Volume

1

CELLTRUST® CORPORATION

Software Development Kit

Application Programming Interface

CELLTRUST CORPORATION

Application Programming interface

20701 N. Scottsdale Rd • Suite 107-451
Scottsdale, AZ 85255
Phone 480.515.5200 • Fax 480.699.9491

Table of Contents

Chapter

1

Introduction

CellTrust Secure Gateway is a Secure SMS, standard SMS, email and voice gateway that facilitates communication between your application and carriers, mobile handsets, landlines, and Email recipients around the world.

The purpose of this manual is to document all Application Programming Interfaces (APIs) provided by CellTrust Secure gateway. An Application Programming Interface allows your programs to connect to CellTrust Secure Gateway and utilize one or more of its functionalities such as:

- Send/Receive secure text message
- Send/Receive text message
- Verify a mobile number
- Lookup carrier of a phone number
- Text to voice and IVR
- Send Emails campaigns
- Track a flight

In this chapter we will provide an overview of CellTrust Secure Gateway. For a complete definition of terms used in the mobile industry please refer to Appendix A.

Send/Receive secure text message

Please refer to HTTP(S), Web Services, and JAVA SDK for more information about this API.

Send/Receive text message

Please refer to HTTP(S), Web Services, and JAVA SDK for more information about this API.

Verify a mobile number

Please refer to HTTP(S) for more information about this API.

Lookup carrier of a phone number

Please refer to HTTP(S) for more information about this API.

Text to voice and IVR

Please refer to HTTP(S), Web Services, and JAVA SDK for more information about this API.

Send Emails campaigns

Please refer to HTTP(S), Web Services, and JAVA SDK for more information about this API.

Track a flight

Please refer to Web Services for more information about this API.

Terms and Conditions

By using this manual and all services described in this manual, you agree to be bound by terms and conditions specified in the Terms and Conditions at the end of this manual. Please read the Terms and Conditions section before you proceed.

CellTrust SMS Gateway

An SMS Gateway facilitates communication between your application and carriers around the world. CellTrust Gateway was designed to accomplish this as effortlessly as possible. With a single interface and one time

integration, we aim to have your application up and testing, when possible, within minutes.

SMS Gateway functionality

SMS Gateways are connected to carriers' networks on one side, and software applications on the other side.

A request that contains a message and a phone number can be submitted to a gateway (through its API), and the gateway will find the country and carrier network associated with the phone number. The message is then passed to the respective carrier and sent on to the recipient's phone number. The diagram below demonstrates the message flow between a typical application, CellTrust SMS Gateway and mobile users around the world:

CellTrust SMS Gateway is also capable of processing messages sent from mobile handsets: A message will be sent to a mobile carrier from the handset, it will be routed to the CellTrust Gateway. The CellTrust Gateway either pushes it back to an application or stores it in a database and provides access to it to be viewed using CellTrust Gateway web portal or retrieved via API. Adding Mobile Originated Message processing makes the gateway '2-Way SMS Gateway'.

CellTrust SMS Gateway Benefits

- The only gateway in North America providing the option of SMS, SecureSMS, Data, Voice and Email
- Fully equipped with ready to roll-out campaign management modules: Use our powerful secure website and cutting-edge

mobile marketing tools to help you design, execute, measure, report and enhance the capability of your mobile strategy

- Redundant Gateway: CellTrust addresses the single point of failure challenge with a unique redundant gateway
- Global coverage with one integration: A onetime integration connects you to virtually any phone number across 700 carriers in over 218 countries
- Support of Secure SMS: CellTrust SMSGateway can be upgraded to CellTrust SecureSMS Gateway, the first of its kind globally, featuring government standard encryption and end-to-end privacy with receive/receipt confirmation to over 218 countries and 700 carriers
- Benchmarking Gateways : A comparison of SMS Gateways across North America

Using CellTrust SMS Gateway

Connecting to CellTrust 2-Way SMS Gateway is straightforward: Based on the technology you have used within your own application, you can choose from several of the mobile industry's most popular APIs to send and receive messages: HTTP/S, Java, .NET, Web Services, or SMTP. You can also use our powerful web tools and plug and play scripts without the need for developing any code to send or receive messages.

Getting Started

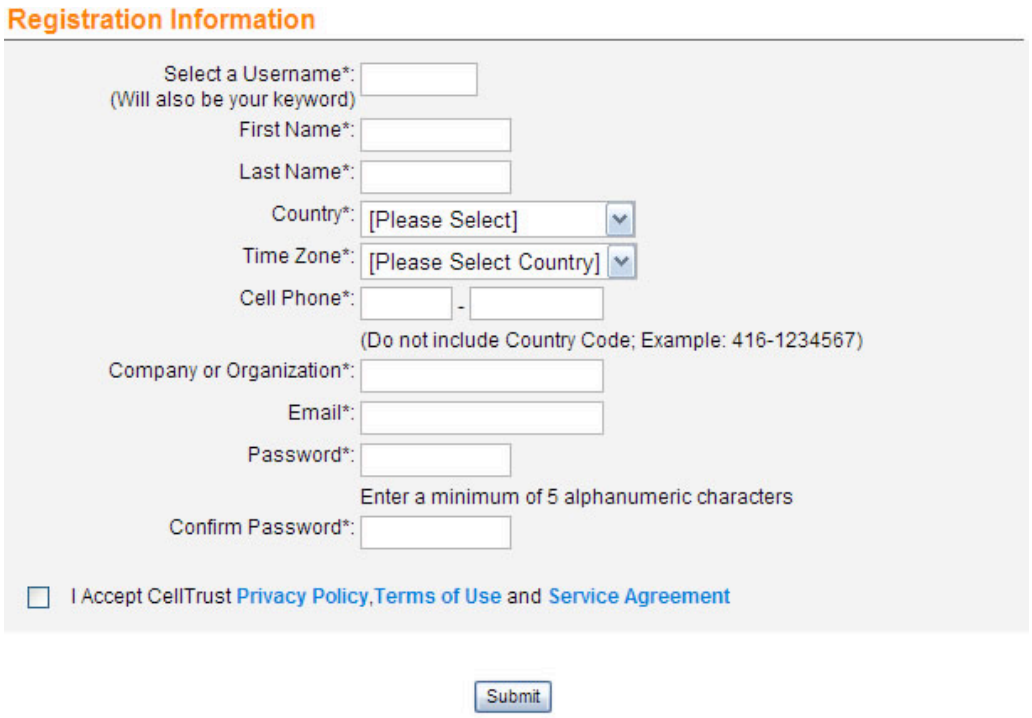
This chapter describes how to start utilizing CellTrust SMS Gateway APIs.

Overall Steps

1. Register for a new account
2. Select the appropriate protocol
3. White list your server on the CellTrust SMS Gateway
4. Integrate selected API with your application
5. Test MO and MT messages

Register for a new account

Registering for an account is free and simple. Go to http://www.celltrust.com and click on “Free Gateway Trails”. CellTrust will provide free test credits for developers to try out sending and receiving messages.



The username is also used as your default keyword. All messages that are sent through the shared short code start with a keyword. The keyword helps the recipient to identify different programs running on a shared short code. Besides the default keyword, users can select other keywords.

Note

Once username is created it can be used for both APIs and logging into the CellTrust Gateway portal.

Note

Please see Glossary for definition of terms that you may not be familiar with such as "keyword" or "short code".

The Cell phone number is required because the CellTrust SMS gateway sends confirmation of password changes to the registered mobile number on the account. The email address provided during registration is used for all alerts and reports.

Select the appropriate Protocol

Once your account is activated by Sales, the next is to select the appropriate API. CellTrust® provides the following APIs:

- HTTP and HTTPS
- Web Services
- Java
- SMTP

The simplest protocol to use is HTTP for integrating with PHP, HTML, .Net, and JAVA application. .Net developers can also use CellTrust® Web Service API. Java application can integrate by simply using CellTrust Java SDK. CellTrust® also provided Email to SMS integration using SMTP protocol. Once the suitable protocol is selected you refer to the appropriate section of this manual for details relating to your selected protocol.

White list your Server

The most affective way to secure your account is to limit the number of servers that can use it. This is achieved by registering your server with the CellTrust SMS Gateway. The process for white listing a server is:

1. Login to CellTrust Gateway portal
2. Click on My Gateway and then on Account Module
3. Enter your IP addresses in Server Validation Section. Use comma to separate IP addresses if you need to white list more than one server.

Figure 2-1 White List your server

Note

Make sure to White List all servers communicating with CellTrust SMS Gateway. Remove all spaces between IP Addresses and commas.

If you are not sure of your IP address, you can use web sites such as http://whatismyipaddress.com to find out your IP address. Make sure you go to this site from the server you want to White List.

CellTrust Gateway Key Components

This chapter discusses key features of the CellTrust Gateway. This chapter will provide the context to help you use CellTrust Gateway APIs.

Templates

Templates play an important role in the CellTrust Gateway and application. Highly flexible, they can be used for several purposes: for frequently used messages, pulling and processing data fields (in the form of tags) for integration to third party systems, and also for message personalization - whereby each recipient can see his or her name in the message.

How does it work?

CellTrust Gateway provides online tools to create and save templates in the system. Each template can be loaded using the CellTrust Gateway Web Portal or can be utilized via API.

For example, if you want to integrate SMS to your "Contact Us" form, you can easily create a template online and for each field in your "Contact Us" form create one tag in the template. Each time a Contact Us form is submitted by a user, you can capture all the data in the fields and it is automatically passed to the gateway. The values are placed in the tags with data and sent to you via SMS.

Premium SMS

Premium SMS is when mobile subscribers are billed for third party content and services directly on their cell phone bill by their wireless carrier. Revenues are split between the content provider and the carrier.

How does Premium SMS Work?

