AnnyWay Enterprise SMS Gateway

HTTP SMS Open Interface specification

Document Version: 2.01
Software Version: 3.9
Document Language: English

Document:	SMS Gateway HTTP SMS Open Interface specification
Document version:	2.01
Software version:	3.9
Department/Country:	BUC/Česká Republika
Author:	Vojtěch Pithart
Created date:	2007-10-01
Last revision by:	
Last revision date:	2016-12-20
State:	final
Limitations:	ready for production use

This document is protected by copyright. All rights, including those of translation, of reprinting and of copying using photo-mechanical or electronic means, are reserved. Protected trademarks, registered names etc., are not identified in the text. The absence of such a designation does not mean that a name is free of copyright within the context of the trade and brand name legislation. The names of persons and companies which are used as examples are purely fictitious.

Limitation of liability

The information contained in this document has been carefully checked, and as such may be considered to be reliable. However, we cannot undertake to guarantee that information specified in this document is without error. In particular, no commitment has been made as to whether the products which have been described are or are not suitable for particular purposes.

T-Mobile Czech Republic a.s. reserves the right to make changes to the products and product information. T-Mobile Czech Republic a.s. does not accept any further liability which results from the use of the products here described. The issuing of this document does not constitute any kind of license to use the products detailed, neither from T-Mobile Czech Republic a.s. nor from third parties.

T-Mobile Czech Republic a.s.
Tomíčkova 2144/1
148 00 Praha 4
Czech Republic

http://www.t-mobile.cz/

History of changes

Version	*Changed by*	*Date*	*Comment*
1.1	Vojtech Pithart	1.5.2005	init
1.2	Vojtech Pithart	1.3.2008	MO and DR specifications updated
1.4.	Petr Hejduk	10.4.2008	Graphic design and correction
1.5draft	Vojtech Pithart	15.4.2008	
1.5draftRC 1	Vojtech Pithart	6.5.2008	Data structures of AT SM and DR defined
1.5draftRC 2	Vojtech Pithart	22.5.2008	Client-side timing rules; server CONF push
1.5draftRC 3	Vojtech Pithart	23.5.2008	HTTP request for AO SM extended with "force-tts" parameter; cleint side timing updated.
1.6	Vojtech Pithart	28.5.2008	
1.7	Vojtech Pithart	3.6.2008	<delivery result> definition updated (for delivery results); new argument online_since for presence mode.
1.8	Vojtech Pithart	9.6.2008	final for AWEG server 3.0
1.9	Vojtech Pithart	24.7.2008	REPORT one-line format extended with “original timestamp”
1.10	Vojtech Pithart	28.7.2008	103: return value definition; Service Messages
1.11	Vojtech Pithart	12.8.2008	Retun code 320 defined; service messages updated
1.12	Vojtech Pithart	2.9.2008	Added “309 too many connections” and “310” premium numbers barred
1.13	Vojtech Pithart	10.2.2009	AWEG terminated (MO-AT) message one-line format – separator changed to “;”
1.14	Vojtech Pithart	21.1.2011	Added 321 “High Priority is not allowed for user”
1.15	Vojtech Pithart	30.5.2011	High priority request (_prio)
2.01	Vojtech Pithart	26.8.2011	Character encoding (UTF-8); Optional parameter cstoascii; concatenated messages. Alphanumeric sender - parameter use_anumber.
2.01	Marcela Spicanova	20.12.2016	Max. number of characters in SMS

About

AWEG stands for SMS Gateway, the main end-user oriented SMS connectivity solution.

This document is a reference guide for HTTP based interface of AWEG.

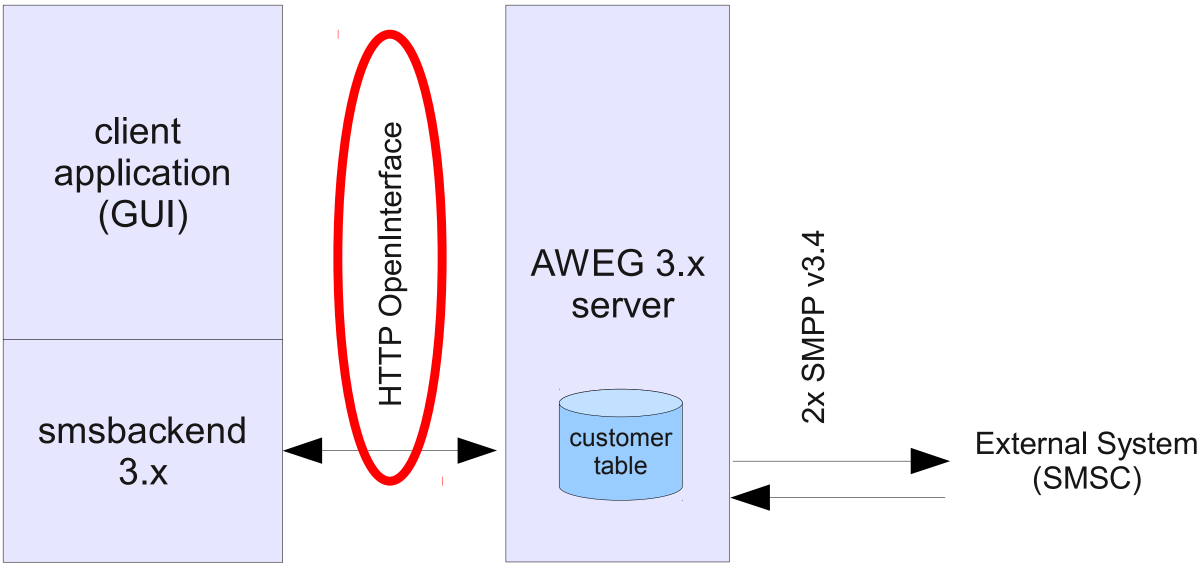
Features:

