



www.routomessaging.com
support@routotelecom.com

HTTP MMS Gateway User Guide

The information contained in this document is proprietary and copyright and for the sole purpose of informing customers about the above service. The service is owned by Routo Telecommunications Ltd, 48 Charlotte Street, London, W1T 2NS, United Kingdom.

Preface:

Please register for [SMS/MMS and HLR Lookup account](#) for testing our SMS, MMS and HLR services and integration to our SMS, MMS APIs. The following features and services are available:

- Straightforward, SMS API, MMS API and HLR API integration
- Send and receive SMS and MMS using HTTP and SMPP and make HLR requests using HTTP
- Free sample code on how to send/receive SMS and MMS and make HLR lookups
- Free 24 hours support; our support to answer any questions
- Minimal SMS, MMS and HLR development time
- Deploy with confidence; we have over 6 years of experience in Text Messages Integration
- Sending and Receiving (with delivery reports) of SMS and MMS

Table of contents:

1	CHANGE HISTORY	4
2	INTRODUCTION	5
3	SENDING MMS USING ASP	6
3.1	MMS content is located on Customers Internet Web Servers	6
3.2	MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway	7
4	SENDING MMS USING C#	9
4.1	MMS content is located on Customers Internet Web Servers	9
4.2	MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway	11
5	SENDING MMS USING PERL	13
5.1	MMS content is located on Customers Internet Web Servers	13
5.2	MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway	13
6	SENDING MMS USING PHP	16
6.1	MMS content is located on Customers Internet Web Servers	16
6.2	MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway	17
7	SENDING MMS USING JAVA	19
7.1	MMS content is located on Customers Internet Web Servers	19
7.2	MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway	21

List of code snippets:

Snippet 1: Sending MMS using ASP – the first method	6
Snippet 2: Sending MMS using ASP – the second method	7
Snippet 3: Example of hex encoding	8
Snippet 4: MMS content is located on Customers Internet Web Servers	9
Snippet 5: MMS-gif C# example	11
Snippet 6: Hex encoding example: The GIF image	12
Snippet 7: Sending MMS using Perl – the first method	13
Snippet 8: Sending MMS using Perl – the second method	14
Snippet 9: Hex encoding example	14
Snippet 10: Example of HEX encoding	15
Snippet 11: Sending MMS using PHP – the first method	16
Snippet 12: Sending MMS using PHP – the second method	17
Snippet 13: Hex encoding example: The GIF image	17
Snippet 14: Example of HEX encoding for the particular GIF file	18
Snippet 15: Sending MMS using Java when the message is located on the server	19
Snippet 16: MMS content using Java submitted to RoutoMessaging SMS Gateway	22

List of tables:

Table 1: Change history	4
Table 2: MMS message body parameters	7
Table 3: Required parameters for the message body	10
Table 4: Message body parameters	13
Table 5: message body parameters	16
Table 6: required parameters for the message body	20

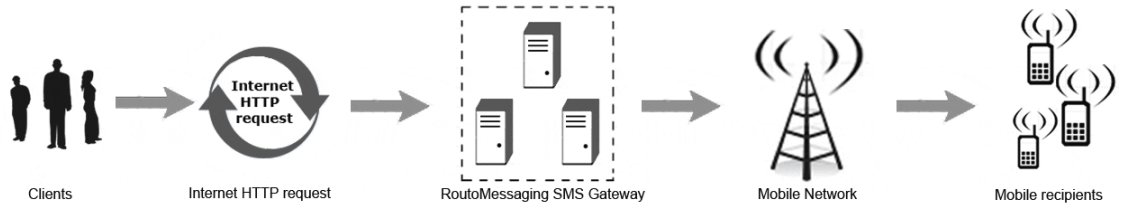
1 Change History

Date	Changes	Section
2000-09-09	Initial release	All

Table 1: Change history

2 Introduction

This document contains detailed information about the methods for implementing the RotoMessaging MMS service through HTTP protocol.



For more details or examples not included in this user guide please contact us through email at support@routotelecom.com or live chat available at www.routomessaging.com.