Premium SMS is a highly effective micro-payment solution because it is seamless from the mobile end users' perspective. Mobile subscribers are required to double-opt-in as part of the transaction process and text message. Asked if they want to subscribe or download content or services

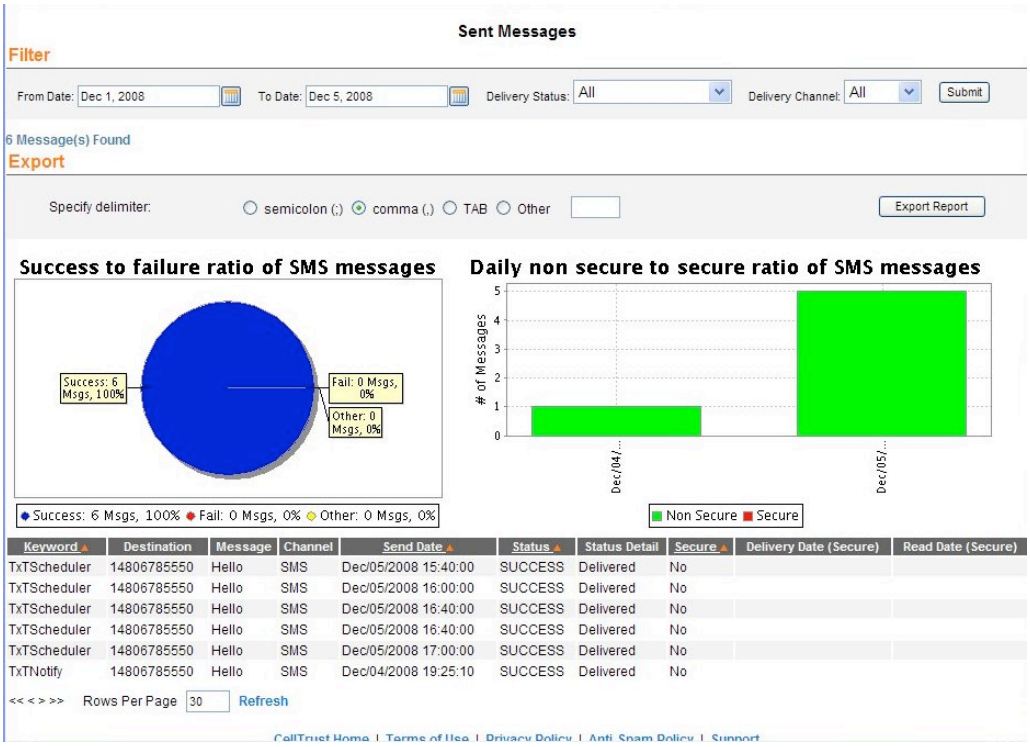
for a specific price requiring confirmation with Yes or No, they are then asked again for another round of Yes or No confirmation - thus the term "double opt-in". The regular billing cycle is based on a calendar month and payment of premium rate revenue is normally made within 45-60 days.

Using CellTrust Premium SMS

Turn-key, commercial-grade CellTrust Premium SMS is ready to roll out on several pre-approved short codes without long regulatory delays. A double-opt-in module is built in for ease of use and regulatory compliance. Broadcast real-time content to large numbers of mobile subscribers, including text, audio, video, images, J2ME/BREW apps, Flashlite, and voice calls (either text-to-speech or pre-recorded) directly to your customers' mobiles with complete double opt-in capability and mobile billing through carriers (pre-approved for ringtones in the USA). Content can be sent to individual phone numbers, or keywords, or groups of keywords. Multimedia content can be uploaded and triggered or scheduled for later download via our WAP tools.

Reports

Whether you use APIs or CellTrust Gateway Web Portal to send your messages, you use reports provided on the Web Portal to audit and troubleshoot your messages.


Sent Messages
Filter
From Date: Dec 1, 2008
To Date: Dec 5, 2008
Delivery Status: All
Delivery Channel: All
Submit
6 Message(s) Found
Export
Specify delimiter:
semicolon (;)
comma (,)
TAB
Other
Export Report
Success to failure ratio of SMS messages
Success: 6 Msgs, 100%
Fail: 0 Msgs, 0%
Other: 0 Msgs, 0%
Success: 6 Msgs, 100% Fail: 0 Msgs, 0% Other: 0 Msgs, 0%
Daily non secure to secure ratio of SMS messages
# of Messages
Dec/04/...
Dec/05/...
Non Secure
Secure
Keyword
Destination
Message
Channel
Send Date
Status
Status Detail
Secure
Delivery Date (Secure)
Read Date (Secure)
TxTScheduler 14806785550 Hello SMS Dec/05/2008 15:40:00 SUCCESS Delivered No
TxTScheduler 14806785550 Hello SMS Dec/05/2008 16:00:00 SUCCESS Delivered No
TxTScheduler 14806785550 Hello SMS Dec/05/2008 16:40:00 SUCCESS Delivered No
TxTScheduler 14806785550 Hello SMS Dec/05/2008 16:40:00 SUCCESS Delivered No
TxTScheduler 14806785550 Hello SMS Dec/05/2008 17:00:00 SUCCESS Delivered No
TxTNotify 14806785550 Hello SMS Dec/04/2008 19:25:10 SUCCESS Delivered No
<< < > >>
Rows Per Page 30
Refresh
CellTrust Home | Terms of Use | Privacy Policy | Anti-Spam Policy | Support

Chapter

2

HTTP(S) APIs

This chapter describes all HTTP and HTTPS APIs

The following format is used throughout this document:

<…>	Required parameters
{…}	Optional parameters
[…]	Value of parameter

Send Secure and Regular SMS

Purpose

To send Secure SMS or standard SMS message to one or more people.

Syntax

```
https://gateway.celltrust.net/TxTNotify/SecureSMS?
<SMSDestination=[Phone #]&
Username=[username]&
Password=[password]&
CustomerNickname=[nickname]&
CarrierId=[carrier ID]&
Message=[message]>
{ AllowInsecure=[true|false]&
DeliveryAck=[true|false]&
ReadAck=[true|false]&
Shortcode=[shortcode]}
```

-or-

http://gateway.celltrust.net/TxTNotify/SecureSMS?
<SMSDestination=[Phone #]&
Username=[username]&
Password=[password]&
CustomerNickname=[nickname]&
CarrierId =[carrier ID]&
Message=[message]>
{ AllowInsecure=[true|false]&
DeliveryAck=[true|false]&
ReadAck=[truelfalse]&
Shortcode=[shortcode]}>

Description

This API can be used to send Secure and regular SMS messages. Secure SMS messages require that the destination phone have a SecureSMS agent. Regular SMS can be sent to any mobile phone capable of receiving SMS messages.

Mandatory Parameters

SMSDestination

The destination mobile phone number that will receive the message. Note that it must include country code and only digits are accepted, for example:

14805551212 for US and Canada

342475551212 for international

Username

The user name is used to authenticate your account. The same username as created during registration.

Password

The password is used to authenticate your account. The same password as created during registration.

Message

The message to be delivered to the handset.

Optional Parameters

CarrierID

Carriers are identified by a unique carrier ID. For a list of carrier IDs, please refer to the Carrier Codes table on page 28. Though CarrierID is an optional parameter for MTs, it is strongly recommended that you provide the CarrierID of the Phone number in the API call.

*Note: You will be charged an additional fraction of credit for every message you send without the CarrierID. You may use Preview API to determine the carrier for a mobile phone number.

Shortcode

It specifies the short code to be used to deliver the message. Relevant only if you have more than one dedicated short code.

AllowInsecure

It defines if the message can be delivered as regular SMS in case secure agent is not installed on the handset. Possible values are “true” or “false”. If omitted, default value is “true”.

DeliveryAck

It defines if secure agent should attempt to respond with confirmation upon receiving the message. Possible values are “true” or “false”. If omitted, default value is “false”.

ReadAck

It defines if secure agent should attempt to respond with confirmation after user opens the message. Possible values are “true” or “false”. If omitted, default value is “false”.

XMLResponse

Set this parameter to true to get a XML response. By default, this API responds with a text string. If request is accepted by the server, the following string is returned: “Message accepted for delivery.”

Response Format

The response is attached to HTTP Response that is returned synchronously for every HTTP Request submitted to Secure SMS gateway. Please note that message delivery is asynchronous process that involves a number of steps. The message ID received from Secure SMS gateway means that message was accepted for delivery. It doesn't necessarily mean

the message is delivered to recipient. Use message ID returned in XML Response for further message tracking and updates of its status.

The DTD for XML Response:

```
<?xml version="1.0"?>

<!-- root element        -->
<!ELEMENT SecureSMSResponse     (Error | MessageId)>

<!-- Error is returned in case any failure happened during validation or
processing of request  -->
<!ELEMENT Error                 (ErrorCode,ErrorString)>
<!ELEMENT ErrorCode             (#PCDATA)>
<!ELEMENT ErrorString           (#PCDATA)>
<!-- In case message is accepted by gateway, Message ID is returned ( can
be used for -->
<!-- further message tracking -->
<!ELEMENT MessageId             (#PCDATA)>
```

Carrier Codes

Id	Carrier Name	Type	Id	Carrier Name	Type
1	att	0	33	immix_pc_mngmnt	0
2	t-mobile	0	34	west_central	0
3	cingular	0	35	revol	1
4	verizon	1	36	qwest_corp	1
5	sprint	1	37	nextech	1
6	nextel	3	38	alaska	0
7	alltel	1	39	all_west	1
8	dobson	3	40	breakaway	1
9	us-cellular	2	41	cox	1

10	cincinnati	2	42	farmers	0
11	cingular	2	43	gci_corp	0
12	cingular-blue	2	44	inland	1
13	cricket	1	45	illinois_valley	0
14	hickory	2	46	nntc	1
15	iowa	0	47	scutah	1
16	metro	2	48	ubet	1
17	midwest	2	49	syringa	1
18	npi	0	50	thumb	1
19	qwest	1	51	aliant	1
20	suncom	2	52	bell	1
21	centennial	1	53	fido	0
22	voce	0	54	mts	1
23	virgin_us	1	55	northerntel	1
24	boost_iden	3	56	rogers	0
25	boost_cdma	3	57	rogers	0
26	cellular_south	3	58	sasktel	1
27	ntelos	1	59	telebec	1
28	cellcom	2	60	telus	1
29	rural	2	61	virgin_ca	1
30	bluegrass_cellular	1	101	united	1
31	cellular_one_east_central	0	102	west_central	0

32	East_Kentucky	1			

***Note:** *Radio Types are defined as 0=GSM, 1=CDMA, 2=TDMA, 3=IDEN*

Notes

All communication with handset is SMS based, and no data plan is required.

Encryption increases the length of the message, so your account can be charged more credits then if the message is delivered through regular gateway.

Based on regulations, application installed on cell phone cannot initiate any message without explicit confirmation from handset owner. As so, while agent will always try to send delivery and/or read confirmations as requested by customer application, the final result will totally depend on handset owner

To ensure complete security always use HTTPS for sending Secure SMS messages

CellTrust SMS Gateway supports both GET and POST requests

Error Codes

201	wrong or missing customer nickname
204	missing destination (phone number is required)
205	missing message (message text is required)
206	security violation
207	service disabled
216	secure agent not installed on handset (identified by provided cell phone number)
401	server error

Example

https://gateway.celltrust.net/TxTNotify/SecureSMS?
Username=USERNAME&Password=PASSWORD& CarrierId =5&
SMSDestination=16024038599&CustomerNickname=NICKNAME&
Message=Hello&AllowInsecure=false&ReadAck=true

The above example sends message “Hello” to phone number 16024038599. The carrier is sprint whose carrier ID is 5.If the phone doesn’t have an agent in this case the message would not be sent since AllowInsecure is set to false. Since ReadAck is set to true, a message is pushed back when the message is read.