- Sending of MT SMS from client/application via HTTP
- Receiving MO SMS by client/application via HTTP
- Receiving DR by client/application via HTTP

The service is provided by either

- T-Mobile Czech Republic a.s., or
- third party, such as national fixed operator.

On the provider's side, there is a customer database, which contains information about usernames and passwords permitted to use the service. Administration of such a database is described in separate document.



HTTP OpenInterface is covered in this document

Sending of AO-MT SMS via HTTP

Sending of one SMS is done via calling HTTP GET request with all required parameters in URI, after the ? (question mark). Except network HTTP error codes, the sending result is returned as at least one line of text/plain response body.

Message (smstext key of URI) can be up to ~1530[1] characters: text will be split up to 10 SMS.

Sending of bulk (more than one message/receiver pair at a time) is performed by sending separate HTTP GETs one after another, withe the *same* "bulk" key. See below.

Web request definition

The submit URL looks like[2]:

```
https://<server>:<port>/<uri>?key=value&key2=value2&...
```

The arguments are listed in table below. All the values are URLescaped[3] - all nonprintable and URL-reserved characters are coded as % (percent sign) followed by 2 hexadecimal digits of ASCII code.

Example

The example below shows how user John send a message to the mobile station +420602127001. This example uses command line tool “wget” which is available on Linux and most of other platforms as well.

```
  [linux] $ wget -O- "https://aweg.t-mobile.cz/?auth=John:xxxx&receiver=
%2B420602127001&smstext=This%20is%20testing%20message%21" 2>/dev/null

 200 [1] bodypart, accepted as [10172622]
 102:63
 103:reports=1
```

The blue part is the command typed into console. Green part is exact form of returned text; for reference, see the paragraph “Return values and error handling” below.

Note: for testing purposes, there is a special receiver number “999999999”. Any message sent to this number is accepted by the server, discarded and not billed.

[1] Maximum number of bodyparts of concatenated message is 10 by default.

[2] The actual address of <server>, <port> and <uri> is assigned by T-mobile

[3] I recommend to use Perl function uri_escape, see http://search.cpan.org/~gaas/URI-1.35/URI/Escape.pm

Reference

List of URL parameters for AO-MT SMS			
Name	**M/O**[1]	**Examples**	**Meaning**
auth	M	Jack1:secretword12 Jack1_prio:secretword12	Plaintext username:password, concatenated with colon, of user that sends (and pays) the message. High priority Request for high-priority message may be expressed by adding "_prio" to the username. Message with high priority might be extra charged.
receiver	M	+420602127001	Telephone number of SMS recipient. Accepted formats are: – international E.164 form with leading "+" or "00" – short national form[2] (usually 9 digits)
smstext	M	Hello%20world	Message text itself; the total length can exceed 160 (70) characters, in which case message will be split to the parts of 153 (67) characters. See also: option cstoascii. Message text encoding to be used is UTF-8.
report	O	1	If set to 1, delivery-report is requested for this message.
use_anumber	O	1	1 ... message has real A-number in sender 0 ... shared shortcode is used as an A-number in sender use_anumber has precedence over use_alphanum
use_alphanum	O	1	1 … message sende will be alphanumeric string (is provisioned for AWEG customer) 0 … no alphanumeric sender
bulk	O	11223456 11223456end	Pseudorandom number identifying the bulk (group of messages send at a time). This is set by smsbackend to the same value for all of messages within one bulk. String "end" appended to the number identifies last message of the bulk. If not used, the bulk is considered as closed by the server after 15 minutes.
cstoascii	O	1	If set to 1, message text is filtered through "cstoascii" program. Accents of Central European characters are removed. Since AWEG version 3.9, default is 0 (no filtering). For AWEG v3.8 and older, there was hardcoded cstoascii=1.
be	M	1.1/27	Version and build number of smsbackend. Free format up to 20 ASCII characters. This is used for logging and for customer technical support. When implementing this protocol in any application, feel free to submit this

[1] (M)andatory or (O)ptional

[2] In this case, international prefix is prefixed into the recipient number. This is set on AWEG server, e.g. "00420"

			information.
fe	M	b-SMS/1.34cz	Identification and version end-user application. Free format up to 30 ASCII characters. Both "fe" and "be" keys should be sent only with first request within one bulk.

