

# **SMPP INTEGRATION MANUAL**

# CONTENTS

- Introduction..... 3
- SMPP INTRODUCTION ..... 3
  - SMPP parameters*..... 4
  - COMMAND STATUS & GSM ERROR CODES*..... 6
  - SCHEDULED DELIVERY*..... 6
- NUMBER LOOKUP OVER SMPP SPECIFICATION ..... 7
- FLASH NOTIFICATIONS OVER SMPP SPECIFICATION ..... 11
- SMPP COMMAND STATUS & GSM ERROR CODES ..... 12
  - SMPP Command status*..... 12
  - SMPP GSM error codes*..... 13

# Introduction

---

This document will provide instruction and examples how to use SMPP communication interface.

## SMPP INTRODUCTION

---

The Short Message Peer-to-Peer (SMPP) is an open, industry standard protocol used by the telecommunication industry for exchanging SMS messages between Short Message Service Centres (SMSC) and an SMS application systems. The protocol is a level-7 TCP/IP protocol, which allows fast delivery of SMS messages.

The connection between the application and the SMPP server is **SMPP version 3.4** (version 3.3 is not supported).

## SMPP parameters

Name	Description
<i>system_id</i>	<b>Required.</b> Provided for each client.
<i>password</i>	<b>Required.</b> Provided for each client. Maximum password length is 8 characters.
<i>IP address</i>	<b>Required.</b> Connection point: 107.20.199.106
<i>port</i>	<b>Required.</b> 8888 (primary and secondary) / 8887 (ssl)
<i>timeout (keep alive or msg)</i>	<b>Required.</b> 30 sec
<i>system_type</i>	<b>Optional.</b> <r:route_code>

**Important:** You are allowed to bind as **transmitter**, **receiver** or **transceiver**. In order to receive delivery reports, you must bind as **transceiver** or **receiver**. You'll receive delivery reports only if your route provides delivery reporting. Delivery reports will be sent equally over all of your currently available sessions capable of receiving them (**transceiver** or **receiver**).

By default, you are allowed to bind with **4 sessions**.

### *PDUS SUPPORTED:*

- bind\_transmitter
- bind\_receiver
- bind\_transceiver
- unbind
- submit\_sm
- deliver\_sm
- enquire\_link

## DELIVERY REPORT FORMAT

### Format

```
"id:<message_id> sub:<message_sub> dlvrld:<message_dlvrld>  
submit date:<message_submit_date> done date:<message_done_date>  
stat:<message_stat> err:<message_err>"
```

### DELIVERY STATUSES (MESSAGE\_STAT):

- DELIVRD
- EXPIRED
- UNDELIV
- ACCEPTD
- UNKNOWN
- ENROUTE
- REJECTD

### DATA CODING SCHEME

If you set DCS 0 or DCS 1 when sending messages, we will treat that as **default GSM7 encoding** (SMSC Default Alphabet or IA5).

For **Latin1** (ISO-8859-1) please use DCS 3 and DCS 8 for sending messages as **Unicode** (ISO/IEC-10646).

If needed, the content of messages can be processed in **Latin1** (ISO-8859-1) even with DCS 0, and in that case a simple adjustment is needed on account level.

## COMMAND STATUS & GSM ERROR CODES

Each request sent to our system is to receive an acknowledgement in `submit_sm resp PDU` and it is fully compliant with SMPP v 3.4 standard. A few proprietary platform command statuses with platform specific delivery report error code are described in the SMPP Command Status & GSM Error Codes section.

## SCHEDULED DELIVERY

Scheduled delivery is supported over SMPP protocol using the **relative time format**.

### *EXAMPLE:*

“070605040302100R” – this would mean that message will be delivered in 7 years, 6 months, 5 days, 4 hours, 3 minutes, 2 seconds and 1 tenth of second from now.

# NUMBER LOOKUP OVER SMPP SPECIFICATION

---

Using SMPP account, it is possible to request **Number Lookup** data (IMSI). In order to use Number Lookup, you can use your default `system_id` and password, setting `system_type = "HLR"` (without quotation marks) in Bind PDU.

**SubmitSM PDU** is used for submitting the Number Lookup request, having `destAddress` parameter set to the required destination address. All other parameters will be ignored (`srcAddress`, TON/NPI, etc).

Number Lookup subsystem will respond using a regular `SubmitSMResp`, containing `message-id` reference.

Once the Number Lookup request is being finalised on the system, you will receive **DeliverSM PDU**, containing:

- IMSI for the required `destAddress` **or**
- error code in case of failure.

**DeliverSM** will contain:

- short message data with our regular delivery report
- IMSI part ("IMSI:xxxxxxxx")
- serving MSC
- additional optional info fields depending on your package.

