

TrueSenses.com

XML-Interface specifications

Version 1.02

2003-07-03

Table of contents

1.	General	3
2.1	Content of the XML document	3
2.2	Example document	4
2.3	Transmission of XML document	4
2.4	<Originator value="x" />	5
2.5	<Destination>	5
3.0	<SMS>	6
3.1	<messagewaiting type="x" onoff="y" sendcontent="z" />	7
3.2	<EMS>	8
3.3	<predefinedsound type="x" />	9
3.4	<predefinedanimation type="x" />	9
3.5	<IMAGE>	10
3.6	<NOKIALOGO>	12
3.7	<NOKIAMULTIPART>	13
3.8	<NOKIARINGTONES>	14
3.9	<WAPPUSH>	15
5.0	XML-Responses	16
6.0	BI-Directional PHP-Example truesenses_2way.php	17

Revision history

03/07/03 Added section 3.9 <WAPPUSH>

Service overview

TrueSenses.com offers various access methods in order to send SMS (short message service).

For simple text-messages (includes flashing SMS) TrueSenses recommends the HTTP(s) or E-Mail interface as described on our site www.truesenses.com under "Help / Download".

The TrueSenses XML-Interface allows you to send more complex content such as:

- Ringtones (RTTTL) for Nokia mobilephones
- Operator logos for Nokia mobilephones
- Multipart messages (text and image) for Nokia mobilephones and some other vendors.
- EMS logos (operator logos) for all vendors supporting EMS on their handsets.
- EMS messages (formatted SMS messages) for all vendors supporting EMS on their handsets.

The XML-Interface also allows you to send normal SMS and flashing SMS.

You can specify up to 1000 recipients per XML-File and you can set the originator ID to any number or to any alphanumeric combination of up to 11 characters.

1. General

The gateway is listening on the following addresses:

- | | | |
|---------|--|----------------|
| ▪ HTTP | Server 1: www.truesenses.com/cgi-bin/smsgateway.cgi | HTTP-Port 80 |
| ▪ HTTPS | Server 2 : secure.simmcomm.ch/cgi-bin/smsgateway.cgi | HTTPS-Port 443 |

In order to access the gateway you must have:

- A valid TrueSenses account / password
- **Enabled** HTTP-Authentication for your host under "Configuration / HTTP-Authentication"

2.1 Content of the XML document

The XML document has to start with the <Message> tag and must end with the </Message> tag. Please note that all tags in this document are **CaSe-sEnSitive**.

2.2 Example document

The following document is a simple example in order to send one SMS to two different subscribers by using originator-id "Test":

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Message>
  <Destination>
    <GSM number="+41790000001" />
    <GSM number="+41790000002" />
  </Destination>

  <ORIGINATOR value="Test"/>

  <SMS>
    <CONTENT>This message is just a test</CONTENT>
  </SMS>
</Message>
```

2.3 Transmission of XML document

The XML document must be sent to the gateway(s) mentioned in "1. General". The following shell-script will send the document by using curl as the HTTP-handler.

```
-- begin of sendxml.sh --
#!/bin/bash

/usr/bin/curl -s "https://secure.simmcomm.ch/cgi-
bin/smsgateway.cgi?FUNCTION=EXTERN_XMLUPLOAD&ACCOUNT=$1&PASSWORD=$2" -T $3
-- end of sendxml.sh --
```

The script requires the following parameters:

- \$1) TrueSenses Account-Name
- \$2) TrueSenses Account-Password
- \$3) PathtoXMLfile (XML-File you want to send)

The script returns the XML-output of the gateway to the standard output.

2.4 <Originator value="x" />

This tag is mandatory!

Specifies a numeric (any number) or alphanumeric (up to 11 characters) originator.

Any abuse of this feature will be tracked and prosecuted according to our general terms and conditions. We will close such accounts immediately without prior notification. Thank you for your understanding.

Example:

```
<Originator value="+41790000001" />
```

2.5 <Destination>

This tag is mandatory!

Define at least one up to 1000 destinations where the following content blocks should be sent to. The phonenumber should be addressed international like "+41790000001", however "0041790000001" is also understood by the gateway.

Example:

```
<Destination>  
  <GSM number="+41790000001" />  
</Destination>
```

Example for multiple destinations:

```
<Destination>  
  <GSM number="+41790000001" />  
  <GSM number="+41790000002" />  
  <GSM number="+41790000003" />  
</Destination>
```

3.0 <SMS>

Sends a SMS message.

Example:

```
<SMS>  
  <CONTENT>This is a testmessage </CONTENT>  
  <CONTENT>which continues with a linefeed here  
  and ends here</CONTENT>  
</SMS>
```

The text for the SMS is extracted from within the <CONTENT> tags. Any linefeed inside the <CONTENT> tag will be converted to a linefeed on the mobilephone.

In order to send special characters like Umlauts (ü,ö,ä), the SMS must be HEX-encoded.

Example for "This is a text containing Ü Ö Ä" as HEX-encoded:

```
<SMS encoded="YES">  
  <CONTENT>This+is+a+text+containing+%DC+%D6+%C4</CONTENT>  
</SMS>
```

Optional parameters:

<flashing />	Sets message class to 0 (flashing SMS)
<long />	Sets long-SMS mode to ON. Accepts >160 characters.

Example for flashing/long SMS:

```
<SMS>  
  <flashing />  
  <long />  
  <CONTENT>This is a flashing SMS which could be very, very long</CONTENT>  
</SMS>
```

3.1 <messagewaiting type="x" onoff="y" sendcontent="z" />

This tag is only valid inside the <SMS> tag!

Sets or clears the messagewaiting indicator on the mobilephone.

Available modes:

- VOICE on / off
- FAX on / off
- DATA on / off
- OTHER on / off

Example with text:

```
<SMS>  
  <messagewaiting type="VOICE" onoff="ON" sendcontent="YES">  
  <CONTENT>This text will be sent together with the messagewaiting flag</CONTENT>  
</SMS>
```

Example without text:

```
<SMS>  
  <messagewaiting type="VOICE" onoff="ON" sendcontent="NO">  
</SMS>
```

ONLY one messagewaiting mode may be set / cleared at a time. Set sendcontent to "YES" if you are sending <CONTENT> together with your messagewaiting indicator. The <messagewaiting> tag must always come before the <CONTENT> tag.

In order to send a so called silent SMS (for tracking purposes), simply set one of the messagewaiting modes to "OFF". The mobilephone will not alert any incoming SMS of this type.

3.2 <EMS>

Sends a EMS formatted message.

Example:

```
<EMS>
  <CONTENT><BOLD><CENTER>SCI
  </CENTER></BOLD><LEFT>Left
  </LEFT><RIGHT>Right
  </RIGHT><COLOR = "darkyellow">Yellow</COLOR></CONTENT>
</SMS>
```

The text for the EMS is extracted from within the <CONTENT> tags. Any linefeed inside the <CONTENT> tag will be converted to a linefeed on the mobilephone.

Changing attributes requires to have a linefeed at the end of the old attribute, as you cannot align text both left and right on the same line.

Inside the <CONTENT> tags you are able to apply the following formatting attributes:

<LEFT>	Align text left
<CENTER>	Align text centered
<RIGHT>	Align text right
<LARGE>	Large text
<SMALL>	Small text
<BOLD>	Bold text
<ITALIC>	<i>Italic text</i>
<UNDERLINE>	<u>Underline text</u>
<STRIKETHROUGH>	Strikethrough text
<COLOR value="x">	Colored text
<BGCOLOR value="x">	Colored background

The following colors are supported:

- black/darkgrey/darkred/darkyellow/darkgreen/darkcyan/darkblue/darkmagenta/grey/white/brightred
- brightyellow/brightgreen/brightcyan/brightblue/brightmagenta

Please note that some EMS mobilephones might not support all tags like COLOR.

In order to send special characters like Umlauts (ü,ö,ä), the EMS must be HEX-encoded.

Example for “This is a text containing Ü Ö Ä” as HEX-encoded:

```
<EMS encoded="YES">  
  <CONTENT>This+is+a+text+containing+%DC+%D6+%C4</CONTENT>  
</EMS>
```

Optional parameters:

<flashing /> Sets message class to 0 (flashing EMS)

3.3 <predefinedsound type="x" />

This tag is only valid inside the <EMS> tag!

Plays a predefined sound stored in the mobilephone.

Available types:

- 0..255

Example:

```
<EMS>  
  <CONTENT>Listen to this:<predefinedsound type="3" /></CONTENT>  
</EMS>
```

3.4 <predefinedanimation type="x" />

This tag is only valid inside the <EMS> tag!

Shows a predefined animation stored in the mobilephone.

Available types:

- 0..255

Example:

```
<EMS>  
  <CONTENT>Look at this:<predefinedanimation type="5" /></CONTENT>  
</EMS>
```

3.5 <IMAGE>

This tag is only valid inside the <EMS>, <NOKIALOGO> and <NOKIAMULTIPART> tags!

Sends an EMS-Image (or operator logo) to the mobilephone.

Available sources:

- HTTP Instructs the gateway to download the image from the given source.
example: <http://www.simmcomm.ch/nokialogo.gif>
- BINARY Instructs the gateway to take the HEX-encoded image
example: HEX-encoded image (%c1%c2%c3%c4%c5...)

Available types:

- GIF Source is GIF
- JPG Source is JPG
- PNG Source is PNG

Available options for protected:

- YES Disallows the recipient to re-distribute the image
- NO Allows the recipient to re-distribute the image

Available options for autosave:

- YES Asks the recipient to auto-save the image
- NO Do not ask the recipient to save the image

Available options for width:

- -unspecified- Autosize the image
- 1..x Width of the image

Available options for height:

- -unspecified- Autosize the image
- 1..x Height of the image

Example of combined text and image:

```
<EMS>
  <CONTENT>Before
  <IMAGE source="HTTP" type="GIF">http://www.simmcomm.ch/nokialogo.gif</IMAGE>
  After</CONTENT>
</EMS>
```

Example of copyprotected and autosave image:

```
<EMS>
  <CONTENT>
  <IMAGE source="HTTP" type="GIF" protected="YES" autosave="YES">http://www.simmcomm.ch/nokialogo.gif</IMAGE>
  </CONTENT>
</EMS>
```

Example of specified width and height:

```
<EMS>
  <CONTENT>
  <IMAGE source="HTTP" type="GIF" width="72" height="14">http://www.simmcomm.ch/nokialogo.gif</IMAGE>
  </CONTENT>
</EMS>
```

Example of HEX-encoded image:

```
<EMS>
  <CONTENT>
  <IMAGE source="BINARY" type="GIF">%bi%na%ry%da%ta%he%re</IMAGE>
  </CONTENT>
</EMS>
```

3.6 <NOKIALOGO>

*This tag must be treated separate as like <SMS>, <EMS>, ..
This tag includes the <IMAGE> tag as described in 3.5 <IMAGE>.*

Sends an operator logo to a NOKIA mobilephone.

Available options for MNC:

- 2 digit number Specify the MobileNetworkCode

Available options for MCC:

- 3 digit number Specify the MobileCountryCode

The above settings are required as NOKIA mobilephones only display operator logos when the phone is logged in to the specified network.

*A complete list of MNC/MCC operator codes is available at <http://www.gsmworld.com/gsminfo/>
Please note that the MobileCountryCode has nothing to do with the international prefix. In the example below, Switzerland has 228 as MobileCountryCode and not +41 as when dialing a Swiss number!*

Example of operator logo for provider Swisscom (01) , Switzerland (228):

```
<NOKIALOGO mcc="228" mnc="01">  
  <IMAGE source="HTTP" type="GIF">http://www.simmcomm.ch/nokialogo.gif</IMAGE>  
</NOKIALOGO>
```

For more information regarding the <IMAGE> tag, please refer to section 3.5.

3.7 <NOKIAMULTIPART>

*This tag must be treated separate as like <SMS>, <EMS>, ..
This tag includes the <IMAGE> tag as described in 3.5 <IMAGE>.*

Sends a picture message (text and image) to a NOKIA mobilephone. Some models of other vendors like SonyEricsson are able to receive this messagetype but are unable to forward it.

Example of picture message with text:

```
<NOKIAMULTIPART>  
  <IMAGE source="HTTP" type="GIF">http://www.simmcomm.ch/nokialogo.gif</IMAGE>  
  <CONTENT>And some text here</CONTENT>  
</NOKIAMULTIPART>
```

In order to send special characters like Umlauts (ü,ö,ä), the NOKIAMULTIPART must be HEX-encoded.

Example for "This is a text containing Ü Ö Ä" as HEX-encoded:

```
<NOKIAMULTIPART encoded="YES">  
  <IMAGE source="HTTP" type="GIF">http://www.simmcomm.ch/nokialogo.gif</IMAGE>  
  <CONTENT>This+is+a+text+containing+%DC+%D6+%C4</CONTENT>  
</NOKIAMULTIPART>
```

For more information regarding the <IMAGE> tag, please refer to section 3.5.

3.8 <NOKIARINGTONE>

This tag must be treated separate as like <SMS>, <EMS>, ..

Sends a RTTTL-ringtone to a NOKIA mobilephone.

Available sources:

- HTTP Instructs the gateway to download the ringtone from the given source.
example: <http://www.simmcomm.ch/ringtone.txt>
- BINARY Instructs the gateway to take the HEX-encoded ringtone
example: HEX-encoded ringtone (%c1%c2%c3%c4%c5...)

Available types:

- RTTTL Source is RTTTL-Ringtone

Example of ringtone:

```
<NOKIARINGTONE>  
  <RINGTONE source="HTTP" type="RTTTL">http://www.simmcomm.ch/ringtone.txt</RINGTONE>  
</NOKIARINGTONE>
```

Example of HEX-encoded ringtone (no linefeeds within RINGTONE!):

```
<NOKIARINGTONE>  
  <RINGTONE source="BINARY" type="RTTTL">  
  KnightRider%3Ad%3D4%2Co%3D5%2Cb%3D125%3A16e%2C16p%2C16f%2C16e%2C16e%2C16p%2C16e%2C  
  16e%2C16f%2C16e%2C16e%2C16d%23%2C16e%2C16e%2C16e%2C16e%2C16p%2C16f%2C16e%2C1  
  6e%2C16p%2C16f%2C16e%2C16f%2C16e%2C16e%2C16e%2C16d%23%2C16e%2C16e%2C16e%2C16d%2C16  
  p%2C16e%2C16d%2C16d%2C16p%2C16e%2C16d%2C16e%2C16d%2C16d%2C16d%2C16c%2C16d%2C16d%2  
  C16d%2C16d%2C16p%2C16e%2C16d%2C16d%2C16p%2C16e%2C16d%2C16e%2C16d%2C16d%2C16d%2C16  
  c%2C16d%2C16d%2C16d</RINGTONE>  
</NOKIARINGTONE>
```

3.9 <WAPPUSH>

This tag must be treated separate as like <SMS>, <EMS>, ..

Sends a WAP Push-Link including messagetext to a mobilephone.

Available options for LINK:

- URL HEX-encoded URL to load in WAP-Browser on phone
Must (currently) always start with http://www.

example as plain-text:

`http://www.wapbrowse.com/cgi-bin/test.cgi?mylogin=test¶m1=test`

example as HEX-encoded:

`http%3A%2F%2Fwww.wapbrowse.com%2Fcgi-bin%2Ftest.cgi%3Fmylogin%3Dtest%26param1%3Dtest`

Available options for SIID:

- SIID Provide the WAP Service Indication

Example of wappush:

```
<WAPPUSH link="http%3A%2F%2Fwww.wapbrowse.com" siid="sci">  
  <CONTENT>This is the link</CONTENT>  
</WAPPUSH>
```

In order to send special characters like Umlauts (ü,ö,ä), the WAPPUSH must be HEX-encoded.

Example for "This is a text containing Ü Ö Ä" as HEX-encoded:

```
<WAPPUSH link="http%3A%2F%2Fwww.wapbrowse.com" siid="sci" encoded="YES">  
  <CONTENT>This+is+a+text+containing+%DC+%D6+%C4</CONTENT>  
</WAPPUSH>
```

Attention: The LINK together with the messagetext must fit into 1 SMS. Please consider not to use too long links or text.

5.0 XML-Responses

In case of a general-error the gateway will respond like this example:

```
<?xml version="1.0"?>
<Reply>
<Error>Invalid login</Error>
</Reply>
```

In case of a successful reception the gateway will respond like this example:

```
<?xml version="1.0"?>
<Reply>
<Result destination="+41790000001" number="0041790000001" mode="SMS" type="OK">990701170214</Result>
<Result destination="+41790000002" number="0041790000002" mode="NOKIARINGTON" type="OK">990701170215</Result>
<Result destination="+41790000003" number="0041790000003" mode="SMS" type="ERROR">Error message here</Result>
<Result destination="+41790000004" number="0041790000004" mode="SMS" type="OK">990701170216, 990701170217</Result>
</Reply>
```

Description of destination:

- Number in exact format as provided by the XML-document

Description of number:

- Number as parsed by the XML-gateway.

Description of mode:

- SMS
- EMS
- NOKIARINGTON
- NOKIALOGO
- NOKIAMULTIPART

Description of type:

- OK
The message was sent successfully. The messageid(s) are returned. If a message was split into multiple SMS, each messageid stands for 1 SMS. Multiple ID's are coma-separated.
- ERROR
In case of an error, the error message is written as plain-text where the messageid stands.

6.0 BI-directional PHP-example truesenses_2way.php:

This script allows you to send all messagetypes of this specification document to mobilephones by using PHP. The script is designed for a 2-way solution, which acts on incoming keyword SMS.

You can download this php script from:

http://www.truesenses.com/website/downloads/truesenses_2way.php

For a detailed description, please look at the code-header below:

```
<html>
<head>
<title>TrueSenses Response</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<?php

/*
*****
* 2-way mobile messaging script, (c) 2003 TrueSenses.com - http://www.truesenses.com
*
* -> This script allows you to operate a 2-way SMS service and send ringtones, logos
* -> and much more directly from your site.
*
* -> This script does not need any other execution rights like executing sub-
* -> programs and should therefore run on almost every server supporting PHP.
*
* -> For outgoing SMS you will need to have a TrueSenses.com account.
* -> For incoming SMS you will need to activate the keyword service for as low as
* -> CHF 5.- (Swiss Francs) per keyword and month including unlimited incoming SMS.
*
* You will need to configure $truesenses_account,$truesenses_password and
* $truesenses_origin
*
* This script is pre-configured to act on incoming SMS messages sent by the
* TrueSenses.com SMS-Gateway. It will compare the incoming keyword and send back
* the requested content.
*
* If you ONLY want to use this script to send messages (without 2-way service),
* simply modify the MAIN program code to discard the incoming parameters provided
* by the gateway: $NUMBER (origin number, international format) and $MESSAGE
*
* The following message-modes are supported:
*
* - SMS
* - EMS
* - EMS Images
* - NOKIA Ringtones
* - NOKIA Logos
* - NOKIA Multipart messages (Picture Message with text)
*
* In order to install the 2-way SMS script follow these steps:
*
* 1. Open an account on TrueSenses.com
* 2. Configure this script with your Account/Password and upload it to your server
* 3. Buy some outgoing SMS credits
* 4. Go to Configuration / Keyword services and add a keyword
```

```

* 4.1 Click on the keyword and activate PUSH to forward incoming SMS to this script
*   example: http://www.[yoursitename].com/truesenses_2way.php
*   method: select POST method - dont forget to click on "Activated"
* 4.2 Activate the keyword by clicking "Activate keyword"
* 5. Your service is now up and running
*
* Send the following SMS to +41-79 381 60 10 or the number given by TrueSenses.com
*
* yourkeyword SMS
* -> You will receive a SMS back to your phone.
*
* yourkeyword SMSFLASH
* -> You will receive a flashing class 0 SMS back to your phone.
*
* yourkeyword NOKIARINGTONE
* -> You will receive a ringtone for your NOKIA phone, downloaded from HTTP-Source.
*
* yourkeyword NOKIARINGTONEHEX
* -> You will receive a ringtone for your NOKIA phone, included as hex-encoded.
*
* yourkeyword NOKIALOGO
* -> You will receive an operator logo for your NOKIA phone, downloaded from HTTP-
* -> source. IMPORTANT: configure the mnc and mmc settings first according to the
* -> destination provider. Otherwise the logo will not show up.
*
* yourkeyword NOKIAMULTIPART
* -> You will receive a picture message with a text for your NOKIA phone, downloaded
* -> from a HTTP-source.
*
* yourkeyword EMSLOGO
* -> You will receive a picture message for your NON-Nokia phone like SonyEricsson
* -> and all other vendors who support the EMS-standard on their phones. The image
* -> is by default copy protected and cannot be sent to another phone. You can de-
* -> activate this by setting protected = "NO".
*
*****
*/

```

