SMS Pro API

Interface specification

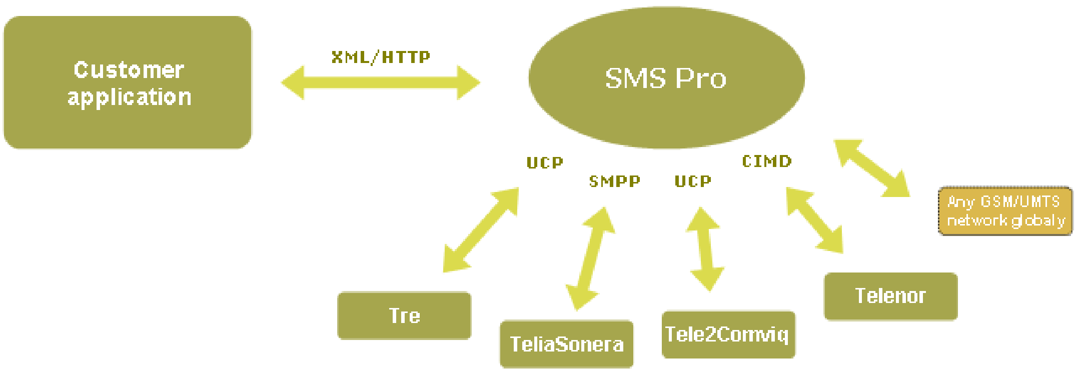
Version 3.3.0

Contents:

1 Introduction

This document describes how to communicate with the SMS Pro API-platform in order to send and receive SMS-messages and perform billing of endusers. The platform consists of two individual server functions, SMS Pro and Billing Pro as described below.

The platform is interconnected with all Swedish mobile operators including their service providers, hence an agreement with Telenor Sverige AB will facilitate billing and SMS-functionality to and from all Swedish mobile phone users regardless of operator and type of subscription.



SMS Pro also provides connectivity to almost any mobile network globally, in the case where the subscriber shall not be charged a premium fee.

In addition, a variety of functions are available, such as, sending binary messages, replacing originating number with another number or a text, or to enclose the name and address of the subscriber who sent the message.

2 Protocols for interconnection, authentication and security

The interconnection between the customer application and the Pro platform is normally made with XML over the Internet. In addition, a limited set of parameters are supported with HTTP GET. For further information see section 6.6.

A request is always authenticated through username and password, which are supplied in the appendix to the agreement.

Additional credentials are forwarded through *HTTP Basic Authentication* as described in Appendix 1.

If implementation of *HTTP Basic Authentication* for some reason is not possible, then authentication can be made through a fixed IP-address, which is supplied by the customer in the appendix to the agreement.

If authentication is made through a fixed IP-address, then HTTP or HTTPS is supported in both directions.

3 Content format

The content is coded into standard XML-templates and posted from the customer application to the Pro platform with XML over HTTP POST, or as mentioned above HTTPS POST.

4 XML References

The XML-syntax conforms to current specification. For reference please refer to the W3C specifications at http://www.w3.org/TR/REC-xml.

5 Abbreviations

API	Application Programming Interface
AO	Application Originated
AT	Application Terminated
CA	Content Aggregator
CIMD	Computer Interface to Message Distribution
CP	Content Provider
HTTP	Hypertext Transfer Protocol
MO	Mobile Originated
MSISDN	Mobile Station Integrated Service Digital Network
MT	Mobile Terminated
MTB	Mobile Terminated Billing
SMPP	Short Message Peer-to-Peer
SMS	Short Message Service
SMSC	Short Message Service Center
SSL	Secure Socket Layer
UCP	Universal Computer Protocol
URL	Uniform Resource Locator
XML	Extensive Mark-up Language

6 Mobile Terminated (MT) SMS

The following examples shows what SMS Pro expects from the customer application to result in a Mobile Terminated (MT) SMS. These examples show MT SMS without premium charge. Premium charge may be added according the Mobile Terminated Billing (MTB) section 7 or Billing Pro section 10 below.

Several different types of messages and combinations are available. Section 6.1, 6.2 and 6.3 show XML which results in normal text-messages while 6.4 and 6.5 shows two examples of binary messages.

The first example in section 6.1 has most of the available parameters exemplified, while the following sections only shows the basic set with the addition of the specific function in question.

For a more detailed description of the elements and attributes, please refer to table 1 below.

Note that there are three parameters to be used at the customer's convenience, SubID1, SubID2 and SubID3. These parameters are stored in SMS Pro database for post-processing of the customer's traffic data.

Note that SMS Pro will respond in the same HTTP-session according to section 6.7.

The URL to which the customer sends the request is,
http://sms-pro.net/services/**customername**/sendsms, where "customername" is replaced by the customer's CustomerID, which is to be found in the appendix to the agreement.

It is recommended to send only to one MSISDN in one XML-request. However if the same message shall be delivered to more than one receiver, it may be done as a batch according to below. Note that delivery status can not be used for the individual receivers and that one small error in for example a mobile telephone number will discard the whole XML-request. It is recommended to limit the number of recipients in one batch to about 100.