Return values and error handling:

Return value on TCP/IP level:

The TCP layer should be connected. Each other state on TCP level (network unreachable, no route to host, connection refused, timeout etc.) should be treated as temporary error and message should be retried.

Return value on SSL level:

The SSL connection should be established. Client application must not check the certificate issuer (even self-signed certificate may be used) and match of hostname with certificate's common name[1]. Each other state on SSL level should be treated as temporary error and message should be retried.

Return value on HTTP level:

Each successful request returns HTTP code “200 OK”. Each other state on HTTP level should be treated as temporary error and message should be retried.

Return value on Application level

AWEG server returns a short document of Content-Type: text/plain, pre-formatted for machine processing. Each line is terminated by LF (\n) character.

```
<return code> <additional info>  (first line - mandatory)
102:<daily_limit_rest>           (informational)
103:<server_feature_list>        (informational list of server-side features)
<Some more human-readable notes> (informational)
```

<return code>   3-digit number describing whether message is accepted or not. See return codes reference table below.

<additional info>   see “2xx codes – positive acknowledgment” paragraph below.

<daily_limit_rest>   Remaining quota of messages remaining for today; decimal positive number.
<server_feature_list>Listing of server-side features configured for particular user.[2]

Return codes reference table:

1xx - informational message
2xx - positive acknowledgment
3xx - negative acknowledgment (errors)

[1] unless agreed with T-mobile – service provider

[2] The “103” line may occur multiple times within one response

1xx codes (informational):

100 Common informational message
101 Localized welcome message (per user)
102:<n> Daily limit rest (no. of messages left today)
103:<key1>=<value1>[;<key1>=<value1>[;...]] Server-side features and configuration; See “service messages – response” below.
104:<n> Used together with return code 309; client should try to reconnect after <n> seconds.

2xx codes (positive acknowledgment):

20x	previous command was completed successfully (generic ACK)
200	... with no quota information
201	... with daily quota note (level1 reached - warning)
202	... with daily quota note (level2 reached - next message will be rejected)
209	timeout – server has nothing to return now.

2xx code also contains additional information. In case of successful message transfer (return codes 200-209), this contains information about number of parts the message was split into, and an unique ID of each part. The message is formatted as follows:

```
200 [1] bodypart, accepted as [10172622]
```

or

```
200 [3] bodyparts, accepted as [f0011424 f0011425 f0011426]
```

The first block in [square brackets] is number of parts.
The second block in [square brackets] is space-delimited list of unique IDs of messages, as received by AWEG server. Unique ID is 8-digit hexadecimal number (range from 0x00000000 to 0xffffffff).

3xx codes (negative acknowledgment):

300	generic error or system failure
301	message text not accepted
302	Daily limit level2 reached (sms rejected)
303	Message not ready for sending
304	user not authorized
305	Error storing billing record
307	internal communication error (VCD)
308	internal communication error (GPCS)
309	too many connections, please connect again later (see also: code 104)
310	premium-SMS destination number barred for user
311	free testing SMS count reached
312	invalid GPCS card number
313	GPCS card balance is not sufficient
314	GPCS account is expired
315	GPCS: No matching route found to destination
320	Service message processing error
321	High Priority is not allowed for user

Service messages

Service message is a special kind of AO message sent by AWEG client. Such a message contains fixed string “service007” instead of destination number.

Service message definition - request

Receiver:	“service007”
Text:	command1[;command2[;...]]

Commands:

VOID

no operation; the “103:” response returns current settings

SET:smReceiveAT=1

user's profile property 'smReceiveAT' is set to “1” - AWEG server will receive AT SMS for this user

SET:smReceiveAT=0

user's profile property 'smReceiveAT' is set to “0” - AWEG server will no longer receive AT SMS for this user

SET:smATExpiry=<n>h or <m>d

user's profile property 'smATExpiry' is set to <n> - AT message expiry will be set to <n> hours or <m> days

SET:smForwardType=none

user's profile property 'smForwardType' is emptied – AT message forwarding will be disabled

SET:smForwardType=MFU

user's profile property 'smForwardType' is set to “MFU” – any incoming (AT) message for this user is forwarded.

SET:smForwardType=MFO

user's profile property 'smForwardType' is set to “MFO” – any incoming (AT) message is forwarded when user is offline.

SET:smForwardTargetNo=<digits>

user's profile property 'smForwardTargetNo' is set – it's destination number for forwarded messages. Empty string removes the value.

SET:smForwardTargetEmail=<address>

user's profile property 'smForwardTargetEmail' is set – it's destination e-mail address for forwarded messages.. Empty string removes the value.

Service message definition – response

On success, server returns “200 OK” as for AO-MT message.

