XML Protocol Specifications

Version 1.5.2
07/08/2009

mpulse
mobile payment & messaging

1. Revisions

Rev#	Date	Remark
1.0	11-05-2004	Initial revision
1.2.1	15-09-2004	Suppressed applications' response
1.3	15-09-2004	Added binary data support
1.3.1	29-09-2004	Added a note on keywords conventions Added a note on the format of <data/> when TP-UDH is provided Changed the possible values of the <type/> element and an explanation on how it is handeld by MG.
1.3.2	06-12-2004	Added support for sending URLs
1.3.3	02-02-2005	Added support for Mobile Number Portability through the use of operator identifiers
1.3.4	24-06-2005	Added a note on some terminology at the beginning of section 2 Conventions
1.3.5	06-09-2005	The shortcode element is no longer present in delivery receipts
1.3.6	13-06-2006	Added a note on the timeout for the sending of MO messages
1.3.7	22-08-2006	Updates to reflect new shortcode and MSISDN formats
1.4.0	15-10-2007	Added ability to request and send delivery reports for failed messages
1.5.0	19-02-2008	Added support for message priority, message expiration, and split (concatenated) messages
1.5.1	05-03-2008	Clarifications on MSISDN format to use and error handling. DTDs for the XML fragments that are used.
1.5.2	07-08-2009	Changing the possible values of the operator elements to the more general MCC/MNC format.

2. Conventions

This document describes the XML format used for message exchanges between Mpulse "mroute" and "mcharge" gateways and customers' applications. Throughout the rest of this document, Mpulse gateways will be abbreviated "MG" and external applications will be referred to as "EA".

The following expressions are allowed within content descriptions:

- [0-9a-z] : the ASCII lowercase letters, plus the ten decimal digits ;
- HEX : the hexadecimal digits ([0-9a-fA-F]) ;
- ISO-8859-1 : latin characters[1] ;
- 'TEXT' : the text TEXT ;
- HEX | 'TEXT' : either hexadecimal digits, or the text TEXT.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described by RFC 2119 (http://www.ietf.org/rfc/rfc2119.txt).

---

[1] For more information, please refer to Special characters encoding, on page 6.