6.1 MT text-SMS

This example will result in a normal text-message to two subscribers.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <password>xxxxxxxx</password>
        <sub_id_1>Department</sub_id_1>
        <sub_id_2>Group</sub_id_2>
        <sub_id_3>System</sub_id_3>
        <request_id>ABC123</request_id>
        <start_delivery>201411011015</start_delivery>
        <valid_until>201411011315</valid_until>
        <status_delivery_url>http://domain.net/receipt</status_delivery_url>
    </header>
    <payload>
        <sms account="71700">
            <message><![CDATA[Text message to mobile]]></message>
            <to_msisdn>+46708651058</to_msisdn>
            <to_msisdn>+46734252604</to_msisdn>
        </sms>
    </payload>
</mobilectrl_sms>
```

6.2 MT text-SMS with alphanumeric originating address

Normally the account number is shown to the subscriber as the originating number, like 71700. However sometimes it may be beneficial to replace the number with a text-string, if for example the subscriber is not supposed to reply to the message.

This example will result in a normal text-message to two subscribers. The originating address displayed to the subscribers will be MARKETPLACE. Note that the text is case sensitive.

A maximum of 11 characters may be used. These may be ASCII 32-126, except 36, 64, 93, 95, 96 and 125. In other words a-z, A-Z, 0-9 and a few punctuation marks.

This is an optional feature which must be ordered from and enabled by Telenor.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <password>xxxxxxxx</password>
        <from_alphanumeric>MARKETPLACE</from_alphanumeric>
    </header>
    <payload>
        <sms account="71700">
            <message><![CDATA[Text message to mobile]]></message>
            <to_msisdn>+46708651058</to_msisdn>
        </sms>
    </payload>
</mobilectrl_sms>
```

6.3 MT text-SMS with numeric originating address

This example will result in a normal text-message to two subscribers. The originating address displayed in the mobile phone will be the mobile number (MSISDN) exemplified +46734252604. If the subscriber chooses to reply to the message it will be forwarded to +46734252604.

It is recommended to use the proper MSISDN format, for example +46734252604 rather than the domestic format 0734252604. Some mobile phones and operators will otherwise cause problems.

The MSISDN in this field is not cleaned in any way by SMS Pro. The field may only contain a decimal value with an optional leading "+". The exact format will be displayed to the receiver of the message.

This is an optional feature which must be ordered from and enabled by Telenor.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <password>xxxxxxxx</password>
        <from_msisdn>+46734252604</from_msisdn>
    </header>
    <payload>
        <sms account="71700">
            <message><![CDATA[Text message to mobile]]></message>
            <to_msisdn>+46708651058</to_msisdn>
        </sms>
    </payload>
</mobilectrl_sms>
```

Note that if both numeric and alphanumeric originator is supplied, only the alphanumeric originator will be forwarded to the mobile phone.

6.4 MT special-SMS

In addition to plain text SMS messages, SMS Pro allows the sending of more complex types of SMS, e.g. ring tones. SMS Pro will split the message in as many SMS messages as required and handle the creation of corresponding header information.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <password>xxxxxxxx</password>
    </header>
    <payload>
        <sms account="71700" destination_port="5505" binary="true">
            <message>024A3A5505E195B...D1D1C000</message>
            <to_msisdn>+46733946030</to_msisdn>
        </sms>
    </payload>
</mobilectrl_sms>
```

6.5 MT complex-SMS

For even more fine-grained control over the sent content, raw_sms can be used. When using raw_sms, you have full control of every aspect of the final SMS message. You supply the header, so if the content is longer than what can be sent in one SMS, you must send it as several and supply the correct header for each one.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <password>xxxxxxxx</password>
    </header>
    <payload>
        <raw_sms account="71700" binary="true" msg_class="1" pid="0">
            <header>06050415811581</header>
            <message>024A3A5505E195B...D1D1C000</message>
            <to_msisdn>+46733946030</to_msisdn>
        </raw_sms>
    </payload>
</mobilectrl_sms>
```

6.6 MT text-SMS using HTTP GET

In some cases it might be beneficial to use HTTP GET instead of POST. For example when testing the connection to SMS Pro with a ordinary browser. This simple method only allows sending ordinary text-messages with none of the above mentioned features.

Use the following URL to send a HTTP GET message,

http://sms-pro.net/services/**customer**/sendmsg?msisdn=**46734252604**&message=**Text message to mobile**&password=**customer-pwd**&sub_id_1=**Sub-ID1**&sub_id_2=**Sub-ID2**

All supported parameters are shown in the example above.

Replace the parameters in bold with your own data.

Please see http://www.w3.org/Addressing/URL/4_Recommentations.html for help with the format of the message parameter.

The message can be sent to several MSISDN, separate by semicolon.

The response from SMS Pro is the same as for a normal HTTP POST request, according to section 6.7.

Table 1, Elements and attributes for all types of MT SMS.

Element	**Attribute**	**Required**	**Values (* = default)**	**Comment**
customer_id		Yes	ASCII, max 30 chars	Customer identification.
password		Yes	ASCII	Customer verification.
sub_id_1		No	ASCII, max 20 chars	Supplementary information
sub_id_2		No	ASCII, max 20 chars	Supplementary information
sub_id_3		No	ASCII, max 20 chars	Supplementary information
request_id		No	ASCII, max 30 chars	Customer defined ID of outgoing message.
from_msisdn		No	Decimal (can have a leading '+'), max 20 chars.	Will be seen as an "originating address" by the receiver. Can only be used by enabled customers with enabled accounts.
from_alphanumeric		No	ASCII, max 11 chars. Valid chars are 32-126 inclusive, except for 36, 64, 93, 95, 96 and 125. That is, included are a few punctuation marks, space, a-z, A-Z and 0-9.	Will be seen as an "alphanumeric originating address" by the receiver. Can only be used by enabled customers with enabled accounts.
start_delivery		No	Date and time formatted as YYYYMMDDhhmm	The time when the message should be delivered.
valid_until		No	Date and time formatted as YYYYMMDDhhmm	The time when SMS Pro should stop trying to deliver the message.
status_delivery_url		No	URL	The URL to where a status report shall be delivered. When not supplied the status report is delivered to the default URL.
sms	try_only_once	No	false* or true	If true, only one attempt to deliver the message will be made. Overrides valid_until.
sms raw_sms	account	No	71700 etc.	Identifies the account through which the message should be sent.
sms raw_sms	binary	No	false* or true	Set to true if content of message is 8-bit. Default (false) indicates 7-bit.
sms raw_sms	msg_class	No	0, 1*, 2 or 3	Message class, as defined in GSM03.38.
sms raw_sms	pid	No	0*	Protocol identifier, as defined in GSM03.40.
sms	destination_port	No	Decimal. E.g. 5505 (ring tone), 5506 (operator logo)	Destination port.
sms	source_port	No	Decimal. 0* or source port	Source port. If this attribute is used, destination_port must be used as well.
message		Yes	160 characters if 7-bit, 140 if 8-bit (binary)	Content to send. Exceeding the limit will divide the message into several SMS with 153 characters in each.
to_msisdn		Yes	+46 and normally 9 (or 13) digits in Sweden. Maximum 20 digits.	Mobile phone number with country prefix. Foreign prefixes allowed depending on agreement.
header		No		Header, as defined in GSM03.40. Always 8-bit. Only allowed in raw_sms.

6.7 SMS Pro response to MT SMS request

When the customer has sent a Mobile Terminated (MT) as described above in this section, SMS Pro will respond in the same HTTP-session to acknowledge that the request has been received and that the request was interpreted correctly. Note that this does not mean that the message has been sent to the subscriber. Please see section 0 and 6.9 for status requests.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_response>
    <customer_id>CUSTOMER</customer_id>
    <request_id> ABC123</request_id>
    <mobilectrl_id>5aa434:eac0a56a0b:-7ffe</mobilectrl_id>
    <status>0</status>
    <code>0</code>
    <errorcode>0</errorcode>
    <message>The MT SMS request is being processed.</message>
    <datetime>2014-11-01 09:58:57</datetime>
</mobilectrl_response>
```