```

$truesenses_account = "YourAccountName"; // TrueSenses.com Account-Name
$truesenses_password = "YourPassword"; // Account-Password
$truesenses_origin = "TrueSenses"; // Message-Origin, 11 alpha chars or numeric number
$debug = 0; // debug-output 0 = off, 1 = on

```

```

//
// open a http-channel, transmit data and return received buffer
function sendToHost($host,$method,$path,$data,$useragent,$poststr)
{
    global $debug;

    if (empty($method)) { // default GET
        $method = 'GET';
    }
    $method = strtoupper($method);
    $fp = fsockopen($host, 80, $errno, $errstr, 30);
    if (!$fp) {
        if ($debug == 1) {
            echo("Could not open connection<br>$errstr ($errno)<br>\n");
        }
    } else {
        if ($method == 'GET')
            $path .= '?' . $data;
    }
}

```

```

fputs($fp, "$method $path HTTP/1.0\r\n");
fputs($fp, "Host: $host\r\n");
fputs($fp, "Content-Type: application/x-www-form-urlencoded\r\n");
if ($method == 'POST') {
    fputs($fp, "Content-Length: " . strlen($data) . "\r\n");
} else {
    if (strlen($poststr)>0) { // if something to POST on GET method
        fputs($fp, "Content-Length: " . strlen($poststr) . "\r\n");
    }
}
if (strlen($useragent)>0) {
    fputs($fp, "User-Agent: $useragent\r\n");
} else {
    fputs($fp, "User-Agent: TrueSensesPHP\r\n");
}
fputs($fp, "Connection: close\r\n");
fputs($fp, "\r\n");
if ($method == 'POST') {
    fputs($fp, $data);
} else { // GET method
    if (strlen($poststr)>0) { // if something to POST on GET method
        fputs($fp, $poststr);
    }
}

$buf = "";
$line = "";
$chr = "0";
$contentsize = "0";
// echo("reading response\n");
while (!feof($fp) && (strlen($buf)<=100000) && ($chr < 500)) { // read max 100000 bytes or 500 lines
    $chr = $chr + 1;
    $line = fgets($fp, 10000); // read one line
    $buf .= $line; // append to buffer
}

$contentpos = strpos(strtoupper($buf), "CONTENT-TYPE");
if ($contentpos > 0) {
    // echo("FOUND contenttype in ::$buf::<br>\n");
    $buf = substr($buf, $contentpos, strlen($buf) - $contentpos);
    // echo("new buffer::$buf::<br>\n");
    // $buf = substr($buf, strpos($buf, "\n") + 1, strlen($buf) - strpos($buf, "\n"));
    $buf = substr($buf, 27, strlen($buf) - (27 + 1)); // assumes Content-Type: text/html (27 chars)
} else {
    if ($debug == 1) {
        echo("NO contenttype found in ::$buf::<br>\n");
    }
    return "ERROR";
}

// delete everything else than the content from the buffer
// $buf = substr($buf, (strlen($buf) - $contentsize), $contentsize - 1);

// echo("buffer::$buf::<br>\n");
// echo("after response\n");

fclose($fp);
return $buf; // return the buffer
}
} // sendToHost

```

```
//
// compose the XML-message and transfer it to sendToHost
function sendXML($destination, $origin, $middle)
{
    global $truesenses_account;
    global $truesenses_password;

    $poststring = "<?xml version='1.0'?>\n";
    $poststring .= "<Message>\n";

    $poststring .= "<Destination>\n";
    $poststring .= "<GSM number='\$destination' />\n";
    $poststring .= "</Destination>\n";

    $poststring .= "<ORIGINATOR value='\$origin' />\n";

    $poststring .= $middle; // insert middle part here

    $poststring .= "</Message>\n";