3.   Mobile-Originated messages (MO SMS)

MG sends messages to EA in XML documents, which are contained within an http POST variable, namely xmlData. This variable can then be handled by a script, client-side. Here is an example message :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<nvxms>
 <message>
  <shortcode>64444</shortcode>
  <sender>352621123456</sender>
  <operator>270/01</operator>
  <data>Please send me your brochure</data>
  <date>1061304840</date>
  <type>TEXT</type>
 </message>
</nvxms>
```

Important: The HTTP connections with the EA will timeout after 20 seconds. If no HTTP 200 code has been received within this time, MG will store the message and retry at a later point of time. Please make sure that the processing time for a message does not exceed this timeout value.

Currently messages will always be sent in the ISO-8859-1 charset, hence the presence of the XML Prolog which will prevent XML parsers to choke on accented characters.

Element / Attribute name	Description	Possible Values
nvxms	The root element of each message exchanged between MG and an EA	
message	Holds details of an incoming message. For the time being, there is only one message element	
shortcode	The shortcode to which the message is destinated	5-digit numbers starting by 64 or 67, such as 64234
sender	The MSISDN of the message's originator	A mobile number in international format, e.g. 352691123456
operator	The network from which the message is coming	270/01 for Luxgsm 270/77 for Tango 270/99 for Orange
data	Text of the incoming message	ISO-8859-1 strings
udh	The User Data Header	HEX-ASCII characters
date	Date at which the message has been received	UNIX timestamps[2]
type	Type of the incoming message	Currently this element will always contain the text element 'TEXT'

---

[2] The number of seconds since the Epoch, i.e. Jan 1st 1970.

3.1. The operator element

Since Mobile Number Portability has become a legal obligation in the member countries of the EU, an extra field indicating the operator that a MSISDN belongs to. We're using the popular MCC/MNC format which contains a Mobile Country Code (MCC) and a Mobile Network Code (MNC) that uniquely identify network operators.

http://en.wikipedia.org/wiki/Mobile_Network_Code contains a pretty comprehensive list of existing codes. The three Luxembourg network operators are using the following values for the <operator> field:

- 270/01 means a LuxGSM subscriber
- 207/77 means a Tango subscriber
- 270/99 means a Orange subscriber

With this information, one can build statistical reports, provide value-added services, and implement whichever logic one wants on a per-operator basis.

4. Mobile-Terminated messages (MT SMS)

MG supports the sending of text and binary messages. The XML format also provides a shorthand for sending Wap-Push URLs.

Element / Attribute name	Description	Possible Values
login	The username of your MG account	
password	The password of your MG account	
nvxms	The root element of each message exchanged between MG and an EA	
message	Holds details of an outgoing message. For the time being, there is only one message element	
shortcode	The shortcode from which the message is sent	5-digit numbers
receiver	Destination address. There can be any number of destinations	A mobile number in international format (ITU-T E.164) – no leading + or 00
data	Text of the messages	Latin1 characters or HEX-ASCII data for binary messages
type	Type of message	TEXT for text messages BINARY for binary messages URL for Wap-Push messages
udh	The User Data Header for binary messages	HEX-ASCII characters
deliveryreport	Optional: the type of delivery reports requested for this message	NONE: no delivery reports ALL: all delivery reports DELIVERED: reports only for

		delivered messages
absolute-time-to-live	Set an absolute expiration date for this message. If the message could not be delivered by the specified date, it will be marked as expired and removed from the delivery queue. The maximum value depends on the operator but is usually 1-2 days.	Format: yyyy/MM/dd HH:mm:ss Z
relative-time-to-live	Same as above, except that the value is the number of seconds from the moment the message has been sent.	Integer e.g. 86400 will set the TTL for this message to 1 day
absolute-scheduled-delivery	Set an absolute scheduled delivery time. This message will be delivered by the operators who support this feature no earlier than the specified date.	Format: yyyy/MM/dd HH:mm:ss Z
relative-scheduled-delivery	Same as above, except that the value is the number of seconds from the moment the message has been sent.	Integer e.g. 86400 will set the scheduled delivery for this message to 1 day
split	If set to true, messages longer than 160 characters will be split and sent as concatenated messages. Each message will contain at most 152 payload characters. Each generated message will be billed separately. Note: only available for TEXT-type messages.	*true* or *false*
priority	Allows modification of the priority of the message in the Mpulse Gateway and operator queues. Example: Bulk messages should be sent at a lower priority than time critical alert messages.	Integer between 0 and 9, 9 being the highest. Default is 4. Maximum value is defined in the sender account.

As of version 1.3, MG supports binary MT messages. This can be easily achieved by:
1. setting the type element to have the 'BINARY' text node as a child ;
2. setting the udh element to have the correct TP-UDH value (see GSM 03.40). As soon as TP-UDH is provided, the data element MUST be hex-encoded; MG won't handle any conversion task.

4.1. The udh element

This element allows full control over the whole TP-UD value, client-side. By specifying a value to the udh element, one can address extended aspects such as message concatenation, pictures, ringing tones, EMS features, etc. For Wap-Push or concatenated messages, it is recommended to use the standard message features described above in chapter 4.

4.2. The type element

By default, an SMS is considered pure text and is sent verbatim to the mobile destination(s). This is also the case when this element contains the value 'TEXT'. The application SHOULD specify the value 'BINARY' when it has to send binary content, such as logos and/or ringtones. In this case, the TP-UDH must be provided by means of the udh element (see section above).

The type element allows the application to send URLs, by specifying the value 'URL' in it, offering a possibility to request the user to download content on a WAP website. The data element will hold the URL of the resource that has to be accessed by the user via his WAP connection; it could be anything that the mobile is able to interpret: a ringtone, a logo, et al.

As of version 1.3.2, the mobile will display the text "Download" pointing to the given URL. A future release will allow the application to specify a custom text. The URL will be sent in a WAP Push message by MG, and MG will handle message encryption, so there's no need to set the udh element. Here's a simple example of AO message:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<nvxms>
 <login>user123</login>
 <password>AAAAA</password>
 <message>
  <shortcode>xxxx</shortcode>
  <receiver>352621123456</receiver>
  <data>http://www.luxsms.lu/download.php?id=3306</data>
  <type>URL</type>
 </message>
</nvxms>
```

Note: the format of the URL must conform to approriate encoding rules ([RFC1738, RFC2369]) concerning special characters, that is:

- Basically, it is safe to translate any character not within [a-zA-Z0-9] as long as it is not a special character used for its special purpose (e.g.: the `&' must not be converted if it is used as an URL parameters' separator);
- Translation consists of a `%' symbol followed by the two-digit hexadecimal representation of the [Latin1] code point for the character.
- As part of an XML element, XML encoding rules still apply: `&' must be replaced by its corresponding parameter entity `&' (only when used as an URL parameters' separator)[3].

5. Responses

When MT messages are posted to MG, the client application will receive in response a list of message identifier, i.e. one identifier per recipient in the message. These ids are unique system-wide, so the application has not to worry about collisions, as they won't ever occur.
The response sent back to the client is an XML document which content is described below:

---

[3] This is the only case in fact : `<', `'', `"' and `>' are not reserved characters as per [RFC1738, RFC2369], so they must always be converted to their `%'-counterpart.

