct tags (4.2) Update mapping tables with new TLVs and mark Ericsson specific mappings (4.3.1.1, 4.3.1.2, 4.3.2.3, 4.3.2.4) – incorporate SMSC HLR TLVs Mark that IMSI query is optional part of MT flow (2.2.2) Change Field to Field (parameter tables) Add Acronyms table

SUMMARY

	Page
1. - PROTOCOL OVERVIEW	5
2. - SMPP+ EXCHANGES	6
2.1. - SMPP+ FOR USSD PHASE 1	6
2.1.1. - <i>MO USSD</i>	6
2.2. - SMPP+ FOR USSD PHASE 2	7
2.2.1. - <i>MO USSD</i>	7
2.2.2. - <i>MT USSD</i>	9
3. - SMPP+ PROTOCOL DESCRIPTION	11
3.1. - GENERAL ASPECTS OF THE SMPP+ PROTOCOL	11
3.2. - NETWORK LEVEL CONNECTION	11
3.3. - TIMERS DEFINITIONS	12
3.4. - BIND OPERATION	12
3.4.1. - " <i>BIND_TRANSMITTER</i> " PDU	12
3.4.2. - " <i>BIND_TRANSMITTER_RESP</i> " PDU	13
3.4.3. - " <i>BIND_RECEIVER</i> " PDU	13
3.4.4. - " <i>BIND_RECEIVER_RESP</i> " PDU	14
3.4.5. - " <i>BIND_TRANSCEIVER</i> " PDU	14
3.4.6. - " <i>BIND_TRANSCEIVER_RESP</i> " PDU	15
3.5. - UNBIND OPERATION	16
3.5.1. - " <i>UNBIND</i> " PDU	16
3.5.2. - " <i>UNBIND_RESP</i> " PDU	16
3.6. - GENERIC_NACK PDU	16
3.6.1. - " <i>GENERIC_NACK</i> " PDU	16
3.7. - SUBMIT_SM OPERATION	17
3.7.1. - " <i>SUBMIT_SM</i> " PDU	17
3.7.2. - " <i>SUBMIT_SM_RESP</i> " PDU	19
3.8. - DELIVER_SM OPERATION	20
3.8.1. - " <i>DELIVER_SM</i> " PDU	20
3.8.2. - " <i>DELIVER_SM_RESP</i> " PDU	23
3.9. - ENQUIRE_LINK OPERATION	23
3.9.1. - " <i>ENQUIRE_LINK</i> " PDU	23
3.9.2. - " <i>ENQUIRE_LINK_RESP</i> " PDU	24
3.10. - OTHERS SMPP v3.4 PDU	24
4. - ANNEXES	24

4.1. - ANNEX 1: LIST OF SPECIFIC SMPP+ ERROR CODES	24
4.2. - ANNEX 2: LIST OF SPECIFIC SMPP+ ATTRIBUTES	25
THE FOLLOWING SECTIONS CONTAIN SMPP+ SPECIFIC DEFINITIONS AND USAGE OF NON-STANDARD OR EXTENDED	
SMPP PARAMETERS	25
4.2.1. - <i>ussd_service_op</i>	25
4.2.2. - <i>ussd_session_id</i>	26
4.3. - ANNEX 3: SMPP+ TO GSM MAP USSD MESSAGES/PARAMETERS MAPPING	28
4.3.1. - <i>DELIVER_SM</i>	28
4.3.2. - <i>SUBMIT_SM</i>	30

REFERENCES

Ref n°	Identifier	Title
1		SMPP Protocol Specification v3.4 – 12 Oct-1999 Issue 1.2
2	SysRS	Comverse USSD GW Service Definition document 1.4
3	ISO 639	Language Identifier Prefixes
4	TIA/EIA 136-710A	Short Message Service – Cellular Messaging Teleservice
5		Comverse USSD GW GSM MAP Interface 1.3

ACRONYMS

Name	Description
ERC	Ericsson Specific
GW	Gateway
HLR	Home Location Register
IMSI	International Mobile Subscriber Identity
MAP	Mobile Application Part
SCCP	Signaling Connection Control Part
SMPP	Short Message Peer to Peer Protocol
USSD	Unstructured Supplementary Services
VLR	Visited Location Register

1. - PROTOCOL OVERVIEW

This document describes the SMPP+ protocol used between the Comverse USSD GW and USSD applications implementing a USSD Phase 1 or USSD Phase 2 application logic.

Using the SMPP protocol, a USSD application system called the 'External Short Message Entity' (ESME) may initiate an application layer connection with a USSD GW over a TCP/IP network connection and may then send USSD messages and receive USSD messages to and from the USSD GW respectively.

SMPP+ means SMPP v3.4 protocol (document [1]) + additional features needed to implement a good and fair link between the participating parties and in order to pass USSD Specific control information between the ESME and the USSD GW. The SMPP+ is based on the Comverse SMPP implementation (SMPP v3.4) described in document [2].

This document defines specifications of the SMPP+ protocol and specifies the command and response format to be used when implementing an SMPP+ protocol interface.

This document is divided as follows:

- SMPP+ exchanges between the different platforms
- SMPP+ PDU description with the specific SMPP+ behavior

2. - SMPP+ EXCHANGES

The SMPP+ is used between the USSD GW and the following platforms:

- USSD application

The document describes also messaging flow between the USSD application, USSD GW and the mobile subscriber.

SMPP is based on the exchange of request and response protocol data units (PDUs) between the ESME and the USSD GW over an underlying TCP/IP network connection. The SMPP+ protocol defines:

- A set of operations and associated Protocol Data Units (PDUs) for the exchange of short messages between an ESME and a USSD GW
- The data that an ESME application can exchange with an USSD GW during SMPP operations.

Note: Every SMPP operation must consist of a request PDU and associated response PDU. The receiving entity must return the associated SMPP response to an SMPP PDU request.

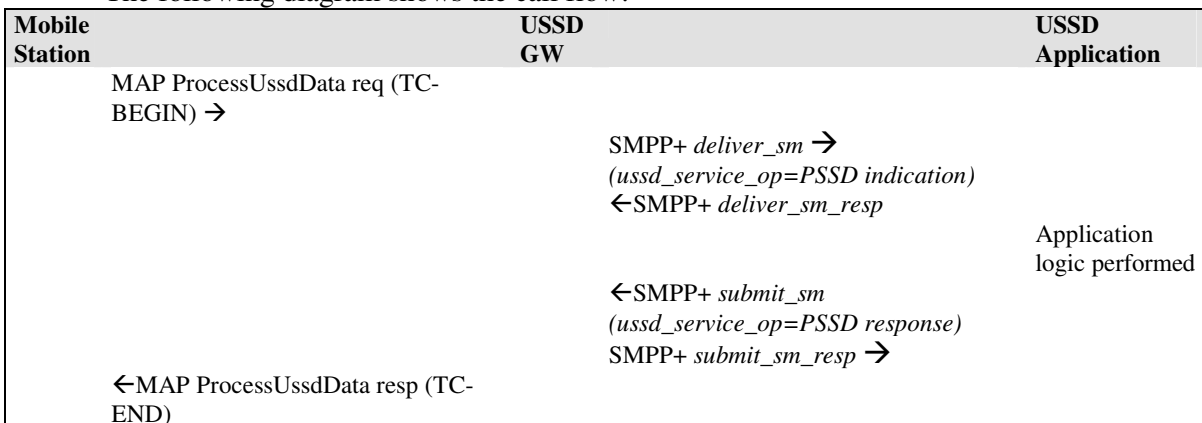
A USSD session may be initiated by the Mobile or the USSD application. However, as a rule, only the USSD application terminated the USSD session (via specific values of the *ussd_service_op* optional parameter in the SUBMIT_SM PDU). In certain error situations the USSD GW may inform the USSD application that the mobile used is disconnected from the USSD session (via specific values of the *ussd_service_op* optional parameter in the DELIVER_SM PDU). See additional explanations embedded in 3.7.1 and 3.8.1.

2.1. - SMPP+ for USSD Phase 1

2.1.1. - MO USSD

The mobile originated USSD Phase 1 session consists of PSSD sent from the mobile station through GSM USSD operations to the USSD GW and forwarded to the USSD application.

The following diagram shows the call flow:

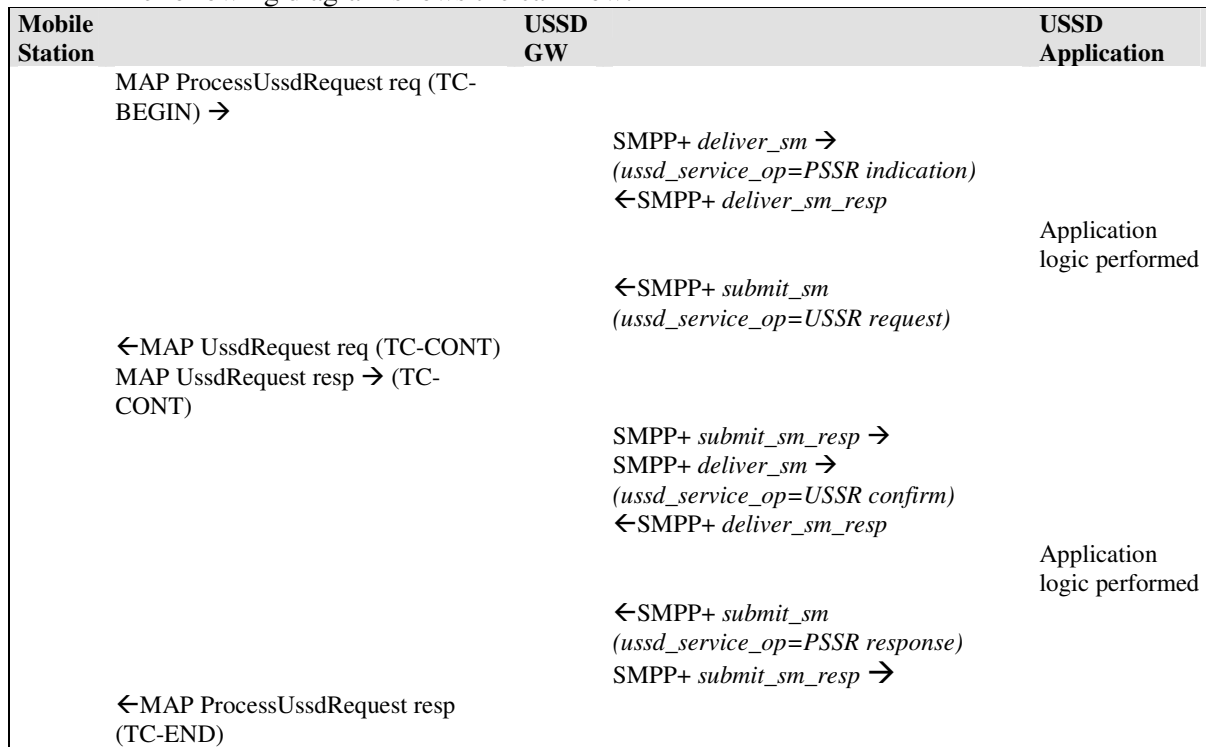


2.2. - SMPP+ for USSD Phase 2

2.2.1. - MO USSD

The mobile originated USSD Phase 2 session consists of a PSSR set from the mobile station to the USSD application through the USSD GW. The application can either close the session with a response or request for additional information by issuing a USSR request.