Success Response Example

```
<SecureSMSResponse>
   <MessageId>AAXF505566459PM14</MessageId>
</SecureSMSResponse>
```

Error Response Example

```
<SecureSMSResponse>
   <Error>
      <ErrorCode>216</ErrorCode>
      <ErrorMessage>Secure agent not installed for the number
....</ErrorMessage>
   </Error>
</TxTNotifyResponse>
```

Sample Code

HTML Example

```
<html><body>
<form name="myform"
action="http://gateway.celltrust.net/TxTNotify/SecureSMS?">
Enroll In Text Messaging services (Enter Your Phone Number):
<input type='text' name='PhoneDestination'>
Enter  the carrier  id of the  phone number:
<input type='text' name= CarrierId >
<input type='hidden' name='Message'
value='You+are+now+enrolled'>
<input type='hidden' name='CustomerNickname'
value='YOURNICKNAME'>
<input type='hidden' name='Username' value='YOURUSERNAME'>
<input type='hidden' name='Password' value='YOURPASSWORD'>
<A href="javascript: submitform()">Submit</A>
</form>
<SCRIPT language="JavaScript">
function submitform()
{
document.myform.submit();
```

```
}
</SCRIPT>
</body></html>
```

Perl Example

```
use strict;
use LWP::UserAgent;
use HTTP::Request::Common;

my $message = '';
my $userAgent = LWP::UserAgent->new();
my $response = $userAgent->request(POST
'http://gateway.celltrust.net/TxTNotify/
SecureSMS?PhoneDestination=YOURPHONE& CarrierId
=CARRIERID&Message=test&Username=YOURUSERNAME&Pass
word=PASSWORD&CustomerNickname=YOURNICKNAME&XMLR
esponse=true');

print $response->error_as_HTML unless $response->is_success;
print $response->as_string;
```

PHP Example

```
<?php
$URL = "https://gateway.celltrust.net/TxTNotify/SecureSMS?"

//your username generally
$USERNAME = "YOUR USERNAME ";
$PASSWORD = "YOUR PASSWORD ";
$NICKNAME = "YOURNICKNAME";
//NOTE COUNTRY CODE IS REQUIRED, EXAMPLE US IS
14805551212
$DESTINATION ="YOURPHONENUMBER";
//Carrier ID of the Carrier is required.
$CARRIER= "CARRIER ID";
// If you are sending to multiple phone numbers, the carrier ID need
//not be declared separately. It can be declared this way:
//$DESTINATION ="16024038599-5,14806781400-11";
$MESSAGE = "This is a test message";
$XMLRESPONSE = "true";

$DATA  = "&PhoneDestination=".$DESTINATION. "& CarrierId
=".$CARRIER."&Message=".$MESSAGE.
```

```
"&Username=".$USERNAME."&Password=".$PASSWORD."&Custo
merNickname=".$NICKNAME."&XMLResponse=".$XMLRESPONSE;


echo do_post_request($URL, $DATA);

//should work in PHP 4 and 5
function do_post_request($url, $data, $optional_headers = null) {
  $start = strpos($url,'//')+2;
  $end = strpos($url,'/',$start);
  $host = substr($url, $start, $end-$start);
  $domain = substr($url,$end);
  $fp = pfsockopen($host, 80);
  if(!$fp) return null;
  fputs ($fp,"POST $domain HTTP/1.1\n");
  fputs ($fp,"Host: $host\n");
  if ($optional_headers) {
          fputs($fp, $optional_headers);
  }
  fputs ($fp,"Content-type: application/x-www-form-urlencoded\n");
  fputs ($fp,"Content-length: ".strlen($data)."\n\n");
  fputs ($fp,"$data\n\n");
  $response = "";
  while(!feof($fp)) {
          $response .= fgets($fp, 1024);
  }
  fclose ($fp);
  return $response;
}
 ?>
```

VB Example

```
<%@ Page Language="VB" src="httputils.cs"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>

<%

    'this example uses VB to interact with csharp object. it was originally
    'developed in c#, porting should not be difficult.  should run on basic
    '.net server.  If web service, an d smtp do not work, you can try http
    'post using this class
```

```
    'vars for message
    Dim strNickname = "YOURNICKNAME"
    Dim strUsename = "YOURUSERNAME"
    Dim strPassword = "YOURPASSWORD"
    Dim strMessage = Session("Message")
    Dim strDestination = Session("OriginatorAddress")
    Dim str CarrierId = "CARRIER ID"

    Dim strXMLResponse = "true"
    Dim strTemplate = "test"
    Dim strWAPPushURL = "http://google.com"
    Dim strRedirectURL = "http://google.com"
    Dim strAdminEmail = "admin@google.com"
    Dim strDestPort = "0"
    Dim strShortcode = Session("Shortcode")
    Dim strPricePoint = ""
    Dim strDoubleOptin = "false"

    'response returned by httppost
    Dim strResponse = ""
    'defined in c# source file
    Dim myhttputils As New httputils()
    strResponse =
httputils.HttpPost("http://gateway.celltrust.net/TxTNotify/SecureSMS?
    "PhoneDestination=" + strDestination +"&CarrierId=" + strCarrierId
+ "&Nickname=" + strNickname + "&Message=" + strMessage +
"&Username=" + strUsername + "&Password=" + strPassword +
"&XMLResponse=" + strXMLResponse)

    'check the response
    If (strResponse <> "") Then
        Response.Write("Response: " + strResponse)
    Else
        Response.Write("XML Response null or empty!")
    End If
%>

</head>
<body></body>
</html>
```

Note: you can download httputils.cs from
http://www.celltrust.com/docs/httputils.zip

Receive Secure SMS Messages

Purpose

To receive Secure Mobile Originated (MO), delivery, and read acknowledge messages.

Description

This section describes how to receive messages sent by the Micro Client. There are 3 types of messages:

- MO Messages (replies, new messages)
- Delivery acknowledge (message was delivered to Micro Client)
- Read acknowledge (message was opened in the Micro Client)

The CellTrust Gateway can be configured to push an incoming message to your server and application. This is achieved by setting a Pushback URL in the CellTrust Gateway Web Portal under SecureSMS and Preferences. In preferences, you can specify one URL for delivery and read acknowledgements and another for messages. Messages are pushed to the URL in real time as HTTP Form Post requests. The parameter section defines the parameters available in the form post.

Note: Please note that CellTrust performs a Form POST, your code cannot utilize the querystring collection, instead you need to access the Form collection. The querystring collection will work when your testing and putting the URL and parameters in your browser but when CellTrust Posts it will be a Form POST and posted data is accessed in the Form collection.

Parameters

PhoneNumber

The mobile phone number that initiated the message.

IsSecure

This is set to true if the message was delivered securely and set to false if the message was delivered in clear text.

Text

The content of the message sent from the handset.

AcceptedDate

This the date and time when the message was received by the CellTrust Gateway.

SentDate

This the date and time when the message was sent by the Micro Client.

AckType

Acknowledge message type – can be “delivered” or “read”

MessageId

Message ID of the message being acknowledged – corresponds to message ID returned by gateway after message is submitted.

AgentRequestId

In case of MO message – this is the internal id of the message submitted by micro-client (used when your application is generating ACK message)

OriginalMTMessageId

Message ID of the original secure message that is being replied by this MO message

Notes

Please note that CellTrust® performs a Form POST, your code cannot utilize the querystring collection, instead you need to access the Form collection. The querystring collection will work when your testing and putting the URL and parameters in your browser but when CellTrust® Posts it will be a Form POST and posted data is accessed in the Form collection.

All parameter names as well as the URL are case sensitive!

To ensure complete security always use HTTPS for sending SMS messages and white list your server.

Examples

Reply post:

PhoneNumber: 14805551212
IsSecure: true
Text: got test message 3
AcceptedDate: 20/Nov/08 03:54:08

Delivery Acknowledge post:

PhoneNumber: 14805551212
IsSecure: true
AckType: delivered
MessageId: AZXF2146903663PM247771
SentDate: 20/Nov/08 03:53:20
AcceptedDate: 20/Nov/08 03:53:43

Read Acknowledge post:

PhoneNumber: 14805551212
IsSecure: true
AckType: delivered
MessageId: AZXF2146903663PM247771
SentDate: 20/Nov/08 03:54:20
AcceptedDate: 20/Nov/08 03:54:45

Sample Code

PHP Example

```
<?php
//================================================
=======
//          Celltrust sms Response gateway Test script
//         Put this script in the web server directory
//================================================
=======
        error_reporting(E_ALL);
        set_time_limit(0);
        ob_implicit_flush();

       // debug variables

        $query    = $_SERVER['QUERY_STRING'];
        $uri      = $_SERVER['REQUEST_URI'];
        $remadr   = $_SERVER['REMOTE_ADDR'];
```

```
    //    create the text output

      $text  = "==================================\n\r";
   foreach (array_keys($_POST) as $key)
   {
            $$key = $_POST[$key];
            $text .= "$key is ${$key}\n\r";
   }
      $text .= "==================================\n\r";
   $text .= print_r ($_POST);


      // write it into the logfile now.
      // to read the logfile in realtime use: tail -f /tmp/ssms_response

       $fdesc = fopen("tmp/ssms_response","a");

       fwrite($fdesc, $text."\n");

       fclose($fdesc);
?>
```

C# Example

```
protected void Page_Load(object sender, EventArgs e) {

  string customernickname, keyword, option, data, message,
responsetype;
  string originatoraddress, acceptedtime, deliverytype, carrier,
networktype;

  // capture Form POST values to variables, unassigned values will be
NULL
  PhoneNumber = Request.Form["PhoneNumber"];
  IsSecure = Request.Form["IsSecure"];
  text = Request.Form["Text"];
  acceptedtime = Request.Form["AcceptedDate"];
  sent = Request.Form[“SentDate"];
  type = Request.Form["AckType"];

  // do something with the message information
  SaveMsgToDB(PhoneNumber, IsSecure, text, acceptedtime, sent,
responsetype, type, acceptedtime);
}
```

Get Message Status of SecureSMS

Purpose

To obtain the status of one or more SecureSMS messages.

Syntax

```
https://reports.celltrust.net/txtreport/SecureSMSReport
?
< Password=[PASSWORD]&
ReportType=[ReportType]&
Username=[username]&>
{ RequestId=[RequestID]
MessageId=[MessageID]&
FromDate=[Date+Time]&
ToDate=[Date+Time]}
```

-or-

```
http://reports.celltrust.net/txtreport/SecureSMSReport?
< Password=[PASSWORD]&
ReportType=[ReportType]&
Username=[username]&>
{ RequestId=[RequestID]
MessageId=[MessageID]&
FromDate=[Date+Time]&
ToDate=[Date+Time]}
```

Description

This API can be used to obtain the status of one or more messages sent using the CellTrust Gateway. Status can be obtained for a list of MessageIds or specific date range. These message statuses can be returned by the Reporting Module:

- SUCCESS – carrier confirmed message delivery
- FAILURE – message couldn't be delivered to carrier (for example, provided number is not cell phone number)
- ENROUTE – message delivered to carrier, but final delivery status is not confirmed
- UNDELIVERABLE – carrier was unable to deliver the message to recipient

- Delivered – the Micro client acknowledge the receipt of the secure sms message
- Read – the Micro client acknowledged that the message was opened by the user on the destination hand set.

In case of FAILURE or UNDELIVERABLE, some additional description will be provided in “Reason” field.

Mandatory Parameters

Username

This parameter is used to identify your account. It is typically your default keyword or username setup during registration.

Password

This parameter is used to identify your account in conjunction with username.

MessageId

This is a list of comma separated message ids. Each message sent through CellTrust gateway is assigned a unique id. The message id is return as a XML tag and is only available if XMLResponse is set to true.

FromDate

This specifies the start of the period of the report. For example: FromDate=03%2F01%2F2008+00%3A00%3A00 indicates 3/1/2008 at midnight.