RotoMessaging provides the following two servers for sending messages through **HTTP** protocol:

- **sm5c5.routotelecom.com**
- **sm5c6.routotelecom.com**

A detailed description of the connection parameters is available in the document "Connecting to RotoMessaging.pdf" at: <http://www.routomessaging.com/sms-api.pmx> In order to send the SMS messages to the RotoMessaging SMS Gateway the customer is required to have the following:

- an SMS account with RotoMessaging
- available credit on the SMS account with RotoMessaging

Note: The servers are not limited to any platforms/languages. In this document we have provided typical examples for connecting to our SMS gateway and sending text messages using ASP, PHP, Perl and JAVA scripting languages in order to get you started.

3 Sending MMS using ASP

The RoutoMessaging SMS Gateway supports 2 methods to send MMS:

- **MMS content is located on Customers Internet Web Servers** – see section 3.1
- **MMS content is submitted to RoutoMessaging SMS Gateway** - see section 3.2

3.1 MMS content is located on Customers Internet Web Servers

The first method is for the case when the MMS content is located on Customers Internet Web Servers. If this method is used, the notification message containing the URL of the MMS content is sent to the handset in order to initiate MMS download (see the ASP example below).

```
<%
dim from
dim subject
dim messClass
dim messSize
dim messExpiry
dim expiryType
dim messUrl
' creating object
set Sms = Server.CreateObject("RoutoTelecom.Sender")
' setting parameters
Sms.Username = "your_username"
Sms.Password = "your_password"
Sms.MobileNo = "44791232321"
Sms.Owner = "44791232321"
Sms.SMSType = "mmsNotification"
from = "44791232321"
subject = "mms_test"
messClass = "personal"
messSize = "4516"
messExpiry = "259200"
expiryType = "relative"
messUrl = "http://www.somedomain.com/somefile.mms"
Sms.Message=from & vbCrLF & subject & vbCrLF & messClass&
vbCrLF & messSize & vbCrLF & messExpiry & vbCrLF &
expiryType & vbCrLF & messUrl
' sending SMS and printing result
SmsResult = Sms.Send
Response.Write SmsResult
set Sms = Nothing
%>
```

Snippet 1: Sending MMS using ASP – the first method

All parameters from message body are required. Here is the list of parameters with possible values:

Parameter	Description
from	the address of the sender
subject	the subject of the message
messClass	message class (possible values are: personal, informational, advertising or auto)
messSize	the size of the MMS message in bytes
messExpiry	the expiration of the MMS message on the MMSC in seconds
expiryType	the expiration type (possible values: absolute, relative)
messUrl	the location of the message up to the 255 characters length

Table 2: MMS message body parameters

3.2 MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway

The second method is for the case when the MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway for onward delivery to the handset.

Example: MMS message type is set to **mms-<file type>** and to **hex encode** the message.

Currently supported message types are:

- mms-gif
- mms-jpg
- mms-jar
- mms-wav

Below is the MMS-gif ASP example:

```
<%
' creating object
set Sms = Server.CreateObject("RoutoTelecom.Sender")
' setting parameters
Sms.Username = "your_username"
Sms.Password = "your_password"
Sms.MobileNo = "44792727272722"
Sms.Owner = "44792727272722"
Sms.SMSType = "mms-gif"
Sms.Message="47494638376148001C00800001000000FFFFFF2C00000000
48001C000008FE0003081C487020000005132A5CC850E0C18710234674887
021C289090F526CD8B022478217355AA4783163008D1E3F823C29B1E54396
2F158A2C99F2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2B
CE91365C5A12881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E6
46A04B4932749AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308
DCA94CB36E85BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68DCC
B7AA5CC1664373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F0679
25865AAAE4DF8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4BA85CF
9E2E9BF2EE9DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE8542A57B9
676DBF2C5D32B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C53F6E146E0
5AE441061E79744907604BEFE9765F7E8D4527A083B2D936D88201F2F7DB8
5201A889F561E6E681E63F2A5A8D588ED4518DD4721AAB81F47FE4D651175
D32914D0003B"
' sending SMS and printing result
SmsResult = Sms.Send
Response.Write SmsResult
set Sms = Nothing
%>
```

