HW group s.r.o. Rumunská 26/12, Praha 2, 120 00, Czech Republic
Phone: +420 222 511 918, E-mail: sales@hwg.cz, support@hwg.cz
www.HW-group.com

How to send SMS via HWg-SMS-GW3

HWg-SMS-GW3 is a SMS gateway for sending text message alarms over a LAN. HWg-SMS-GW3 is primarily intended for use with HW group products and software. However, its communication interfaces are sufficiently documented for use with third-party systems. This Application Note shows how to use HWg-SMS-GW3 in such third-party systems.

Methods of sending text messages:

Sending text messages using the netGSM protocol

The netGSM protocol is based on SOAP; that is, sending a XML file using HTTP POST. For a description of the netGSM protocol, see: http://hw-group.us/product-version/netgsm

In a nutshell, HTTP POST is used to send a XML file (service.xml) with the following structure:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <pos:QueueAdd>
      <Queue>GsmOut</Queue>
      <Gsm>
        <Cmd>SMS</Cmd>
        <Nmr>123456789</Nmr>
        <Text>Hello World!</Text>
      </Gsm>
    </pos:QueueAdd>
  </soapenv:Body>
</soapenv:Envelope>
```

For a description of individual parameters, see the above-mentioned protocol description.

Uploading a XML file using cURL

cURL is a command-line tool to transfer data using protocols such as HTTP, FTP and more. cURL consists of two parts – the curl.exe console application, and the libcurl.dll library that implements all the curl functions. The library can be used for writing your own scripts (bindings are available for many languages). cURL is available free of charge at https://curl.haxx.se/ (https://curl.haxx.se/download.html); however, attention must be paid to

choose the correct version to download. The 32-bit version will not work in a 64-bit Windows environment, even though it starts OK!

cURL is started from the command line with the following command:

```
curl -X POST -T service.xml 192.168.100.169
```

where:

-X defines the transfer method, in this case POST
-T specifies that a file is uploaded and what is its name (the file must be in the same folder as the curl.exe utility; otherwise, the path must be given)
IP address is the address of HWg-SMS-GW3

If the HWg-SMS-GW3 is secured with a username and a password, the credentials must be specified with the *-u* parameter:

```
curl -X POST -T service.xml -u user:pass 192.168.100.169
```

The response is:

```
C:\Users\volmr\Desktop\curl>curl -X POST -T service.xml -u user:pass 192.168.100.169
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total   Spent    Left  Speed
  0   375    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0<?xml version="1.0"
encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:pos="poseidonService.xsd">
<soapenv:Header/>
<soapenv:Body>
<pos:QueueAddResponse>
<Rslt>1</Rslt>
<ID>169</ID>
</pos:QueueAddResponse>
</soapenv:Body>
</soapenv:Envelope>
100   667  100   292  100   375    292    375  0:00:01  0:00:01 --:--:--   657

C:\Users\volmr\Desktop\curl>
```

Uploading a XML file using PHP

PHP is one of the most popular open source web scripting language. Sending a SMS from your own website is very simple; an example script follows:

```
<?php

$type = "SMS";
$nmr = "+420111222333";
$text =  "Hello world!";

$host = "mysmsgw.domain.com";
$port = 80;
$user = "username";
$pass = "password";

$xml = "<soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\">
        <soapenv:Header/>
        <soapenv:Body>
                <pos:QueueAdd>
                        <Queue>GsmOut</Queue>
                        <Gsm>
                                <Cmd>".$type."</Cmd>
                                <Nmr>".$nmr."</Nmr>
                                <Text>".$text."</Text>
                        </Gsm>
                </pos:QueueAdd>
```

```
        </soapenv:Body>
</soapenv:Envelope>";

$fp = @fsockopen($host, $port, $errno, $errstr, 30);
if (!$fp) {
    echo "$host:$port: $errstr ($errno)\r\n";
}
else {
    fwrite($fp, "POST /service.xml HTTP/1.0\r\n");
    fwrite($fp, "User-Agent: MyPHPTest\r\n");
    fwrite($fp, sprintf("Host: %s\r\n", $host));
    fwrite($fp, sprintf("Authorization: Basic %s\r\n", base64_encode($user . ":" . $pass)));
    fwrite($fp, sprintf("Content-Length: %d\r\n\r\n", strlen($xml)));

    fwrite($fp, $xml);
    while (!feof($fp)) {
        echo fgets($fp, 128);
    }
    fclose($fp);
}

