

Orange GDEP

GDEP communication protocol

[Supported device types and versions](#)

[Communication line configuration](#)

[Communication station configuration](#)

[I/O tag configuration](#)

[Literature](#)

[Changes and modifications](#)

[Document revisions](#)

Supported device types and versions

This protocol is used for sending and receiving SMS messages and MMS ones, processing WAP requirements, and playing SMS games. The protocol is built on the standards such as TCP, HTTP (or HTTPS), and XML. GAP server (GAME Platform) on the Orange side and an application (D2000 KOM process) are the communication partners. The implementation in D2000 System supports the following messages:

Table 1

| Message | Direction | Meaning |
|---------------|----------------------|--|
| HELLO | KOM->GAP | Logging on (name + password) to the GAP platform when starting the connection. |
| INQUIRE_LINK | KOM->GAP GAP->KOM | Testing the functionality of connection after a longer period of inactivity. |
| INQ_RSP | KOM->GAP GAP->KOM | Response to INQUIRE. |
| RECEIVE (sms) | GAP->KOM | Receiving SMS from GAP server. |
| RCV_RSP (sms) | KOM->GAP | Confirmation of receiving SMS by D2000 KOM. |
| SEND (sms) | KOM->GAP | Sending SMS message. |
| SND_RSP (sms) | GAP->KOM | Confirmation of receiving SMS by GAP server. |
| ACCEPTED | GAP->KOM | Confirmation of receiving SMS or MMS by SMS/MMS center. |
| ACC_RSP | KOM->GAP | Response to ACCEPTED from D2000 KOM. |
| RECEIPT (sms) | GAP->KOM | Confirmation of receiving SMS by an end-user. |
| REC_RSP | KOM->GAP | Response to RECEIPT from D2000 KOM. |
| BY | KOM->GAP | End of connection by D2000 KOM. |
| BY_RSP | GAP->KOM | Response to BY from GAP server. |
| NAK | GAP->KOM | An unacceptable message is declined. |

The following messages have not been implemented yet:

Table 2

| Message | Direction | Meaning |
|-------------------|-----------|---|
| RECEIVE (wap) | GAP->KOM | Receiving WAP requirement from GAP server. |
| RCV_RSP (wap) | KOM->GAP | WAP requirement by D2000 KOM. |
| RECEIVE (mms-xml) | GAP->KOM | Receiving MMS from GAP server. |
| RCV_RSP (mms-xml) | KOM->GAP | Confirmation of receiving MMS by D2000 KOM. |
| SEND (mms-xml) | KOM->GAP | Sending MMS. |
| SND_RSP (mms-xml) | GAP->KOM | Confirmation of receiving MMS by GAP server. |
| RECEIPT (mms) | GAP->KOM | Confirmation of receiving MMS by an end-user. |
| SCORE | KOM->GAP | Changing the player's score. |

| | | |
|----------|----------|--|
| SCO_RSP | GAP->KOM | Response to SCORE from GAP server. |
| NICK | KOM->GAP | Administration of nicknames: creation, canceling, adding, deleting, and searching. |
| NICK_RSP | GAP->KOM | Response to NICK from GAP server. |

Communication line configuration

Communication line category: [TCP/IP-TCP](#)

TCP Parameters:

- Host: String max. 80 characters – server name in INET format (name or numerical address a.b.c.d).
- Port: TCP port number (1 to 65535).
- Line number: Not used, enter any number.

Note: GDEP protocol is built over HTTP or HTTPS protocol. D2000 KOM process does not include HTTPS communication. It uses the [stunnel](#) program (modular method). The stunnel program is able to wrap both the client and server communication in the SSL layer. As GDEP protocol needs the implementation of both client and server on each side (both GAP server and D2000 KOM), you must start two stunnel processes:

- one of them is in a server mode, which will wait on the connection of the GAP server. After processing the SSL layer, it will send data to the defined port, on which D2000 KOM is waiting as a GDEP server (see the [Server Port](#) parameter).
- the other one is in a client mode, which waits on the defined port. D2000 KOM (client-side) will be connected with it. After wrapping data in the SSL layer, the stunnel will send them to a target computer and port (server port of GAP server). Therefore, in the **Host** parameter, there will be *localhost* and in the **Port** parameter, there will be the port number on which the stunnel process is waiting in a client mode.