<b>Optional Info Fields</b>	<b>Type</b>	<b>Hex</b>	<b>Decimal</b>
<i>Original network name</i>	TLVString	0x1412	5138
<i>Original network prefix</i>	TLVString	0x140B	5131
<i>Original country</i>	TLVString	0x1422	5154
<i>Original country code</i>	TLVString	0x1423	5155
<i>Original country prefix</i>	TLVString	0x1424	5156
<i>Ported network name</i>	TLVString	0x1413	5139
<i>Ported country prefix</i>	TLVString	0x1442	5186
<i>Ported network prefix</i>	TLVString	0x143e	5182
<i>Ported network country name</i>	TLVString	0x143f	5183
<i>Is number ported</i>	TLVInt	0x1421	5153
<i>Roaming network name</i>	TLVString	0x1414	5140
<i>Roaming network prefix</i>	TLVString	0x1419	5145
<i>Roaming country name</i>	TLVString	0x1415	5141
<i>Roaming country code</i>	TLVString	0x1417	5143
<i>Roaming country prefix</i>	TLVString	0x1420	5152
<i>MCCMNC</i>	TLVString	0x1416	5142



Price per message For compatibility reasons, price per message is multiplied by 100	TLVInt	0x1418	5144
Serving HLR	TLVString	0x1409	5129
Is number correct	TLVInt	0x1425	5157

INFO: Besides DeliverSM.shortMessage, we included IMSI also as an extra-optional parameter:

SMPP\_VENDOR\_SPECIFIC\_IMSI = 0x1403

#### EXAMPLE:

In case that Number Lookup request was successful, **DeliverSM** will be as follows (IMSI 21910110053751):

#### Groovy

```
addr: 0 0 38591xxxxxxx
addr: 0 0 0000000000
msg: id:40072910491427628 sub:001 dlvr:001 submit date:1007291049 done date:1007291049 stat:D
ELIVRD err:000
IMSI:219101100935850 MSC:38591016 HLR:38591xxxxxxx ORN:VipNet PON:VipNet RON:VipNet ROC:HR MCC
MNC:21910
opt: (oct: (tlv: 1059) 030000) (byte: (tlv: 1063) 2) (str: (tlv: 30) 40072910491427628) (str:
(tlv: 5129)38591xxxxxxx) (str: (tlv: 5138) VipNet) (str: (tlv: 5139) VipNet) (str: (tlv: 5140)
VipNet) (str: (tlv:5141) Croatia ) (str: (tlv: 5143) HR) (str: (tlv: 5142) 21910) (int: (tlv:
5144) 1) (str: (tlv: 5145) 91) (str: (tlv: 5152) 385) (int: (tlv: 5153) 1) (str: (tlv: 5154) C
roatia ) (str: (tlv: 5155) HR) (str: (tlv: 5156) 385) (int: (tlv: 5157) 1) ) (extraopt: (oct:
(tlv: 5123) 323139313031313030393335383530) (oct: (tlv: 5126) 3338353931303136) )
```

If an error occurred, **DeliverSM** will be as follows:

### Groovy

```
addr: 0 0 385915369423
```

```
addr: 0 0 0000000000
```

```
msg: id:40072910491419819 sub:001 dlvr:001 submit date:1007291049 done date:1007291049 stat:U  
NDELIV err:001
```

```
IMSI: MSC: ORN:VipNet MCCMNC:
```

```
opt: (oct: (tlv: 1059) 030001) (byte: (tlv: 1063) 5) (str: (tlv: 30) 40072910491419819) (str:  
(tlv: 5138) VipNet) (str: (tlv: 5142) ) (int: (tlv: 5144) 1) (int: (tlv: 5153) 0) (str: (tlv:  
5154) Croatia ) (str: (tlv: 5155) HR) (str: (tlv: 5156) 385) (int: (tlv: 5157) 1) )
```

# FLASH NOTIFICATIONS OVER SMPP SPECIFICATION

---

You can use your SMPP account to send **Flash notifications**. Such notifications are immediately displayed on a mobile phone screen upon arrival and aren't stored in the memory of the device.

In order to use Flash notifications, you can use your default `system_id` and password, setting `system_type = "NSMS"` (without quotation marks) in Bind PDU.

Procedure for submitting Flash notifications is exactly the same as for normal SMS, using **SubmitSM PDU**. system will automatically convert your message into the Flash notification using message parameters you have submitted.

Delivery reports will be sent to you using **DeliverSM PDU**.

**NOTE:** Long SMS feature is not supported for Flash notifications.

# SMPP COMMAND STATUS & GSM ERROR CODES

---

## SMPP Command status

Command status is received as a response for **Submit\_SM**, on special events, illustrated in the table below.