Snippet 2: Sending MMS using ASP – the second method

The following is the Hex encoding example:



Hex encoded:

```
47494638376148001C00800001000000FFFFFF2C0000000048001C00000  
8FE0003081C487020000005132A5CC850E0C18710234674887021C28909  
0F526CD8B022478217355AA4783163008D1E3F823C29B1E543962F158A2  
C99F2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2BCE913  
65C5A12881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E646A  
04B4932749AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308D  
CA94CB36E85BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68DC  
CB7AA5CC1664373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F0  
67925865AAAE4DF8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4B  
A85CF9E2E9BF2EE9DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE85  
42A57B9676DBF2C5D32B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C5  
3F6E146E05AE441061E79744907604BEFE9765F7E8D4527A083B2D936D8  
8201F2F7DB85201A889F561E6E681E63F2A5A8D588ED4518DD4721AAB81  
F47FE4D651175D32914D0003B
```

type=mms-gif

Snippet 3: Example of hex encoding

4 Sending MMS using C#

The RoutoMessaging SMS Gateway supports 2 methods to send MMS:

- **MMS content is located on Customers Internet Web Servers** – see section 4.1
- **MMS content is submitted to RoutoMessaging SMS Gateway** - see section 4.2

4.1 MMS content is located on Customers Internet Web Servers

The first method is when the MMS content is located on Customers Internet Web Servers. If this method is used, a notification message containing the URL of the MMS content is sent to the handset in order to initiate MMS download (see the example below).

```
public partial class sendsms : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
    RoutoSMSTelecom routo = new RoutoSMSTelecom();

    routo.SetUser("your_username");
    routo.SetPass("your_password");
    routo.SetNumber("44791212121212");
    routo.SetType("mmsNotification");

    string from = "44792727272722";
    string subject = "mms_test";
    string messClass = "personal";
    string messSize = "4516";
    string messExpiry = "259200";
    string expiryType = "relative";
    string messageUrl = "http://somedomain.com/somefile.mms";

    routo.SetMessage("from + "\r\n" + subject + "\r\n" +
messClass + "\r\n" + messSize + "\r\n" + messExpiry + "\r\n" +
expiryType + "\r\n" + messageUrl");

    string header = routo.Send();

    sms.InnerText = header;
}
}
```

Snippet 4: MMS content is located on Customers Internet Web Servers

All parameters from the message body are required. The following is the list of parameters with possible values.

parameter	description
from	the address of the sender
subject	the subject of the message
messClass	the message class; the possible values are: <ul style="list-style-type: none"> • personal • informational • advertising • auto
messSize	the size of the MMS message in bytes
messExpiry	the expiration of the MMS message on the MMSC in seconds
expiryType	the expiration type; the possible values are: <ul style="list-style-type: none"> • absolute • relative
expiryType	the expiration type; the possible values are: <ul style="list-style-type: none"> • absolute • relative
messUrl	the location of the message up to the 255 characters length

Table 3: Required parameters for the message body

4.2 MMS content is hex encoded and submitted to RotoMessaging SMS Gateway

The second method is when the MMS content is hex encoded and submitted to RotoMessaging SMS Gateway for onward delivery to the handset.

Example: MMS message type is set to **mms-<file type>** and to **hex encode** the message.

Currently supported message types are:

- mms-gif
- mms-jpg
- mms-jar
- mms-wav

Below is the MMS-gif MMS-gif C# example:

```
public partial class sendsms : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        RotoSMSTelecom routo = new RotoSMSTelecom();

        routo.SetUser("your_username");
        routo.SetPass("your_password");
        routo.SetNumber("44791212121212");
        routo.SetType("mms-gif");

        routo.SetMessage("47494638376148001C00800001000000FFFFFF2C000000004
8001C00000
8FE0003081C487020000005132A5CC850E0C18710234674887021C289090F
526CD8B022478217355AA4783163008D1E3F823C29B1E543962F158A2C99F
2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2BCE91365C5A1
2881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E646A04B49327
49AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308DCA94CB36E8
5BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68DCCB7AA5CC1664
373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F067925865AAAE4D
F8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4BA85CF9E2E9BF2EE9
DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE8542A57B9676DBF2C5D3
2B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C53F6E146E05AE441061E7
9744907604BEFE9765F7E8D4527A083B2D936D88201F2F7DB85201A889F56
1E6E681E63F2A5A8D588ED4518DD4721AAB81F47FE4D651175D32914D0003
B");

        string header = routo.Send();
        sms.InnerText = header;
    }
}
```