Table 2, Elements and attributes for SMS Pro response to MT SMS request.

Element	Attribute	Required	Values (* = default)	Comment
customer_id		Yes	ASCII, max 30 chars	Customer identification.
request_id		No	ASCII, max 30 chars	Customer defined ID of outgoing message.
mobilectrl_id		Yes		Transaction identification.
status		Yes	= 0 (ok) not = 0 (not ok)	Acknowledge that the XML was received ok (see below).
code		No		SMS Pro internal values. For reference only.
errorcode		No		SMS Pro internal values. For reference only.
message		Yes	ASCII	Readable information explaining the status.
datetime		Yes	ASCII	The time when SMS Pro received the request.

6.8 Delivery status - Requested

The customer may have interest in finding out if a message has been delivered to the mobile phone. For this purpose SMS Pro provides an interface where the customer may ask for the status of a message.

Note that we can never tell if the message has been opened or read by the recipient, only if it has been received and acknowledged by the mobile phone.

The following XML shall be sent to http://sms-pro.net/services/**customer**/status. “Customer” in the URL must be replaced with the customer’s CustomerID.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_delivery_status_request>
    <customer_id>CUSTOMER</customer_id>
    <status_for type="mobilectrl_id">12f95de:11817630961:77ec</status_for>
</mobilectrl_delivery_status_request>
```

In case of a successfully delivered message, SMS Pro will respond with the following XML in the same session,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_delivery_status>
    <mobilectrl_id>12f95de:11817630961:77ec</mobilectrl_id>
    <status>0</status>
    <delivery_status since="20140217144142">0</delivery_status>
    <message>SMS SENT</message>
</mobilectrl_delivery_status>
```

In case of an unsuccessfully delivered message, SMS Pro will respond with the following XML in the same session,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_delivery_status>
    <mobilectrl_id>12f95de:11817630961:77ec</mobilectrl_id>
    <status>0</status>
    <delivery_status since="20140216110118">-2</delivery_status>
<message>SMS FAILED</message>
</mobilectrl_delivery_status>
```

Please see table 3 for more details in the case where SMS Pro was unable to deliver the message, unable to retrieve the status of the message or unable to parse the XML from the customer.

6.9 Delivery status - Automatic

As described in section 6.8, the customer may request the delivery status of a message. However the delivery status can also be sent automatically to the customer.

Note that we can never tell if a message has been opened or read by the recipient, only if it has been received and acknowledged by the mobile phone.

Automatic delivery status of a message is only sent to the customer if the customer has announced a specific URL for the purpose in the agreement.

However if the customer has a multiple server system where the status reports are requested to different servers, then the preferred URL may be forwarded in the MT SMS request as described in section 6.1-6.5.

In case of a successfully delivered message, SMS Pro will send the following XML to the customer,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_delivery_status>
    <mobilectrl_id>12f95de:11817630961:77ec</mobilectrl_id>
    <status>0</status>
    <delivery_status since="20140217144142">0</delivery_status>
    <message>SMS SENT</message>
</mobilectrl_delivery_status>
```