The following diagram shows the call flow:



During this type of session there is a possibility that the USSD GW must disconnect the session based on MAP connectivity disconnection.

The following flow shows the usage of USSREL indication by the GW to signify this disconnection. In this message *deliver_sm* does not contain any *short_message* – the USSREL only identifies session disconnection.

Mobile Station	USSD GW	USSD Application
MAP ProcessUssdRequest req (TC-BEGIN) →	SMPP+ <i>deliver_sm</i> → (<i>ussd_service_op</i> =PSSR indication) ←SMPP+ <i>deliver_sm_resp</i>	Application logic performed
←MAP UssdRequest req (TC-CONT) MAP UssdRequest resp → (TC-CONT)	←SMPP+ <i>submit_sm</i> (<i>ussd_service_op</i> =USSR request)	
	SMPP+ <i>submit_sm_resp</i> → SMPP+ <i>deliver_sm</i> → (<i>ussd_service_op</i> =USSR confirm) ←SMPP+ <i>deliver_sm_resp</i>	Application logic in process
MAP close (TC-END / TC-ABORT)→	SMPP+ <i>deliver_sm</i> → (<i>ussd_service_op</i> =USSREL indication) ←SMPP+ <i>deliver_sm_resp</i>	

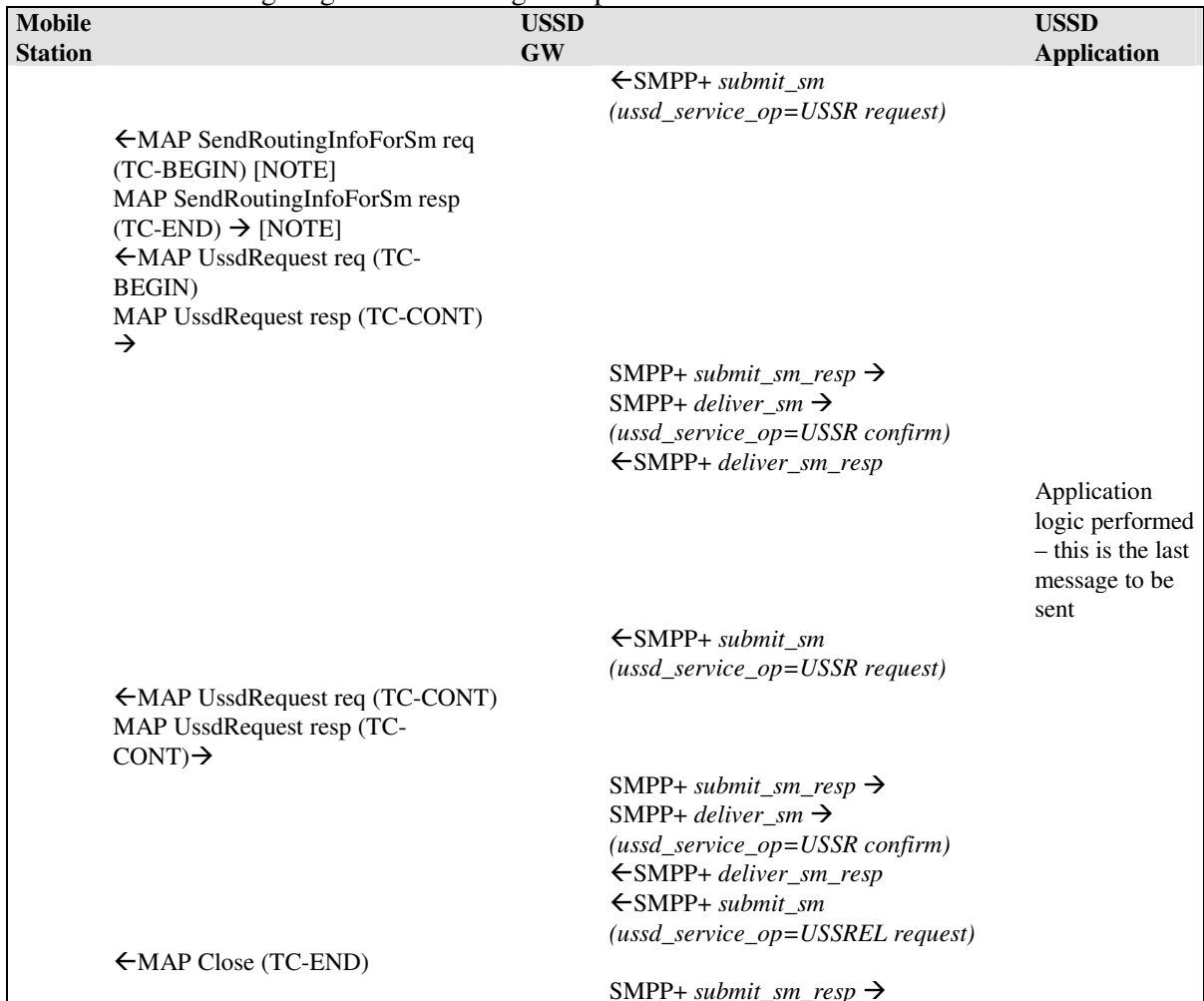
2.2.2. - MT USSD

The mobile terminated USSD Phase 2 session starts with a USSR/USSN sent from the USSD application through the USSD GW to the mobile station. Based on the response of the mobile subscriber the session may continue with another USSR/USSN query. The application decides when the mobile user response terminates the session.

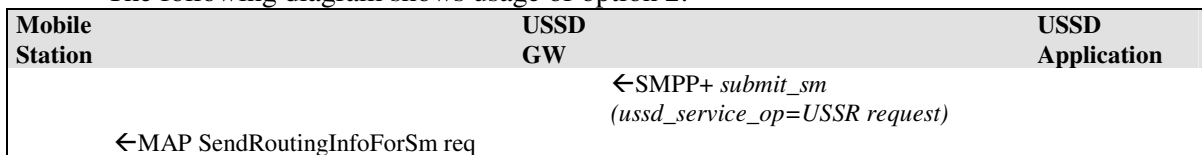
The application can terminate with several options:

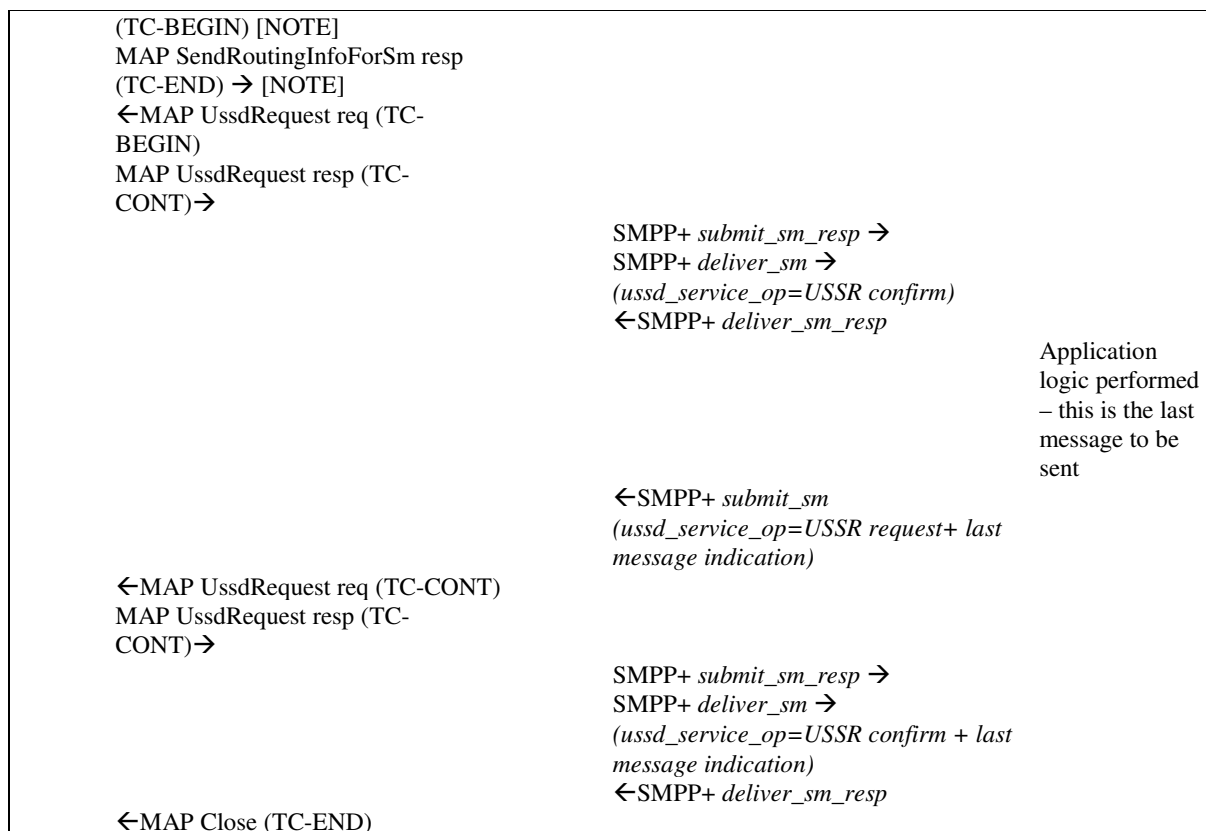
1. Send a “USSREL request” – this enables the application to close the session without sending a message to the mobile subscriber
2. Send a “USSR request + last message indication” – this instructs the USSD GW to close the session after the response to the “USSR request” is received. In this case the USSD GW will forward the response in a “USSR confirm + last message indication” deliver_sm message.
3. Send a “USSN request + last message indication” – this instructs the USSD GW to close the session after the confirmation from the mobile as been received.