Snippet 5: MMS-gif C# example

Hex encoding example: The GIF image:

Hex encoded:

```
47494638376148001C00800001000000FFFFFF2C0000000048001C00000  
8FE0003081C487020000005132A5CC850E0C18710234674887021C28909  
0F526CD8B022478217355AA4783163008D1E3F823C29B1E543962F158A2  
C99F2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2BCE913  
65C5A12881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E646A  
04B4932749AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308D  
CA94CB36E85BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68DC  
CB7AA5CC1664373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F0  
67925865AAAE4DF8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4B  
A85CF9E2E9BF2EE9DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE85  
42A57B9676DBF2C5D32B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C5  
3F6E146E05AE441061E79744907604BEFE9765F7E8D4527A083B2D936D8  
8201F2F7DB85201A889F561E6E681E63F2A5A8D588ED4518DD4721AAB81  
F47FE4D651175D32914D0003B  
type=mms-gif
```

Snippet 6: Hex encoding example: The GIF image

5 Sending MMS using Perl

The RoutuMessaging SMS Gateway supports 2 methods for sending an MMS:

- **MMS content is located on Customers Internet Web Servers** – see section 5.1
- **MMS content is submitted to RoutuMessaging SMS Gateway** - see section 5.2

5.1 MMS content is located on Customers Internet Web Servers

The first method is for the case when the MMS content is located on Customers Internet Web Servers.

If this method is used, the notification message containing the URL of the MMS content is sent to the handset in order to initiate MMS download (see the example below).

```
$from = "447927272722";
$subject = "mms_test";
$messClass = "personal";
$messSize = "4516"; //message size in bytes
$messExpiry = "259200"; //expiration time in seconds
$expiryType = "relative";
$messUrl =
"http://www.somedomain.com/somefile.mms";
# setting up message body
"$from\r\n$subject\r\n$messClass\r\n$messSize\r\n$messExpiry\r\n$expiryType\r\n$messUrl";
my $response = $ua->post(
'http://smc5.routotelecom.com/cgi-bin/SMSsend',
{
number => $number,
user => $username,
pass => $password,
type => 'mmsNotification',
message => $message
}
);
```

Snippet 7: Sending MMS using Perl – the first method

All parameters from the message body are required. The following is the list of parameters with possible values.

Parameter	Description
from	the address of the sender
subject	the subject of the message
messClass	message class (possible values are: personal, informational, advertising or auto)
messSize	the size of the MMS message in bytes
messExpiry	the expiration of the MMS message on the MMSC in seconds
expiryType	the expiration type (possible values: absolute, relative)
messUrl	the location of the message up to the 255 characters length

Table 4: Message body parameters

5.2 MMS content is hex encoded and submitted to RoutuMessaging SMS Gateway

The second method is for the case when the MMS content is hex encoded and submitted to RoutuMessaging SMS Gateway for onward delivery to the handset.

Example: MMS message type is set to **mms-<file type>** and to **hex encode** the message.