On error, server returns “315 <human_readable_message>”; The error message tries to explain what was wrong with the request. It should be presented to the end-used.

The response of service message also contains the “103” line. The “103” line can occur multiple times within one response. The format is:

```
103:key=value[;key2=value2[;...]]
```

The “key” is ASCII string of those characters: [A-Za-z0-9_].

The “value” is uri-escaped string (UTF8 in general).
The trailing “;” is optional.

Example:

```
103:S_reports=1
103:U_anumber=420234456100;U_smForwardTargetEmail=john@example.org;
```

The S_.... configuration items are returned within each AO-MT SMS request

Well-known system-wide configuration items			
key	**defined values**	**example**	**description**
S_reports	0 or 1	1	AO-MT Delivery reports are supported by server (1) or not (0).
S_smATExpiryList	string	3h/1d/3d/7d	slash-separated list of Nh (hours) or Nd (days)

The U_.... configuration items are returned within “service” AO SMS request (destination=”service007”)

Well-known customer-specific configuration items (a.k.a. user profile)			
key	**values**	**example**	**description**
U_anumber	digits 0-9	420234493147	User's A-number configured on server side. E.164 format (international without +). This may be empty (a.k.a. subscriber without A-number).
U_smReceiveAT	0 or 1	1	The end-user wants to receive AT messages within AWEG platform (1) or not (0).
U_smATExpiry	positive integer	72	Expiry timeout for AT messages (hours).
U_smForwardType	“MFU“ or „MFO“ or “none”	MFU	This is forwarding type definition. Empty = user have no message forwarding active. MFU (message forward unconditional) = any incoming (AT) message for this user is forwarded. MFO (message forwarding on offline) = any incoming (AT) message is forwarded when user is offline. MFO is applied also in condition when delivery of AT message fails within N seconds (see smForwardDlvrTimeout). Message forwarding will become inactive when none of smForwardTargetNo or smForwardTargetEmail is set.
U_smForwardTargetNo	digits 0-9	724014566	Destination number for forwarding; this is used if smForwardType is MFU or MFO. Both smForwardTarget{No,Email} can be used simultaneously.
U_smForwardTargetEmail	e-mail address	john@doe.net	Destination e-mail address for forwarding; This is used if smForwardType is MFU or MFO.Both smForwardTarget{No,Email} can be used simultaneously.
U_smForwardDlvrTimeout	positive integer	90	Delivery timeout for conditional forward. If user have smForwardType=MFO, user is online, but incoming (AT) message is not delivered to user within (90) seconds (network congestion etc.), user is considered as offline and message forwarding takes place.
U_useAnumber	0 or 1	1	Default setting of „use Anumber“ for AO-MT messages.

Receiving MO-AT messages (synchronous)

MO-AT (mobile originated, aweg terminated) messages can be received in on-line mode. This mode is in-depth explained in “Client-server presence via HTTP” below. One of entities client can receive in this mode is short message line staring by “SM:” and its format is defined in “AWEG terminated (MO-AT) message one-line format” inside this document.

Example of received message:

```
SM:000001e1;+420604999887;+420234493147;20080603141057;GSM7;Hello%20world
```

Message from +420604999887 to +420234493147 (number associated to aweg client) with text “Hello world”.

Once client receives such a line, it must confirm this with another (parallel) HTTP request with “ack=” parameter. The not-confirmed message is being re-sent to the client every ~15 seconds.

Example how clients confirms reception of a message

```
  [linux] $ wget -O- "https://aweg.t-mobile.cz/longtime?
auth=John:xxx&ack=M:000001e1"

 INFO: ACK-deleting messages 000001e1
```

MO-AT Message lifecycle

- AWEG server is statefull, with per-user message queue
- Message queue has a configurable expiry time (default = 72 hours)
- Messages from the queue are delivered to the client in a first possible time, after the clients conects (sets the presence up)
    - Messages are deleted from queue after client acknowledges reception of message (see ack= parameter of client-server presence)
- When message forwarding takes place, forwarding is performed immediatelly (regardless of client's presence)
    - forwarded message is placed into sm_out spooldir, as if it were originated by aweg user
    - billing record is created

Receiving Delivery Reports (synchronous)

Delivery reports for messages previously sent by aweg uesr can be received in on-line mode. This mode is in-depth explained in “Client-server presence via HTTP” below. One of entities client can receive in this mode is delivery report line starting by “REPORT:” and its format is defined in “AWEG terminated delivery report one-line format” inside this document.

Example of received delivery report:

```
REPORT:000001e5,03143918,2,20080602143845,20080603143900,+420724010456
```

The message with ID “000001e5” was DELIVERED (state=2) at 3.6.2008 14:39.

Once client receives such a report, it must confirm it with another (parallel) HTTP request with “ack=” parameter. The not-confirmed report is being re-sent to the client every ~60 seconds.