The following diagram shows usage of option 1:

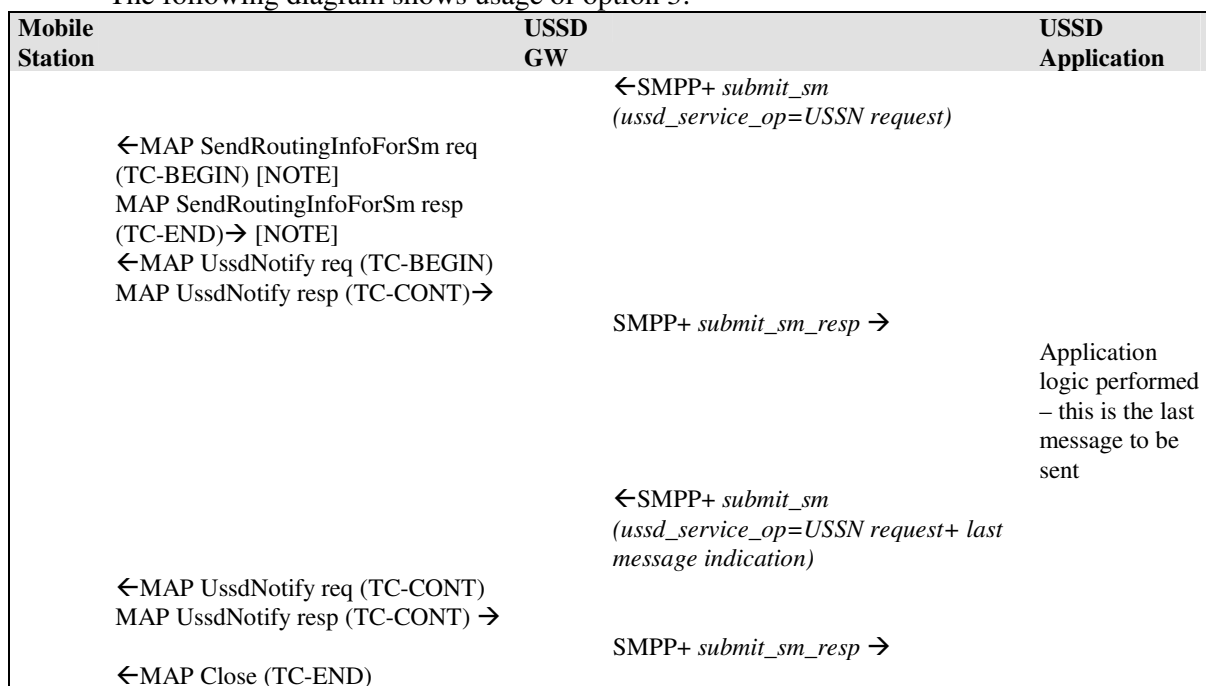



The following diagram shows usage of option 2:





The following diagram shows usage of option 3:



 **NOTE** : This transaction (IMSI query - SendRoutingInfoForSm) may be skipped if the network (HLR) does not require IMSI as Destination Reference in the initial UssdRequest/UssdNotify message.

An important error situation to consider in the MT USSD is when the application initiates a session but the USSD GW is unable to fulfill the request (internal or network problems). In this case – the error is indicated in the response to the submit_sm message and the session is not initiated.

Mobile Station	USSD GW	USSD Application
	← SMPP+ <i>submit_sm</i> (<i>ussd_service_op</i> =USSN request) Error SMPP+ <i>submit_sm_resp</i> (<i>command_status</i> = <i>ESME_RUSSDSESSIONTERMABN</i>) →	

3. - SMPP+ PROTOCOL DESCRIPTION

3.1. - General aspects of the SMPP+ protocol

SMPP+ is based on the SMPPv3.4 protocol.

SMPP+ and SMPPv3.4 share:

- the same encoding/decoding rules
- the same constants (command IDs, command status, ...)
- the same operations, parameters, type of parameters

SMPP+ only makes additions to the protocol (new constants, TLVs...) or change the meaning of some attributes or operations.

3.2. - Network level connection

The underlying transport interface between the SMSC and ESME should be based on a TCP/IP network connection. SMPP+ is an application layer protocol and is not intended to offer transport functionality. It is therefore assumed that the underlying network connection will provide reliable data transfer from point to point including packet encoding, windowing, flow control and error handling.

Thus, at the SMPP+ level, the ESME and SMSC should treat the network connection as a reliable transport which manages delivery and receipt of SMPP PDUs.

Connection between the USSD GW and ESMEs is established over a TCP/IP Network.

TCP/IP connection is always initiated by ESME. ESME don't accept incoming SMPP+ connection from the USSD GW (the SMPP OUTBIND operation is not supported). ESMEs are always on the client side of the TCP/IP connection. Connection is established to the USSD GW SMPP interface specific port and IP address.

The USSD GW must accept multiple TCP/IP connections from equipment that hosts multiple ESMEs. Moreover there may be multiple connections per *system_id* (unique identifier of the ESME set in the BIND operation).

An application may choose one of 2 binding options:

- One or more connections of TRX bind option (transceiver)
- One or more pairs of TX & RX connection combinations (transmitter and receiver)

The USSD GW contains a USSD traffic routing algorithm. Each ESME is configured with a specific USSD traffic routing criteria. In case multiple ESMEs are configured with the same routing criteria the USSD GW SMPP interface shall provide load sharing between them.

3.3. - Timers definitions

SMPP+ supports the same timers as defined in SMPPv3.4 (for detailed description see section 7.2).

SMPP+ gives the following recommended values for these timers:

Timer	Value on the USSD GW	Recommended value on the ESME	Notes
			all timeout value must be tunable by ESME
<i>session_init_timer</i>	2 seconds	-	This timer is not active on an ESME. USSD GW waits for bidirectional BIND in order to cancel the timer.
<i>enquire_link_timer</i>	60 seconds	60 seconds	
<i>inactivity_timer</i>	-	60 seconds	<i>enquire_link</i> requests are used to avoid this timer to be reached. USSD GW does not use this timer.
<i>response_timer</i>	Up to 30 seconds	Up to 1800 seconds	USSD requests trigger user interaction on the mobile station, thus may take a long time to complete.

3.4. - BIND Operation

The purpose of the SMPP+ bind operation is to register an instance of an ESME with the USSD GW system and request an SMPP+ session over this network connection for the submission or delivery of USSD operations. Thus, the Bind operation may be viewed as a form of USSD GW login request to authenticate the ESME entity wishing to establish a connection.

An ESME can bind to the USSD GW in one of the following ways:

- The ESME will use the SMPP+ BIND_TRANSMITTER and BIND_RECEIVER operations. The ESME must use both operations on two different connections to bind to the USSD GW. A single connection will leave that connection blocked until another connection is bound.
- The ESME will bind using a BIND_TRANCEIVER on a single connection

Dynamic registration via the *address_range* parameter is not supported in the USSD GW.

3.4.1. - "BIND_TRANSMITTER" PDU

SMPP+ BIND_TRANSMITTER is fully compliant with SMPPv3.4 BIND_TRANSMITTER

An ESME bound as a Transmitter is authorized to send USSD operations to the USSD GW and to receive the corresponding SMPP responses from the USSD GW.

BIND_TRANSMITTER is always sent by the ESME.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x00000002 (<i>bind_transmitter</i>)
	<i>command_status</i>	NULL
	<i>sequence_number</i>	set to a session unique monotonically increasing sequence number (0x00000001 to 0x7FFFFFFF)
B O D Y	<i>system_id</i>	set to the string identifying the ESME to the USSD GW
	<i>password</i>	set to the password string associated with the <i>system_id</i> on the USSD GW
	<i>system_type</i>	NULL (ignored by the USSD GW SMPP interface)
	<i>interface_version</i>	0x34
	<i>addr_ton</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)
	<i>addr_npi</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)
	<i>address_range</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)

Notes:

- The *system_id* is unique per ESME. It is defined in the USSD GW configuration.
- The *password* is associated with the *system_id* and is defined in the USSD GW configuration.

3.4.2. - "BIND_TRANSMITTER_RESP" PDU

SMPP+ BIND_TRANSMITTER_RESP is fully compliant with SMPPv3.4 BIND_TRANSMITTER_RESP.

BIND_TRANSMITTER_RESP is always sent by the USSD GW.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x80000002 (<i>bind_transmitter_resp</i>)
	<i>command_status</i>	Indicate status (success or error) of original <i>bind_transmitter</i> request.
	<i>sequence_number</i>	set to sequence number of original <i>bind_transmitter</i> request.
B O D Y	<i>system_id</i>	set to the string identifying the USSD GW to the ESME
	OPTIONAL PARAMETERS for BIND_TRANSMITTER_RESP	
	<i>sc_interface_version</i>	0x34

According to SMPPV3.4, SMPP+ assumes that the body portion is not present if the *command_status* field contains non-zero value (if it is present, it will be ignored).

3.4.3. - "BIND RECEIVER" PDU

ESME Receiver

An ESME bound as a Receiver is authorized to receive USSD operations from the USSD GW and to return the corresponding USSD messages responses to the USSD GW.

SMPP+ BIND_RECEIVER fully compliant with SMPP v3.4 BIND_RECEIVER.

BIND_RECEIVER is always sent by the ESME.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x00000001 (<i>bind_receiver</i>)
	<i>command_status</i>	NULL
	<i>sequence_number</i>	set to a session unique monotonically increasing sequence number (0x00000001 to 0x7FFFFFFF)
B O D Y	<i>system_id</i>	set to the string identifying the ESME to the USSD GW
	<i>password</i>	set to the password string associated with the <i>system_id</i> on the USSD GW
	<i>system_type</i>	NULL (ignored by the USSD GW SMPP interface)
	<i>interface_version</i>	0x34
	<i>addr_ton</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)
	<i>addr_npi</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)
	<i>address_range</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)

Notes:

- The *system_id* is unique per ESME. It is defined in the USSD GW configuration.
- The *password* is associated with the *system_id* and is defined in the USSD GW configuration.

3.4.4. - "BIND RECEIVER_RESP" PDU

SMPP+ BIND_RECEIVER_RESP is fully compliant with SMPPv3.4 BIND_RECEIVER_RESP.

BIND_RECEIVER_RESP is always sent by the USSD GW.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x80000001 (<i>bind_receiver_resp</i>)
	<i>command_status</i>	Indicate status (success or error) of original <i>bind_receiver</i> request.
	<i>sequence_number</i>	set to sequence number of original <i>bind_receiver</i> request.
B O D Y	<i>system_id</i>	set to the string identifying the USSD GW to the ESME
	OPTIONAL PARAMETERS for BIND_RECEIVER_RESP	
	<i>sc_interface_version</i>	0x34

According to SMPPV3.4, SMPP+ assumes that the body portion is not present if the *command_status* field contains non-zero value (if it is present, it will be ignored).

3.4.5. - "BIND TRANSCEIVER" PDU

ESME Transceiver

An ESME bound as a Transceiver is authorized to receive USSD operations from the USSD GW and to return the corresponding USSD messages responses to the USSD GW. It is also authorized to send USSD operations to the USSD GW and receive responses.

SMPP+ BIND_TRANSCEIVER fully compliant with SMPP v3.4 BIND_TRANSCEIVER. BIND_TRANSCEIVER is always sent by the ESME.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x00000009 (<i>bind_transceiver</i>)
	<i>command_status</i>	NULL
	<i>sequence_number</i>	set to a session unique monotonically increasing sequence number (0x00000001 to 0x7FFFFFFF)
B O D Y	<i>system_id</i>	set to the string identifying the ESME to the USSD GW
	<i>password</i>	set to the password string associated with the <i>system_id</i> on the USSD GW
	<i>system_type</i>	NULL (ignored by the USSD GW SMPP interface)
	<i>interface_version</i>	0x34
	<i>addr_ton</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)
	<i>addr_npi</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)
	<i>address_range</i>	NULL (defined in the USSD GW SMPP interface ESME Configuration)

Notes:

- The *system_id* is unique per ESME. It is defined in the USSD GW configuration.
- The *password* is associated with the *system_id* and is defined in the USSD GW configuration.