Currently supported message types are:

- mms-gif
- mms-jpg
- mms-jar
- mms-wav

Below is the MMS-gif Perl example.

```
my $response = $ua->post(
    'http://smsc5.routotelecom.com/cgi-bin/SMSsend',
    {
        number => $number,
        user => $username,
        pass => $password,
        type => 'mms-gif',
        message =>
        '47494638376148001C00800001000000FFFFFF2C0000000048001C0000
        08FE0003081C487020000005132A5CC850E0C18710234674887021C2890
        90F526CD8B022478217355AA4783163008D1E3F823C29B1E543962F158A
        2C99F2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2BCE91
        365C5A12881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E646
        A04B4932749AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308
        DCA94CB36E85BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68D
        CCB7AA5CC1664373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F
        067925865AAAE4DF8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4
        BA85CF9E2E9BF2EE9DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE8
        542A57B9676DBF2C5D32B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C
        53F6E146E05AE441061E79744907604BEFE9765F7E8D4527A083B2D936D
        88201F2F7DB85201A889F561E6E681E63F2A5A8D588ED4518DD4721AAB8
        1F47FE4D651175D32914D0003B'
    }
);
```

Snippet 8: Sending MMS using Perl – the second method

Hex encoding example:



The GIF image:

```
Hex encoded:
47494638376148001C00800001000000FFFFFF2C0000000048001C00000
8FE0003081C487020000005132A5CC850E0C18710234674887021C28909
0F526CD8B022478217355AA4783163008D1E3F823C29B1E543962F158A2
C99F2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2BCE913
65C5A12881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E646A
04B4932749AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308D
CA94CB36E85BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68DC
CB7AA5CC1664373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F0
67925865AAAE4DF8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4B
A85CF9E2E9BF2EE9DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE85
42A57B9676DBF2C5D32B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C5
3F6E146E05AE441061E79744907604BEFE9765F7E8D4527A083B2D936D8
8201F2F7DB85201A889F561E6E681E63F2A5A8D588ED4518DD4721AAB81
F47FE4D651175D32914D0003B
type=mms-gif
```

Snippet 9: Hex encoding example

The Hex encoding is simple. Below is the small script which HEX encodes any file given as first parameter. The Output file is the second parameter.

```
#!/usr/bin/perl
open(F, "<".$ARGV[0]) or die($!);
open(O, ">".$ARGV[1]) or die($!);
while(<F>){
```

```
s/({1})/sprintf("%02X", ord($1))/ges;  
print O $_;  
}  
close F;  
close O;
```

Snippet 10: Example of HEX encoding

6 Sending MMS using PHP

The RoutoMessaging SMS Gateway supports 2 methods for sending an MMS:

- **MMS content is located on Customers Internet Web Servers** – see section 6.1
- **MMS content is submitted to RoutoMessaging SMS Gateway** - see section 6.2

6.1 MMS content is located on Customers Internet Web Servers

The first method is for the case when the MMS content is located on Customers Internet Web Servers. If this method is used, the notification message containing the URL of the MMS content is sent to the handset in order to initiate MMS download (see the PHP example below).

```
<?php
include("RoutoTelecomSMS.php");
$sms = new RoutoTelecomSMS;
$sms->SetUser("your_username");
$sms->SetPass("your_password");
$sms->SetNumber("447927272722");
$sms->SetType("mmsNotification");
$from = "447927272722";
$subject = "mms_test";
$messClass = "personal";
$messSize = "4516"; //message size in bytes
$messExpiry = "259200"; //expiration time in seconds
$expiryType = "relative";
$messUrl =
"http://www.somedomain.com/somefile.mms";
// setting up message body
$sms->SetMessage("$from\r\n$subject\r\n$messClass\r\n$messSize\r\n$messExpiry\r\n$expiryType\r\n$messUrl");
// send message and print result
$smsresult = $sms->Send();
print $smsresult;
?>
```

Snippet 11: Sending MMS using PHP – the first method

All parameters from the message body are required. The following is the list of parameters with possible values.

Parameter	Description
from	the address of the sender
subject	the subject of the message
messClass	message class (possible values are: personal, informational, advertising or auto)
messSize	the size of the MMS message in bytes
messExpiry	the expiration of the MMS message on the MMSC in seconds
expiryType	the expiration type (possible values: absolute, relative)
messUrl	the location of the message up to the 255 characters length

Table 5: message body parameters

6.2 MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway

The second method is for the case when the MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway for onward delivery to the handset.

Example: MMS message type is set to **mms-<file type>** and to **hex encode** the message.