?>
```

Simply fill out the address, username and password for your HWg-SMS-GW3 and everything works right away. The $type, $nmr and $text variables can be of course passed from other scripts.

Sending a SMS from Nagios

Nagios uses scripts written in Perl. The Netways company from Germany has provided the notify-poseidon-sms.pl script that can send up to 160 characters with this command:

```
./notify-poseidon-sms.pl -H 192.0.2.10 -M "Test message" -D 123456789
```

where:
-H – (Host) IP address of HWg-SMS-GW3
-M – (Message) message text
-D – (Destination) recipient's telephone number

A limitation is that the script can only send SMS and cannot ring the recipient's phone number; however, this can be easily changed, see e.g. notify-poseidon-call.pl (see the attachments). Also, it does not support http authentication.

Example of use in Windows:

```
C:\Users\volmr\Desktop\curl>perl notify-poseidon-sms.pl -H 192.168.100.169 -M "Hello World!" -D
777232759
OK, message sent with ID 1 to '777232759'

C:\Users\volmr\Desktop\curl>
```

Sending text messages using HTTP GET

HTTP GET is a method where all parameters are passed directly in the URL. This method is often used for simple applications because it allows for simple debugging and is easy to code. One simply invokes the address with the necessary parameters:

```
values.xml?Cmd=SMS&Nmr=00420123456789&Text=Hello World!
```

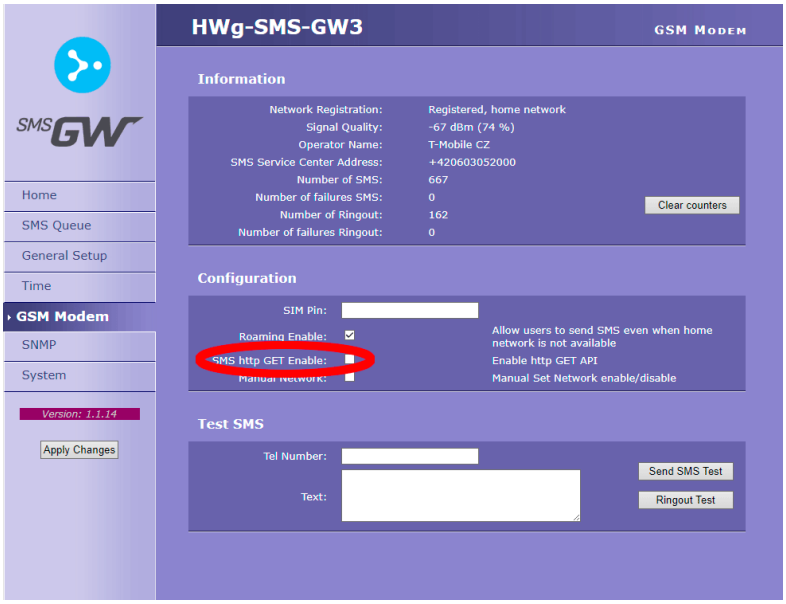
where:
Values.xml – file in the GW that the command should process. It is either values.xml or service.xml
Cmd – command that instructs what the GW should do: SMS – send a text message, CALL –

ring the number
Nmr – phone number in national or international format. It is recommended to use 00 instead of the "+" sign so that there is no need to replace the "+" with an entity
Text – message text encoded in UTF-8

A disadvantage is that the HTTP GET method, natively used to display WWW pages, is difficult to secure against misuse (one would need to set WWW Security, which then prevents unrestricted browsing of the HWg-SMS-GW3 web interface). And even then, the login credentials are sent in the URL in the open:

```
http://user:pass@192.168.1.1/values.xml?Cmd=SMS&Nmr=777232759&Text=Hello 20World!
```

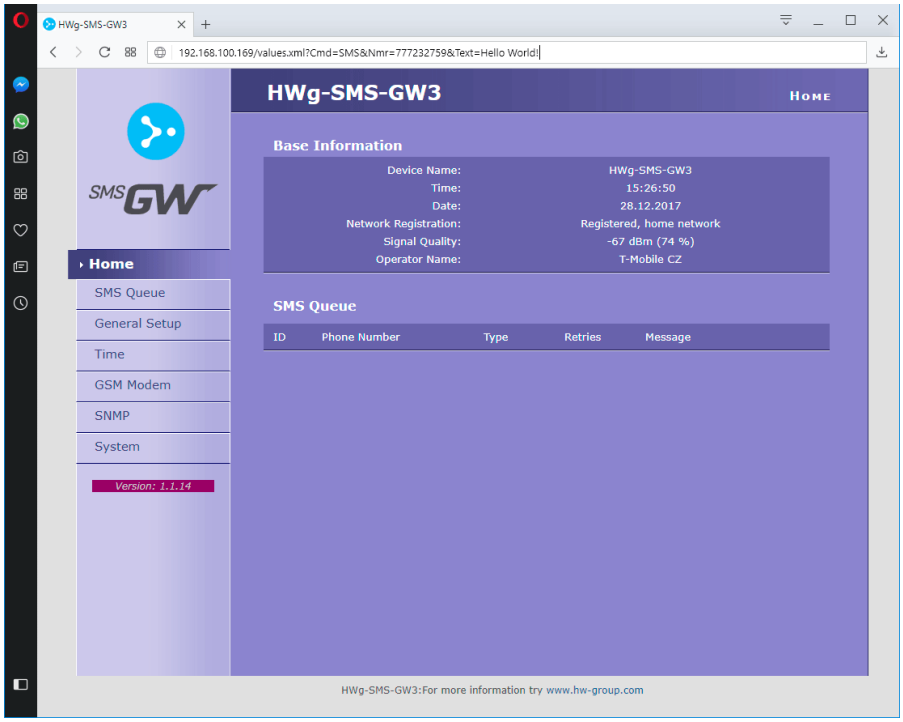
For these reasons, the sending of messages via HTTP GET is disabled by default, and must be manually enabled at the GSM Modem tab if it is needed:



Sending a SMS from a web browser

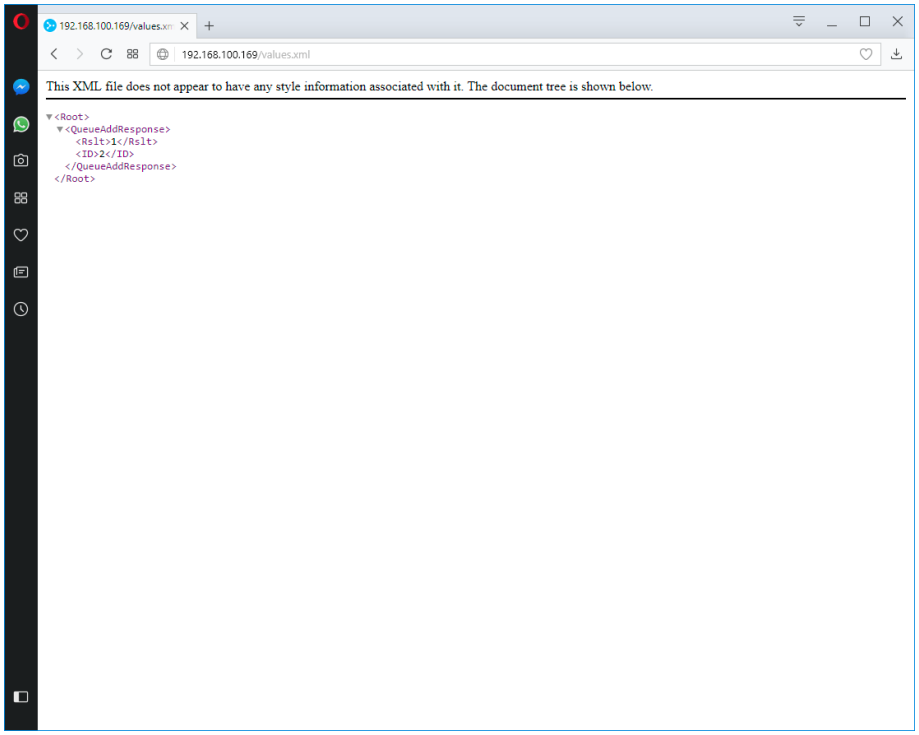
Enter the URL of HWg-SMS-GW3 including the requested commands into the address field of the web browser:

```
http://192.168.100.169/values.xml?Cmd=SMS&Nmr=777232759&Text=Hello World!
```



After sending, a XML confirmation appears:

```
<Root>
  <QueueAddResponse>
    <Rslt>1</Rslt>
    <ID>2</ID>
  </QueueAddResponse>
</Root>
```



Sending SMS using Wget

GNU Wget is a command-line utility that serves as a simple but powerful http client. It implements file transfer over HTTP, HTTPS and FTP protocols. It is available at https://www.gnu.org/software/wget/ and Windows versions are available at http://gnuwin32.sourceforge.net/packages/wget.htm

Wget is very simple to use – simply enter the same command that would be entered in the browser address field. However, the entire URL must be enclosed in double quotes:

```
wget "http://192.168.100.169/values.xml?Cmd=SMS&Nmr=777232759&Text=Hello World!"
```

The result looks like this:

```
C:\Users\volmr\Desktop\wget>wget "http://192.168.100.169/values.xml?Cmd=SMS&Nmr=777232759&Text=Hello
World!"
--2017-12-28 15:56:35--  http://192.168.100.169/values.xml?Cmd=SMS&Nmr=777232759&Text=Hello%20World!
Connecting to 192.168.100.169:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: unspecified [text/xml]
Saving to: 'values.xml@Cmd=SMS&Nmr=777232759&Text=Hello%20World!.1'

values.xml@Cmd=SMS&Nmr=777232     [ <=>                                                ]     118  --.-
KB/s    in 0s

2017-12-28 15:56:35 (5.48 MB/s) - 'values.xml@Cmd=SMS&Nmr=777232759&Text=Hello%20World!.1' saved
[118]


C:\Users\volmr\Desktop\wget>
```

Sending SMS using cURL

The cURL utility is used in the same way as the wget utility. This is the command:

```
curl "http://192.168.100.169/values.xml?Cmd=SMS&Nmr=777232759&Text=Hello World!"
```

And this is the application's response:

```
C:\Users\volmr\Desktop\curl>curl "http://192.168.100.169/values.xml?Cmd=SMS&Nmr=777232759&Text=Hello
World!"
<?xml version="1.0" encoding="utf-8"?>
<Root>
<QueueAddResponse>
<Rslt>1</Rslt>
<ID>4</ID>
</QueueAddResponse>
</Root>
C:\Users\volmr\Desktop\curl>
```

Of course, the main result is that the message is sent.

Sending a SMS from PRTG

PRTG Network Monitor is a complex network monitoring solution that combines Paessler's expertise with a complete set of monitoring functions, intuitive and easy-to-use UI and the most advanced monitoring tool suitable for all types of networks.

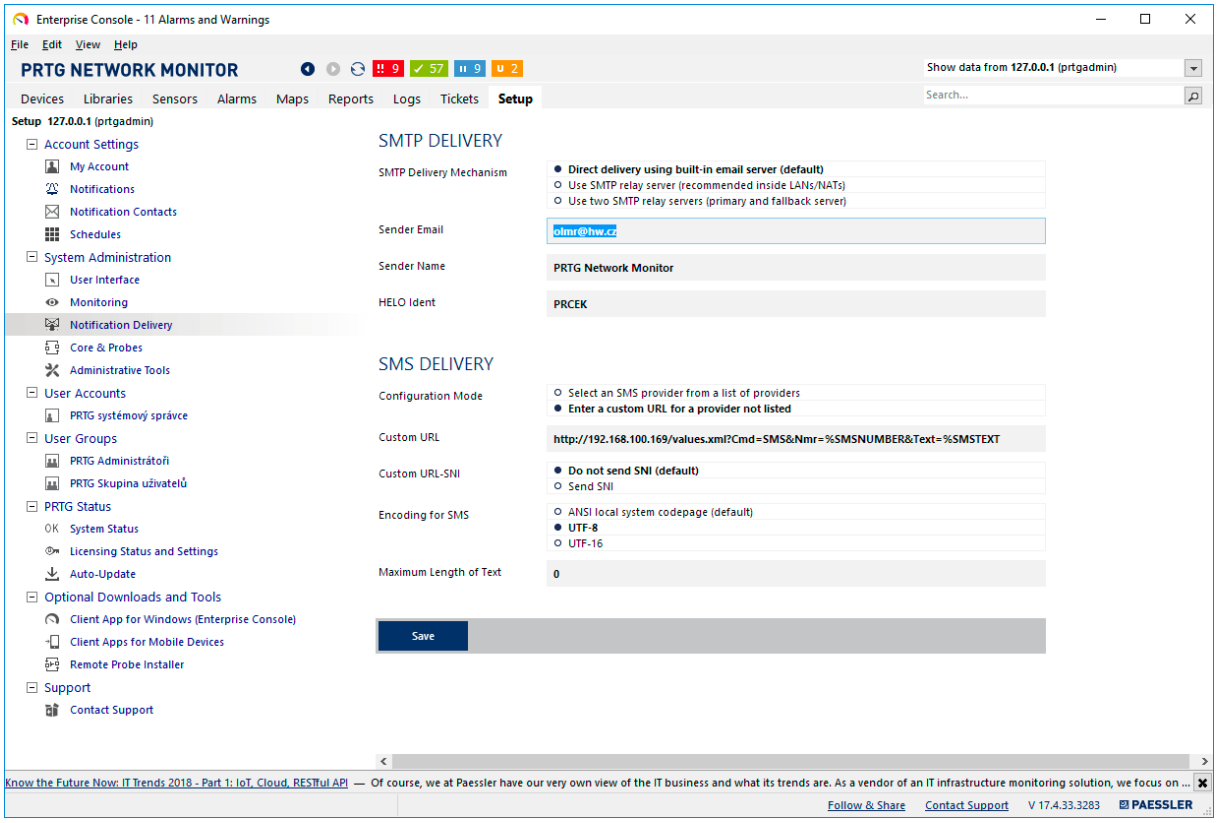
PRTG Network Monitor can send text messages using the HTTP GET method. The setting up only involves specifying a URL for sending the messages. This is done with the Enterprise Console (application or WWW interface) in Setup > System Administration > Notification Delivery. The Custom URL field should contain:

```
http://IP_address/values.xml?Cmd=SMS&Nmr=%SMSNUMBER&Text=%SMSTEXT
```

This ensures that the SMSNUMBER and SMSTEXT system variables are inserted at the correct places in the URL.

And if the HWg-SMS-GW3 is protected with a username and a password, the credentials are specified as in the previous cases:

```
http://user:pass@IP_adresa/values.xml?Cmd=SMS&Nmr=%SMSNUMBER&Text=%SMSTEXT
```



Sending a SMS using the SNMP Write function (snmpset)

Starting from firmware version 1.1.3, HWg-SMS-GW3 supports the sending of SMS and dialing of numbers using SNMP. SNMP write, or more exactly SNMP SET function, is used for that purpose. However, it is necessary to write all the 3 required OIDs for the recipient's number, message text and function (SMS/ring/SMS+ring). The OIDs are the following:

- *msgQueueNmr (OID .1.3.6.1.4.1.21796.4.10.2.2.0)* - (mandatory) Recipient's phone number in national or international format. The international format can use either "+" or "00".
- *msgQueueCmd (OID .1.3.6.1.4.1.21796.4.10.2.3.0)* - (mandatory) Action: 1=sms, 2=ring, 3=sms+ring
- *msgQueueText (OID .1.3.6.1.4.1.21796.4.10.2.1.0)* - (mandatory) Message text. Can be omitted if the requested action is ring only.

In a concrete example, the individual commands could look like this:

```
snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.2.0 i: 777232759
snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.1.0 s "Hello World!"
snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.3.0 i: 1
```

When the commands are entered in the command line, the following appears:

```
C:\usr\bin>snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.2.0 i: 777232759
SNMPv2-SMI::enterprises.21796.4.10.2.2.0 = INTEGER: 777232759

C:\usr\bin>snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.1.0 s "Hello World!"
SNMPv2-SMI::enterprises.21796.4.10.2.1.0 = STRING: "Hello World!"

C:\usr\bin>snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.3.0 i: 1
SNMPv2-SMI::enterprises.21796.4.10.2.3.0 = INTEGER: 1

C:\usr\bin>
```

The command to ring the number then looks like this:

```
snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.2.0 i: 777232759
snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.3.0 i: 2
```

and the response:

```
C:\usr\bin>snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.2.0 i: 777232759
SNMPv2-SMI::enterprises.21796.4.10.2.2.0 = INTEGER: 777232759

C:\usr\bin>snmpset -v 1 -c private 192.168.100.169 1.3.6.1.4.1.21796.4.10.2.3.0 i: 2
SNMPv2-SMI::enterprises.21796.4.10.2.3.0 = INTEGER: 2

C:\usr\bin>
```