Element / Attribute name	Description	Possible Values
nvxms	The root element of each message exchanged between MG and an EA	
result	The element containing the message identifiers	
receiver	Identifies one receiver of the message	
number	The MSISDN of the receiver	A mobile number in international format
id	The identifier of the message, related to the receiver	A long number

Here is an example of response sent back to the client :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<nvxms>
 <result>
  <receiver>
   <number>352621123456</number>
   <id>45617645</id>
  </receiver>
  <receiver>
   <number>352691654321</number>
   <id>45617646</id>
  </receiver>
 </result>
</nvxms>
```

5.1. Error response

In case an error occurred while validating the data submitted to MG, it will provide an error description in the following format:

```
<nvxms>
 <error>
  <text>Error description.</text>
 </error>
</nvxms>
```

6. Delivery notifications

For MT messages, delivery notifications are sent to the originating application the same way incoming messages are. With the message identifier, an application can know exactly when a message has been received by which recipient.

Element / Attribute name	Description	Possible Values
nvxms	The root element of each message exchanged between MG and an EA	
report	The element containing the status report	
timestamp	Time at which the message has been delivered	UNIX timestamp
id	The message identifier related to the recipient	A long number
status	The status of the original message. The possible values are determined by the settings of the sender account or by the <deliveryreport> parameter in the original message.	**DELIVERED**: message has been delivered to the handset **NOTDELIVERED**: message has not been delivered to the handset **REJECTED**: message has not been accepted by the mobile operator **BILLING_ERROR**: the premium MT message could not be billed because of lack of credit **UNKNOWN**: no further information has been provided by the mobile operator **EXPIRED**: the time to live for the message has expired **DELETED**: the message has been removed from the queue by the mobile operator **BLACKLISTED**: the destination number cannot receive messages from this service

Here is an example delivery report :

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nvxms>
 <report>
  <timestamp>1234551987</timestamp>
  <id>45617645</id>
  <status>DELIVERED</status>
 </report>
</nvxms>
```

7. Special characters encoding

Conforming to the XML specification, each XML document MUST start with a *processing instruction* who taughts the reader about some properties of the document.

Each document sent to MG MUST start with :

<?xml version="1.0" encoding="ISO-8859-1"?>

Some characters are considered illegal within the context of XML elements, because the parser might interpret them as control sequences and hence produce parsing errors. In order to avoid that, the following 5 characters SHOULD be replaced by their corresponding *entity reference*, as shown in the table below :

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

Note: only « < » and « & » are strictly illegal in XML, while apostrophes and quotation marks need to be encoded only within attribute values.

8. DTDs

The document type definitions of the XML documents used in this specification will help you validate your XML fragments before submitting them to MG.

8.1. MO messages

```
<!ELEMENT data ( #PCDATA ) >
<!ELEMENT date ( #PCDATA ) >
<!ELEMENT message ( shortcode, sender, operator, data, date, type ) >
<!ELEMENT nvxms ( message ) >
<!ELEMENT operator ( #PCDATA ) >
<!ELEMENT sender ( #PCDATA ) >
<!ELEMENT shortcode ( #PCDATA ) >
<!ELEMENT type ( #PCDATA ) >
```

8.2. MT messages

```
<!ELEMENT absolute-time-to-live ( #PCDATA ) >
<!ELEMENT absolute-scheduled-delivery ( #PCDATA ) >
<!ELEMENT data ( #PCDATA ) >
<!ELEMENT deliveryreport ( #PCDATA ) >
<!ELEMENT login ( #PCDATA ) >
<!ELEMENT message ( shortcode, receiver+, udh, data, deliveryreport, absolute-time-to-live, relative-
time-to-live, absolute-scheduled-delivery, relative-scheduled-delivery, split, priority, type ) >
<!ELEMENT nvxms ( login, password, message ) >
<!ELEMENT password ( #PCDATA ) >
<!ELEMENT priority ( #PCDATA ) >
<!ELEMENT receiver ( #PCDATA ) >
<!ELEMENT relative-time-to-live ( #PCDATA ) >
<!ELEMENT relative-scheduled-delivery ( #PCDATA ) >
<!ELEMENT shortcode ( #PCDATA ) >
<!ELEMENT split ( #PCDATA ) >
<!ELEMENT type ( #PCDATA ) >
<!ELEMENT udh ( #PCDATA ) >
```

8.3. MT message response

```
<!ELEMENT id ( #PCDATA ) >
<!ELEMENT number ( #PCDATA ) >
<!ELEMENT nvxms ( result ) >
<!ELEMENT receiver ( number, id ) >
<!ELEMENT result ( receiver+ ) >
```

8.4. Error response

```
<!ELEMENT error ( text ) >
<!ELEMENT nvxms ( error ) >
<!ELEMENT text ( #PCDATA ) >
```

8.5. Delivery report

```
<!ELEMENT id ( #PCDATA ) >
<!ELEMENT nvxms ( report ) >
<!ELEMENT report ( timestamp, id, status ) >
<!ELEMENT status ( #PCDATA ) >
<!ELEMENT timestamp ( #PCDATA ) >
```