ToDate

This specifies the end of the period of the report. For example: ToDate=03%2F05%2F2008+00%3A00%3A00 indicates 3/5/2008 at midnight.

ReportType

The type of requested report. Currently only “MessageStatusReport” type is supported.

Optional Parameters

RequestId

Client-generated request ID – this ID will be added as an attribute to generated response and can be used for tracking by client application.

Response Format

HTTP Response provides results of request processing in XML format. The report is attached to HTTP Response that is returned synchronously for every HTTP Request submitted to Secure SMS Report gateway.

The DTD for XML Response:

```
<?xml version="1.0"?>

<!-- root element        -->
<!ELEMENT  SecureMTMessageStatusReport    (Error |
SecureMTMessage*)>
<!-- version represents current report version -->
<!-- request_id is optional parameter as provided in request -->
<!-- total shows total number of messages in report -->
<!ATTLIST  SecureMTMessageStatusReport   version  (1.0)       "1.0"
                                                 request_id    CDATA
#IMPLIED
                                                 total    CDATA
#IMPLIED>
<!-- Error is returned in case any failure happened during validation or
processing of request  -->
<!ELEMENT Error              (ErrorCode,ErrorString)>
<!ELEMENT ErrorCode           (#PCDATA)>
<!ELEMENT ErrorString         (#PCDATA)>


<!-- Each secure message is represented by  SecureMTMessage
element-->
<!ELEMENT  SecureMTMessage
(Text,GatewaySendDate,DeliveredDate?,ReadDate?,ToNumber,Encr
ypted,Status)-->
<!-- decrypted message text-->
<!ELEMENT Text            (#PCDATA)>
<!-- date message was submitted by gateway-->
<!ELEMENT   GatewaySendDate         (#PCDATA)>
<!--  Time when message was received by secure agent (as reported
by handset) – if provided-->
<!ELEMENT   DeliveredDate          (#PCDATA)>
<!--  Time when message was opened by handset owner (as reported
by handset) – if provided -->
<!ELEMENT   ReadDate          (#PCDATA)>
```

```
<!--  Recipient of the message -->
<!ELEMENT   ToNumber          (#PCDATA)>
<!--  Flag defines if the message was sent as encrypted (can be “true” or “false”) -->
<!ELEMENT   Encrypted         (#PCDATA)>
<!--  Status of the message -->
<!ELEMENT   Status            (#PCDATA)>
```

Notes

If the HTTP Request is generated using HTML Form, the value of “Username” field can be easily read through “Get Page Source” feature of WEB Browser. CellTrust® highly recommends you use redirection through CGI script in this case as it will allow you to hide all authentication-related information from the Web Browser users, and also allows usage of IP identification for better protection.

Example

In this example we are requesting the status of all messages from 3/1/2008 to 6/4/2008.

```
https://reports.celltrust.net/txtreport/SecureSMSReport?Username=USERNAME&Password=PASSWORD&&FromDate=03%2F01%2F2008+00%3A00%3A00&ToDate=06%2F04%2F2008+00%3A00%3A00&RequestId=123456
```

Success Response Example

```
<SecureMTMessageStatusReport request_id="123456" total="2" version=”1.0”>
   <SecureMTMessage>
<Text>test secure</Text>
<GatewaySendDate>06/02/2008 15:21:09</GatewaySendDate>
<ToNumber>11231234567</ToNumber>
<Encrypted>false</Encrypted>
<Status>ENROUTE</Status>
   </SecureMTMessage>
   <SecureMTMessage>
<Text>Hello. secure world!!</Text>
<GatewaySendDate>06/03/2008 06:56:11</GatewaySendDate>
```

```
<DeliveredDate>06/03/2008 07:01:49</DeliveredDate>
<ReadDate>06/03/2008 07:03:05</ReadDate>
<ToNumber>13216543210</ToNumber>
<Encrypted>true</Encrypted>
<Status>READ</Status>
   </SecureMTMessage>
</SecureMTMessageStatusReport>
```

Error Response Example

```
<SecureMTMessageStatusReport>
   <Error>
<ErrorCode>207</ErrorCode>
<ErrorString>Secure SMS service is not enabled</ErrorString>
   </Error>
</SecureMTMessageStatusReport>
```

Sample Code

Perl Example

```
use strict;
use LWP::UserAgent;
use HTTP::Request::Common;

my $message = '';

my $userAgent = LWP::UserAgent->new();

my $response = $userAgent->request(POST
'http://reports.celltrust.net/txtreport/
SecureSMSReport?MessageId=MESSGAEID&Username=YOURUS
ERNAME&Password=PASSWORD&
ReportType=MessageStatusReport ');

print $response->error_as_HTML unless $response->is_success;
print $response->as_string;
```

PHP Example

```
<?php
$URL = "https://reports.celltrust.net/txtreport/SecureSMSReport?";

//your username generally
```

```
$USERNAME = "YOUR USERNAME";
$PASSWORD = "PASSWORD";

//NOTE COUNTRY CODE IS REQUIRED, EXAMPLE US IS
14805551212
$DESTINATION =" MessageStatusReport ";
$XMLRESPONSE = "MESSAGEID";

$DATA  =
"&ReportType=".$DESTINATION."&Username=".$USERNAME.
."&Password=".$PASSWORD ."&MessageId=".$XMLRESPONSE;


echo do_post_request($URL, $DATA);

//should work in PHP 4 and 5
function do_post_request($url, $data, $optional_headers = null) {
  $start = strpos($url,'//')+2;
  $end = strpos($url,'/',$start);
  $host = substr($url, $start, $end-$start);
  $domain = substr($url,$end);
  $fp = pfsockopen($host, 80);
  if(!$fp) return null;
  fputs ($fp,"POST $domain HTTP/1.1\n");
  fputs ($fp,"Host: $host\n");
  if ($optional_headers) {
          fputs($fp, $optional_headers);
  }
  fputs ($fp,"Content-type: application/x-www-form-urlencoded\n");
  fputs ($fp,"Content-length: ".strlen($data)."\n\n");
  fputs ($fp,"$data\n\n");
  $response = "";
  while(!feof($fp)) {
          $response .= fgets($fp, 1024);
  }
  fclose ($fp);
  return $response;
}
 ?>
```

VB Example

```
<%@ Page Language="VB" src="httputils.cs"%>
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<title></title>

<%

    'this example uses VB to interact with csharp object. it was originally
    'developed in c#, porting should not be difficult.  should run on basic
    '.net server.  If web service, an d smtp do not work, you can try http
    'post using this class

    'vars for message
    Dim strNickname = "YOURNICKNAME"
    Dim strPassword= "PASSWORD"
    Dim strMessageID = "MESSAGEID"
    Dim strPassword = "YOURPASSWORD"
    Dim strMessage = Session("Message")
    Dim strDestination = Session("OriginatorAddress")
    Dim strXMLResponse = "true"
    Dim strTemplate = "test"
    Dim strWAPPushURL = "http://google.com"
    Dim strRedirectURL = "http://google.com"
    Dim strAdminEmail = "admin@google.com"
    Dim strDestPort = "0"
    Dim strShortcode = Session("Shortcode")
    Dim strPricePoint = ""
    Dim strDoubleOptin = "false"
    Dim strFromDate =”01-01-2008”
    Dim strToDate=”02-02-2008”

    'response returned by httppost
    Dim strResponse = ""

    'defined in c# source file
    Dim myhttputils As New httputils()
    strResponse =
httputils.HttpPost("http://reports.celltrust.net/txtreport/
SecureSMSReport?",
  "MessageId=" + strMessageID + "&ReportType=
MessageStatusReport " + "&Username=" + strNickname +
"&Password=" + strPassword +
+ "&FromDate=" + strFromDate + "&ToDate=" + strToDate)

    'check the response
```

```
    If (strResponse <> "") Then
       Response.Write("Response: " + strResponse)
    Else
       Response.Write("XML Response null or empty!")
    End If
%>

</head>
<body></body>
</html>
```

Note: you can download httputils.cs from
http://www.celltrust.com/docs/httputils.zip

Send Standard SMS or Email

Purpose

To send standard SMS message to one or more people including WAP push*, premium message, double opt-in, and other standard SMS variations.

Syntax

```
https://gateway.celltrust.net/TxTNotify/TxTNotify?
<PhoneDestination=[Phone #]&
carrierId=[carrierID]&
Username=[username]&
Password=[password]&
Message=[message]&
Nickname=[nickname]>
{ TemplateName=[template name]&
Subject=[subject]&
EmailDestination=[email address]&
WapPushURL=[url]&
AdminEmail=[email address]&
DestPort=[port number]&
XMLResponse=[true|false]&
Shortcode=[shortcode]&
PricePoint =[amount]&
DoubleOptin =[true|false]&
redirectURL=[url]}
```

-or-

```
http://gateway.celltrust.net/TxTNotify/TxTNotify?
<PhoneDestination=[Phone #]&
carrierId=[carrierID]&
Username=[username]&
Password=[password]&
Message=[message]&
Nickname=[nickname]>
{ TemplateName=[template name]&
Subject=[subject]&
EmailDestination=[email address]&
WapPushURL=[url]&
AdminEmail=[email address]&
DestPort=[port number]&
XMLResponse=[true|false]&
Shortcode=[shortcode]&
PricePoint =[amount]&
DoubleOptin =[true|false]&
redirectURL=[url]}
```

Description

This API can be used to send regular SMS and email messages. The following types of standard SMS messages are supported:

- Text Message
- WAP Push*
- Premium message
- Double Opt-in

Mandatory Parameters

PhoneDestination

The destination mobile phone number that will receive the message. Note that it must include country code and only digits are accepted, for example:

14805551212 for US and Canada

342475551212 for international

EmailDestination

The destination email address(s) that will receive the message. More than one address can be passed as comma delimited list.

Username

The user name is used to authenticate your account. The same username as created during registration.

Password

The password is used to authenticate your account. The same password as created during registration.

Message

The message to be delivered to the handset. If your message contains URL sensitive characters, then the message must be "URL encoded".

Nickname

This is the keyword that is appended to the beginning of your message. When using a shared short code, this string identifies your company as the sender of the message. You must own this keyword, in order to be able to send a message from it. When you first register for an account your username and keyword are the same string. You can view a list of your keywords by logging in to CellTrust gateway web portal (My Gateway/Manage Keywords).

Optional Parameters

CarrierID

Carriers are identified by a unique carrier ID. For a list of carrier IDs, please refer to the Carrier Codes table on page 28. Though CarrierID is an optional parameter for MTs, it is strongly recommended that you provide the CarrierID of the Phone number in the API call.

*Note: You will be charged an additional fraction of credit for every message you send without the CarrierID. You may use Preview API to determine the carrier for a mobile phone number.

Shortcode

It specifies the short code to be used to deliver the message. Relevant only if you have more than one dedicated short code.

TemplateName

This is the name of a predefined template that will be used to create and send a message. Developers can go to CellTrust gateway web portal and change the content of the message, parameters and formatting.

Subject

This is the subject of an email. This parameter only applies to Email messages and not Text Messages.