    $sendresp = sendToHost('www.truesenses.com','get','/cgi-bin/smsgateway.cgi',
"FUNCTION=EXTERN_XMLUPLOAD&ACCOUNT=$truesenses_account&PASSWORD=$truesenses_password", "", "$poststring");

    return $sendresp; // return response
} // sendXML

// MAIN program

// Look if incoming message contains a space (keyword detected)
//
// Incoming message will contain the full message (no keyword required) if you
// are hosting a dedicated access number on TrueSenses
//
$pos = strpos ($MESSAGE, " ");

if ($pos === false) { // no space in message detected, keyword is message
    $KEYWORD = "$MESSAGE";
    $MSGONLY = "$MESSAGE";
} else { // space detected, extract keyword
    $KEYWORD = substr($MESSAGE, 0, $pos);
    $MSGONLY = substr($MESSAGE, ($pos + 1), (strlen($MESSAGE) - ($pos)));
}

if ($debug == 1) {
    echo("Keyword::$KEYWORD:: Msgonly::$MSGONLY:: Fullmessage::$MESSAGE::<br>\n");
}

if (strcmp(strtoupper($MSGONLY), "SMS") == 0) {

    // sends a normal SMS

    $xmlmsg = "<SMS encoded='YES'>\n";
    // add the following line if you want to send long-SMS (spawning over multiple SMS)
    // $xmlmsg .= "<long />\n";
    $xmlmsg .= "<CONTENT>";
    $xmlmsg .= urlencode("Your keyword was $KEYWORD and your message was $MSGONLY"); // hex-encode the content
    $xmlmsg .= "</CONTENT>\n";
    $xmlmsg .= "</SMS>\n";

    $rcbuf = sendXML($NUMBER, $truesenses_origin, $xmlmsg);

    if ($debug == 1) {
        echo("RESPONSE::$rcbuf:\n");
    }
} // SMS reply
```

```

if (strcmp(strtoupper($MSGONLY), "SMSFLASH") == 0) {

    // sends a flashing SMS

    $xmlmsg = "<SMS encoded=\"YES\">\n";
    $xmlmsg .= "<flashing />\n"; // set msgclass 0 flag
    // add the following line if you want to send long-SMS (spawning over multiple SMS)
    // $xmlmsg .= "<long />\n";
    $xmlmsg .= "<CONTENT>";
    $xmlmsg .= urlencode("Your keyword was $KEYWORD and your message was $MSGONLY"); // hex-encode the content
    $xmlmsg .= "</CONTENT>\n";
    $xmlmsg .= "</SMS>\n";

    $rcbuf = sendXML($NUMBER, $truesenses_origin, $xmlmsg);

    if ($debug == 1) {
        echo("RESPONSE::$rcbuf:\n");
    }
} // SMS reply

if (strcmp(strtoupper($MSGONLY), "NOKIARINGTONE") == 0) {

    // sends a nokiaringtone - input must be in RTTTL format
    //

    // assemble XML message

    $http_location = "http://www.simmmcomm.ch/ringtone.txt"; // location of the ringtone

    $xmlmsg = "<NOKIARINGTONE>\n";
    $xmlmsg .= "<RINGTONE source=\"HTTP\" type=\"RTTTL\">$http_location</RINGTONE>\n";
    $xmlmsg .= "</NOKIARINGTONE>\n";

    $rcbuf = sendXML($NUMBER, $truesenses_origin, $xmlmsg);

    if ($debug == 1) {
        echo("RESPONSE::$rcbuf:\n");
    }
} // NOKIARINGTONE

if (strcmp(strtoupper($MSGONLY), "NOKIARINGTONEHEX") == 0) {

    // sends a nokiaringtone as hex include - input must be in RTTTL format
    //

    // assemble XML message

    $http_location =
urlencode("KnightRider:d=4,o=5,b=125:16e,16p,16f,16e,16e,16p,16e,16e,16f,16e,16e,16e,16d#,16e,16e,16e,16p,16f,16e,16e,16f,16e,
16e,16e,16d#,16e,16e,16e,16d,16p,16e,16d,16d,16p,16e,16d,16e,16d,16d,16d,16c,16d,16d,16d,16d,16p,16e,16d,16d,16p,16e,16d,16d,16d,16
c,16d,16d,16d"); // free demo ringtone from the internet

