

SMS Gateway Router

HTTPTX Interface Description

ADVENAGE GmbH
Blumenhagenstr. 10
D-30167 Hannover
Germany

September 2009

Version 1.1 September 2009

Copyrights, trademarks and acknowledgments.
Windows, Windows 2000, Windows XP, Windows 2003 Server are copyright
of Microsoft Corporation.
All other copyrights and trademarks are the property of their respective owners.

ADVENAGE GmbH 2008-2009

Contents

1 About this guide.....	4
1.1 References.....	4
2 HTTPTX interface.....	5
2.1 Overview.....	5
2.2 Actors.....	5
2.3 Context.....	5
2.4 Technical Details.....	6
2.4.1 Delivery of Text SMS MT.....	7
2.4.2 Delivery of binary SMS MT.....	8
3 User Agent Examples for User.....	9
3.1 Text SMS Delivery.....	9
3.1.1 Example using WGET and PHP.....	9
3.1.2 Example using CURL and PHP.....	9
3.2 Binary SMS Examples.....	10
3.2.1 Example for concatenated SM with UDH.....	10
3.2.2 Example for WAP Push SM with UDH.....	10

1 About this guide

This guide covers the description of the HTTPX interface to SMS Gateway Router for 3rd parties.

1.1 References

- [1] Short Message Peer to Peer Protocol Specification v3.4; Oct 1999; SMPP Developers Forum
- [2] ETSI TS 100 901 V7.3.0; Digital cellular telecommunications system (Phase 2+); Technical Realization of the Short Message Service (SMS); Point-to-Point (PP); GSM 03.40 version 7.3.0 Release 1998.

2 HTTPTX interface

2.1 Overview

The HTTPTX interface to the GSM Gateway Router is an easy to use interface for delivery of SM-MT (Text SMS and binary SMS). The interface is suitable to deliver low volumes of short messages (typically up to 5 SMS per second) via a simple HTTP(S) interface. Only few parameters need to be supplied: MSISDN and the text or binary contents of the message. For binary SMS also the data coding may be specified.

This interface was designed to provide business support system suppliers and other third parties with a simple interface to deliver SM-MT.

From the technical perspective SMS delivery via this interface uses a HTTP-POST operation with two parameters delivered in the request body: MSISDN and message text or the binary contents. To protect data against unwanted transformation, data shall be delivered in URL encoded ISO Latin 1 coding.

Due to the simple interface, simple UNIX scripts can be used for short message delivery. The sections below include some examples.

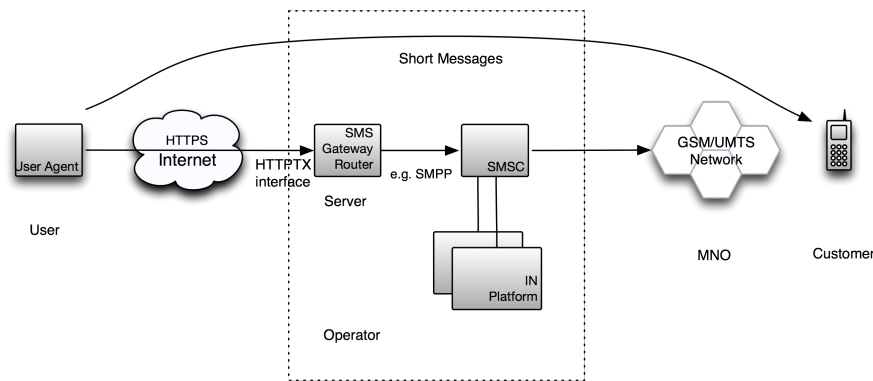
Configuration details are left to a configuration form. This form shall be supplied by the SMS Gateway Router operating party to the HTTPTX interface using party.

2.2 Actors

Actor	Description
User	Party using the HTTPTX interface to deliver SM-MT
User Agent	The software being used by the User to deliver SM-MT
Operator	The party operating the SMS Gateway Router and providing the HTTPTX interface to the user.
Server	The technical system (which might be distributed) running the SMS Gateway Router service and offers the HTTPTX interface to the User.
Customer	The party receiving SM-MT from the User via the Operator and MVNOs and/or MNOs in between.