In case of an unsuccessfully delivered message, SMS Pro will send the following XML to the customer,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_delivery_status>
    <mobilectrl_id>12f95de:11817630961:77ec</mobilectrl_id>
    <status>0</status>
    <delivery_status since="20140217144142">-2</delivery_status>
    <message>SMS FAILED</message>
</mobilectrl_delivery_status>
```

The customer is requested to acknowledge the XML with a HTTP-200. Any additional information transferred will be disregarded.

Table 3, Message, Status and Delivery status for Requested and Automatic Delivery status.

Message	**Delivery status - Requested**	**Delivery status - Automatic**
Message = SMS SENT	Status = 0 Delivery status = 0	Status = 0 Delivery status = 0
Message = SMS PARSED	Status = 0 Delivery status = 2	n/a
Message = SMS IN PROCESS	Status = 0 Delivery status = 1	n/a
Message = SMS FAILED	Status = 0 Delivery status = -2	Status = 0 Delivery status = -2
Message = SMSC FAILED	Status = 0 Delivery status = -2	Status = 0 Delivery status = -2
Message = No SMS found matching requested value	Status = 0 Delivery status = -1	n/a
Message = “Description of error in XML request”	Status = -2 Delivery status = n/a	n/a
Message = SMS STATUS UNKNOWN	Status = 0 Delivery status = -2	Status = 0 Delivery status = -2
Message = Internal error	Status = 0 Delivery status = -3	Status = 0 Delivery status = -3

7 Mobile Terminated Billing (MTB) SMS

Mobile Terminated Billing (MTB) SMS provides the customer with the possibility to charge the subscriber at the same time as an SMS is sent out.

MTB SMS is activated by setting the attribute **account** to **72700**. All MTB SMS are to be sent through the same MTB-account. The customer defines the tariff in the attribute **premiumrate**.

When SMS Pro receives a MTB SMS request, the subscriber will first be charged the requested amount. If the billing is successful the message will be sent, but if the billing fails, the SMS will not be sent.

7.1 MTB SMS request

Below is an example of a MTB SMS request. In addition almost all elements and attributes of MT SMS may be used (as described in section 6). Restrictions are made to manipulation of the originator (in this case 72700) due to that traceability of premium billed messages otherwise decreases.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <password>xxxxxxxx</password>
    </header>
    <payload>
        <sms account="72700" premiumrate="5">
            <message>Text which will cost the receiver 5 kr.</message>
            <to_msisdn>+46733946030</to_msisdn>
            <description>SMS-ticket</description>
        </sms>
    </payload>
</mobilectrl_sms>
```

Table 4, Elements and attributes for MTB SMS – Additions to Table 1.

Element	Attribute	Required	Values (* = default)	Comment
sms	account	Yes	72700	72700 is the account to use for MTB SMS. All other accounts are used for MT SMS.
sms	premiumrate	Yes	1-200	Attribute premiumrate is required for MTB SMS. Granularity is 1 öre. Separator must be ".". For example 19.95. Unit is SEK.
to_msisdn		Yes	+46 and normally 9 (or 13) digits in Sweden. Maximum 20 digits.	There must be only one to_msisdn element when sending MTB SMS.
description		No	ASCII, max 40 chars Betaltjänst *	Description of the purchase. Will be printed on the end-user's invoice if supported by the end-user's operator.

7.2 SMS Pro response to MTB SMS request

When the customer has sent a MTB SMS request according to 7.1 above, SMS Pro will respond in the same HTTP-session to acknowledge that the request has been received and that it was interpreted correctly. Note that this does not confirm that the requested charge has been applied or that the message has been sent to the subscriber.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_response>
    <customer_id>CUSTOMER</customer_id>
    <request_id> ABC123</request_id>
    <mobilectrl_id>5aa434:eac0a56a0b:-7ffe</mobilectrl_id>
    <status>0</status>
    <code>0</code>
    <errorcode>0</errorcode>
    <message>The MT SMS request is being processed.</message>
    <datetime>2014-11-01 09:58:57</datetime>
</mobilectrl_response>
```

Table 5, Elements and attributes for SMS Pro response to MTB SMS request.

Element	**Attribute**	**Required**	**Values (* = default)**	**Comment**
customer_id		Yes	ASCII, max 30 chars	Customer identification.
Request_id		No	ASCII, max 30 chars	Customer defined ID of outgoing message.
mobilectrl_id		Yes		Transaction identification.
status		Yes	= 0 (ok) not = 0 (not ok)	Acknowledge that the XML was received ok (see below).
code		No		SMS Pro internal values. For reference only.
errorcode		No		SMS Pro internal values. For reference only.
Message		Yes	ASCII	Readable information explaining the status.
Datetime		Yes	ASCII	The time when SMS Pro received the request.

7.3 Result of MTB request

SMS Pro sends the result of the billing operation to the customer only if the customer has announced a specific URL for the purpose in the agreement.

Example of a successful billing operation,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<content_billing_result>
    <customer_id>CUSTOMER</customer_id>
    <mobilectrl_id>5aa434:eac0a56a0b:-7ffe</mobilectrl_id>
    <status pending=”true”>0</status>
    <statuscode>0</statuscode>
    <message operator="tre">Purchase OK</message>
    <datetime>2014-10-29 16:58:51</datetime>
</content_billing_result>
```