WapPushURL

If this parameter is present, the message will be considered to be WAP Push* message. The end user will be notified that there is downloaded content in the message and can choose to download the content. The end user must have a data connection to use this feature on their phone. In this case Message parameter will appear as a link on recipient cell phone.

NOTE: *Some carriers (such as Verizon Wireless in US and other Non-GSM carriers) do not allow sending WAP Push* messages. Please review the glossary at the end of this document for more details*

redirectURL

If present, this URL is used as the destination page after message has been processed. This option is useful for HTML forms so that a new page is displayed once the message has been processed by the CellTrust gateway.

AdminEmail

If present, an email is sent to the email address specified in this parameter. The email message contains phone and/or email destination, status of the message and content of the message.

Sample email content:

Message: hello Phone Destinations: 14805155200 Email Destinations: test@test.com Result: Success

XMLResponse

Set this parameter to true to get a XML response. By default, this API responds with a text string. If request is accepted by the server, the following string is returned: “Message accepted for delivery.”

DestPort

This parameter specifies the destination port on the mobile phone. Typically this is used to “wake up” an application listening on a specific port on a mobile device. This is referred to as PushRegistry on J2ME phones. Windows Mobile and Brew phones do not support sending a SMS message to a port.

PricePoint

Price point value for premium messages in cents (like 199 for $1.99 price point). Your account must be enabled for price points for this to work. The short code specified must be approved for a premium program for this feature to be active.

DoubleOptin

Double opt-in is the process by which an end user will confirm their acceptance to be charged on their mobile phone bill for a service to be delivered to their mobile phone. Premium or SMS billing is the process for charging for a service delivered to the phone. In order to charge a user an additional fee, user should first receive a ‘Charge Notification’ and reply with OK or ACCEPT or similar words to accept the charge. The original content must remain on hold until acceptance message is received and then it can be sent to the user followed with a ‘Charge confirmation’ message.

If set to “true”, server will start “double opt-in” process. CellTrust gateway will send a message to the mobile phone to confirm the charge specified in PricePoint parameter. If end user accepts the charge by texting OK back to your short code, the content link will be sent to the end user.

Dynamic tags

In addition to default parameters, there are dynamic “tag” parameters for each template. As long as these tags are defined in the Message Template, the application can pass these parameters with the values to the CellTrust Gateway and it will replace tags with provided values in the message content. Please refer to CellTrust Gateway web portal under Modules --> Templates to create a template and setup dynamic tags.

For example if you use ${app_name} in your template then you can pass app_name=test_app in the API. In the generate message ${app_name} will be replaced with test_app.

Response Format

The response is attached to HTTP Response that is returned synchronously for every HTTP Request submitted to Secure SMS gateway. Please note that message delivery is asynchronous process that involves a number of steps. The message ID received from Secure SMS gateway means that message was accepted for delivery. It doesn't necessarily mean the message is delivered to recipient. Use message ID returned in XML Response for further message tracking and updates of its status.

The DTD for XML Response:

```
<?xml version="1.0"?>

<!-- root element        -->
<!ELEMENT TxTNotifyResponse     (Error, MsgResponseList)>

<!-- Error is returned in case request didn't pass validation   -->
<!-- on "network" level ( missing parameter, security problem etc) -->
<!ELEMENT Error                 (ErrorCode,ErrorString)>
<!ELEMENT ErrorCode             (#PCDATA)>
<!ELEMENT ErrorString           (#PCDATA)>

<!-- MsgResponseList contains specific results of request -->
<!-- processing. Note that TxTNotify allows to submit SMS and Email -->
<!-- messages in the same request, and so the response can contain more
-->
<!-- then one result ( in fact, one per delivery type) -->
<!ELEMENT MsgResponseList       (MsgResponse*)>

<!-- MsgResponse has one attribute that defines its delivery type -->
<!ELEMENT MsgResponse
(Status,(ErrorCode,ErrorString)?|MessageId?)>
<!ATTLIST MsgResponse           type    (SMS|EMAIL)   #REQUIRED>

<!-- Status specifies result of submit operation. Only two possible -->
<!-- values are defined ( ACCEPTED, or FAILURE ) -->
<!ELEMENT Status                (#PCDATA)>

<!-- In case of FAILURE, error values are returned -->
<!ELEMENT ErrorCode             (#PCDATA)>
<!ELEMENT ErrorString           (#PCDATA)>

<!-- In case of ACCEPTED, Message ID is returned ( can be used for -->
```

<!-- further message tracking -->
<!ELEMENT MessageId (#PCDATA)>

Notes

All communication with handset is SMS based, and no data plan is required.

All parameter names as well as the URL are case sensitive!

To ensure complete security always use HTTPS for sending SMS messages and white list your server.

CellTrust SMS Gateway supports both GET and POST requests

In case of error, the CellTrust Gateway will redirect the message to the redirectURL with a parameter "error". It is the responsibility of the customer to process the error and provide the user with a friendly message. For example, if the "destination" parameter passed to the gateway is a 9-digit number instead of 10-digit, the CellTrust Gateway will redirect the message to the redirectURL with error parameter set to Invalid phone number, for example:

http://www.yourcompany.com/msgSendConfirmed.php?error=Invalid%20Phone%20Number

Error Codes

201	wrong or missing customer nickname
204	missing destination (phone number is required)
205	missing message (message text is required)
206	security violation
207	service disabled
302	missing mailing address (valid mailing address is required for sending emails)
401	server error

Examples

In this example a message is sent using template "Hello" to the phone number 14805155200. The carrier for this phone number is qwest with a carrier ID 19. It also sends the same message to email address name@destinationdomain.com. Hello template must be created on

the CellTrust Gateway Web Portal. In this example two dynamic tags are used.

https://gateway.celltrust.net/TxTNotify/TxTNotify?
Username=ABC&Password=XXX&CustomerNickname=ABC&
PhoneDestination=14805551212&carrierId=19&
EmailDestination=name@destinationdomain.com&
TemplateName=Hello&FirstTag=FirstTagValue&SecondTag=SecondTagVal
ue&redirectURL=http://www.yourcompany.com/msgSendConfirmed.asp

Sample template content for the above message:

${FirstTag} was received by ${SecondTag}.

This example demonstrates sending SMS to multiple phone numbers. When sending to multiple phone numbers, the carrier ID is included in the PhoneDestination field separated from phone number by a '-':

https://gateway.celltrust.net/TxTNotify/TxTNotify?
Username=ABC&Password=XXX&CustomerNickname=ABC&
PhoneDestination=14805551212-19,16024038599-5,14806781400-
11&Message=you+are+a+champion

This example demonstrates using URL encoding in a message parameter and setting XMLResponse to true to receive a XML response instead of a string.

https://gateway.celltrust.net/TxTNotify/TxTNotify?Username=ABC&Pa
ssword=XXX&CustomerNickname=ABC&PhoneDestination=148055
51212&carrierId=19&AdminEmail=admin@company.com&**Message
=Hello,%20World!&XMLResponse=true**

This example demonstrates sending a WAP Push* message to a phone:

https://gateway.celltrust.net/TxTNotify/TxTNotify?Username=ABC&Pa
ssword=XXX&CustomerNickname=ABC&PhoneDestination=148055
51212&carrierId=19&AdminEmail=admin@company.com&Message=
Push%20Me&**WapPushURL=http://my.wappushurl.com**

NOTE: *Some carriers (such as Verizon Wireless in US and other Non-GSM carriers) do not allow sending WAP Push* messages. Please review the glossary at the end of this document for more details*

This example demonstrates using premium SMS message with a double opt-in. Premium messages will charge the recipient of the message specified in the PricePoint parameter. The short code must be approved for premium messaging by CellTrust and carriers.

https://gateway.celltrust.net/TxTNotify/TxTNotify?Username=ABC&Password=XXX&CustomerNickname=ABC&PhoneDestination=14805551212&carrierId=19&Message=Download%20from%20http://d2c.myringtones.com&**PricePoint=299&DoubleOptin=true**

Success Response Example

```
<TxTNotifyResponse>
  <MsgResponseList>
    <MsgResponse type="SMS">
      <Status>ACCEPTED</Status>
      <MessageId>AAXF505566459PM14</MessageId>
    </MsgResponse>
  </MsgResponseList>
</TxTNotifyResponse>
```

Error Response Example

```
<TxTNotifyResponse>
<Error>
  <ErrorCode>201</ErrorCode>
  <ErrorString>Customer not found</ErrorString>
  </Error>
  </TxTNotifyResponse>
  </Error>
</TxTNotifyResponse>
```

Sample Code

HTML Example

```
<html><body>
<form name="myform"
action="http://gateway.celltrust.net/TxTNotify/TxTNotify?">
Enroll In Text Messaging services (Enter Your Phone Number):
<input type='text' name='PhoneDestination'>
Enter the carrier ID of  the phone:
<input type='text' name='carrierId'>
<input type='hidden' name='Message'
```

```
value='You+are+now+enrolled'>
<input type='hidden' name='CustomerNickname'
value='YOURNICKNAME'>
<input type='hidden' name='Username' value='YOURUSERNAME'>
<input type='hidden' name='Password' value='YOURPASSWORD'>
<A href="javascript: submitform()">Submit</A>
</form>
<SCRIPT language="JavaScript">
function submitform()
{
document.myform.submit();
}
</SCRIPT>
</body></html>
```

Note: There are two potential problems in regards to the Web form: 1) anyone that views the page source can access all of the proprietary information required to successfully submit requests, and 2) the Web form provides an easy way to create a script that is able to submit multiple requests using parameters defined in the form. Both these problems could compromise your account and allow unauthorized users to send messages through your account from your web site. We strongly recommend that you use programming methods to block application from hitting your web form multiple times and separate your web form from the module sending the message.