Example how clients confirms reception of a delivery report

```
  [linux] $ wget -O- "https://aweg.t-mobile.cz/longtime?
auth=John:xxx&ack=R:000001e5"

 INFO: ACK-deleting reports 000001e5
```

Delivery report lifecycle

- AWEG server is statefull, with per-user report queue
- Report queue has a configurable expiry time (default = 32 days)
- Reports from the queue are delivered to the client in a first possible time, after the clients conects (sets the presence up)
- Reports are delivered to the client via two channels:
  - Asynchronous on-request (v2.0 compatibility): client can get all reports (specified by timeframe) via HTTP request; there is no ACK mechanism from client to server thus reports **are not deleted** after get[1].
  - Synchronous: (v3.0+): Delivery reports are passed to the client's HTTP response (see client-server presence) in real time; there is ACK mechanism defined, therefore reports **are deleted** from queue after client's ACK.
- Both mechanisms (Async and Sync) for getting delivery reports can be combined

[1] all reports are deleted after configurable timeout (32 days) regardless how many times client got them

Client-server presence via HTTP (long-timeout polling)

Presence is used to express an client's current network availability (offline or online), and to notify server of that availability. Client which is connected (has a real-time presence) can receive messages and delivery reports in realtime. Presence information is also used as a condition for conditional message forwarding.

Message/report transfer is done by constantly requesting a web service from client. Client invokes Web Service via HTTP/HTTPS which times out (waits) until a message is received, or until a maximum time is reached. After the connection is closed by server, client should open a new one. The timing scheme of this procedure is descripted below.

Web request definition

```
https://<server>:<port>/<uri>/longtime?auth=<username>:<password>&...
```

The arguments are explained in “reference” block below. This request returns either all message(s) waiting for user, or nothing (timeout).

Example

The example below shows how user John polls for incoming messages. Examples below uses command line tool “wget” which is available on Linux and most of other platforms as well. The blue part is the command typed into console. Green part is exact form of returned text; for reference, see the paragraph “Return values and error handling” below.

Client polls for messages – no messages are received within 900 seconds:

```
  [linux] $ wget -O- "https://aweg.t-mobile.cz/longtime?
auth=John:xxxxx&sleep=900" 2>/dev/null

 209 TIMEOUT, please connect again
```

(a) Client polls for messages – two messages and one report were waiting (returned immediately)

```
  [linux] $ wget -O- "https://aweg.t-mobile.cz/longtime?
auth=John:xxxxx&sleep=900" 2>/dev/null

 SM:5000ab11,+420602123456,495495495,20080331235959,UTF-8,This%20is
%20testing%20message%20A%21
 SM:5000ab12,+420602123456,495495495,20080401000025,UTF-8,This%20is
%20testing%20message%20B%21
 REPORT:1001f6a8,30163840,1,20080312180701,20080312180713,+4915702100210
```

(b) Clients confirms two received messages to the server (with no polling)

```
  [linux] $ wget -O- "https://aweg.t-mobile.cz/longtime?
auth=John:xxxxx&ack=M:5000ab11,5000ab12" 2>/dev/null

 INFO: ACK-deleting messages 5000ab11,5000ab12
```

(c) Clients confirms two received messages and one delivery report to the server (with no polling)

```
  [linux] $ wget -O- "https://aweg.t-mobile.cz/longtime?
auth=John:xxxxx&ack=M:5000ab11,5000ab12,R:5114c1dd" 2>/dev/null

 INFO: ACK-deleting messages 5000ab11,5000ab12
 INFO: ACK-deleting report 5114c1dd
```

Reference

List of URL parameters for MO-AT SMS (aweg-terminated)			
Name	**M/O[1]**	**Examples**	**Meaning**
auth	M	Jack1:secretword12	username:password of user that receives the message.
ack	O	M:f006ab56,M:f006ab57,R:54007ddc	List of unique ID of messages (M) or delivery reports (R) previously received by client; such a message is considered as delivered (to client) and deleted by AWEG server. Comma-separated list of 10-digit snippets (M: or R: and 8-digit hexadecimal numbers). Maximum 256 IDs at a time. Prefix M: or R: can be omitted in second and subsequent snippets; in such a case, the same prefix as in previuos snippet is assumed. With ack, there are no messages returned by the server.
sleep	O	900	Maximum time (in seconds) for a request. After this time, server returns “209 TIMEOUT” and closes the connection. The default timeout is set in server's configuration.
limit	O	1	Integer number; default 1000. Maximum number of messages returned by server within a single request. Parameter limit=0 can be used to receive no message – only meaningful for sending “ack”.
online_since	O	20080422115959	Timestamp when client went online for first time. Client's application (like smsbackend) should remember this timestamp and send the same value within all HTTP requests.
be	O	v3.0/r63/2008-05-22	Version and build number of smsbackend. Free format up to 20 ASCII characters. This is used for logging and for customer technical support. When implementing this protocol in any application, feel free to submit this information.
fe	O	b-SMS/1.34cz	Identification and version end-user application. Free format up to 30 ASCII characters.