Communication station configuration

Communication protocol: **GDEP**

Station address: not used

Note: There must be exactly ONE communication station on the line.

Station protocol parameters

You can set the following station protocol parameters:

Table 3

| Key word | Full name | Meaning | Unit | Default value |
|----------|-----------------------------|---|------|---------------|
| RD | Relogin Delay | Waiting before repeated attempt to log on to GAP server. | sec | 10 |
| TKI | TCP KeepInit option | KeepInit setting of TCP protocol. | - | False |
| CE | Station Communication Error | A number of failed attempts to connect to the server when the station switches to StCOMERR status. | - | 1 |
| HE | Station Hard Error | A number of failed attempts to connect to the server when the station switches to StHARDERR status. | - | 3 |
| U | User Name | User name to log on to GAP server. | - | - |
| P | User Password | The password to log on to the GAP server. | - | - |
| HH | HTTP Host | IP address or server name, which is specified in the HTTP request. | - | - |
| HP | HTTP Port | Port of server, which is specified in the HTTP request. | - | 0 |
| SP | Server Port | The port on which KOM is listening as a server. | - | 1000 |

| | | | | |
|----------------------------------|--|---|-----|----|
| TSI | Timeout Send Inquire | After this timeout, KOM will send INQUIRE message (0=disabled). | sec | 90 |
| TRI | Timeout Receive Inq_Rsp | Till this timeout, KOM will wait for INQ_RSP (response to INQUIRE). | sec | 10 |
| TRA | Timeout Receive Accept | After this timeout, the KOM process denotes the message as being expired (confirmation level 2), if an ACCEPTED message does not come. | sec | 10 |
| TRR | Timeout Receive Receipt | After this timeout, the KOM process denotes the message as being expired (confirmation level 3), if a RECEIPT message does not come. | sec | 60 |
| K1 H1 P1 K2 H2 P2 | KOM Name 1 Host Address 1 Port Number 1 KOM Name 2 Host Address 2 Port Number 2 | When using in D2000 System redundancy (two D2000 KOM processes): if the name of workstation, on which D2000 KOM is running, is the same than <i>K1</i> (or <i>K2</i>) parameter, when connecting to GAP server, <i>HELLO</i> message will contain the optional parameters <i>host</i> and <i>port</i> with values <i>H1</i> and <i>P1</i> (or <i>H2</i> and <i>P2</i>). This way, the GAP server knows on which of two D2000 KOM processes shall connect. | | |

I/O tag configuration

I/O tag is defined by its address, which must correspond with the address specified in the tables below. I/O tag names can be different.

SMS receiving: After receiving SMS, the values of the following I/O tags will be set (if I/O tags have been configured).

Table 4

| Address | Type | I/O tag name | Meaning |
|---------|------|------------------|---|
| 1 | TxtO | Rcv_Id | A 64-bit ID of received SMS (RECEIVE message). After processing of the message by an application, the D2000 KOM process should be informed, by writing into this I/O tag, to confirm the receiving by RCV_RSP message and make accessible next SMS to the application (if it is in a queue). [message sequence ID on the Game Platform (64-bit number displayed decimally in ASCII format). It is used for duplicate receipt detection when a session has been re-established after the connection failure.] |
| 2 | TxtI | Rcv_Serv_Id | Serv_id parameter of the received message. [service ID that this message belongs to.] |
| 3 | TxtI | Rcv_AppAddr | Appaddr parameter of the received message. [destination number. It can be a short or long number dedicated to the service (External Application). It is the identification of the game. It is never encrypted.] |
| 4 | TxtI | Rcv_AppAddrType | The type attribute of appaddr parameter. |
| 5 | TxtI | Rcv_Msisdn | Msisdn parameter of the received message. [source number. Can be encrypted, see the type attribute.] |
| 6 | TxtI | Rcv_MsisdnType | A type attribute of msisdn parameter. |
| 7 | Ci | Rcv_Employee | Parameter employee of the received message. [if application have enabled mark of the employee, then element employee has value 0 – if source MSISDN is NOT an employee 1 – if source MSISDN is employee] |
| 8 | Ci | Rcv_Dcs | Parameter dcs of the received message. [parameter data_coding of the GDEP protocol (DCS) [SMPP PS v3.4]. (8-bit integer defined in the ASCII). Default = 0.] |
| 9 | Ci | Rcv_Esm | Parameter esm of the received message. [parameter esm of the SMPP protocol (DCS) [SMPP PS v3.4]. (8-bit integer defined in the ASCII). Default = 0.] |
| 10 | TxtI | Rcv_Receive_Date | Parameter receive_date of the received message. [time of receive SMS to the SMS center.] |