Example of an unsuccessful billing operation,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<content_billing_result>
    <customer_id>CUSTOMER</customer_id>
    <mobilectrl_id>84:175ae09:12a9rt6cd82:11b2</mobilectrl_id>
    <status pending="false">-2</status>
    <statuscode>103</statuscode>
    <message operator="telenor"> The end-user account balance is low.</message>
    <datetime>2014-08-24 09:16:00</datetime>
</content_billing_result>
```

The customer is requested to acknowledge this XML with a HTTP-200. Any additional information transferred will be disregarded.

Table 6, Elements and attributes for result of MTB SMS.

Element	**Attribute**	**Required**	**Values (* = default)**	**Comment**
customer_id		Yes	ASCII, max 30 chars	Customer identification.
mobilectrl_id		Yes		Transaction identification, same as received as reply to the original MTB SMS request.
status		Yes	0 indicates success, other values failure.	Status of the billing.
status	pending	Yes	True or false	Indicates if the billing request is pending or not.
statuscode		Yes	= 0 (ok, but only if status also is 0)not = 0 (see table 7)	Gives a code which indicates the reason for that the billing failed.
message		Yes	ASCII	Readable information explaining the status.
message	operator	Yes		The operator involved in the billing. Not always presented.
datetime		Yes	ASCII	The time when SMS Pro received the request.

Table 7, Description of statuscodes in case of a negative MTB response.

Statuscode	**Description**
0	Purchase OK (but only if status also is 0).
101	Purchase not performed. Reason undefined.
102	The end-user does not exist.
103	The end-user account balance is low.
104	The end-user is not allowed to purchase premium services.
105	Invalid amount.
106	Invalid VAT.
107	Invalid currency.

8 Mobile Originated (MO) SMS

The following sections shows the XML sent to the customer as a result of a MO SMS.

Section 8.4 describes expected response from the customer. If this response is omitted or faulty SMS Pro will retry to send the message a number of times.

Normally MO SMS is delivered as readable text, according to GSM7/8 character set. However the use of smileys in MO SMS has increased. Smileys based on normal characters will be forwarded as normal characters, but smileys based on emojis are by default deleted from the message so that all readable characters can be forwarded as readable text. For further options please see section 8.3.

8.1 MO SMS

The following example shows what SMS Pro sends to the customer application as a result of a Mobile Originated (MO) SMS.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_received_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <mobilectrl_id>5aa434:eac0a56a0b:-845c</mobilectrl_id>
        <datetime>2014-10-25 13:43:19</datetime>
    </header>
    <payload>
        <sms account="71700" premiumrate="0">
            <message>Text message from mobile</message>
            <from_msisdn operator="telenor">+46708651052</from_msisdn>
        </sms>
    </payload>
</mobilectrl_received_sms>
```

8.2 MO SMS with dictionary details

The following example shows what SMS Pro sends to the customer application as a result of a Mobile Originated (MO) SMS with activated dictionary details. This is an optional feature which must be ordered from and enabled by Telenor.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_received_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <mobilectrl_id>5aa434:eac0a56a0b:-845c</mobilectrl_id>
        <datetime>2014-10-25 13:43:19</datetime>
    </header>
    <payload>
        <sms account="71700" premiumrate="0">
            <message>Text message from mobile</message>
            <from_msisdn operator="telenor">+46708111222</from_msisdn>
        </sms>
        <extrainfo>
            <firstname>Kalle</firstname>
            <lastname>Anka</lastname>
            <address>Kvackvägen 8</address>
            <postalcode>12345</postalcode>
            <city>ANKEBORG</city>
            <personinfostatus>0</personinfostatus>
        </extrainfo>
    </payload>
</mobilectrl_received_sms>
```

8.3 MO SMS with emojis and other binary messages

The use of smileys has increased in the market. Previously smileys have been made up from normal characters.

Example :-)

Messages with these smileys have just been forwarded as normal characters. However smileys based on emojis must be forwarded in hexadecimal notation to be properly displayed.

Example 👍

Emojis have been defined for several different operating systems and platforms. It is therefore difficult to support each and every emoji character. SMS Pro supports emojis in MO SMS based on Twitters character set, the so called Twemoji.

When an SMS contains emojis mixed with normal text it will be received from the mobile operator in hexadecimal notation, like in option 3 below. However if the customer expects normal text, the message will make no sense without extra conversion in the customer application. Therefore SMS Pro will by default erase all emojis from all messages and forward the plain text.

SMS Pro supports three ways of encoding messages with emojis.

The following MO SMS is sent by a mobile phone: Hello 👍

The message can be forwarded in accordance with any of the following options,

1. <message>Hello</message>
2. <message>Hello &#x1F44D;</message>
3. <message>48656C6C6F20D83DDC4D</message>

Option 1, where the emoji is deleted, is default. The second and third option needs to be ordered from and enabled by Telenor.

Note for option 1,

If a message only contains emojis, those will be deleted and the message will be empty. The message tag will then have the following content.

<message>[MESSAGE WITH NO READABLE CHARACTERS]</message>

Note for option 2,

Incoming hex-code is converted byte by byte to match the corresponding Unicode character.

Note for option 3,

If a message is sent in hexadecimal notation the attribute "binary" will be added and get the value "true". The attribute binary is only added if the message is binary. Otherwise it is not added.

Example,
<sms binary="true">
<message>48656C6C6F20D83DDC4D</message>

8.4 Customer response to MO SMS

SMS Pro requires a delivered message to be acknowledged by the customer application in the same HTTP-session.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_received_sms_response>
    <customer_id>CUSTOMER</customer_id>
    <mobilectrl_id>5aa434:eac0a56a0b:-845c</mobilectrl_id>
    <status>0</status>
</mobilectrl_received_sms_response>
```

Table 8, Elements and attributes for MO SMS.

Element	Attribute	Required	Values	Comment
customer_id		Yes	ASCII, max 30 chars	Used for identification.
mobilectrl_id		No	ASCII, max 40 chars	Transaction identification.
datetime		No	ASCII	Date and time when the request was received by SMS Pro.
sms	premiumrate	No	0	Premium charge for old MOB-accounts. Will always be set to Zero (0).
sms	account	No	71700, 72700 etc.	Identifies the account through which the message was received.
sms	destination_address	No	Numeric (can have leading '+'), operator dependant	The actual number (destination address) the message was sent to. Only for enabled customers.
sms	binary	No	true	Value is true if the message is encoded in hexadecimal notation.
message		Yes	ASCII	Message from mobile. Text Convert binary to text
from_msisdn		Yes	+46 and normally 9 digits in Sweden. Maximum 20 digits.	Mobile phone number.
from_msisdn	operator	No	E.g. tele2, telia, telenor, tre	Operator identification.
firstname		No	ASCII	First name
lastname		No	ASCII	Sir name
address		No	ASCII	Street address
postalcode		No	ASCII	Zip code
city		No	ASCII	City
personinfostatus		No	= 0 (ok) = 1 (no data available) = -2 (look-up failed)	Result of retrieval of personal information.
status		Yes	= 0 (ok) not = 0 (not ok)	Acknowledge that the XML was received ok.