Perl Example

```
use strict;
use LWP::UserAgent;
use HTTP::Request::Common;

my $message = '';
my $userAgent = LWP::UserAgent->new();

my $response = $userAgent->request(POST
'http://gateway.celltrust.net/TxTNotify/
TxTNotify?PhoneDestination=YOURPHONE&carrierId=CARRIERID&
Message=test&Username=USERNAME&Password=PASSWORD&C
ustomerNickname=YOURNICKNAME&XMLResponse=true');

print $response->error_as_HTML unless $response->is_success;
print $response->as_string;
```

PHP Example

```
<?php
$URL = "https://gateway.celltrust.net/TxTNotify/TxTNotify?";

//your account details with CellTrust
$NICKNAME = "YOURNICKNAME";
$USERNAME="YOURUSERNAME";
$PASSWORD="YOURPASSWORD";

//NOTE COUNTRY CODE IS REQUIRED, EXAMPLE US IS
14805551212
$DESTINATION ="YOURPHONENUMBER";
$CARRIERID = "PHONENUMBERS CARRIER ID";
// If you are sending to multiple phone numbers, the carrier ID need
//not be declared separately. It can be declared this way:
//$DESTINATION ="16024038599-5,14806781400-11";
$MESSAGE = "This is a test message";
$XMLRESPONSE = "true";

$DATA =
"&PhoneDestination=".$DESTINATION."&carrierId=".$CARRIERID."&
Message=".$MESSAGE."&CustomerNickname=".$NICKNAME."&Us
ername=".$USERNAME."&Password=".$PASSWORD."&XMLRespo
nse=".$XMLRESPONSE;

echo do_post_request($URL, $DATA);

//should work in PHP 4 and 5
function do_post_request($url, $data, $optional_headers = null) {
  $start = strpos($url,'//')+2;
  $end = strpos($url,'/',$start);
  $host = substr($url, $start, $end-$start);
  $domain = substr($url,$end);
  $fp = pfsockopen($host, 80);
  if(!$fp) return null;
  fputs ($fp,"POST $domain HTTP/1.1\n");
  fputs ($fp,"Host: $host\n");
  if ($optional_headers) {
          fputs($fp, $optional_headers);
  }
  fputs ($fp,"Content-type: application/x-www-form-urlencoded\n");
  fputs ($fp,"Content-length: ".strlen($data)."\n\n");
  fputs ($fp,"$data\n\n");
```

```
  $response = "";
  while(!feof($fp)) {
          $response .= fgets($fp, 1024);
  }
  fclose ($fp);
  return $response;
}

?>
```

Java Example

```
import java.net.URLEncoder;
import java.net.*;
import java.io.*;

/*
* Performs HTTP Post Using Java to use celltrust HTTP/Notify Api
*
*/
public class HTTPNotifyPost
{
   public static void main(String[] args)
   {
      //uncomment to send a secure message
      //replace YOURUSERNAME with the username at CellTrust
      //replace YOURPASSWORD with the password at CellTrust
      //replace YOURNICKNAME with the nickname at CellTrust
      //replace PHONENUMBER with the  phone number, be sure to
use country code
      //"+" and "011" is not neccessary for phone number
      //replace MESSAGE with the message you want to send to the
user(s)

SendSMS("YOURUSERNAME","YOURPASSWORD","YOURNICKN
AME","PHONENUMBER","CARRIER ID","MESSAGE");
         }
 public static void SendSMS(String username, String password, String
nickname, String phone, String carrierId, String message)
   {
      try
      {
         String query = "Username=" +
URLEncoder.encode(username);
         query += "&";
```

```
            query += "Password=" + URLEncoder.encode(password);
            query += "&";
            query += "CustomerNickname=" +
URLEncoder.encode(nickname);
            query += "&";
            query += "PhoneDestination=" + URLEncoder.encode(phone);
            query += "&";
            query += "carrierId=" + URLEncoder.encode(carrierId);
            query += "&";
            query += "Message=" + URLEncoder.encode(message);
            query += "&";
            query += "XMLResponse=true";
            query += "&";
            query += "AllowInsecure=false";


        // Send data

        URL myurl = new
URL("http://gateway.celltrust.net/TxTNotify/TxTNotify");
        URLConnection conn = myurl.openConnection();
        conn.setDoOutput(true);
        OutputStreamWriter wr = new
OutputStreamWriter(conn.getOutputStream());
        wr.write(query);
        wr.flush();

        // Get the response
        BufferedReader rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        String line;
        while ((line = rd.readLine()) != null) {
            System.out.println("Line is: " + line);
        }
        wr.close();
        rd.close();
    }
    catch (Exception e)
    {
   System.out.println("Error: " + e.toString());
    }

    }
```

```
}
```

VB Example

```
<%@ Page Language="VB" src="httputils.cs"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>

<%

    'this example uses VB to interact with csharp object. it was originally
    'developed in c#, porting should not be difficult.  should run on basic
    '.net server.  If web service, an d smtp do not work, you can try http
    'post using this class

    'vars for message
    Dim strUsername = "YOURUSERNAME"
    Dim strPassword = "YOURPASSWORD"
    Dim strNickname = "YOURNICKNAME"
    Dim strcarrierId = "CARRIER ID"

    Dim strMessage = Session("Message")
    Dim strDestination = Session("OriginatorAddress")
    Dim strXMLResponse = "true"
    Dim strTemplate = "test"
    Dim strWAPPushURL = "http://google.com"
    Dim strRedirectURL = "http://google.com"
    Dim strAdminEmail = "admin@google.com"
    Dim strDestPort = "0"
    Dim strShortcode = Session("Shortcode")
    Dim strPricePoint = ""
    Dim strDoubleOptin = "false"

    'response returned by httppost
    Dim strResponse = ""

    'defined in c# source file
    Dim myhttputils As New httputils()
    strResponse =
httputils.HttpPost("http://gateway.celltrust.net/TxTNotify/TxTNotify?", _
```

```
    "PhoneDestination=" + strDestination + "&carrierId=" +
strcarrierId+"&Username=" + strUsername +"&Password=" +
strPassword +"&Nickname=" + strNickname +"&Message=" +
strMessage +  "&XMLResponse=" + strXMLResponse)

    'check the response
    If (strResponse <> "") Then
        Response.Write("Response: " + strResponse)
    Else
        Response.Write("XML Response null or empty!")
    End If
%>

</head>
<body></body>
</html>
```

Note: you can download httputils.cs from
http://www.celltrust.com/docs/httputils.zip

Receive Standard SMS Messages

Purpose

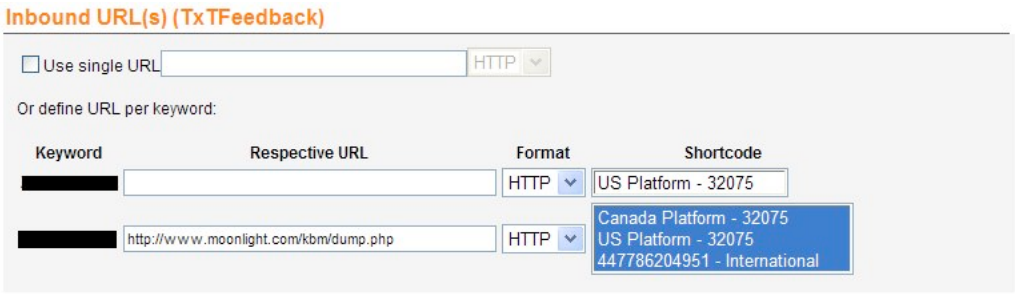
To receive standard Mobile Originated (MO) messages.

Description

This section describes how to receive messages sent by any phone capable of sending a text message.

The CellTrust Gateway can be configured to push an incoming message to your server and application. This is achieved by setting a Pushback URL in the CellTrust Gateway Web Portal under MyGateway and Account Module. In Account Module, you can specify an URL for each keyword. Messages are pushed to the URL in real time as HTTP Form Post requests. The parameter section defines the parameters available in the form post.

Note: CellTrust Gateway performs a Form POST, your code cannot utilize the querystring collection, instead you need to access the Form collection. The querystring collection will work when you're testing and putting the URL and parameters in your browser but when CellTrust Posts it will be a Form POST and posted data is accessed in the Form collection.



Parameters

CustomerNickname

The keyword that was used in the MO.

ResponseType

This can be set to Normal or admin. Stop and Start messages are set to admin type.

Option and Keyword

These fields are associated with Campaign Module. If you setup a campaign in the CellTrust Gateway Web Portal, these parameters will be filled based on your campaign.

Message

The full content of the message including keyword.

NOTE: Whether the short or long code is a shared or dedicated code for a specific SMS account on the Gateway, the entire message body of the text message will be provided within the “Message” parameter posted to the inbound URL of incoming SMS messages.

Data

The content of the message after the keyword.

OriginatorAddress

The phone number of the mobile phone that sent the message.

AcceptedTime

Date and time that message was received by the server.

DeliveryType

This can be set to SMS, Email, or Voice.

Carrier

Carrier of the sender of the message. This parameter is only filled for US carriers.

NetworkType

Carrier's network such as GSM or CDMA. This parameter is only filled for US carriers.

Notes

Please note that CellTrust performs a Form POST, your code cannot utilize the querystring collection, and instead you need to access the Form collection. The querystring collection will work when you're testing and putting the URL and parameters in your browser but when CellTrust Posts it will be a Form POST and posted data is accessed in the Form collection.

All parameter names as well as the URL are case sensitive!

To ensure complete security always use HTTPS for sending SMS messages and white list your server.

Examples

```
CustomerNickname:    NICKNAME
ResponseType:        NORMAL
Keyword:
Option:
Message:             NICKNAME test message
Data:                test message
OriginatorAddress:   14805155200
AcceptedTime:        20Nov,08 15:34:30
DeliveryType:        SMS
Carrier:             t-mobile
NetworkType:    gsm
```

Sample Code

PHP Example

```
<?php
//=================================================
=======
//        Celltrust sms Response gateway Test script
//        Put this script in the web server directory
```

```
//=================================================
=======
        error_reporting(E_ALL);
        set_time_limit(0);
        ob_implicit_flush();

       // debug variables

        $query    = $_SERVER['QUERY_STRING'];
        $uri      = $_SERVER['REQUEST_URI'];
        $remadr   = $_SERVER['REMOTE_ADDR'];

       //    create the text output

       $text  = "==================================\n\r";
   foreach (array_keys($_POST) as $key)
   {
            $$key = $_POST[$key];
            $text .= "$key is ${$key}\n\r";
   }
       $text .= "==================================\n\r";
   $text .= print_r ($_POST);


       // write it into the logfile now.
       // to read the logfile in realtime use: tail -f /tmp/ssms_response

        $fdesc = fopen("tmp/ssms_response","a");
        fwrite($fdesc, $text."\n");
        fclose($fdesc);
?>
```

VB Example

```
protected void Page_Load(object sender, EventArgs e) {
            string customernickname, keyword, option, data, message,
responsetype;
  string originatoraddress, acceptedtime, deliverytype, carrier,
networktype;

  // capture Form POST values to variables, unassigned values will be
NULL
  customernickname = Request.Form["CustomerNickname"];
  keyword = Request.Form["Keyword"];
  option = Request.Form["Option"];
```

```
  data = Request.Form["Data"];
  message = Request.Form["Message"];
  responsetype = Request.Form["ResponseType"];
  originatoraddress = Request.Form["OriginatorAddress"];
  acceptedtime = Request.Form["AcceptedTime"];
  deliverytype = Request.Form["DeliveryType"];
  carrier = Request.Form["Carrier"];
  networktype = Request.Form["NetworkType"];

  // do something with the message information
  SaveMsgToDB(customernickname, keyword, option, data,
message, responsetype, originatoraddress, acceptedtime,
deliverytype, carrier, networktype);
}
```

Get Message Status

Purpose

To obtain the status of one or messages.

Syntax

```
https://reports.celltrust.net/txtreport/TxTReport?
<MessageId=[MessageID]&
ReportType=[ReportType]&
Nickname=[nickname]>
{ RequestId=[RequestID]}
```

-or-

```
http://reports.celltrust.net/txtreport/TxTReport?
<MessageId=[MessageID]&
ReportType=[ReportType]&
Nicknae=[nickname]>
{ RequestId=[RequestID]}
```

Description

This API can be used to obtain the status of one or more messages sent using the CellTrust Gateway. These message statuses can be returned by the Reporting Module:

- SUCCESS – carrier confirmed message delivery
- FAILURE – message couldn't be delivered to carrier (f.i., provided number is not cell phone number)
- ENROUTE – message delivered to carrier, but final delivery status is not confirmed
- UNDELIVERABLE – carrier was unable to deliver the message to recipient

In case of FAILURE or UNDELIVERABLE, some additional description will be provided in “Reason” field.