| | | | |
|----|------|-------------|--|
| 11 | TxtI | Rcv_Coding | Attribute coding of the content parameter. [type of coding in the element content. Currently used are: <i>text</i> ASCII text, <i>base64</i> content is coded by code base64 – this is used for binary data.] |
| 12 | TxtI | Rcv_Content | Parameter content of the received message. [nothing but data. Data sent by the subscriber to the External Application (SMS).] |
| 13 | TxtI | Rcv_MsgPart | Parameter msg_part of received message (trio ref_num/seg_num/tot_seg). [<i>ref_num</i> - contain an originator generated reference number so that a segmented short message may be reassembled into a single original message <i>seg_num</i> - sequence number of the particular message within the concatenated message <i>tot_seg</i> - indicating the total number of fragments] |

SMS sending: When sending an SMS message, you should set the values for the following I/O tags (if they have been configured). The message will be sent after setting the value of the I/O tag [Snd_AppDefId](#).

The symbol (o), stated at the I/O tag means that the tag is optional (it may not be configured because GDEP protocol denotes this attribute as optional).

Table 5

| Address | Type | I/O tag name | Meaning |
|---------|------|-------------------------|--|
| 101 | TxtO | Snd_Serv_Id | Parameter serv_id of the sent message. [service ID that this message belongs to.] |
| 102 | TxtO | Snd_Rcv_Msg_Id (o) | Parameter rcv_msg_id of the sent message. [ID of the received message (RECEIVE) that this message is a response to] |
| 103 | TxtO | Snd_AppAddr | Parameter appaddr of the sent message. [source number. It can be either a short or long (MSISDN) number assigned to the service.] |
| 104 | Co | Snd_AppAddrType (o) | A type attribute of appaddr parameter. |
| 105 | TxtO | Snd_Msisdn | Parameter msisdn of the sent message. [destination number. In the case of a mobile telephone number, it can be encrypted. In the case of a short number, the destination number is the short number only. Can be encrypted, the type is present in the type attribute.] |
| 106 | Co | Snd_MsisdnType (o) | A type attribute of msisdn parameter. |
| 107 | TxtO | Snd_BillMsisdn (o) | Parameter billmsisdn of the sent message. [MSISDN to which the message will be billed (this tag is only used with specific applications, where the billed subscriber is not the recipient of the message. This option must be approved at the Game Platform by agreement with the marketing dept.) Can be encrypted, the type is present in the type attribute.] |
| 108 | Co | Snd_BillType (o) | A type attribute of billmsisdn parameter. |
| 109 | Co | Snd_Flags (o) | Parameter flags of the sent message. [32-bit numbers displayed decimally in ASCII format. Flags are combined with the OR binary function. Unused bits are reserved. <i>SMS_FLAG_RECEIPT</i> bit 1 - delivery receipt requested <i>SMS_FLAG_REPLACE</i> bit 2 - SM replacement requested on SIM card position under bits 4-6 <i>SMS_FLAG_POSITION</i> bit 4-6 - SIM position for SM replacement position range: from 1 to 7, 0 is unused] |
| 110 | TxtO | Snd_Expiration (o) | Parameter expiration of sent message. [validity period of the SM in the GSM time encoding (i.e. YYMMDDhhmmss), or the character „-“, if not defined] |
| 111 | TxtO | Snd_Delivery (o) | Parameter delivery of the sent message. [delivery time of the SM in the GSM time encoding (i.e. YYMMDDhhmmss), or the character „-“, if not defined.] |
| 112 | TxtO | Snd_BillKey | Parameter billkey of the sent message. [billing key, 4 characters long string consisting of (0..9) and (A .. Z). Each application will have 4 keys to be used in this field] |
| 113 | TxtO | Snd_EventBilling_Id (o) | Parameter evbill-id of the sent message. [service identifier in the event billing in case this type of billing method should be used.] |
| 114 | TxtO | Snd_EventBillingSeq (o) | Parameter evbill-id-seq of the sent message. [service identifier in the event billing in case this type of billing method should be used.] |
| 115 | Co | Snd_Dcs (o) | Parameter dcs of the sent message. [parameter data_coding of the SMPP protocol (DCS) [SMPP PS v3.4]. (8-bit integer defined in the ASCII). Default = 0.] |
| 116 | Co | Snd_Esm (o) | Parameter esm of the sent message. [parameter esm of the SMPP protocol (DCS) [SMPP PS v3.4]. (8-bit integer defined in the ASCII). Default = 0.] |