2.3 Context

The context of SM-MT delivery via the HTTPTX interface from the User via the Operator to the Customer is depicted by the figure below:



The user may deploy any kind of Web enabled application to send SM-MT. The portfolio might include solutions like a simple Web form, script applications up to full Java or native applications. The particular application chosen is called user agent.

The user agent shall send HTTP or HTTPS post requests to the server including the MSISDN of the customer and the text or contents to be sent to the customer in URL encoded ISO Latin 1.

After the request was processed by the server, the user agent receives an acknowledgment. This can either confirm delivery of the message or might report a message delivery failure. For this purpose the HTTP return code is used. A return code of 200 acknowledges successful delivery, any other code reports a delivery failure.

The server forwards the SM usually to a SMSC. The SMSC delivers the message to the subscriber, either directly via SS7 or via other protocols.

A typical delay between user request and message delivery to the subscriber is 10 seconds. If the MS of the subscriber is switched off, the SMSC will typically try to deliver the message during the next 48 hours. If this fails, the message will be discarded silently.

2.4 Technical Details

Important configuration parameters for the user agent are defined by the operator. They include:

- External IP address or host name of the server.
- External IP port of the server.
- Access path to the HTTPTX interface on the server.
- Transport protocol to be used (commonly HTTPS, but HTTP might be used in combination with an IP VPN as well).
- User credentials, meaning user-name and password preventing unauthorized access to the interface

All this information is collected in a URI of the form:

PROT://USERNAME:PASSWORD@HOSTNAME:PORT/PATH_TO_HTTPTX_INTERFACE

A typical example might be:

```
https://mustermann:secret@www.operator.com/httptx/httptx
```

2.4.1 Delivery of Text SMS MT

MSISDN of the customer and the text of the message are transmitted as variables in a HTTP POST request, named MSISDN and TEXT. Both shall be provided in the ISO Latin 1 character set in URL encoded format. Message delivery to the customer is based on the ISO Latin 1 coding scheme. If the text message length exceeds 140 characters, the message is segmented automatically into several chunks of 134 bytes each and delivered as concatenated short messages using ISO Latin 1 data coding. For details refer to [2]. The maximum length of a single short message is 140 characters.