Mandatory Parameters

Nickname

This parameter is used to identify your account. It is typically your default keyword or username.

MessageId

This is a list of comma separated message ids. Each message sent through CellTrust gateway is assigned a unique id. The message id is return as a XML tag and is only available if XMLResponse is set to true.

ReportType

The type of requested report. Currently only “MessageStatusReport” type is supported.

Optional Parameters

RequestId

Client-generated request ID – this ID will be added as an attribute to generated response and can be used for tracking by client application.

Response Format

The Reporting Module data are returned inside the PMResponse document generated by the CellTrust Gateway. The DTD for the relevant part of PMResponse XML document is described below.

The DTD for XML Response:

```
<!-- PMResponse encloses all supported responses and data reports
-->
<!ELEMENT PMResponse
(GenerateDate,(RTMessageStatusResponse|RTMessageHistoryRes
ponse))>
<!-- Please always make sure you parse correct PMResponse version
-->
<!-- request_id field will contain the value of RequestId passed inside
HTTP Request -->
<!ATTLIST PMResponse     pm_version  (1_0)       "1_0"
                         request_id    CDATA   #IMPLIED>

<!-- the exact date of this response creation -->
<!ELEMENT GenerateDate (DateTime)>


<!-- RT Message History Response -->
<!-- Allows to receive information about all messages submitted during
-->
<!-- predefined period  -->
<!ELEMENT RTMessageHistoryResponse
(FromDate,ToDate,RTMessageSet)>

<!ELEMENT FromDate (DateTime)>

<!ELEMENT ToDate (DateTime)>

<!-- String representing timestamp -->
<!ELEMENT DateTime (DateFormat,Date)>

<!-- Date/Time format    -->
<!ELEMENT DateFormat (#PCDATA)>

<!-- The date itself   -->
<!ELEMENT Date (#PCDATA)>

<!-- RT Message Status Response   -->
<!-- Allows to receive information about specific messages identified
by -->
<!-- provided message ID -->
<!ELEMENT RTMessageStatusResponse (RTMessageSet)>
```

```
<!-- Placeholder for multiple messages. “RT” stays for “Recipient-
Terminated”  -->
<!-- and is used as opposite to “Mobile-Terminated” message, as it
covers both -->
<!-- SMS and Email messages       -->
<!ELEMENT RTMessageSet (RTMessage)>

<!-- Contains all information per single message   -->
<!ELEMENTRTMessage  (RTMessageId, DeliveryType,
RTMessageBody, RTMessageSubject?,  RTMessagePart+)>

<!-- Element represents RT message string ID ( as used f.i. in report
request )-->
<!ELEMENT RTMessageId (#PCDATA)>

<!-- Element represents Delivery Type ( currently SMS or Email ) -->
<!ELEMENT DeliveryType (#PCDATA)>

<!-- Element represents RT message body -->
<!ELEMENT RTMessageBody (#PCDATA)>

<!-- Element represents RT message subject (optional for emails) -->
<!ELEMENT RTMessageSubject (#PCDATA)>

<!-- in case of SMS, long messages can be split to number of shorter
messages -->
<!-- that are sent one after another. Each message is represented by
RTMessagePart -->
<!-- in case message was sent to more then one destination, each
message part will -->
<!-- have associated list of recipients.       -->
<!ELEMENT RTMessagePart
(RTMessagePartText,RTMessagePartSendEvent+)>
<!-- Each part has order number inside message     -->
<!ATTLIST RTMessagePart part_number  CDATA   #REQUIRED>

<!-- Element represents RT message part text -->
<!ELEMENT RTMessagePartText (#PCDATA)>

<!-- Specific info for delivery of this message part to specific recipient -
->
<!ELEMENT RTMessagePartSendEvent
(Recipient,Status,Reason?)>
```

```
<!-- Element represents Status of send event -->
<!ELEMENT Status (#PCDATA)>

<!-- Element represents failure description -->
<!ELEMENT Reason (#PCDATA)>

<!-- Recipient information  -->
<!ELEMENT Recipient (FirstName?,LastName?,Destination)>

<!-- Recipient first name (if known ) -->
<!ELEMENT FirstName  (#PCDATA)>

<!-- Recipient last name (if known ) -->
<!ELEMENT LastName   (#PCDATA)>

<!-- Destination ( phone number or email ) -->
<!ELEMENT Destination   (#PCDATA)>
```

Notes

If the HTTP Request is generated using HTML Form, the value of "Nickname" field can be easily read through "Get Page Source" feature of WEB Browser. CellTrust highly recommends you use redirection through CGI script in this case as it will allow you to hide all authentication-related information from the Web Browser users, and also allows usage of IP identification for better protection.

Example

In this example we are requesting the status of message AZXF1129154692PM24775.

```
http://reports.celltrust.net/txtreport/TxTReport?Nickname=YOURNICK
NAME&MessageId=AZXF1129154692PM247751&ReportType=Mes
sageStatusReport
```

Success Response Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<PMResponse pm_version="1_0">
<GenerateDate>
```

```
<DateTime>
  <DateFormat>dd/MMM/yyyy HH:mm:ss</DateFormat>
  <Date>20/Nov/2008 21:24:08</Date>
  </DateTime>
  </GenerateDate>
<RTMessageStatusResponse>
<RTMessageSet>
<RTMessage>
  <RTMessageId>AZXF1129154732PM247751</RTMessageId>
  <RTMessageBody>test sent to 14805551212</RTMessageBody>
<RTMessagePart part_number="1">
  <RTMessagePartText>test sent to
14805551212</RTMessagePartText>
<RTMessagePartSendEvent>
<Recipient>
  <Destination>14805551212</Destination>
  </Recipient>
  <Status>SUCCESS</Status>
  </RTMessagePartSendEvent>
  </RTMessagePart>
  <DeliveryType>SMS</DeliveryType>
  </RTMessage>
  </RTMessageSet>
  </RTMessageStatusResponse>
  </PMResponse>
```

Error Response Example

```
<?xml version="1.0" ?>
<TxTNotifyResponse>
<Error>
  <ErrorCode>208</ErrorCode>
  <ErrorString>Missing report type</ErrorString>
  </Error>
  </TxTNotifyResponse>
```

Sample Code

Perl Example

```
use strict;
use LWP::UserAgent;
use HTTP::Request::Common;
```

```
my $message = '';

my $userAgent = LWP::UserAgent->new();

my $response = $userAgent->request(POST
'http://reports.celltrust.net/txtreport/
TxTReport?MessageId=MESSAGEID&Nickname=YOURNICKNAME
& ReportType=MessageStatusReport ');

print $response->error_as_HTML unless $response->is_success;

print $response->as_string;
```

PHP Example

```
<?php
$URL = "https://reports.celltrust.net/txtreport/ TxTReport?";

//your username generally
$NICKNAME = "YOURNICKNAME";

//NOTE COUNTRY CODE IS REQUIRED, EXAMPLE US IS
14805551212
$DESTINATION =" MessageStatusReport ";
$XMLRESPONSE = "MESSAGEID";

$DATA  =
"&ReportType=".$DESTINATION."&Nickname=".$NICKNAME."&Mes
sageId=".$XMLRESPONSE;


echo do_post_request($URL, $DATA);

//should work in PHP 4 and 5
function do_post_request($url, $data, $optional_headers = null) {
  $start = strpos($url,'//')+2;
  $end = strpos($url,'/',$start);
  $host = substr($url, $start, $end-$start);
  $domain = substr($url,$end);
  $fp = pfsockopen($host, 80);
  if(!$fp) return null;
  fputs ($fp,"POST $domain HTTP/1.1\n");
  fputs ($fp,"Host: $host\n");
```

```
  if ($optional_headers) {
          fputs($fp, $optional_headers);
  }
  fputs ($fp,"Content-type: application/x-www-form-urlencoded\n");
  fputs ($fp,"Content-length: ".strlen($data)."\n\n");
  fputs ($fp,"$data\n\n");
  $response = "";
  while(!feof($fp)) {
          $response .= fgets($fp, 1024);
  }
  fclose ($fp);
  return $response;
}
 ?>
```

VB Example

```
<%@ Page Language="VB" src="httputils.cs"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>

<%

    'this example uses VB to interact with csharp object. it was originally
    'developed in c#, porting should not be difficult.  should run on basic
    '.net server.  If web service, and smtp do not work, you can try http
    'post using this class

    'vars for message
    Dim strNickname = "YOURNICKNAME"
    Dim strMessageID = "MESSAGEID"
    Dim strPassword = "YOURPASSWORD"
    Dim strMessage = "Message"
    Dim strDestination = "OriginatorAddress"
    Dim strXMLResponse = "true"
    Dim strTemplate = "test"
    Dim strWAPPushURL = "http://google.com"
    Dim strRedirectURL = "http://google.com"
    Dim strAdminEmail = "admin@google.com"
    Dim strDestPort = "0"
    Dim strShortcode = "Shortcode"
    Dim strPricePoint = ""
    Dim strDoubleOptin = "false"
```

```
    'response returned by httppost
    Dim strResponse = ""

    'defined in c# source file
    Dim myhttputils As New httputils()
    strResponse =
httputils.HttpPost("http://reports.celltrust.net/txtreport/TxTReport?",
  "MessageId=" + strMessageID + "&ReportType=
MessageStatusReport " + "&Nickname=" + strNickname )

    'check the response
    If (strResponse <> "") Then
        Response.Write("Response: " + strResponse)
    Else
        Response.Write("XML Response null or empty!")
    End If
%>

</head>
<body></body>
</html>
```

Note: you can download httputils.cs from
http://www.celltrust.com/docs/httputils.zip

Schedule Standard SMS or Email

Purpose

To schedule to send a standard SMS message or email to one or more people.

Syntax

```
https://gateway.celltrust.net/pmsgw/TxTScheduler?
<PhoneDestination=[Phone #]&
carrierId=[Carrier ID]&
Username=[Username]&
Password=[Password]&
StartYear=[start year]&
StartMonth=[start month]&
StartDay=[start day]&
StartHour=[start hour]&
StartMinute=[start minute]&
```

```
Message=[message]&
CustomerNickname=[nickname]>
{ TemplateName=[template name]&
EmailDestination=[email address]&
AdminEmail=[email address]&
redirectURL=[url]}
```

-or-