Value (HEX/DEC)	Description
0x00000022 / 34	Network not covered
0x000000FF / 255	Account has insufficient balance
0x0000000a	Invalid_Source_Address
0x0000000c	Duplicate_Message_ID
0x000004a1	System_Error or Channel_Disabled

## SMPP GSM error codes

<b>Id</b>	<b>Short description</b>	<b>Is permanent</b>
0	NO_ERROR	NULL
1	EC_UNKNOWN_SUBSCRIBER	1
5	EC_UNIDENTIFIED_SUBSCRIBER	0
6	EC_ABSENT_SUBSCRIBER_SM	0
9	EC_ILLEGAL_SUBSCRIBER	1
10	EC_BEARER_SERVICE_NOT_PROVISIONED	0
11	EC_TELESERVICE_NOT_PROVISIONED	1
12	EC_ILLEGAL_EQUIPMENT	1
13	EC_CALL_BARRED	0
20	EC_SS_INCOMPATIBILITY	0
21	EC_FACILITY_NOT_SUPPORTED	0
27	EC_ABSENT_SUBSCRIBER	0
31	EC_SUBSCRIBER_BUSY_FOR_MT_SMS	0
32	EC_SM_DELIVERY_FAILURE	0
33	EC_MESSAGE_WAITING_LIST_FULL	0
34	EC_SYSTEM_FAILURE	0

Id	Short description	Is permanent
35	EC_DATA_MISSING	1
36	EC_UNEXPECTED_DATA_VALUE	1
51	EC_RESOURCE_LIMITATION	0
71	EC_UNKNOWN_ALPHABET	1
72	EC_USSD_BUSY	1
255	EC_UNKNOWN_ERROR	1
256	EC_SM_DF_memoryCapacityExceeded	0
257	EC_SM_DF_equipmentProtocolError	0
258	EC_SM_DF_equipmentNotSM_Equipped	0
259	EC_SM_DF_unknownServiceCentre	0
260	EC_SM_DF_sc_Congestion	0
261	EC_SM_DF_invalidSME_Address	0
262	EC_SM_DF_subscriberNotSC_Subscriber	0
500	EC_PROVIDER_GENERAL_ERROR	0
502	EC_NO_RESPONSE	0
503	EC_SERVICE_COMPLETION_FAILURE	0
504	EC_UNEXPECTED_RESPONSE_FROM_PEER	0

<b>Id</b>	<b>Short description</b>	<b>Is permanent</b>
507	EC_MISTYPED_PARAMETER	0
508	EC_NOT_SUPPORTEDED_SERVICE	0
509	EC_DUPLICATED_INVOKE_ID	0
511	EC_INITIATING_RELEASE	0
1024	EC_OR_appContextNotSupported	0
1025	EC_OR_invalidDestinationReference	0
1026	EC_OR_invalidOriginatingReference	0
1027	EC_OR_encapsulatedAC_NotSupported	0
1028	EC_OR_transportProtectionNotAdequate	0
1029	EC_OR_noReasonGiven	0
1030	EC_OR_potentialVersionIncompatibility	0
1031	EC_OR_remoteNodeNotReachable	0
1152	EC_NNR_noTranslationForAnAddressOfSuchNature	0
1153	EC_NNR_noTranslationForThisSpecificAddress	0
1154	EC_NNR_subsystemCongestion	0
1155	EC_NNR_subsystemFailure	0
1156	EC_NNR_unequippedUser	0

Id	Short description	Is permanent
1157	EC_NNR_MTPfailure	0
1158	EC_NNR_networkCongestion	0
1159	EC_NNR_unqualified	0
1160	EC_NNR_errorInMessageTransportXUDT	0
1161	EC_NNR_errorInLocalProcessingXUDT	0
1162	EC_NNR_destinationCannotPerformReassemblyXUDT	0
1163	EC_NNR_SCCPfailure	0
1164	EC_NNR_hopCounterViolation	0
1165	EC_NNR_segmentationNotSupported	0
1166	EC_NNR_segmentationFailure	0
1281	EC_UA_userSpecificReason	0
1282	EC_UA_userResourceLimitation	0
1283	EC_UA_resourceUnavailable	0
1284	EC_UA_applicationProcedureCancellation	0
1536	EC_PA_providerMalfunction	0
1537	EC_PA_supportingDialogOrTransactionReleased	0
1538	EC_PA_resourceLimitation	0



<b>Id</b>	<b>Short description</b>	<b>Is permanent</b>
1539	EC_PA_maintenanceActivity	0
1540	EC_PA_versionIncompatibility	0
1541	EC_PA_abnormalMapDialog	0
1792	EC_NC_abnormalEventDetectedByPeer	0
1793	EC_NC_responseRejectedByPeer	0
1794	EC_NC_abnormalEventReceivedFromPeer	0
1795	EC_NC_messageCannotBeDeliveredToPeer	0
1796	EC_NC_providerOutOfInvoke	0
2048	EC_TIME_OUT	0
2049	EC_IMSI_BLACKLISTED	1
2050	EC_DEST_ADDRESS_BLACKLISTED	1
2051	EC_InvalidMscAddress	0
4096	EC_invalidPduFormat	1
4097	EC_NotSubmittedToGMSC	1
4100	EC_Cancelled	1
4101	EC_ValidityExpired	1
4102	EC_NotSubmittedToSmppChannel	0