9 Destination Number Trigger (DNT)

In some service applications it may be beneficial to avoid using key-words to determine to which customer a specific MO-message shall be sent. One solution is of course to open a customer specific SMS-account and send all MO-messages to the same URL regardless of the contents of the message. However it is more cost efficient if several customers can share the same account. This can be achieved if the destination is determined based on the destination number instead of a key-word in the message.

The solution is to match the destination number to a predefined URL, specific for a certain customer and regardless of the contents of the message. This is done in the function Destination Number Trigger (DNT) in SMS Pro.

The DNT uses an SMS-account with a normal MSISDN (+46 709 032599). At the end of the account number a few digits are added (+46 709 032599 xxxx). These digits will be matched to the list of URLs in SMS Pro and the message will be forwarded to the customer who owns the destination suffix (exemplified as xxxx above).

By using an account with a MSISDN account-number this function has no geographical limit. It can be used to and from any subscription globally.

The DNT is especially useful when a dialogue with the end-user is initiated with a MT-message. This is due to that the account number will be longer than a normal mobile phone number and therefore a bit cumbersome to enter for the end-user. When the dialogue is initiated through a MT-message, the end-user simply replies to the originating number.

One customer normally has one DNT-account and one URL defined, but several DNT-accounts may be defined for one customer.

The XML-format for the MT-message is the same as described in section 6.3 and the MO-message is the same as for a normal MO-message, as described in section 0. However when a customer has more than one suffix to the destination number (exemplified as xxxx above), then the parameter destination_address should be activated for the customer. This is done to enable the customer to differentiate between MO messages to different numbers.

This is an optional feature which must be ordered from and enabled by Telenor.

The following example shows what SMS Pro sends to the customer application as a result of a Mobile Originated (MO) SMS through the DNT.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_received_sms>
    <header>
        <customer_id>CUSTOMER</customer_id>
        <mobilectrl_id>5aa434:eac0a56a0b:-845c</mobilectrl_id>
        <datetime>2014-10-25 13:43:19</datetime>
    </header>
    <payload>
        <sms account=”FSG2MO” premiumrate=”0” destination_address ”467090325991234”>
            <message>Text message from mobile</message>
            <from_msisdn operator=”telenor”>+46708651052</from_msisdn>
        </sms>
    </payload>
</mobilectrl_received_sms>
```

10 Billing Pro

The Billing Pro functionality makes it possible to perform a charge request without sending any SMS. Charging requests are only forwarded to the respective operator’s charging interface. This function facilitates charging also when services are ordered by the end-user through other channels like MMS, WAP or WEB.

There is also a fundamental difference between a *Billing Pro request* and a *MTB SMS request* (as describe in section 7 above). Billing Pro will always keep the HTTP-session open and wait for the response from the operator, before acknowledging the request to the customer. Hence the customer will always be informed if the billing was successful or not prior to the closing of the session.

10.1 Billing Pro request

The following example shows what Billing Pro expects from the customer application to result in a charge request to the operator where the subscription is listed.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<billing_request>
    <customer>
        <customer_id>CUSTOMER</customer_id>
        <password>xxxxxxxx</password>
        <request_id></request_id>
        <sub_id_1></sub_id_1>
        <sub_id_2></sub_id_2>
        <sub_id_3></sub_id_3>
    </customer>
    <bill>
        <type>debit</type>
        <msisdn>+46734252604</msisdn>
        <currency>SEK</currency>
        <amount>1000</amount>
        <vat>2500</vat>
        <content_type>2</content_type>
        <description></description>
    </bill>
</billing_request>
```

Table 9, Elements and attributes for a Billing Pro request.

Element	Attribute	Required	Values (* = default)	Comment
customer_id		Yes	ASCII, max 30 chars	Customer identification.
password		Yes	ASCII	Customer verification.
request_id		No	ASCII, max 30 chars	Customer defined ID of billing request.
sub_id_1		No	ASCII, max 20 chars	Supplementary information
sub_id_2		No	ASCII, max 20 chars	Supplementary information
sub_id_3		No	ASCII, max 20 chars	Supplementary information
type		No	debit* or credit	Defines if the request is a debit or credit. This function will be taken into use when all operators support s the credit function.
msisdn		Yes	+46(and 9 digits)	Mobile phone number with country prefix. Foreign prefixes allowed depending on agreement.
currency		No	SEK *	For future use.
amount		Yes	100-20000	The amount to be charged is defined in öre. The maximum amount to be charged is 200 kr.
vat		No	2500 *	The VAT-rate for the purchase. The unit is 100 times the VAT percent. Only 25% is supported.
content_type		Yes	0-6	Describes the carrier over which the content is transferred. 0=MMS, 1=WAP, 2=SMS, 3=WEB, 4=VOICE, 5=VIDEO, 6=LBS
description		No	ASCII, max 40 chars Betaltjänst *	Description of the purchase. Will be printed on the end-user’s invoice if supported by the end-user’s operator.

10.2 Response to Billing Pro request

Note that the session, through which the request was made, will not be closed until a response is received from the operator and the result of the request is forwarded to the customer.