Currently supported message types are:


- mms-gif
- mms-jpg
- mms-jar
- mms-wav

Below is the MMS-gif PHP example:

```
<?php
include("RoutoTelecomSMS.php");
$sms = new RoutoTelecomSMS;
$sms->SetUser("your_username");
$sms->SetPass("your_password");
$sms->SetNumber("447927272722");
$sms->SetType("mms-gif");
$sms->SetMessage
("47494638376148001C00800001000000FFFFFF2C0000000048001C00000
8FE0003081C487020000005132A5CC850E0C18710234674887021C289090F
526CD8B022478217355AA4783163008D1E3F823C29B1E543962F158A2C99F
2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2BCE91365C5A1
2881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E646A04B49327
49AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308DCA94CB36E8
5BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68DCCB7AA5CC1664
373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F067925865AAAE4D
F8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4BA85CF9E2E9BF2EE9
DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE8542A57B9676DBF2C5D3
2B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C53F6E146E05AE441061E7
9744907604BEFE9765F7E8D4527A083B2D936D88201F2F7DB85201A889F56
1E6E681E63F2A5A8D588ED4518DD4721AAB81F47FE4D651175D32914D0003
B");
// send message and print result
$smsresult = $sms->Send();
print $smsresult;
?>
```

Snippet 12: Sending MMS using PHP – the second method

The following is the Hex encoding example: The GIF image:



```
Hex encoded:
47494638376148001C00800001000000FFFFFF2C0000000048001C00000
8FE0003081C487020000005132A5CC850E0C18710234674887021C28909
0F526CD8B022478217355AA4783163008D1E3F823C29B1E543962F158A2
C99F2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2BCE913
65C5A12881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E646A
04B4932749AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308D
CA94CB36E85BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68DC
B7AA5CC1664373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F0
67925865AAAE4DF8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4B
A85CF9E2E9BF2EE9DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE85
42A57B9676DBF2C5D32B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C5
3F6E146E05AE441061E79744907604BEFE9765F7E8D4527A083B2D936D8
8201F2F7DB85201A889F561E6E681E63F2A5A8D588ED4518DD4721AAB81
F47FE4D651175D32914D0003B
type=mms-gif
```

Snippet 13: Hex encoding example: The GIF image

The following is the small script for hex encoding of the particular GIF file:

```
<?php
// URL of some GIF file for sending
$imgurl = "http://www.domain.com/some_logo.gif";
// reads GIF using standard PHP file system functions
$gifFile = fopen( $imgurl, "r" );
if (!$gifFile) exit;
$gifContent = "";
while ( !feof($gifFile) ) {
    $gifContent .= fread( $gifFile, 1024 );
}
fclose( $gifFile );
// hex encode GIF data
$encodedImage='';
for ($i=0; $i < strlen($gifContent); $i++)
{
    $encodedImage .= sprintf("%02X",(ord($gifContent[$i])));
}
?>
```

Snippet 14: Example of HEX encoding for the particular GIF file

7 Sending MMS using Java

The RoutoMessaging SMS Gateway supports 2 methods to send MMS:

- **MMS content is located on Customers Internet Web Servers** – see section 7.1
- **MMS content is submitted to RoutoMessaging SMS Gateway** - see section 7.2

7.1 MMS content is located on Customers Internet Web Servers

The first method is when the MMS content is located on Customers Internet Web Servers. If this method is used, a notification message containing the URL of the MMS content is sent to the handset in order to initiate MMS download (see the example below)

```
<html>
  <head>
    <title>Routo</title>
  </head>
  <body>
    <h1>Routo Telecom</h1>
    <br/><br/>
<!-- Include Routo SMS class-->
<jsp:useBean id="sms" class="routosms.RoutoSMSTelecom" />
<!-- Setting SMS parameters-->
<% routosms.RoutoSMSTelecom routo = new
routosms.RoutoSMSTelecom();%> <!-- creating object>
<% routo.SetUser("your_username");%>
<% routo.SetPass("your_password");%>
<% routo.SetNumber("44792727272722");%>
<% routo.SetType("mmsNotification");%>
<% String messageUrl = "http://www.somedomain.com/somefile.mms";%>
<% String from = "44792727272722"; %>
<% String subject = "mms-test"; %>
<% String messClass = "personal";%>
<% String messSize = "4516"; %>
<% String messExpiry = "259200"; %>
<% String expiryType= "relative"; %>
<% routo.SetMessage(from + "\r\n" + subject + "\r\n" + messClass +
"\r\n" + messSize + "\r\n" + messExpiry + "\r\n" + expiryType +
"\r\n" + messageUrl);%>
<!-- Send SMS and print result-->
<%=routo.Send()%>
  </body>
</html>
```