3.4.6. - "BIND_TRANSCEIVER_RESP" PDU

SMPP+ BIND_TRANSCEIVER_RESP is fully compliant with SMPPv3.4 BIND_TRANSCEIVER_RESP.

BIND_TRANSCEIVER_RESP is always sent by the USSD GW.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x80000009 (<i>bind_transceiver_resp</i>)
	<i>command_status</i>	Indicate status (success or error) of original <i>bind_transceiver</i> request.
	<i>sequence_number</i>	set to sequence number of original <i>bind_transceiver</i> request.
B O D Y	<i>system_id</i>	set to the string identifying the USSD GW to the ESME
	OPTIONAL PARAMETERS for BIND_TRANSCEIVER_RESP	
	<i>sc_interface_version</i>	0x34

According to SMPPV3.4, SMPP+ assumes that the body portion is not present if the *command_status* field contains non-zero value (if it is present, it will be ignored).

3.5. - UNBIND Operation

The purpose of the SMPP *unbind* operation is to deregister an instance of an ESME from the USSD GW and to inform the USSD GW that the ESME no longer wishes to use this network connection.

3.5.1. - "UNBIND" PDU

SMPP+ UNBIND is fully compliant with SMPPv3.4 UNBIND.

UNBIND is either sent by the USSD GW or the ESME.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x00000006 (<i>unbind</i>)
	<i>command_status</i>	NULL
	<i>sequence_number</i>	set to a session unique monotonically increasing sequence number (0x00000001 to 0x7FFFFFFF)

3.5.2. - "UNBIND_RESP" PDU

SMPP+ UNBIND_RESP is fully compliant with SMPPv3.4 UNBIND_RESP.

UNBIND_RESP is either sent by the USSD GW or the ESME.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x80000006 (<i>unbind</i>)
	<i>command_status</i>	indicate outcome of original <i>unbind</i> request
	<i>sequence_number</i>	set to sequence number of original <i>unbind</i> request.

The ESME/USSD GW will consider the *unbind* as successful regardless of the *command_status* value, and will terminate the network level connection.

3.6. - GENERIC_NACK PDU

This is a generic negative acknowledgement to an SMPP PDU submitted with an invalid message header.

3.6.1. - "GENERIC_NACK" PDU

SMPP+ GENERIC_NACK is fully compliant with SMPPv3.4 GENERIC_NACK.

GENERIC_NACK is either sent by the USSD GW or the ESME.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x80000000 (<i>generic_nack</i>)
	<i>command_status</i>	Error code corresponding to reason for sending the <i>generic_nack</i> .
	<i>sequence_number</i>	set to sequence number of original PDU or to NULL if the original PDU cannot be decoded.


Possible values for *command_status* are:

ESME_RINVMSGLEN : Message Length is invalid
 ESME_RINVCMDID : Invalid Command ID

3.7. - SUBMIT_SM Operation

The *submit_sm* operation is used by an ESME to submit a USSD operation to the USSD GW for onward transmission to a specified mobile station (MS).

3.7.1. - "SUBMIT_SM" PDU

SMPP+ SUBMIT_SM differs from SMPPv3.4 SUBMIT_SM by the meaning of several fields. Differences are highlighted with a warning .

SUBMIT_SM is always sent by the ESME.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x00000004 (<i>submit_sm</i>)
	<i>command_status</i>	NULL
	<i>sequence_number</i>	set to a session unique monotonically increasing sequence number (0x00000001 to 0x7FFFFFFF)
M A N D A T O R Y P A R A M E T E R	<i>service_type</i>	The <i>service_type</i> parameter can be used to indicate the SMS Application service associated with the message. Specifying the <i>service_type</i> allows the ESME to: <ul style="list-style-type: none"> • avail of enhanced messaging services such as “replace by service” type • To control the teleservice used on the air interface. The value of “USSD” shall be used.
	<i>source_addr_ton</i>	Type of Number for source address. Ignored by USSD GW, set to 0 (Unknown).
	<i>source_addr_npi</i>	Numbering Plan Indicator for source address. Ignored by USSD GW, set to 0 (Unknown).
	<i>source_addr</i>	Address of SME which originated this message. (may include sub-address digit) Ignored by USSD GW, set to NULL (Unknown).
	<i>dest_addr_ton</i>	Type of Number for destination address. In first application initiating session set to subscriber ton, otherwise set to 0 (Unknown).
	<i>dest_addr_npi</i>	Numbering Plan Indicator for destination address. In first application initiating session set to subscriber npi, otherwise set to 0 (Unknown).

S	<i>destination_addr</i>	Destination address of this USSD session (may include sub-address digit) In first application initiating session set to subscriber address, otherwise set to NULL (Unknown).
	<i>esm_class</i>	Indicates Message Mode & Message Type. ▼ 0x00 shall be used
	<i>protocol_id</i>	Protocol Identifier (if known). ▼ 0x00 shall be used
	<i>priority_flag</i>	Designates the priority level of the message. ▼ 0x00 shall be used
	<i>schedule_delivery_time</i>	The short message is to be scheduled by the SMSC for delivery. ▼ NULL shall be used
	<i>validity_period</i>	The validity period of this message. ▼ NULL shall be used
	<i>registered_delivery</i>	Indicator to signify if an SMSC delivery receipt or an SME acknowledgement is required. ▼ 0x00 shall be used.
	<i>replace_if_present</i>	Flag indicating if submitted message should replace an existing message. ▼ 0x00 shall be used
	<i>data_coding</i>	Defines the encoding scheme of the USSD operation user data (<i>short_message</i>). ▼ IA5, 7-bit character set (English), 8-bit binary and 16-bit (UCS2) coding schemes are supported.
	<i>sm_default_msg_id</i>	Indicates the short message to send from a list of predefined short messages stored on the SMSC. Set to NULL.
	<i>sm_length</i>	length in octets of the <i>short_message</i> user data
	<i>short_message</i>	USSD Phase 1 supports up to 200 octets of <i>short_message</i> and USSD Phase 2 supports up to 160 octets.

The following Optional parameters are relevant for the USSD interface:

OPTIONAL PARAMETERS for SUBMIT_SM		
	Field Name	Value / Description

ussd_service_op	<p>The <i>ussd_service_op</i> parameter is required to define the USSD service operation when SMPP is being used as an interface to a (GSM) USSD system.</p> <p>Possible values in <i>submit_sm</i>:</p> <ul style="list-style-type: none"> PSSD response PSSR response USSR request USSN request, USSREL request USSR request + last message indication USSN request + last message indication. <p>▼ The USSREL request value denotes that the USSD application wishes to terminate the USSD session with the mobile station.</p> <p>The following values indicate session termination request from the application: PSSD response, PSSR response, USSREL, USSR request + last message indication, USSN request + last message indication.</p> <p>NOTE: In the USSR request + last message indication the application must wait for a USSR confirm + last message indication from the USSD GW.</p> <p>NOTE: last message indication values are allowed only in application originated USSD sessions. Mobile originated USSD session must terminate with PSSD response or PSSR response.</p>
▼ ussd_session_id	<p>▼ The session ID between the USSD GW and the USSD Application. When the USSD application starts a USSD session (by issuing a USSR request or USSN request for example) it does not include this parameter and receives an allocated session id from the USSD GW in the <i>submit_sm_resp</i> PDU.</p> <p>See Annex 2 for more details.</p>
language_indicator	Contains <i>language_indicator</i> as defined in [4] 4.10

Other SMPP optional fields are not set.

3.7.2. - "SUBMIT_SM_RESP" PDU

SMPP+ SUBMIT_SM_RESP differs from SMPPv3.4 SUBMIT_SM_RESP by the meaning of several fields.

SUBMIT_SM_RESP is always sent by the USSD GW.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x80000004 (<i>submit_sm_resp</i>)
	<i>command_status</i>	indicate outcome of original <i>submit_sm</i> request
	<i>sequence_number</i>	set to sequence number of original <i>submit_sm</i> request.
B O D Y	<i>message_id</i>	set to with the SMSC message ID associated with the submitted SMS. ▼ This parameter is not used by the USSD GW, set to NULL.
	OPTIONAL PARAMETERS for SUBMIT_SM_RESP	
	Field Name	Value / Description
	<i>ussd_session_id</i>	▼ USSD GW allocated session id. Returned when the <i>submit_sm</i> initiated a USSD session. See Annex 2 for more details.

According to SMPPV3.4, SMPP+ assumes that the body portion is not present if the *command_status* field contains non-zero value (if it is present, it will be ignored).

3.8. - DELIVER SM Operation

The *deliver_sm* is issued by the USSD GW to send a USSD operation to an ESME.
In addition the USSD GW may signal the discontinuation of a USSD session with this operation.

3.8.1. - "DELIVER SM" PDU

SMPP+ DELIVER_SM differs from SMPPv3.4 DELIVER_SM by the meaning of several fields and by adding two new optional parameters.

DELIVER_SM is always sent by the USSD GW.

Differences are highlighted with a warning ▼.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x00000005 (<i>deliver_sm</i>)
	<i>command_status</i>	NULL
	<i>sequence_number</i>	set to a session unique monotonically increasing sequence number (0x00000001 to 0x7FFFFFFF)
M A N D A T E	<i>service_type</i>	The <i>service_type</i> parameter can be used to indicate the SMS Application service associated with the message. The value of "USSD" shall be used
	<i>source_addr_ton</i>	Type of Number for source address. In first mobile initiating session message set to subscriber's ton. If not known, set to 0 (Unknown).

O R Y P A R A M E T E R S	<i>source_addr_npi</i>	Numbering Plan Indicator for source. In first mobile initiating session message set to subscriber's npi. If not known, set to 0 (Unknown).
	<i>source_addr</i>	Address of SME which originated this message. (may include sub-address digit) In first mobile initiating session message set to subscriber's address. (MSISDN). If not known, set to NULL (Unknown).
	<i>dest_addr_ton</i>	Type of number of destination SME. Not used by USSD GW, set to 0.
	<i>dest_addr_npi</i>	Numbering Plan Indicator of destination SME. Not used by USSD GW, set to 0.
	<i>destination_addr</i>	Destination address of this short message (may include sub-address digit) For mobile terminated messages, this is the number of the recipient MS. Not used by USSD GW., set to NULL (Unknown).
	<i>esm_class</i>	Indicates Message Type and enhanced network services. ▼ 0x00 shall be used
	<i>protocol_id</i>	Protocol Identifier ▼ 0x00 shall be used
	<i>priority_flag</i>	Designates the priority level of the message. ▼ 0x00 shall be used
	<i>schedule_delivery_time</i>	This field is unused for deliver_sm. ▼ NULL shall be used
	<i>validity_period</i>	This field is unused for deliver_sm ▼ NULL shall be used
	<i>registered_delivery</i>	Indicates if an ESME acknowledgement is required. ▼ 0x00 shall be used.
	<i>replace_if_present</i>	Not used in deliver_sm. ▼ 0x00 shall be used
	<i>data_coding</i>	Defines the encoding scheme of the USSD operation user data (<i>short_message</i>). ▼ IA5, 7-bit character set (English), 8-bit binary and 16-bit (UCS2) coding schemes are supported.
	<i>sm_default_msg_id</i>	Unused in deliver_sm. Set to NULL.
	<i>sm_length</i>	length in octets of the <i>short_message</i> user data
	<i>short_message</i>	Up to 200 octets of short message user data using the specified encoding scheme. USSD Phase 1 supports up to 200 octets of <i>short_message</i> and USSD Phase 2 supports up to 160 octets. Note: The message data should be inserted only in the <i>short_message</i>