Example of a successful billing operation,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<billing_result>
    <customer_id>CUSTOMER</customer_id>
    <transaction_id>19c472:11175c2ba94:-8000</transaction_id>
    <operator>tele2</operator>
    <status>0</status>
    <statuscode>0</statuscode>
    <message><![CDATA[Billing performed successfully.]]></message>
    <datetime>2014-01-24 19:30:06</datetime>
</billing_result>
```

Example of an unsuccessful billing operation,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<billing_result>
    <customer_id>CUSTOMER</customer_id>
    <transaction_id>84:175ae09:12a9rt6cd82:11b2</transaction_id>
    <operator>tele2</operator>
    <status>-2</status>
    <statuscode>103</statuscode>
    <message> The end-user account balance is low.</message>
    <datetime>2014-08-24 09:16:00</datetime>
</billing_result>
```

Table 10, Elements and attributes for Response to Billing Pro request.

Element	**Attribute**	**Required**	**Values (* = default)**	**Comment**
customer_id		Yes	ASCII, max 30 chars	Customer identification.
transaction _id		Yes		Transaction identification.
operator		Yes		The operator involved in the billing transaction.
status		Yes	= 0 (ok) not = 0 (not ok)	Acknowledges that the XML was received ok and that the billing was approved by the operator.
statuscode		Yes	= 0 (ok) not = 0 (see table 11)	Gives a code which indicates the reason for that the billing failed.
message		Yes	ASCII	Readable information explaining the statuscode.
datetime		Yes	ASCII	The time when Billing Pro received the request.

Table 11, Description of statuscodes in case of negative Billing Pro response.

Statuscode	**Description**
0	Purchase OK
101	Purchase not performed. Reason undefined.
102	The end-user does not exist.
103	The end-user account balance is low.
104	The end-user is not allowed to purchase premium services.
105	Invalid amount.
106	Invalid VAT.
107	Invalid currency.

11 Keep-alive

It is possible to send a keep-alive request to the Pro platform. The request shall be formed according to the example below and sent to http://sms-pro.net/services/**customer**/alive. "Customer" in the URL must be replaced with the customer's CustomerID.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mobilectrl_sms>
<mobilectrl_alive_request></mobilectrl_alive_request>
```

The Pro platform will respond with the following XML,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobilectrl_alive_check>
<status>0</status>
<message>SMSPro is alive.</message>
</mobilectrl_alive_check>
```

12 Support for STOPP, TEST and INFO

In order to make it easy for a subscriber to terminate any ongoing subscription and to test and obtain information about an account, the customer shall implement the functions STOPP, TEST and INFO, in accordance with the instructions below and also verify that these functions work in a proper way.

12.1 STOPP (and STOP)

When the keyword STOPP (STOP shall also be supportet) is sent (MO→AT) to the customer's account, the customer shall immediately terminate any ongoing subscription service and generate a response (AO→MT) from the service, informing that the service has been terminated.

Note that a STOPP-message will be distributed to all companies having the actual account activated (for example 72700). Hence if the customer does not have an active subscription for the particular subscriber in question, **the customer may at no conditions send any message or error message to the subscriber**.

12.2 TEST

When the keyword TEST is sent (MO→AT) to the customer's account, the customer shall generate a response (AO→MT) from the connected application which received the message. The message shall contain the word "Räksmörgås", in order to test if the swedish characters "åäö" are correct and thereby whether the correct character table is used. **Note that this section is only valid if the customer has its own account.**

12.3 INFO

When the keyword INFO is sent (MO→AT) to the customer's account, the customer shall generate a response (AO→MT) from the connected application which received the message. The message shall contain information regarding the owner of the account. At a minimum, customer name, telephone number and e-mail address (or URL) shall be stated. **Note that this section is only valid if the customer has its own account.**

13 The status tag

The element *status* is used to indicate the outcome of a request. A value of 0 (zero) indicates success in the sense that the XML was well formed and contained values accepted by the Pro Platform. It does not indicate that the message was successfully delivered to the recipient.

Table 12, Values of element status.

Value	Meaning	Comment
1	Error	Indicates error. Only available in result of MTB or Billing Pro requests.
0	Success	The request was successfully accepted.
-2	Error	The request is not correct. No SMS will be sent. This is e.g. the case if the msisdn is invalid.

14 Best practice

14.1 Always check returned status

The Pro Platform will always acknowledge the reception of a request by returning a mobilectrl_response XML. By checking the value of the element *status* you get immediate feedback on the outcome of the initial parsing and validation of the received XML. In case of errors, the element message contains information describing the error. It is <u>not</u> recommended to implement a logic bases on parsing the message string, since Telenor reserves the right to update the content without notice. The application logic should only be based on the value of *status*.

Please note that the above also is applicable for the content_billing_result XML as described in section 7.2.

14.2 Always acknowledge to the Pro Platform

Always acknowledge the reception of a MO SMS. Failing to return a correct response will mark the transaction as failed and will get attention by the surveillance. If the lack of response is due to neglect from the customer, surveillance can be turned off, lowering the chance of proper support.

14.3 Set the content length

Setting the HTTP header field Content-Length gives the receiver a chance to decide when all information in a request has been received. Correctly setting the length of the content in requests and responses ensures optimal transfer through involved servers.

Please refer to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13 for details.

14.4 The use of CDATA to send any character in a message

The XML-specification might not support all characters which the customer wants to send in a text-message. To get around this limitation it is advices to encapsulate all HTTP POST messages with CDATA.

Example,

```
<message><![CDATA[Type your message to the subscriber here.]]></message>
```

14.5 Number of MSISDN in a batch

Sending one message to several recipients in the same XML-request is convenient (as described in section 6). However, if one single MSISDN doesn’t conform to the required format, the whole XML will be considered faulty and hence rejected. Therefore it is advised to restrict the number of recipients in one XML-request to not more than 100.