    $xmlmsg = "<NOKIARINGTONE>\n";
    $xmlmsg .= "<RINGTONE source=\"BINARY\" type=\"RTTTL\">$http_location</RINGTONE>\n";
    $xmlmsg .= "</NOKIARINGTONE>\n";

    $rcbuf = sendXML($NUMBER, $truesenses_origin, $xmlmsg);

    if ($debug == 1) {
        echo("RESPONSE::$rcbuf:\n");
    }
} // NOKIARINGTONEHEX

```

```

if (strcmp(strtoupper($MSGONLY), "NOKIALOGO") == 0) {

    // sends a nokialogo - input must be in GIF, JPG or PNG
    //
    // parameters:
    // mcc = MobileCountryCode for destination (example 228 - switzerland)
    // mnc = MobileNetworkCode for destination (example 01 - swisscom, 02 sunrise, 03 orange)

    // assemble XML message

    $http_type = "GIF";
    $http_location = "http://www.simmmcomm.ch/nokialogo.gif"; // location of the logo

    $destination_mcc = "228"; // mandatory: MobileCountryCode for destination
    $destination_mnc = "01"; // mandatory: MobileNetworkCode for destination

    $xmlmsg = "<NOKIALOGO mcc=\"\$destination_mcc\" mnc=\"\$destination_mnc\">\n";
    $xmlmsg .= "<IMAGE source=\"HTTP\" type=\"\$http_type\"> \$http_location</IMAGE>\n";
    $xmlmsg .= "</NOKIALOGO>\n";

    $rcbuf = sendXML($NUMBER, $truesenses_origin, $xmlmsg);

    if ($debug == 1) {
        echo("RESPONSE::$rcbuf:\n");
    }
} // NOKIALOGO

if (strcmp(strtoupper($MSGONLY), "NOKIAMULTIPART") == 0) {

    // sends a nokiamultipart message (picturemessage) - input must be in GIF, JPG or PNG
    //
    // options:
    // - specify <CONTENT> tag if you want to send some text with your image

    // assemble XML message

    $http_type = "GIF";
    $http_location = "http://www.simmmcomm.ch/nokialogo.gif"; // location of the logo

    $xmlmsg = "<NOKIAMULTIPART encoded=\"YES\">\n";
    $xmlmsg .= "<IMAGE source=\"HTTP\" type=\"\$http_type\"> \$http_location</IMAGE>\n";
    $xmlmsg .= "<CONTENT>";
    $xmlmsg .= urlencode("Your logo"); // hex-encode the content
    $xmlmsg .= "</CONTENT>\n";
    $xmlmsg .= "</NOKIAMULTIPART>\n";

    $rcbuf = sendXML($NUMBER, $truesenses_origin, $xmlmsg);

    if ($debug == 1) {
        echo("RESPONSE::$rcbuf:\n");
    }
} // NOKIAMULTIPART

if (strcmp(strtoupper($MSGONLY), "EMSLOGO") == 0) {

    // sends a logo to NON-nokia, EMS-capable phones like SonyEricsson
    //
    // parameters:
    // autosave = YES/NO (automatically ask user to save image)
    // protected = YES/NO (disallow re-distribution of image)
    // width = x (width of image, do not provide parameter for autosizing)
    // height = y (height of image, do not provide parameter for autosizing)

    // assemble XML message

```

```
$http_type = "GIF";
$http_location = "http://www.simmmcomm.ch/nokialogo.gif"; // location of the logo

$xmlmsg = "<EMS>\n";
// no newline after <content> tag as it would convert to text-enter %0A in ems and would consume space
$xmlmsg .= "<CONTENT><IMAGE source=\"HTTP\" type=\"$http_type\" autosave=\"YES\"
protected=\"NO\">$http_location</IMAGE></CONTENT>\n";
$xmlmsg .= "</EMS>\n";

$rcbuf = sendXML($NUMBER, $truesenses_origin, $xmlmsg);

if ($debug == 1) {
    echo("RESPONSE::$rcbuf:\n");
}
} // EMSLOGO

// send confirmation to TrueSenses.com that we have received the incoming sms
echo("OK<br>\n");

?>
</body>
</html>
```