OPTIONAL PARAMETERS for DELIVER_SM

Field Name	Value / Description
------------	---------------------

O P T I O N A L P A R A M E T E R S	ussd_service_op	<p>The <i>ussd_service_op</i> parameter is required to define the USSD service operation when SMPP is being used as an interface to a (GSM) USSD system.</p> <p>Possible values in <i>deliver_sm</i>:</p> <ul style="list-style-type: none"> PSSD indication PSSR indication USSR confirm USSN confirm USSREL indication USSR confirm + last message indication. <p>▼ The USSREL indication value denotes that the USSD GW wishes to terminate the USSD session with the USSD application.</p> <p>The USSD GW informs the USSD application that the USSD session is terminated using USSREL indication or USSR confirm + last message indication</p>
	▼ussd_session_id	<p>▼ The session ID between the USSD GW and the USSD Application.</p> <p>See Annex 2 for more details.</p>
	language_indicator	Contains <i>language_indicator</i> as defined in [4] 4.10
	imsi	<p>Contains the International Mobile Station Identity – a subscriber identity as defined in the mobile network.</p> <p>May be present only in the first mobile originated session message.</p>
	hlr_address_ton	<p>Contains the type of number of the HLR address.</p> <p>May be present only in the first mobile originated session message.</p>
	hlr_address_npi	<p>Contains the numbering plan indicator of the HLR address.</p> <p>May be present only in the first mobile originated session message.</p>
	hlr_address	<p>Contains the HLR address</p> <p>May be present only in the first mobile originated session message.</p>
	vlr_number_ton	<p>Contains the type of number of the VLR number.</p> <p>May be present only in the first mobile originated session message.</p>
	vlr_number_npi	<p>Contains the numbering plan indicator of the VLR number.</p> <p>May be present only in the first mobile originated session message.</p>

	vlr_number_addr	Contains the VLR number May be present only in the first mobile originated session message.
--	-----------------	--

Other SMPP optional fields are ignored.

3.8.2. - "DELIVER SM RESP" PDU

SMPP+ DELIVER_SM_RESP differs from SMPPv3.4 DELIVER_SM_RESP by the meaning of several fields.

DELIVER_SM_RESP is always sent by the ESME.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x80000005 (<i>deliver_sm_resp</i>)
	<i>command_status</i>	indicate outcome of original <i>deliver_sm</i> request.
	<i>sequence_number</i>	set to sequence number of original <i>deliver_sm</i> request.
B O D Y	<i>message_id</i>	This field is unused and is set to NULL.

The following standard *command_status* codes should also be returned:

Error	Description
ESME_ROK	No error
ESME_RSYSERR	System error

3.9. - ENQUIRE LINK Operation

This message can be sent either the ESME or USSD GW and is used to provide a confidence-check of the communication path between an ESME and an USSD GW. On receipt of this request the receiving party should respond with an *enquire_link_resp*, thus verifying that the application level connection between the USSD GW and the ESME is functioning.

3.9.1. - "ENQUIRE LINK" PDU

SMPP+ ENQUIRE_LINK is fully compliant with SMPPv3.4 ENQUIRE_LINK.

Fields are set as defined in the following table:

H E A D	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x00000015 (<i>enquire_link</i>)

E R	<i>command_status</i>	NULL
	<i>sequence_number</i>	set to a session unique monotonically increasing sequence number (0x00000001 to 0x7FFFFFFF)

3.9.2. - "ENQUIRE_LINK_RESP" PDU

SMPP+ ENQUIRE_LINK_RESP is fully compliant with SMPPv3.4 ENQUIRE_LINK_RESP.

Fields are set as defined in the following table:

H E A D E R	Field Name	Value / Description
	<i>command_length</i>	set to the overall length of the PDU
	<i>command_id</i>	0x80000015 (<i>enquire_link_resp</i>)
	<i>command_status</i>	set to ESME_ROK (Success)
	<i>sequence_number</i>	set to sequence number of original <i>enquire_link</i> request.

3.10. - Others SMPP v3.4 PDU

Other SMPPv3.4 PDU are not supported.

Whenever an ESME receives an unrecognized/unsupported PDU, it will return a *generic_nack* PDU.

4. - ANNEXES

4.1. - ANNEX 1: List OF SPECIFIC SMPP+ ERROR CODES

The following error codes are addition to the SMPPv3.4 specification. All additional codes are defined in the "Vendor specific block".

Error	Code Value	Description
ESME_RUSSDINVLOGIC	0x4B1	Invalid USSD service logic requested
EMSE_RUSSDUNSUPPOPER	0x4B2	An SMPP PDU which is not supported in USSD GW was received
ESME_RUSSDGMSYSFAIL	0x4B3	System Failure error returned from GSM
ESME_RUSSDGSMDATAMISS	0x4B4	Data Missing error returned from GSM
ESME_RUSSDGSMUNEXPDATAVAL	0x4B5	Unexpected Data Value error returned from GSM
ESME_RUSSDGSMUNKNWALPH	0x4B6	Unknown Alphabet error returned from GSM
ESME_RUSSDGSMABSSUB	0x4B7	Absent Subscriber error returned from GSM
ESME_RUSSDGSMILLSUB	0x4B8	Illegal Subscriber error returned from GSM
ESME_RUSSDGSMILLEQUP	0x4B9	Illegal Equipment error returned from GSM
ESME_RUSSDGSMBUSY	0x4BA	USSD Busy error returned from GSM
ESME_RUSSDGSMCALLBAR	0x4BB	Call Barred error returned from GSM
ESME_RUSSDGSMNETFAIL	0x4BC	Network Failure error returned from GSM
ESME_RUSSDGSMTIMEXP	0x4BD	Timeout Expired error returned from

Error	Code Value	Description
		GSM
ESME_RUSSDGSMFACNSUP	0x4BE	Facility Not Supported error returned from GSM
ESME_RUSSDGSMMUNKNSUB	0x4BF	Unknown Subscriber error returned from GSM
ESME_RUSSDGSMTLSRVNPRV	0x4C0	Teleservice Not Provisioned error returned from GSM
ESME_RUSSDSESSIONTERMABN	0x4C1	USSD Session Terminated Abnormally at the responding entity (e.g. ESME in deliver_sm_resp or USSD GW in submit_sm_resp)

4.2. - ANNEX 2: List OF SPECIFIC SMPP+ ATTRIBUTES

The following attribute codes are addition to the SMPPv3.4 specification. All additional codes are defined in the "Vendor specific block".

Tag	Value	Description
<i>ussd_session_id</i>	0x1501	Int type - See 4.2.2. - for full details.
<i>imsi</i>	0x1502	C-Octet String type – see 4.2.3. -for full details
<i>hlr_address_ton</i>	0x1800	Int Type – see 4.2.4. - for full details
<i>hlr_address_npi</i>	0x1801	Int Type4.2.5. - for full details
<i>hlr_address</i>	0x1802	C-Octet String type – see 4.2.6. - for full details
<i>vlr_number_ton</i>	0x1503	Int Type – see 4.2.7. - for full details
<i>vlr_number_npi</i>	0x1504	Int Type – see 4.2.8. - for full details
<i>vlr_number_addr</i>	0x1505	C-Octet String type – see 4.2.9. - for full details

The following sections contain SMPP+ specific definitions and usage of non-standard or extended SMPP parameters.

4.2.1. - ussd_service_op

The *ussd_service_op* parameter is required to define the USSD service operation when SMPP is being used as an interface to a (GSM) USSD system.

Field	Size octets	Type	Description
Parameter Tag	2	Integer	ussd_service_op (0x0501)
Length	2	Integer	Length of Value part in octets
Value	1	Octet String	0 = PSSD indication 1 = PSSR indication 2 = USSR request 3 = USSN request 4 to 15 = reserved 16 = PSSD response 17 = PSSR response

			<p>18 = USSR confirm 19 = USSN confirm 20 to 31 = reserved 32 to 255 = reserved for vendor specific USSD operations:</p> <ul style="list-style-type: none"> ▼ 128 = USSREL request ▼ 129 = USSREL indication ▼ 130 = USSR request + last message indication ▼ 131 = USSN request + last message indication ▼ 146 = USSR confirm + last message indication <p>When USSREL is used in <i>ussd_service_op</i> the <i>short_message</i> should be ignored. It is used only to signify to the receiving entity that the session is terminated.</p> <p>When last message indication is set, it means that this is the last submit_sm / deliver_sm for this session.</p> <p>NOTE: last message indication values are allowed only in application originated USSD sessions. Mobile originated USSD session must terminate with PSSD response or PSSR response.</p> <p>NOTE: The values noted above a numeric; however the parameter type is Octet-String. Thus, for example, encoding of PSSD response (decimal 16) is 0x10.</p>
--	--	--	---

4.2.2. - ussd_session_id

This parameter specifies a unique session ID assigned by the USSD GW. In MT sessions (started by application) the parameter is not included in initiating submit_sm PDU and the allocated *ussd_session_id* is returned to the application in the submit_sm_resp PDU. All subsequent submit/deliver messages within the same USSD session must include this parameter with the value assigned by the USSD GW.

A USSD session starts with PSSD/PSSR indication (MO USSD) or USSR/USSN request (MT USSD).

The USSD session ends with PSSD/PSSR response MO USSD.

USSREL request/indication terminates the session in both MO/MT USSD.

A USSD session also ends when a USSR request – USSR confirm (when last message indication used) cycle ends or upon USSN request + last message indication response. See sections 2.1 and 2.2 for detailed flows.

Field	Size octets	Type	Description
Parameter Tag	2	Integer	ussd_session_id (0x1501)

Length	2	Integer	Length of Value part in octets
Value	4	Integer	Session id range is between 0 and 16,777,215. It is allocated by the USSD GW. When the USSD application initiates a USSD session it does not include the parameter in the submit_sm optional part. The USSD GW returns the session id in the submit_sm_resp.

Notes:

If a specific ESME has multiple connections to the USSD GW, the *ussd_session_id* should be unique across all the connections. Also, a *ussd_session_id* is not bound to a specific connection and the session may continue on a different connection. This enables load sharing between the multiple connections.