Return values and error handling:

Return value on TCP/IP level:

The TCP layer should be connected. Each other state on TCP level (network unreachable, no route to host, connection refused, timeout etc.) should be treated as temporary error. The request should be repeated after 15 seconds delay.

Return value on SSL level:

The SSL connection should be established. Client application must not check the certificate issuer

[1] (M)andatory or (O)ptional

(even self-signed certificate may be used) and match of hostname with certificate's common name[2]. Each other state on SSL level should be treated as temporary error. The request should be repeated after 60 seconds delay.

Return value on HTTP level:

Each successful request returns HTTP code “200 OK”. Each other state on HTTP level should be treated as temporary error. The request should be repeated after 60 seconds delay.

Return value on Application level

AWEG server returns a short document of Content-Type: text/plain, pre-formatted for machine processing. Each line is terminated by LF (\n) character.

Output

First line of output:

```
<return code> <additional info>
```

`<return code>` 3-digit number describing whether message is accepted or not. See return codes reference table.

`<additional info>` Human readable english message

Examples of 1st line of output:

`200 OK`	Login was successful
`304 not authorized (Invalid password for user xxxyyyzzz)`	Username or password problems

Next line(s) of output:

One line contains one entity: short message or delivery report.

- Short message line starts by “SM:” and its format is defined in “AWEG terminated (MO-AT) message one-line format” inside this document.
- Delivery report line starts by “REPORT:” and its format is defined in “AWEG terminated delivery report one-line format” inside this document.
- Config-push line starts by “CONF:” Extensible configuration push in form “CONF:variable=value”. Server can modify client's configuration at runtime. See also: client-side timing rules.
- Informational line starts by “INFO:” and its format is not specified.

CONF string sent by server	*client's action*
CONF:intervalA=NNN	see client-side timing rules below

[2] unless agreed with T-mobile – service provider

CONF:intervalB=NNN	
CONF:intervalC=NNN	
CONF:U_anumber=NNNNNNNNN	Information about client's A-number (format E.164)

Client-side timing rules

Client connected to AWEG server must implement timing structure specified here. There are 3 pre-defined (default) intervals in seconds. These values can be modified by the server.

First HTTP request:

After initialization, AWEG client sends the request with “sleep=0”. This request will be finished imediatelly and server can send “CONF:” push to set the clien's intervals. Result of this request can contain CONF lines as well as SM: or REPORT: lines.

Subsequent HTTP requests:

Normal operation of client is to make one request after another, with timing specified in diagram below. Nevertheless, “CONF:” push can be given by the server at any time during operation. Client must be able to update its interval settings any time.

Interval definition

Interval	*Purpose*	*Client's default start-up value[sec]*	*Overrided by server...*	*... in range [sec]*
A	HTTP request “sleep” time	290	yes	0-3600
B	Client's pause after finished HTTP request	5	yes	1-900
C	Client's pause after failure of HTTP request	30	yes	15-900

“SET local intervals” operation

The intervals are modified by server using “CONF” lines.

CONF string sent by server	*client's action*
CONF:intervalA=NNN	Client MUST set NNN as a "sleep" argument for next request
CONF:intervalB=NNN	Client MUST wait NNN seconds between requests (after successfull http get)
CONF:intervalC=NNN	Client MUST wait NNN seconds between requests (after failed http get)

report:

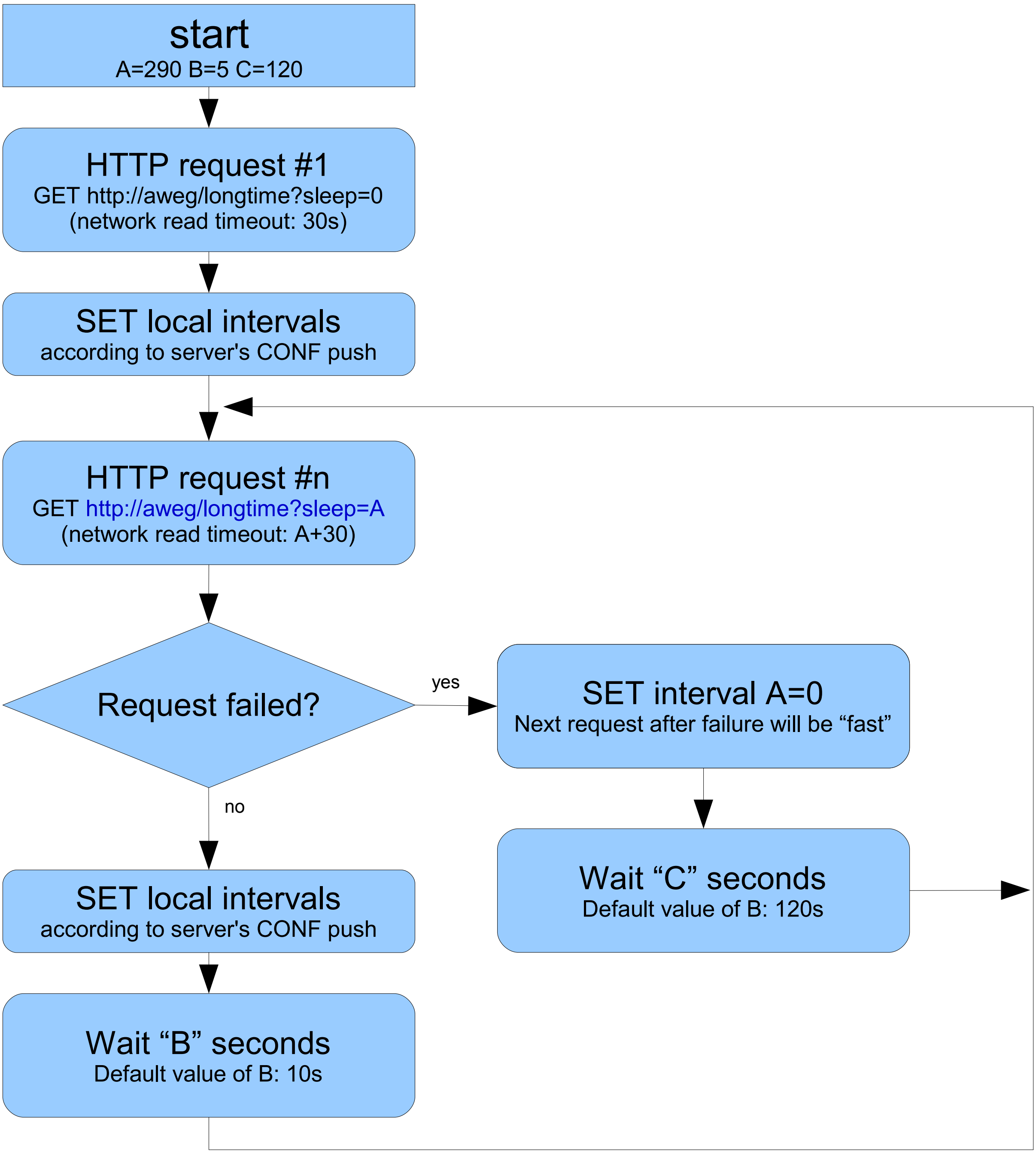

start
A=290 B=5 C=120
HTTP request #1
GET http://aweg/longtime?sleep=0
(network read timeout: 30s)
SET local intervals
according to server's CONF push
HTTP request #n
GET http://aweg/longtime?sleep=A
(network read timeout: A+30)
Request failed?
yes
SET interval A=0
Next request after failure will be “fast”
Wait “C” seconds
Default value of B: 120s
no
SET local intervals
according to server's CONF push
Wait “B” seconds
Default value of B: 10s

Diagram: client's HTTP request timing

Receiving Delivery Reports via HTTP (asynchronous on-request)

Receiving of delivery reports done via calling HTTP GET request with all required parameters in URI, after the ? (question mark). Except network HTTP error codes, the sending result is returned as at least one line of text/plain response body.

Web request definition

The submit URL looks like[1]:

```
https://<server>:<port>/<uri>/report?auth=<login>:<password>&since=<timestamp>
```

The arguments are listed in table below.

Example

The example below shows how user John requests delivery reports for messages previously sent. This example uses command line tool "wget" which is available on Linux and most of other platforms as well.

```
  [linux] $ wget -O- "https://aweg.t-mobile.cz/report?
auth=John:xxxxx&since=20080312180000" 2>/dev/null

 REPORT:1001f6a8,30163840,2,20080312180701,20080312180713,+4915702100210
 REPORT:1001f6a7,30162554,2,20080312180701,20080312180712,+303685541088
```

The blue part is the command typed into console. Green part is exact form of returned text. For reference of REPORT lines, see "AWEG terminated delivery report one-line format"

Reference

List of URL parameters for Delivery Reports			
Name	**M/O[2]**	**Examples**	**Meaning**
auth	M	Jack1:secretword12	username:password of user that requests delivery reports.
since	O	20080425112233	No reports with close-timestamp earlier than specified is not returned. Client should give the timestamp of last "seen". The format id 14-digit string in form YYYYMMDDhhmmss.

[1] The actual address of <server>, <port> and <uri> is assigned by T-mobile

[2] (M)andatory or (O)ptional

Return values and error handling:

Return value on TCP/IP level:

The TCP layer should be connected. Each other state on TCP level (network unreachable, no route to host, connection refused, timeout etc.) should be treated as temporary error and request should be retried.

Return value on SSL level:

The SSL connection should be established. Client application must not check the certificate issuer (even self-signed certificate may be used) and match of hostname with certificate's common name[1]. Each other state on SSL level should be treated as temporary error and request should be retried.