| | | | |
|-----|------|---------------------|--|
| 117 | TxtO | Snd_Score (o) | Parameter score of the sent message. [the game is sending score for the subscriber with – MSISDN. Format: +score – Game Platform adds this score value to the daily counter -score – Game Platform subtracts this score value from the daily counter score – Game Platform replaces the daily counter by this score value (daily counter is a parameter, which is bind to MSISDN, nick, game, and day. This counter will be changed only by GDEP protocol)] |
| 118 | TxtO | Snd_ScoreNick (o) | Attribute nick of score parameter. [– if this attribute will be used it will define the particularly used nick of the subscriber defined by MSISDN. The score will update the daily counter for this MSISDN and nick. If this attribute will not be used daily counter for MSISDN and default value nick="" will be used.] |
| 119 | TxtO | Snd_ScoreFinish (o) | Attribute finished of score parameter. [if this attribute will be used (with the any value, the default value will be empty string ""), the Game Platform will add the new issue to the table C_SCORE, where will be add each player so many time how many will finish the game successfully. Use this attribute only after successful finishing of the game by the player.] |
| 120 | TxtO | Snd_Coding | Parameter content of the sent message. [type of coding in the element content Currently used are: text ASCII text base64 content is coded by code base64 – this is used for binary data.] |
| 121 | TxtO | Snd_Content | Parameter content of the sent message. [nothing but data. Data sent by the External Application (SMS) to the subscriber.] |
| 122 | TxtO | Snd_Category (o) | Parameter category of the sent message. [element can be used to specify the category of content that is to be sent, character string 1-64 bytes long] |
| 123 | Co | Snd_AppDefId | Any positive integer, generated by the application, that (for application) defines the sent SMS. The confirmation of delivery or non-delivery will contain this number (I/O tag Rsp_AppDefId) if the KOM process has not been restarted. |
| 130 | TxtI | Snd_64bit_Id | A 64-bit identifier set by the KOM process when sending an SMS message. As the GDEP protocol uses a unique 64-bit identifier (GDEP server initiates its value when connecting to D2000 KOM process), the confirmation of delivery is marked internally by this 64-bit ID. If the KOM process is restarted, after start-up the GAP server could send the confirmations of delivery, which have been sent before restart. In such a case, D2000 KOM sets Snd_AppDefId to -1. The application may couple the confirmations according to Snd_64bit_Id. |

Confirmation of SMS delivery: After receiving the ACCEPTED, RECEIPT, and NAK messages, the following I/O tags will be set (Rsp_AppDefId will be set as the last).