4.2.3. - *imsi*

Field	Size octets	Type	Description
Parameter Tag	2	Integer	imsi (0x1502)
Length	2	Integer	Length of Value part in octets
Value	Var. max 15	C -Octet String	Contains the IMSI of the mobile subscriber

4.2.4. - *hlr_address_ton*

Field	Size octets	Type	Description
Parameter Tag	2	Integer	hlr_address_ton (0x1800)
Length	2	Integer	Length of Value part in octets
Value	1	Integer	Contains the type of number of the HLR address. It is encoded exactly as defined in SMPP 3.4 [1] 5.2.5

4.2.5. - *hlr_address_npi*

Field	Size octets	Type	Description
Parameter Tag	2	Integer	hlr_address_npi (0x1801)
Length	2	Integer	Length of Value part in octets
Value	1	Integer	Contains the type of number of the HLR address. It is encoded exactly as defined in SMPP 3.4 [1] 5.2.6

4.2.6. - *hlr_address*

Field	Size	Type	Description
-------	------	------	-------------

	octets		
Parameter Tag	2	Integer	hlr_address (0x1802)
Length	2	Integer	Length of Value part in octets
Value	Var. max 21	C-Octet String	Contains the HLR address.

4.2.7. - vlr_number_ton

Field	Size octets	Type	Description
Parameter Tag	2	Integer	vlr_number_ton (0x1503)
Length	2	Integer	Length of Value part in octets
Value	1	Integer	Contains the type of number of the VLR number. It is encoded exactly as defined in SMPP 3.4 [1] 5.2.5

4.2.8. - vlr_number_npi

Field	Size octets	Type	Description
Parameter Tag	2	Integer	vlr_number_npi (0x1504)
Length	2	Integer	Length of Value part in octets
Value	1	Integer	Contains the type of number of the VLR number. It is encoded exactly as defined in SMPP 3.4 [1] 5.2.6

4.2.9. - vlr_number_addr

Field	Size octets	Type	Description
Parameter Tag	2	Integer	vlr_number_addr (0x1505)
Length	2	Integer	Length of Value part in octets
Value	Var. max 21	C-Octet String	Contains the VLR number address.

4.3. - ANNEX 3: SMPP+ to GSM MAP USSD messages/parameters mapping

4.3.1. - DELIVER_SM

4.3.1.1. - DELIVER_SM (ussd_service_op=PSSD indication)

DELIVER_SM (ussd_service_op=PSSD indication)		MAP BeginSubscriberActivity (request)
source_addr_ton source_addr_npi source_addr	←	OriginationEntityNumber or MSISDN <ERC>

<i>imsi</i>	←	<i>IMSI</i>
<i>hlr_address_ton</i> <i>hlr_address_npi</i> <i>hlr_address</i>	←	<i>HLR Address <ERC></i>
<i>vlr_number_ton</i> <i>vlr_number_npi</i> <i>vlr_number_addr</i>	←	<i>VLR Number <ERC></i>

DELIVER_SM (ussd_service_op=PSSD indication)		MAP ProcessUnstructuredSsData (request)
<i>sm_length</i> <i>short_message</i> <i>data_coding=0x01 (IA5)</i>	←	<i>SS-UserData</i>

4.3.1.2. - *DELIVER_SM (ussd_service_op=PSSR indication)*

DELIVER_SM (ussd_service_op=PSSR indication)		MAP OPEN (indication)
<i>imsi</i>	←	<i>DestinationReference (Note 3)</i>
<i>source_addr_ton</i> <i>source_addr_npi</i> <i>source_addr</i>	←	<i>DestinationReference (Note 3)</i>
<i>hlr_address_ton</i> <i>hlr_address_npi</i> <i>hlr_address</i>	←	<i>Origination Reference</i>
<i>vlr_number_ton</i> <i>vlr_number_npi</i> <i>vlr_number_addr</i>	←	<i>VLR Number <ERC></i>
<i>source_addr_ton</i> <i>source_addr_npi</i> <i>source_addr</i>	←	<i>MSISDN <ERC> (Note3)</i>

DELIVER_SM (ussd_service_op=PSSR indication)		MAP ProcessUnstructuredSsRequest (request)
<i>source_addr_ton</i> <i>source_addr_npi</i> <i>source_addr</i>	←	<i>MSISDN (Note 3)</i>
<i>data_coding (Note1)</i> <i>language_indicator (Note2)</i>	←	<i>ussd data coding scheme</i>
<i>sm_length</i> <i>short_message</i> <i>language_indicator (Note2)</i>	←	<i>ussd string</i>

NOTE1: The only values in use are:

- 7 bit character set = 0x00 (formatted in LATIN-1 encoding, for mapping GSM default alphabet to LATIN-1 see http://www.dreamfabric.com/sms/default_alphabet.html)
- 8 bit character set = 0x04
- 16 bit character set (UCS2) = 0x08

NOTE2: Mapping and usage of *language_indicator* is defined in 4.4. -

NOTE3: MSISDN may arrive in one (or more) of the following locations.

- Destination Reference
- MSISDN <ERC>
- MSISDN in component portion

It can't appear in the first 2 locations at the same time.

One of the MSISDN locations is mapped to the source_addr according to a configurable setting. The Destination Reference may contain IMSI (configurable setting) and thus mapped to the imsi parameter. In this case MSISDN will be mapped from one of the other two locations.

4.3.1.3. - DELIVER SM (ussd_service_op=USSR confirm / USSD confirm + last message indication)

DELIVER_SM (ussd_service_op=USSR confirm [+ last message indication])		MAP UnstructuredSsRequest (response)
<i>data_coding</i> (NOTE1) <i>language_indicator</i> (NOTE2)	←	<i>ussd data coding scheme</i>
<i>sm_length</i> <i>short_message</i> <i>language_indicator</i> (NOTE2)	←	<i>ussd string</i>

NOTE1: The only values in use are:

- 7 bit character set = 0x00 (formatted in LATIN-1 encoding, for mapping GSM default alphabet to LATIN-1 see http://www.dreamfabric.com/sms/default_alphabet.html)
- 8 bit character set = 0x04
- 16 bit character set (UCS2) = 0x08

NOTE2: Mapping and usage of *language_indicator* is defined in 4.4. -

4.3.1.4. - DELIVER SM (ussd_service_op=USSN confirm)

No mappings

4.3.1.5. - DELIVER SM (ussd_service_op=USSREL indication)

No mappings

4.3.2. - SUBMIT SM

4.3.2.1. - SUBMIT SM (ussd_service_op=PSSD response)

SUBMIT_SM (ussd_service_op=PSSD response)		MAP ProcessUnstructuredSsData (response)
<i>sm_length</i> <i>short_message</i> <i>data_coding=0x01</i> (IA5)	→	<i>SS-UserData</i>

4.3.2.2. - SUBMIT SM (ussd_service_op=PSSR response)

SUBMIT_SM (ussd_service_op=PSSR response)		MAP ProcessUnstructuredSsRequest (response)
<i>data_coding</i> (NOTE1) <i>language_indicator</i> (NOTE2)	→	<i>ussd data coding scheme</i>
<i>sm_length</i> <i>short_message</i> <i>language_indicator</i> (NOTE2)	→	<i>ussd string</i>

NOTE1: The only values in use are:

- 7 bit character set = 0x00 (formatted in LATIN-1 encoding, for mapping GSM default alphabet to LATIN-1 see http://www.dreamfabric.com/sms/default_alphabet.html)
- 8 bit character set = 0x04
- 16 bit character set (UCS2) = 0x08

NOTE2: Mapping and usage of *language_indicator* is defined in 4.4. -

4.3.2.3. - SUBMIT_SM (ussd_service_op=USSR request / USSR request + last message indication)

Retrieve IMSI for session establishment (only during session initiation)

SUBMIT_SM (ussd_service_op=USSR request [+last message indication])		MAP OPEN (request) sent with MAP SendRoutingInfoForSm (request)
<i>dest_addr_ton</i> <i>dest_addr_npi</i> <i>destination_addr</i>	→	<i>DestinationAddress</i>
		<i>OriginatingAddress (NOTE1)</i>

NOTE1: Taken from USSD GW configuration file

SUBMIT_SM (ussd_service_op=USSR request [+last message indication])		MAP SendRoutingInfoForSm (request)
<i>dest_addr_ton</i> <i>dest_addr_npi</i> <i>destination_addr</i>	→	<i>msisdn</i>

SUBMIT_SM (ussd_service_op=USSR request [+last message indication])		MAP OPEN (request) sent with MAP UnstructuredSsRequest (request) in first application initiating session message
<i>dest_addr_ton</i> <i>dest_addr_npi</i> <i>destination_addr</i>	→	<i>Destination Address and MSISDN <ERC></i> <i>Destination Reference (note1)</i>
		<i>Destination Reference (note1)</i>
		<i>Originating Address (note2)</i>
		<i>Origination Reference (note3)</i>

NOTE1: MSISDN may be set in one of the following locations.

Destination Reference

MSISDN <ERC>

The Destination Reference may include the IMSI based on Address Query result if defined in configuration (this nullifies the use of MSISDN in Destination Reference).

NOTE2: Taken from USSD GW configuration file

NOTE3: Taken from USSD GW configuration file

Actual Request

SUBMIT_SM (ussd_service_op=USSR request [+last message indication])		MAP UnstructuredSsRequest (request)
<i>data_coding (NOTE1)</i> <i>language_indicator (NOTE2)</i>	→	<i>ussd data coding scheme</i>
<i>sm_length</i> <i>short_message</i> <i>language_indicator (NOTE2)</i>	→	<i>ussd string</i>

NOTE1: The only values in use are:

- 7 bit character set = 0x00 (formatted in LATIN-1 encoding, for mapping GSM default alphabet to LATIN-1 see http://www.dreamfabric.com/sms/default_alphabet.html)
- 8 bit character set = 0x04
- 16 bit character set (UCS2) = 0x08

NOTE2: Mapping and usage of *language_indicator* is defined in 4.4. -

4.3.2.4. - SUBMIT_SM (ussd_service_op=USSN Request / USSN request + last message indication)

Retrieve IMSI for session establishment (only during session initiation)

SUBMIT_SM (ussd_service_op=USSN request [+last message indication])		MAP OPEN (request) sent with MAP SendRoutingInfoForSm (request)
<i>dest_addr_ton</i> <i>dest_addr_npi</i> <i>destination_addr</i>	→	<i>DestinationAddress</i>
		<i>OriginatingAddress (NOTE1)</i>

NOTE1: Filled with USSD GW configurable address

SUBMIT_SM (ussd_service_op=USSN request [+last message indication])		MAP SendRoutingInfoForSm (request)
<i>dest_addr_ton</i> <i>dest_addr_npi</i> <i>destination_addr</i>	→	<i>msisdn</i>

SUBMIT_SM (ussd_service_op=USSN request [+last message indication])		MAP OPEN (request) sent with MAP UnstructuredSsNotify (request) in first application initiating session message
<i>dest_addr_ton</i> <i>dest_addr_npi</i> <i>destination_addr</i>	→ → →	<i>Destination Address and</i> <i>MSISDN <ERC></i> <i>Destination Reference (note1)</i>
		<i>Destination Reference (note1)</i>
		<i>Originating Address (note2)</i>
		<i>Origination Reference (note3)</i>

NOTE1: MSISDN may be set in one of the following locations.

Destination Reference

MSISDN <ERC>

The Destination Reference may include the IMSI based on Address Query result if defined in configuration (this nullifies the use of MSISDN in Destination Reference).