A complete trace of an message exchange between user agent and server is given below as detailed description. The SSL trace was discarded.:

```

0000: 50 4f 53 54 20 2f 68 74 74 70 74 78 2f 68 74 74 POST /httptx/htt
0010: 70 74 78 20 48 54 54 50 2f 31 2e 31 0d 0a 41 75 ptx HTTP/1.1..Au
0020: 74 68 6f 72 69 7a 61 74 69 6f 6e 3a 20 42 61 73 thORIZATION: Bas
0030: 69 63 20 5a 33 4a 31 5a 57 34 36 56 57 78 30 63 ic Z3JlZw46Vwx0c
0040: 6d 45 30 54 6d 56 30 0d 0a 55 73 65 72 2d 41 67 mE0TmV0..User-Ag
0050: 65 6e 74 3a 20 63 75 72 6c 2f 37 2e 31 36 2e 34 ent: curl/7.16.4
0060: 20 28 69 36 38 36 2d 73 75 73 65 2d 6c 69 6e 75 (i686-suse-linu
0070: 78 2d 67 6e 75 29 20 6c 69 62 63 75 72 6c 2f 37 x-gnu) libcurl/7
0080: 2e 31 36 2e 34 20 4f 70 65 6e 53 53 4c 2f 30 2e .16.4 OpenSSL/0.
0090: 39 2e 38 65 20 7a 6c 69 62 2f 31 2e 32 2e 33 20 9.8e zlib/1.2.3
00a0: 6c 69 62 69 64 6e 2f 31 2e 30 0d 0a 48 6f 73 74 libidn/1.0..Host
00b0: 3a 20 77 77 77 2e 61 64 76 65 6e 61 67 65 2e 69 : www.advenage.i
00c0: 6e 74 72 61 0d 0a 41 63 63 65 70 74 3a 20 2a 2f ntra..Accept: */
00d0: 2a 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 *.Content-Lengt
00e0: 68 3a 20 38 31 0d 0a 43 6f 6e 74 65 6e 74 2d 54 h: 81..Content-T
00f0: 79 70 65 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e ype: application
0100: 2f 78 2d 77 77 77 2d 66 6f 72 6d 2d 75 72 6c 65 /x-www-form-urle
0110: 6e 63 6f 64 65 64 0d 0a 0d 0a ncoded....
=> Send data, 81 bytes (0x51)
0000: 4d 53 49 53 44 4e 3d 30 31 37 37 38 39 35 36 38 MSISDN=017789568
0010: 36 32 26 54 45 58 54 3d 54 65 73 74 2b 77 69 74 62&TEXT=Test+wit
0020: 68 2b 67 65 72 6d 61 6e 2b 73 70 65 63 69 61 6c h+german+special
0030: 2b 63 68 61 72 61 63 74 65 72 73 2b 25 45 34 25 +characters+%E4%
0040: 46 36 25 46 43 25 43 34 25 44 36 25 44 43 25 44 F6%FC%C4%D6%DC%D
0050: 46 F
<= Recv header, 17 bytes (0x11)
0000: 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d HTTP/1.1 200 OK.
0010: 0a .
<= Recv header, 37 bytes (0x25)
0000: 44 61 74 65 3a 20 54 75 65 2c 20 31 35 20 4a 75 Date: Tue, 15 Ju
0010: 6c 20 32 30 30 38 20 31 39 3a 33 36 3a 30 39 20 1 2008 19:36:09
0020: 47 4d 54 0d 0a GMT..
<= Recv header, 35 bytes (0x23)
0000: 53 65 72 76 65 72 3a 20 41 70 61 63 68 65 2f 32 Server: Apache/2
0010: 2e 32 2e 34 20 28 4c 69 6e 75 78 2f 53 55 53 45 .2.4 (Linux/SUSE
0020: 29 0d 0a )..
<= Recv header, 28 bytes (0x1c)
0000: 54 72 61 6e 73 66 65 72 2d 45 6e 63 6f 64 69 6e Transfer-Encodin
0010: 67 3a 20 63 68 75 6e 6b 65 64 0d 0a g: chunked..
<= Recv header, 44 bytes (0x2c)
0000: 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74 65 Content-Type: te
0010: 78 74 2f 68 74 6d 6c 3b 63 68 61 72 73 65 74 3d xt/html;charset=
0020: 69 73 6f 2d 38 38 35 39 2d 31 0d 0a iso-8859-1..
<= Recv header, 2 bytes (0x2)
0000: 0d 0a ..
<= Recv data, 3 bytes (0x3)
0000: 32 0d 0a 2..
<= Recv data, 2 bytes (0x2)
0000: 4f 4b OK
<= Recv data, 2 bytes (0x2)
0000: 0d 0a ..
<= Recv data, 5 bytes (0x5)
0000: 30 0d 0a 0d 0a 0....

```

Please note:

- The message is delivered to the subscriber in ISO Latin 1 data coding. Depending on the capabilities of the connected SMSC and network not all characters of ISO Latin 1 may be transferred transparently.
- Data must be URL encoded.
- Message length for a single SM-MT is limited to 140 characters. If the message exceeds this length, the message is transferred in multiple SM-MT as concatenated short messages. This might cause additional costs.

- Binary SMS with UDH are supported using this interface, but need to be posted explicitly in this format (next section).
- User can neither define the originating short code or MSISDN of the message nor the global title of the SMSC to be used. This data is provided by the operator. If multiple short codes shall be used, multiple URLs need to be served.
- For test environments operator might decide to deliver SM-MT via other channels then the short code number or MSISDN defined. Customer perception is that the sender MSISDN might be different from the one expected.
- For technical reasons the operator might apply rate limits on SM-MT delivery. For example, test accounts are usually limited to 1 SM per second.
- The Operator might limit distribution of SMS to particular MNCs or customer MSISDN ranges. If a message can not be routed due to such limitations, usually a HTTP return code different from 200 will be returned.