Snippet 15: Sending MMS using Java when the message is located on the server

All parameters from the message body are required. The following is the list of parameters with possible values.

parameter	description
from	the address of the sender
subject	the subject of the message
messClass	the message class; the possible values are: <ul style="list-style-type: none"> • private • informational • advertising • auto
messSize	the size of the MMS message in bytes
messExpiry	the expiration of the MMS message on the MMSC in seconds
expiryType	the expiration type; the possible values are: <ul style="list-style-type: none"> • absolute • relative
expiryType	the expiration type; the possible values are: <ul style="list-style-type: none"> • absolute • relative
messUrl	the location of the message up to the 255 characters length

Table 6: required parameters for the message body

7.2 MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway

The second method is when the MMS content is hex encoded and submitted to RoutoMessaging SMS Gateway for onward delivery to the handset.

Example: MMS message type is set to **mms-<file type>** and to **hex encode** the message.

Currently supported message types are:

- mms-gif
- mms-jpg
- mms-jar
- mms-wav

Below is the MMS-gif Java example:

```
<html>
  <head>
    <title>Routo</title>
  </head>
  <body>
    <h1>Routo Telecom</h1>
    <br/><br/>
    <!-- Include Routo SMS class-->
    <jsp:useBean id="sms" class="routosms.RoutoSMSTelecom" />
    <!-- Setting SMS parameters-->
    <% routosms.RoutoSMSTelecom routo = new
    routosms.RoutoSMSTelecom();%> <!-- creating object-->
    <% routo.SetUser("your_username");%>
    <% routo.SetPass("your_password");%>
    <% routo.SetNumber("44792727272722");%>
    <% routo.SetType("mms-gif");%>
    <%
    routo.SetMessage("47494638376148001C00800001000000FFFFFF2C00000000
    48001C00000
    8FE0003081C487020000005132A5CC850E0C18710234674887021C289090F
    526CD8B022478217355AA4783163008D1E3F823C29B1E543962F158A2C99F
    2A4CD922A57C274393124C68D333DC6F499D3A0C39B4869E2BCE91365C5A1
    2881163D4A33A955A12CB3F6C499F269CD8C5DA5561D5B70E646A04B49327
    49AD5E6D5AB698F32051BD2E0D795505F92DDEB16AFD2B2308DCA94CB36E8
    5BB220EB7A05ABD3A462BD7A0FCF0DBBD5AE60CB8ED5B68DCCB7AA5CC1664
    373DD5A377161C969BD8AB42CBAEC6288AE9DF294D837F067925865AAAE4D
    F8E7DA98B757AB4D7D972D66A6774D03776BB6F6D3C4BA85CF9E2E9BF2EE9
    DB73B7EAD09DB6E50C8AE7B6C733F1B5EF865ECDE8542A57B9676DBF2C5D3
    2B9FBF7C7EFBE4E107A3C74DBF3EE3E5DB75955C53F6E146E05AE441061E7
    9744907604BEFE9765F7E8D4527A083B2D936D88201F2F7DB85201A889F56
    1E6E681E63F2A5A8D588ED4518DD4721AAB81F47FE4D651175D32914D0003
    B");%>
```

```
<!-- Send SMS and print result-->  
<%=routo.Send()%>  
  </body>  
</html>
```

Snippet 16: MMS content using Java submitted to RoutoMessaging SMS Gateway

For more details or examples not included in this user guide please contact us through email at support@routotelecom.com or live chat available at www.routomessaging.com.