NOTE2: Taken from USSD GW configuration file

NOTE3: Taken from USSD GW configuration file

Actual Request

SUBMIT_SM (ussd_service_op=USSN request [+last message indication])		MAP UnstructuredSsNotify (request)
<i>data_coding (NOTE1)</i> <i>language_indicator (NOTE2)</i>	→	<i>ussd data coding scheme</i>
<i>sm_length</i> <i>short_message</i> <i>language_indicator (NOTE2)</i>	→	<i>ussd string</i>

NOTE1: The only values in use are:

- 7 bit character set = 0x00 (formatted in LATIN-1 encoding, for mapping GSM default alphabet to LATIN-1 see http://www.dreamfabric.com/sms/default_alphabet.html)

- 8 bit character set = 0x04
- 16 bit character set (UCS2) = 0x08

NOTE2: Mapping and usage of *language_indicator* is defined in 4.4. -


4.3.2.5. - SUBMIT SM (ussd_service_op=USSREL request)


No mappings

4.4. - ANNEX 4: Parameters value mappings


4.4.1. - language indicator mappings


USSD message may contain USSD string with language identifier associated with either GSM default alphabet or UCS2 encodings (these language codes are taken from ISO 639 [3]). When present the following mapping applies. The SMPP *language_indicator* is based on TIA/EIA 136-710A [4] specification.

 **NOTE** When SMPP is translated into MAP and *data_coding* is set to 8 bit character set, *language_indicator* is ignored. The USSD Data Coding Scheme in the MAP layer will get the value of 0x44 in this case.

 **NOTE** When SMPP is translated into MAP and *data_coding* is set to 7 bit character set (LATIN-1) and *language_indicator* is one of the following European languages (German, English, Italian, Spanish, Dutch, Swedish, Danish, Portuguese, Finnish, Norwegian, Greek, Turkish, Hungarian, Polish, Czech), the mapping is done directly into the coding scheme and *short_message* is transferred without change.

For other languages – the 2 character prefix mapped in the table is added to the *short_message* transferred to MAP.

 **NOTE** When SMPP is translated into MAP and *data_coding* is set to 16 bit character set (UCS2) and *language_indicator* is used, the 2 character prefix mapped in the table is added to the *short_message* transferred to MAP.

 **NOTE** When *language_indicator* is not used the value inserted into the USSD Data Coding Scheme in the MAP layer is according to General Data Coding indication table – e.g. no compression and no message class meaning. The remaining bits are used to specify the coding as received in the *data_coding*. Thus values of 0x40 for Default Alphabet, 0x44 for 8 bit coding and 0x48 for UCS-2 encoding are used.

ISO 639 Prefix (2 char)	GSM MAP Language		SMPP <i>language_indicator</i>	SMPP Language
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	0	Unspecified
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	7	Mandarin
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	29	French Creole

ISO 639 Prefix (2 char)	GSM MAP Language		SMPP <i>language_indicator</i>	SMPP Language
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	31	Navaho
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	38	Miao (Hmong)
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	43	Kru
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	48	Formosan
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	50	Bosnian
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	52	Llocano
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	55	Flemish
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	56	Syrian
	<set encoding scheme to 0x40 if Latin-1 0x48 if UCS2>	<-	60	Cajun
<no prefix / general data coding>		->	0	Unspecified
AA	Afar	->	0	Unspecified
AB	Abkhazian	->	0	Unspecified
AF	Afrikaans	->	0	Unspecified
AM	Amharic	<->	61	Amharic
AR	Arabic	<->	19	Arabic
AS	Assamese	->	0	Unspecified
AY	Aymara	->	0	Unspecified
AZ	Azerbaijani	->	0	Unspecified
BA	Bashkir	->	0	Unspecified
BE	Byelorussian	->	0	Unspecified
BG	Bulgarian	->	0	Unspecified
BH	Bihari	->	0	Unspecified
BI	Bislama	->	0	Unspecified
BN	Bengali Bangla	<->	53	Bengali
BO	Tibetan	->	0	Unspecified
BR	Breton	->	0	Unspecified
CA	Catalan	->	0	Unspecified

ISO 639 Prefix (2 char)	GSM MAP Language		SMPP language_indicator	SMPP Language
CO	Corsican	->	0	Unspecified
CS	Czech	<->	36	Czech
CY	Welsh	->	0	Unspecified
DA	Danish	<->	54	Danish
DE	German	<->	4	German
DZ	Bhutani	->	0	Unspecified
EL	Greek	<->	24	Greek
EN	English American	<->	1	English
EO	Esperanto	->	0	Unspecified
ES	Spanish	<->	3	Spanish
ET	Estonian	->	0	Unspecified
EU	Basque	->	0	Unspecified
FA	Persian	<->	28	Persian
FI	Finnish	<->	46	Finnish
FJ	Fiji	->	0	Unspecified
FO	Faeroese	->	0	Unspecified
FR	French	<->	2	French
FY	Frisian	->	0	Unspecified
GA	Irish	->	0	Unspecified
GD	Gaelic Scots Gaelic	<->	15	Gaelic
GL	Galician	->	0	Unspecified
GN	Guarani	->	0	Unspecified
GU	Gujarati	<->	34	Gujarathi
HA	Hausa	->	0	Unspecified
HI	Hindi	<->	10	Hindi
HR	Croatian	<->	49	Croatian
HU	Hungarian	<->	32	Hungarian
HY	Armenian	<->	30	Armenian
IA	Interlingua	->	0	Unspecified
IE	Interlingue	->	0	Unspecified
IK	Inupiak	->	0	Unspecified
IN	Indonesian	<->	9	Bahasa - Indonesia
IS	Icelandic	->	0	Unspecified
IT	Italian	<->	21	Italian
IW	Hebrew	<->	16	Hebrew
JA	Japanese	<->	17	Nihongo (Japan)

ISO 639 Prefix (2 char)	GSM MAP Language		SMPP <i>language_indicator</i>	SMPP Language
<i>JI</i>	<i>Yiddish</i>	<->	25	<i>Yiddish</i>
<i>JW</i>	<i>Javanese</i>	->	0	<i>Unspecified</i>
<i>KA</i>	<i>Georgian</i>	->	0	<i>Unspecified</i>
<i>KK</i>	<i>Kazakh</i>	->	0	<i>Unspecified</i>
<i>KL</i>	<i>Greenlandic</i>	->	0	<i>Unspecified</i>
<i>KM</i>	<i>Cambodian</i>	<->	33	<i>Mon-khmer (Cambodian)</i>
<i>KN</i>	<i>Kannada</i>	->	0	<i>Unspecified</i>
<i>KO</i>	<i>Korean</i>	<->	8	<i>Kangul (Korea)</i>
<i>KS</i>	<i>Kashmiri</i>	->	0	<i>Unspecified</i>
<i>KU</i>	<i>Kurdish</i>	->	0	<i>Unspecified</i>
<i>KY</i>	<i>Kirghiz</i>	->	0	<i>Unspecified</i>
<i>LA</i>	<i>Latin</i>	->	0	<i>Unspecified</i>
<i>LN</i>	<i>Lingala</i>	->	0	<i>Unspecified</i>
<i>LO</i>	<i>Laothian</i>	<->	27	<i>Laotian</i>
<i>LT</i>	<i>Lithuanian</i>	<->	45	<i>Lithuanian</i>
<i>LV</i>	<i>Latvian Lettish</i>	->	0	<i>Unspecified</i>
<i>MG</i>	<i>Malagasy</i>	->	0	<i>Unspecified</i>
<i>MI</i>	<i>Maori</i>	->	0	<i>Unspecified</i>
<i>MK</i>	<i>Macedonian</i>	->	0	<i>Unspecified</i>
<i>ML</i>	<i>Malayalam</i>	<->	59	<i>Malayalam</i>
<i>MN</i>	<i>Mongolian</i>	->	0	<i>Unspecified</i>
<i>MO</i>	<i>Moldavian</i>	->	0	<i>Unspecified</i>
<i>MR</i>	<i>Marathi</i>	->	0	<i>Unspecified</i>
<i>MS</i>	<i>Malay</i>	->	0	<i>Unspecified</i>
<i>MT</i>	<i>Maltese</i>	->	0	<i>Unspecified</i>
<i>MY</i>	<i>Burmese</i>	->	0	<i>Unspecified</i>
<i>NA</i>	<i>Nauru</i>	->	0	<i>Unspecified</i>
<i>NE</i>	<i>Nepali</i>	->	0	<i>Unspecified</i>
<i>NL</i>	<i>Dutch</i>	<->	20	<i>Dutch</i>
<i>NO</i>	<i>Norwegian</i>	<->	39	<i>Norwegian</i>
<i>OC</i>	<i>Occitan</i>	->	0	<i>Unspecified</i>
<i>OM</i>	<i>Oromo Afan</i>	->	0	<i>Unspecified</i>
<i>OR</i>	<i>Oriya</i>	->	0	<i>Unspecified</i>
<i>PA</i>	<i>Punjabi</i>	<->	47	<i>Punjabi</i>
<i>PL</i>	<i>Polish</i>	<->	22	<i>Polish</i>

ISO 639 Prefix (2 char)	GSM MAP Language		SMPP <i>language_indicator</i>	SMPP Language
<i>PS</i>	<i>Pashto Pushto</i>	->	0	<i>Unspecified</i>
<i>PT</i>	<i>Portuguese</i>	<->	5	<i>Portuguese</i>
<i>QU</i>	<i>Quechua</i>	->	0	<i>Unspecified</i>
<i>RM</i>	<i>Rhaeto-Romance</i>	->	0	<i>Unspecified</i>
<i>RN</i>	<i>Kirundi</i>	->	0	<i>Unspecified</i>
<i>RO</i>	<i>Romanian</i>	<->	44	<i>Rumanian</i>
<i>RU</i>	<i>Russian</i>	<->	18	<i>Russian</i>
<i>RW</i>	<i>Kinyarwanda</i>	->	0	<i>Unspecified</i>
<i>SA</i>	<i>Sanskrit</i>	->	0	<i>Unspecified</i>
<i>SD</i>	<i>Sindhi</i>	->	0	<i>Unspecified</i>
<i>SG</i>	<i>Sangro</i>	->	0	<i>Unspecified</i>
<i>SH</i>	<i>Serbo-Croatian</i>	->	0	<i>Unspecified</i>
<i>SI</i>	<i>Singhalese</i>	->	0	<i>Unspecified</i>
<i>SK</i>	<i>Slovak</i>	<->	40	<i>Slovak</i>
<i>SL</i>	<i>Slovenian</i>	->	0	<i>Unspecified</i>
<i>SM</i>	<i>Samoan</i>	<->	58	<i>Samoan</i>
<i>SN</i>	<i>Shona</i>	->	0	<i>Unspecified</i>
<i>SO</i>	<i>Somali</i>	->	0	<i>Unspecified</i>
<i>SQ</i>	<i>Albanian</i>	->	0	<i>Unspecified</i>
<i>SR</i>	<i>Serbian</i>	<->	42	<i>Serbian</i>
<i>SS</i>	<i>Siswati</i>	->	0	<i>Unspecified</i>
<i>ST</i>	<i>Sesotho</i>	->	0	<i>Unspecified</i>
<i>SU</i>	<i>Sudanese</i>	->	0	<i>Unspecified</i>
<i>SV</i>	<i>Swedish</i>	<->	41	<i>Swedish</i>
<i>SW</i>	<i>Swahili</i>	<->	14	<i>Swahili (East Africa)</i>
<i>TA</i>	<i>Tamil</i>	<->	57	<i>Tamil</i>
<i>TE</i>	<i>Tegulu</i>	->	0	<i>Unspecified</i>
<i>TG</i>	<i>Tajik</i>	->	0	<i>Unspecified</i>
<i>TH</i>	<i>Thai</i>	<->	26	<i>Thai</i>
<i>TI</i>	<i>Tigrinya</i>	->	0	<i>Unspecified</i>
<i>TK</i>	<i>Turkmen</i>	->	0	<i>Unspecified</i>
<i>TL</i>	<i>Tagalog</i>	<->	12	<i>Tagalog (Philippines)</i>
<i>TN</i>	<i>Setswana</i>	->	0	<i>Unspecified</i>
<i>TO</i>	<i>Tonga</i>	->	0	<i>Unspecified</i>
<i>TR</i>	<i>Turkish</i>	<->	51	<i>Turkish</i>