Table 6

| Address | Type | I/O tag name | Meaning |
|--|------|----------------|--|
| 131 | Co | Rsp_AppDefId | Snd_AppDefId of sent message or -1, if the message with this number has not been found in the list of sent and unconfirmed messages (e.g. after restarting of KOM process). After processing the confirmation of delivery, the application must set Rsp_AppDefId on 0. It causes that the confirmation will be sent to the GAP server (ACC_RSP message or REC_RSP, depending on the message on which D2000 KOM responds) and D2000 KOM can set the next value if it has other confirmation of delivery. |
| 132 | Ci | Rsp_ErrNo | The error number: 0 - no error, any other values are mentioned in the literature [1] . If the RECEIPT message has been received, Rsp_ErrNo copies the value of the state , except the value 2 (delivered) and 6 (accepted) - the message is considered to be delivered successfully and Rsp_ErrNo=0. |
| 133 | Ci | Rsp_CnfType | Level of confirmation: <i>level 1</i> - the message has been sent to the GAP server (Rsp_ErrNo=0) or an error occurred when sending (Rsp_ErrNo=-1) <i>level 2</i> - the ACCEPTED message has been received (with zero or nonzero error code, whose value is in Rsp_ErrNo) or timeout Timeout Receive Accept has expired (Rsp_ErrNo=-2) <i>level 3</i> - RECEIPT message has been received or timeout Timeout Receive Receipt has expired (Rsp_ErrNo=-3) |
| 134 | TxtI | Rsp_ErrDesc | Text description of the error (when expiring the timeout) or value err_type when there is the negative message ACCEPT. [err_type - a type of error, one of SMSC, MMSC, INTERNAL <ul style="list-style-type: none"> SMSC means the error occurred in SMSC MMSC means the error occurred in MMSC INTERNAL means the error occurred in GAP] |
| 135 | TxtI | Rsp_ServId | A 64-bit ID of the sent SMS message to which this report relates. |
| <i>The values of these I/O tags are valid only when receiving RECEIPT.</i> | | | |
| 136 | TxtI | Rec_SubmitDate | Parameter submit_date of RECEIPT message. [date of delivery of sent SMS message to SMS center] |
| 137 | TxtI | Rec_DoneDate | Parameter done_date of RECEIPT message. [date of delivery of sent SMS message to destination] |

| | | | |
|-----|------|----------------|---|
| 138 | Ci | Rec_State | <p>Parameter state of RECEIPT message.</p> <p>[state of the message, can have the following values (as described in [2] SMPP 3.4, chapter 5.2.28 message_state):</p> <ul style="list-style-type: none"> • 1 – <i>enroute</i> – the message is in enroute state • 2 – <i>delivered</i> – the message is delivered to the destination • 3 – <i>expired</i> – message validity period has expired • 4 – <i>deleted</i> – the message has been deleted • 5 – <i>undeliverable</i> – the message is undeliverable • 6 – <i>accepted</i> – the message is in an accepted state (i.e. has been manually read on behalf of the subscriber by customer service) • 7 – <i>unknown</i> – the message is in an invalid state • 8 – <i>rejected</i> – the message is in a rejected state <p>]</p> <p>Note: If the RECEIPT message with <i>state</i>=1 (temporary <i>enroute</i> state) has been received, the application waits once more with the timeout Timeout Receive Receipt to receive the RECEIPT message of the final state.</p> |
| 139 | Txtl | Rec_Msisdn | <p>Parameter msisdn of RECEIPT message.</p> <p>[msisdn that the message is delivered to. It can be encrypted.]</p> |
| 140 | Txtl | Rec_MsisdnType | An attribute type of msisdn parameter. |

Identification of communication status:

Table 7

| Address | Type | I/O tag name | Meaning |
|---------|------|------------------|---|
| 199 | Di | GAP_SrvConnected | Information on whether the GAP server is connected to the D2000 KOM process. After breaking the communication, there can happen the situation when the D2000 KOM process will connect to the GAP server, but the GAP server will not connect to the D2000 KOM process. If for example, the value of GAP_SrvConnected is FALSE within a minute after reconnection to the GAP server (the status of the station is StON), we recommend restarting the D2000 KOM . If it fails, contact the responsible person for the GAP server. |

Attribute *type* (Rcv_AppAddrType, Rcv_MsisdnType) can acquire values listed in the table below (the numeric values are entered when sending SMS messages, the text ones when receiving SMS messages):

Table 8

| Value | Number | Meaning |
|--------|--------|---|
| msisdn | 1 | Cell phone number in the format +421905123456. |
| alias | 2 | An encrypted user phone number (encryption is done by GAP server). It is a 32-character string (hexadecimal code). Example of encryption: 421905014940 -> 2F8E47BCAAAB12CA3E19B7DCC2DFDE84. |
| ow | 3 | The specification [1] does not contain a detailed definition, but this type is used only for MMS messages. |

Literature

- [1] Game Data Exchange Protocol Specification 2.4 (provided by Orange Slovensko, a.s.)
[2] SMPP PS v3.4 (Short Message Peer to Peer Protocol Specification v3.4)

Changes and modifications

-

Document revisions

- Ver. 1.0 – May 2, 2006: the protocol implementation



Related pages:

[Communication protocols](#)