Return value on HTTP level:

Each successful request returns HTTP code “200 OK”. Each other state on HTTP level should be treated as temporary error and request should be retried.

Return value on Application level

AWEG server returns a short document of Content-Type: text/plain, pre-formatted for machine processing. One delivery reports is returned on separate line, terminated by LF (\n). One line of output contains one-line format of delivery report. This is defined in paragraph “AWEG terminated delivery report one-line format” inside this document.

[1] unless agreed with T-mobile – service provider

Data structures

AWEG terminated (MO-AT) message one-line format

Definition

```
  SM:<uniqueID>;<sedner>;<receiver>;<timestamp>;<encoding>;<text>
  SM:<uniqueID>;<sedner>;<receiver>;<timestamp>;<encoding>;<text>
  ...
```

SM: Fixed string "SM:" with optional whitespace between colon (:) and first digit of uniqueID.

<uniqueID> 8-digit hexadecimal number, unique for each message. This ID is assigned by AWEG server. Client should ACKnowledge this ID immediately by sending next HTTP request with "ack" option.

<sender> Sender's international MSISDN in E.164 (international) format with leading "+" added. Note that "+" character is URL-escaped into "%2B"

<receiver> Receiver's number. Can be one of:

- international MSISDN in E.164 (international) format with leading "+" added.
- national number or service number (shortcode) without leading "+".

<timestamp> Local date and time of when message was received by SMSC. Format is YYYYMMDDhhmmss.

<encoding> Character encoding used for message <text>. Character encodings are defined by IANA at http://www.iana.org/assignments/character-sets. The default encoding is UTF-8.

<text> URL-escaped string – content of short message.

Example

```
 SM: 5000ab11;%2B420602123456;%2B495495495;20080331235900;UTF-8;This%20is
%20testing%20message%20A%21
```

Short message from +420602123456 to +495495495, sent at 23:59[1]; Message ID is 5000ab11.

AWEG terminated delivery report one-line format

Definition

```
  REPORT:<messageID>,<bulkID>,<delivery result>,<original timestamp>,<final
timestamp>,<receiver msisdn>
```

<messageID>: 8-digit hexadecimal number, unique for each message. This ID is assigned by

[1] This is service centre timestamp (SCTS) from originating SMSC

AWEG server on receiving MT message. See “2xx codes (positive acknowledgment)” above.

<bulkID>: decimal number specifying the bulk (group of MT SMS sent together) - see MT SMS reference, parameter “bulk”. If you are not sure, you can safely ignore this.

<delivery result>: 0 ... no delivery report information (yet)
1 ... ENROUTE: The message is in enroute state.
2 ... DELIVERED Message is delivered to destination *(this is the one mostly seen in wild nature)*
3 ... EXPIRED Message validity period has expired.
4 ... DELETED Message has been deleted.
5 ... UNDELIVERABLE Message is undeliverable
6 ... ACCEPTED Message is in accepted state (i.e. has been manually read on behalf of the subscriber by customer service)
7 ... UNKNOWN Message is in invalid state
8 ... REJECTED Message is in a rejected state

The delivery result conforms to the SMPP message_state, received with SMSC Delivery Receipt via aweg's SMPP-ESME process.

<original timestamp>: Date and time in format YYYYMMDDhhmmss, when the original AO-MT message was sent.

<final timestamp>: Date and time in format YYYYMMDDhhmmss, when final state of message delivery was reached.

<receiver msisdn>: Telephone number of SMS recipient – international E.164 form with leading “+”. The character “+” is not URL-escaped here.

Example

```
REPORT:1001f6a8,30163840,2,20080312180701,20080312180713,+4915702100210
```

Message to +4915702100210 (id 1001f6a8), was delivered at 18:07.

```
REPORT:1001f6a7,30162554,2,20080312180830,20080312180844,+303685541088
```

Message to +303685541088 (id 1001f6a7), was delivered at 18:08.

```
REPORT:1001f6a1,30162554,3,20080312181501,20080315181512,+421910301145
```

Message to +421910301145 (id 1001f6a1), was NOT delivered (due to expiration time); the final state of non-delivery is valid as of 18:15.

Glossary

Glossary	
AWEG3	AnnyWay Enterprise SMS Gateway, second version: client/server solution, which is used for sending short messages (SMS) from desktop PCs or applications to the mobile (GSM) or fixed (PSTN/ISDN) telephones.
SMS	Short Message Service
MT SMS	Mobile Terminated SMS, i.e. when message is being sent **to** mobile station
MO SMS	Mobile Originated SMS, i.e. when message is being sent **from** mobile station
A-number	Telephone number of sender (of message) or of callee (of voice call)
wget	Command-line non-interactive HTTP/FTP client
DR	delivery report
HTTP	hypertext transfer protocol
MSISDN	Mobile station international subscriber number
TTS	Text To Speech - system that "reads" the short text message using synthetic voice
SMPP	Short Message peer to peer; see Protocol Apecification v3.4, Issue 1.2