ISO 639 Prefix (2 char)	GSM MAP Language		SMPP <i>language_indicator</i>	SMPP Language
<i>TS</i>	<i>Tsonga</i>	->	0	<i>Unspecified</i>
<i>TT</i>	<i>Tatar</i>	->	0	<i>Unspecified</i>
<i>TW</i>	<i>Twi</i>	->	0	<i>Unspecified</i>
<i>UK</i>	<i>Ukrainian</i>	<->	35	<i>Ukrainian</i>
<i>UR</i>	<i>Urdu</i>	<->	11	<i>Urdu</i>
<i>UZ</i>	<i>Uzbek</i>	->	0	<i>Unspecified</i>
<i>VI</i>	<i>Vietnamese</i>	<->	23	<i>Vietnamese</i>
<i>VO</i>	<i>Volapuk</i>	->	0	<i>Unspecified</i>
<i>WO</i>	<i>Wolof</i>	->	0	<i>Unspecified</i>
<i>XH</i>	<i>Xhosa</i>	->	0	<i>Unspecified</i>
<i>YO</i>	<i>Yoruba</i>	->	0	<i>Unspecified</i>
<i>ZH</i>	<i>Chinese</i>	<->	6	<i>Cantonese</i>
<i>ZU</i>	<i>Zulu</i>	->	0	<i>Unspecified</i>

4.4.2. - ton (source_addr_npi, dest_addr_npi) mappings

Whenever ton subfield is used the following applies in mappings to/from MAP and SCCP layers.

GSM MAP Nature Of Address		SMPP <i>ton</i>
<i>Unknown (0)</i>	<->	<i>Unknown (0)</i>
<i>International (1)</i>	<->	<i>International (1)</i>
<i>National Significant (2)</i>	<->	<i>National (2)</i>
<i>Network Specific (3)</i>	<->	<i>Network Specific (3)</i>
<i>Subscriber Number (4)</i>	<->	<i>Subscriber Number (4)</i>
<i>--- error ---</i>	<-	<i>Alphanumeric (5)</i>
<i>Reserved (5)</i>	->	<i>Unknown (0)</i>
<i>Abbreviated (6)</i>	<->	<i>Abbreviated (6)</i>
<i>All others</i>	->	<i>Unknown (0)</i>

SCCP Nature Of Address		SMPP <i>ton</i>
<i>--- error ---</i>	<-	<i>Unknown (0)</i>
<i>International (4)</i>	<-	<i>International (1)</i>
<i>National Significant (3)</i>	<-	<i>National (2)</i>
<i>--- error ---</i>	<-	<i>Network Specific (3)</i>
<i>Subscriber Number (1)</i>	<-	<i>Subscriber Number (4)</i>
<i>--- error ---</i>	<-	<i>Alphanumeric (5)</i>
<i>--- error ---</i>	<-	<i>Abbreviated (6)</i>

4.4.3. - npi (source_addr_npi, dest_addr_npi) mappings

Whenever npi subfield is used the following applies in mappings to/from MAP and SCCP layers.

GSM MAP Numbering Plan		SMPP <i>npi</i>
<i>Unknown (0)</i>	<->	<i>Unknown (0)</i>
<i>ISDN/Telephony (1)</i>	<->	<i>ISDN (E163/E164) (1)</i>

GSM MAP Numbering Plan		SMPP <i>npi</i>
<i>Spare (2,5,7)</i>	->	<i>Unknown (0)</i>
<i>Data (3)</i>	<->	<i>Data (X.121) (3)</i>
<i>Telex (4)</i>	<->	<i>Telex (F.69) (4)</i>
<i>Land Mobile (6)</i>	<->	<i>Land Mobile (E.212) (6)</i>
<i>National (8)</i>	<->	<i>National (8)</i>
<i>Private (9)</i>	<->	<i>Private (9)</i>
<i>--- error ---</i>	<-	<i>ERMES (10)</i>
<i>--- error ---</i>	<-	<i>Internet (IP) (14)</i>
<i>--- error ---</i>	<-	<i>WAP Client Id (18)</i>

SCCP Numbering Plan		SMPP <i>npi</i>
<i>Unknown (0)</i>	<-	<i>Unknown (0)</i>
<i>ISDN/Telephony (1)</i>	<-	<i>ISDN (E163/E164) (1)</i>
<i>Data (3)</i>	<-	<i>Data (X.121) (3)</i>
<i>Telex (4)</i>	<-	<i>Telex (F.69) (4)</i>
<i>Land Mobile (6)</i>	<-	<i>Land Mobile (E.212) (6)</i>
<i>--- error ---</i>	<-	<i>National (8)</i>
<i>--- error ---</i>	<-	<i>Private (9)</i>
<i>--- error ---</i>	<-	<i>ERMES (10)</i>
<i>--- error ---</i>	<-	<i>Internet (IP) (14)</i>
<i>--- error ---</i>	<-	<i>WAP Client Id (18)</i>

4.5. - ANNEX 5: SMPP+ Message Examples

The following examples show encoding of SMPP+ messages. The first is a deliver_sm message with *ussd_service_op* = PSSD Indication. The second is a submit_sm message which responds to the PSSD indication with a PSSD response.

PSSD Indication:

```

SmppEncoder: Message deliver_sm encoded:
Header:
  command_length: [85]
  command_id: [5]
  command_status: [0]
  sequence number: [1]
Mandatory Parameters:
  service_type -> c_octet_string len [4] value 'USSD'
  source_addr_ton -> int value [1]
  source_addr_npi -> int value [1]
  source_addr -> c_octet_string len [16] value '012345678901*#A0'
  dest_addr_ton -> int value [0]
  dest_addr_npi -> int value [0]
  destination_addr -> c_octet_string len [0] value ''
  esm_class -> int value [0]
  protocol_id -> int value [0]
  priority_flag -> int value [0]
  schedule_delivery_time -> c_octet_string len [0] value ''

```

```

validity_period -> c_octet_string len [0] value ''
registered_delivery -> int value [0]
replace_if_present_flag -> int value [0]
data_coding -> int value [1]
sm_default_msg_id -> int value [0]
sm_length -> int value [19]
short_message -> octet_string len [19] value '*123456789128050020'
Optional Parameters:
user_message_reference -> int not present
source_port -> int not present
destination_port -> int not present
sar_msg_ref_num -> int not present
sar_total_segments -> int not present
sar_segment_seqnum -> int not present
user_response_code -> int not present
privacy_indicator -> int not present
payload_type -> int not present
message_payload -> octet_string not present
callback_num -> octet_string not present
source_subaddress -> octet_string not present
dest_subaddress -> octet_string not present
language_indicator -> int not present
its_session_info -> octet_string not present
network_error_code -> octet_string not present
message_state -> int not present
receipted_message_id -> c_octet_string not present
ussd_service_op -> octet_string len [1] value '0'
ussd_session_id -> int value [1]

```

```

-----
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
-----
000000: 00 00 00 55 00 00 00 05 00 00 00 00 00 00 00 01 ...U.....
000016: 55 53 53 44 00 01 01 30 31 32 33 34 35 36 37 38 USSD...012345678
000032: 39 30 31 2a 23 41 30 00 00 00 00 00 00 00 00 00 901*#A0.....
000048: 00 00 01 00 13 2a 31 32 33 34 35 36 37 38 39 31 .....*1234567891
000064: 32 38 30 35 30 30 32 30 05 01 00 01 30 15 01 00 28050020....0...
000080: 04 00 00 00 01 .....

```

PSSD Response:

```

SmppDecoder: Message submit_sm decoded:
Header:
  command_length: [74]
  command_id: [4]
  command_status: [0]
  sequence number: [2]
Mandatory Parameters:
  service_type -> c_octet_string len [1] value '2'
  source_addr_ton -> int value [1]
  source_addr_npi -> int value [1]

```



```

source_addr -> c_octet_string len [6] value '111111'
dest_addr_ton -> int value [1]
dest_addr_npi -> int value [1]
destination_addr -> c_octet_string len [6] value '222222'
esm_class -> int value [255]
protocol_id -> int value [255]
priority_flag -> int value [255]
schedule_delivery_time -> c_octet_string len [5] value '12345'
validity_period -> c_octet_string len [5] value '12345'
registered_delivery -> int value [255]
replace_if_present_flag -> int value [255]
data_coding -> int value [255]
sm_default_msg_id -> int value [255]
sm_length -> int value [5]
short_message -> octet_string len [5] value '-48-65-6c-6c-6f'

```

Optional Parameters:

```

user_message_reference -> int not present
source_port -> int not present
source_addr_subunit -> int not present
destination_port -> int not present
dest_addr_subunit -> int not present
sar_msg_ref_num -> int not present
sar_total_segments -> int not present
sar_segment_seqnum -> int not present
more_messages_to_send -> int not present
payload_type -> int not present
message_payload -> octet_string not present
privacy_indicator -> int not present
callback_num -> octet_string not present
callback_num_pres_ind -> int not present
callback_num_atag -> octet_string not present
source_subaddress -> octet_string not present
dest_subaddress -> octet_string not present
user_response_code -> int not present
display_time -> int not present
sms_signal -> int not present
ms_validity -> int not present
ms_msg_wait_facilities -> int not present
number_of_messages -> int not present
alert_on_message_delivery -> octet_string not present
language_indicator -> int not present
its_reply_type -> int not present
its_session_info -> octet_string not present
ussd_service_op -> octet_string len [1] value '-10'
ussd_session_id -> int value [1]

```

```

-----
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
-----
000000: 32 00 01 01 31 31 31 31  31 31 00 01 01 32 32 32    2...111111...222
000016: 32 32 32 00 ff ff ff 31  32 33 34 35 00 31 32 33    222...12345.123

```

000032:	34	35	00	ff	ff	ff	ff	05	48	65	6c	6c	6f	05	01	00	45.....Hello...
000048:	01	10	15	01	00	04	00	00	00	01						