Furthermore the Delivery Status Request as described in section 6.8 and 6.9 can not be used for a message sent in a batch. This is due to that all messages in a batch will be given the same ID and if only one message in the batch has been successful, then a positive response will be returned to the customer for any of the messages in the batch.

14.6 Cleaning of MSISDN

The Pro platform will do it’s best to clean the number forwarded in the tag to_msisdn. Only the digits 0-9 and a possibly leading “+” will be accepted. For international numbers it is mandatory with a leading “+”. Both the normal mobile number series called 0+9 and the telematics number series 0+13 are accepted.

14.7 Flash-SMS

It is possible to deliver flash-SMS with the Pro platform (or class 0 SMS as it is called in ETSI 03.40 specification). Simply set attribute message class to zero in the SMS element according to table 1.

Example,

```
<sms account="71700" msg_class="0">
```

Appendix 1 - Implementation of HTTP Basic Authentication

A request is always authenticated through username and password, which are supplied in the appendix to the agreement.

Additional credentials are forwarded through HTTP Basic Authentication as described in this appendix.

If implementation of HTTP Basic Authentication for some reason is not possible, then authentication can be made through a fixed IP-address, which is supplied by the customer in the appendix to the agreement.

If authentication is made through a fixed IP-address, then HTTP or HTTPS is supported in both directions.

Implementation of HTTP Basic Authentication supports dynamic IP.

All communication must be made using SSL-encryption.

Credentials are sent using ”HTTP Basic Authentication” with an authorization header in the HTTP-request.

The URL to which the customer sends the request is,
https://sms-pro.net:44343/services/**customername**/sendsms where “customername” is replaced by the customer’s CustomerID, which is found in the appendix to the agreement.

HTTP basic authentication is a simple challenge and response mechanism with which a server can request authentication information (a user-ID and password) from a client. The client passes the authentication information to the server in an Authorization header. The authentication information is in base-64 encoding.

If a client makes a request for which the server expects authentication information, the server sends an HTTP response with a 401 status code, a reason phrase indicating an authentication error and a WWW-Authenticate header.

The format of a WWW-Authenticate header for HTTP basic authentication is,

```
WWW-Authenticate: Basic realm="Our Site"
```

The WWW-Authenticate header contains a realm attribute, which identifies the set of resources to which the authentication information requested (that is, the user-ID and password) will apply. Web clients display this string to the end user when they request a user-ID and password. Each realm may require different authentication information. Web clients may store the authentication information for each realm so that the endusers do not need to retype the information for every request.

When the Web client has obtained a user-ID and password, it resends the original request with an Authorization header. Alternatively, the client may send the Authorization header when it makes its original request and this might be accepted by the server, avoiding the challenge and response process.

The format of the Authorization header is,

```
Authorization: Basic userid: password
```

The user-ID and password are encoded using the base-64 encoding scheme.

The RFC 2617, HTTP Authentication: Basic and Digest Access Authentication, available from http://www.ietf.org/rfc/rfc2617.txt, has more detailed information about basic authentication.

Note that the HTTP basic authentication scheme can only be considered a secure means of authentication when the connection between the Web client and the server is secure. If the connection is insecure, the scheme does not provide sufficient security to prevent unauthorized users from discovering and using the authentication information for a server. If there is a possibility of a password being intercepted, basic authentication should be used in combination with SSL, so that SSL encryption is used to protect the user-ID and password information.

Examples

Java

```
package test;

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;

import org.apache.commons.codec.binary.Base64;

public class ConnectToUrlUsingBasicAuthentication {

    public static void main(String[] args) {

       try {
         String webPage = "http://192.168.1.1";
         String name = "admin";
         String password = "admin";

         String authString = name + ":" + password;
         System.out.println("auth string: " + authString);
         byte[] authEncBytes = Base64.encodeBase64(authString.getBytes());
         String authStringEnc = new String(authEncBytes);
         System.out.println("Base64 encoded auth string: " + authStringEnc);

         URL url = new URL(webPage);
         URLConnection urlConnection = url.openConnection();
         urlConnection.setRequestProperty("Authorization", "Basic " +
                      authStringEnc);
         InputStream is = urlConnection.getInputStream();
         InputStreamReader isr = new InputStreamReader(is);

         int numCharsRead;
         char[] charArray = new char[1024];
         StringBuffer sb = new StringBuffer();
         while ((numCharsRead = isr.read(charArray)) > 0) {
                   sb.append(charArray, 0, numCharsRead);
                }
                String result = sb.toString();

                System.out.println("*** BEGIN ***");
                System.out.println(result);
                System.out.println("*** END ***");
            } catch (MalformedURLException e) {
                    e.printStackTrace();
            } catch (IOException e) {
                    e.printStackTrace();
           }
        }
```

Java (Using HttpClient)

```
HttpClient myHttpClient = new HttpClient();
       myHttpClient.getState().setCredentials(
         new AuthScope("sms-pro.net", 44343, "SmsPro auth"),
         new UsernamePasswordCredentials("smsprouser", "smspropwd"));
String post = buildPostMethod(xmlToSend);
myHttpClient.executeMethod(post);
```

C#

```
byte[] AuthBytes = Encoding.ASCII.GetBytes("username:password");
string sAuth = Convert.ToBase64String(AuthBytes);
WebRequest request = WebRequest.Create("https://sms-pro.net:44343/services/
CUSTOMERID/sendsms");
request.Headers.Add("Authorization", "Basic " + sAuth);
request.PreAuthenticate = true;
request.Method = "POST";
request.ContentType = "text/xml";
// fill in other request properties here, like content
WebResponse respose = request.GetResponse();
```