2.4.2 Delivery of binary SMS MT

Binary SMS in this context mean SMS MT with individual UDH. An example of this message type are WAP push messages. To deliver a binary message with UDH, at least two parameters need to be set in the HTTP POST request: MSISDN and UDHSMS. A third optional parameter "DC" allows to define the data coding according to SMPP. If DC is not set, the default value 4 (binary 8 bit data) is assumed for DC as default.

All parameters (MSISDN, UDHSMS, DC) shall be supplied URL encoded. The SMS starts with the UDH. The first byte is the UDHL. The application assumes by default ESM_CLASS=0x40 and DATA_CODING=4.

Please note:

- Message length is limited to 140 bytes including UDH.
- User can neither define the originating short code or MSISDN of the message nor the global title of the SMSC to be used. This data is provided by the operator. If multiple short codes shall be used, multiple URLs need to be served.
- For test environments operator might decide to deliver SM-MT via other channels then the short code number or MSISDN defined. Customer perception is that the sender MSISDN might be different from the one expected.
- For technical reasons operator might apply rate limits on SM-MT delivery. For example, test accounts are usually limited to 1 SM per second.
- Operator might limit distribution of SMS to particular MNCs or customer MSISDN ranges. If a message can not be routed due to such limitations, usually a HTTP return code different from 400 will be returned in these cases.

Definition of commonly used DC (DATA_CODING) codes:

Value	Meaning
0	SMSC default alphabet
1	IA5
2	Octet unspecified (8-bit binary)
3	Latin 1 (ISO-8859-1)
4	Octet unspecified (8-bit binary)

3 User Agent Examples for User

In this section we provide some simple user agents deploying typical UNIX scripting concepts. User shall understand them as examples. Particular implementations might vary.

3.1 Text SMS Delivery

3.1.1 Example using WGET and PHP

This example uses the commonly used applications PHP and “wget” in a script to deliver SM-MT via HTTPTX:

```
#!/bin/sh
# short shell script using PHP and
# wget to send SMS via HTTPTX service
# Usage: wgetdemo msisdn text
#
# (C) 2008 ADVENAGE GmbH, http://www.advenage.com
#
MSISDN=`php -r "echo urlencode(\"$1\").\"\\n\";"`
TEXT=`php -r "echo urlencode(\"$2\").\"\\n\";"`
wget --no-check-certificate --post-data "MSISDN=$MSISDN&TEXT=$TEXT" \
https://mustermann:secret@www.operator.com/httptx/httptx
```

Please note:

- Both, MSISDN and TEXT shall be URL encoded, especially in case the + MSISDN format shall be used, such as +4915775123456.
- If the text exceed 140 characters in length, multiple concatenated SM are generated..
- Application requires previous installation of “php” and “wget”.
- The demo application does not check the return value.

3.1.2 Example using CURL and PHP

This example uses the commonly used applications PHP and “curl” in a script to deliver SM-MT via HTTPTX:

```
#!/bin/sh
# short shell script using PHP and
# curl to send SMS via HTTPTX service
# Usage: wgetdemo msisdn text
#
# (C) 2008 ADVENAGE GmbH, http://www.advenage.com
#
MSISDN=`php -r "echo urlencode(\"$1\").\"\\n\";"`
TEXT=`php -r "echo urlencode(\"$2\").\"\\n\";"`
curl -k -d "MSISDN=$MSISDN&TEXT=$TEXT" https://mustermann:secret@www.operator.com/httptx/httptx
```

Please note:

- Both, MSISDN and TEXT shall be URL encoded, especially in case the + MSISDN format shall be used, such as +4915775123456.
- If the text exceed 140 characters in length, multiple concatenated SM are generated..
- Application requires previous installation of “php” and “wget”.
- The demo application does not check the return value.