```
https://gateway.celltrust.net/pmsgw/TxTScheduler?
<PhoneDestination=[Phone #]&
carrierId =[Carrier ID]&
StartYear=[start year]&
StartMonth=[start month]&
StartDay=[start day]&
StartHour=[start hour]&
StartMinute=[start minute]&
Message=[message]&
CustomerNickname=[nickname]>
{ TemplateName=[template name]&
EmailDestination=[email address]&
AdminEmail=[email address]&
redirectURL=[url]}
```

Description

This API can be used to schedule to send a standard SMS or email messages. A message can be scheduled to be sent on a given day at a specific time.

Mandatory Parameters

PhoneDestination

The destination mobile phone number that will receive the message. Note that it must include country code and only digits are accepted, for example:

14805551212 for US and Canada

342475551212 for international

EmailDestination

The destination email address(s) that will receive the message. More than one address can be passed as comma delimited list.

Message

The message to be delivered to the handset. If your message contains URL sensitive characters, then the message must be “URL encoded”

CustomerNickname

This is the keyword that is appended to the beginning of your message. When using a shared short code, this string identifies your company as the sender of the message. You must own this keyword, in order to be able to send a message from it. When you first register for an account your username and keyword are the same string. You can view a list of your keywords by logging in to CellTrust gateway web portal (My Gateway/Manage Keywords).

StartYear

This parameter specifies the starting year in yyyy format. For example for 2008 use 2008.

StartMonth

This parameter specifies the starting month in mm format. For example use 07 for July.

StartDay

This parameter specifies the starting day in dd format. For example use 01 for the first day of the month.

StartHour

This parameter specifies the starting hour in hh format. This is in 24 hour format. For example, 2 pm is 14.

StartMinute

This parameter specifies the starting minutes in mm format. For example use 20 for 20 minutes

For July 15, 2008 at 5:15 PM use:

StartYear = 2008
StartMonth=07
StartDay=15
StartHour=17
StartMinute=15

Optional Parameters

CarrierID

Carriers are identified by a unique carrier ID. For a list of carrier IDs, please refer to the Carrier Codes table on page 28. Though CarrierID is an optional parameter for MTs, it is strongly recommended that you provide the CarrierID of the Phone number in the API call.

*Note: You will be charged an additional fraction of credit for every message you send without the CarrierID. You may use Preview API to determine the carrier for a mobile phone number.

TemplateName

This is the name of a predefined template that will be used to create and send a message. Developers can go to CellTrust gateway web portal and change the content of the message, parameters and formatting.

redirectURL

If present, this URL is used as the destination page after message has been processed. This option is useful for HTML forms so that a new page is displayed once the message has been processed by the CellTrust gateway.

AdminEmail

If present, an email is sent to the email address specified in this parameter. The email message contains phone and/or email destination, status of the message and content of the message.

Sample email content:

Message: hello Phone Destinations: 14805155200 Email Destinations: test@test.com Result: Success

Dynamic tags

In addition to default parameters, there are dynamic “tag” parameters for each template. As long as these tags are defined in the Message Template, the application can pass these parameters with the values to the CellTrust Gateway and it will replace tags with provided values in the message content. Please refer to CellTrust Gateway web portal under Modules --> Templates to create a template and setup dynamic tags.

For example if you use ${app_name} in your template then you can pass app_name=test_app in the API. In the generate message ${app_name} will be replaced with test_app.

Notes

All communication with handset is SMS based, and no data plan is required.

All parameter names as well as the URL are case sensitive!

To ensure complete security always use HTTPS for sending SMS messages and white list your server.

CellTrust SMS Gateway supports both GET and POST requests

In case of error, the CellTrust Gateway will redirect the message to the redirectURL with a parameter “error”. It is the responsibility of the customer to process the error and provide the user with a friendly message. For example, if the “destination” parameter passed to the gateway is a 9-digit number instead of 10-digit, the CellTrust Gateway will redirect the message to the redirectURL with error parameter set to Invalid phone number, for example:

http://www.yourcompany.com/msgSendConfirmed.php?error=Invalid%20Phone%20Number

Error Codes

Customer not found
Message template not found
Message Template or Message Body required
Missing date parameter
Phone or Email Destination is required
Scheduled date is in the past
server error

Example

In this example a message “Hello” is sent to phone number 14805555200 at 3 pm on December 5th, 2008. This message has to

be scheduled before 3 pm on December 5th, 2008, otherwise the system will generate an error.

https://gateway.celltrust.net/pmsgw/TxTScheduler?PhoneDestination=14805555200&
CarrierId=19&CustomerNickname=ABC&Message=Hello&StartYear=2008&StartMonth=12&StartDay=5&StartHour=15&StartMinute=00

Success Response Example

CellTrust Gateway response with HTTP 200 when successful.

Error Response Example

HTTP Status with an error message is returned when an error occurs. The error will include an error message such as "Scheduled date is in the past".

Sample Code

HTML Example

```
<html><body>
<form name="myform"
action="http://gateway.celltrust.net/pmsgw/TxTScheduler?">
Enroll In Text Messaging services (Enter Your Phone Number):
<input type='text' name='PhoneDestination'>
Enter the Carrier ID:
<input type='text' name=' CarrierId '>
input type='text' name='StartYear'>
input type='text' name='StartMonth'>
input type='text' name='StartDay'>
input type='text' name='StartHour'>
input type='text' name='StartMinute'>
<input type='hidden' name='Message'
value='You+are+now+enrolled'>
<input type='hidden' name='CustomerNickname'
value='YOURNICKNAME'>
<input type='hidden' name='Username' value='YOURUSERNAME'>
<input type='hidden' name='Password' value='YOURPASSWORD'>
<A href="javascript: submitform()">Submit</A>
</form>
<SCRIPT language="JavaScript">
```

```
function submitform()
{
document.myform.submit();
}
</SCRIPT>
</body></html>
```

Note: There are two potential problems in regards to the Web form: 1) anyone that views the page source can access all of the proprietary information required to successfully submit requests, and 2) the Web form provides an easy way to create a script that is able to submit multiple requests using parameters defined in the form. Both these problems could compromise your account and allow unauthorized users to send messages through your account from your web site. We strongly recommend that you use programming methods to block application from hitting your web form multiple times and separate your web form from the module sending the message.

Perl Example

```
use strict;
use LWP::UserAgent;
use HTTP::Request::Common;

my $message = '';

my $userAgent = LWP::UserAgent->new();

my $response = $userAgent->request(POST
'http://gateway.celltrust.net/pmsgw/TxTScheduler?PhoneDestination=
YOURPHONE&CarrierId=CARRIERID&Message=test&Username=Y
OURUSERNAME&Password=YOURPASSWORD&CustomerNickna
me=YOURNICKNAME&StartYear=2008&StartMonth=07&StartDay=1
4&StartHour=17&StartMinute=20');

print $response->error_as_HTML unless $response->is_success;

print $response->as_string;
```

PHP Example

```
<?php
$URL = "https://gateway.celltrust.net/pmsgw/TxTScheduler?";
```

```
//your account details at CellTrust
$NICKNAME = "YOURNICKNAME";
$USERNAME = "YOURUSERNAME";
$PASSWORD = "YOURPASSWORD";


//NOTE COUNTRY CODE IS REQUIRED, EXAMPLE US IS
14805551212
$DESTINATION ="YOURPHONENUMBER";
$CARRIERID="PHONENUMBER'S CARRIER ID";
$MESSAGE = "This is a test message";
$XMLRESPONSE = "true";

$DATA  =
"&PhoneDestination=".$DESTINATION."&CarrierId=".$CARRIERID."
&Message=".$MESSAGE."&CustomerNickname=".$NICKNAME."&U
sername=".$USERNAME."&Password=".$PASSWORD.&StartYear=
2008&StartMonth=07&StartDay=14&StartHour=17&StartMinute=20;


echo do_post_request($URL, $DATA);

//should work in PHP 4 and 5
function do_post_request($url, $data, $optional_headers = null) {
  $start = strpos($url,'//')+2;
  $end = strpos($url,'/',$start);
  $host = substr($url, $start, $end-$start);
  $domain = substr($url,$end);
  $fp = pfsockopen($host, 80);
  if(!$fp) return null;
  fputs ($fp,"POST $domain HTTP/1.1\n");
  fputs ($fp,"Host: $host\n");
  if ($optional_headers) {
          fputs($fp, $optional_headers);
  }
  fputs ($fp,"Content-type: application/x-www-form-urlencoded\n");
  fputs ($fp,"Content-length: ".strlen($data)."\n\n");
  fputs ($fp,"$data\n\n");
  $response = "";
  while(!feof($fp)) {
          $response .= fgets($fp, 1024);
  }
  fclose ($fp);
```

```
  return $response;
}
 ?>
```

VB Example

```
<%@ Page Language="VB" src="httputils.cs"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>

<%

    'this example uses VB to interact with csharp object. it was originally
    'developed in c#, porting should not be difficult.  should run on basic
    '.net server.  If web service, an d smtp do not work, you can try http
    'post using this class

    'vars for message
    Dim strNickname = "YOURNICKNAME"
    Dim strPassword = "YOURPASSWORD"
    Dim strUsername = "YOURUSERNAME"
    Dim strMessage = Session("Message")
    Dim strDestination = Session("OriginatorAddress")
    Dim strcarrierId = "CARRIER ID"
    Dim strXMLResponse = "true"
    Dim strTemplate = "test"
    Dim strWAPPushURL = "http://google.com"
    Dim strRedirectURL = "http://google.com"
    Dim strAdminEmail = "admin@google.com"
    Dim strDestPort = "0"
    Dim strShortcode = Session("Shortcode")
    Dim strPricePoint = ""
    Dim strDoubleOptin = "false"

    'response returned by httppost
    Dim strResponse = ""

    'defined in c# source file
    Dim myhttputils As New httputils()
    strResponse = httputils.HttpPost("http://
gateway.celltrust.net/pmsgw/TxTScheduler?", _
    "PhoneDestination=" + strDestination + "&CarrierId=" + strcarrierId
+ "&Message=" + strMessage +"&Username=" + strUsername +
```

```
"&Password=" + strPassword +  "&Nickname="
+strNickname+&StartYear=2008&StartMonth=07&StartDay=14&Start
Hour=17&StartMinute=20")

    'check the response
    If (strResponse <> "") Then
        Response.Write("Response: " + strResponse)
    Else
        Response.Write("XML Response null or empty!")
    End If
%>

</head>
<body></body>
</html>
```

Note: you can download httputils.cs from
http://www.celltrust.com/docs/httputils.zip

Preview a Phone Number

Purpose

This API can be used to determine if a phone number is a valid mobile phone number. It can also be used to determine the carrier servicing a given phone number. This API is currently only available in United States.

Syntax

```
https://gateway.celltrust.net/preview/PreviewConnector?
<PhoneNumber=[Phone #]&
Username=[username]&
Password=[password]&
RequestType=[carrier]>
{ RequestId=[number] }
```

-or-

```
https://gateway.celltrust.net/preview/PreviewConnector?
<PhoneNumber=[Phone #]&
Username=[username]&
Password=[password]&
```

```
RequestType=[carrier]>
```
{ RequestId=[number] }

Description

This API can be used to obtain the carrier of a phone number.

Mandatory Parameters

PhoneNumber

The phone number you want to investigate. Note that it must include country code and only digits are accepted, for example:

14805551212 for US and Canada

342475551212 for international

Username

The user name is used to authenticate your account. The same username as created during registration.

Password

The password is used to authenticate your account. The same password as created during registration.

RequestType

Currently only “Carrier” request type is support. Carrier request type responds with all available carrier information.

Optional Parameters

RequestId

If specified, this parameter is returned back in the XML response.

Response Format

The response is attached to HTTP Response that is returned synchronously for every HTTP Request submitted to Secure SMS gateway.

The DTD for XML Response:

```
<!ELEMENT PreviewResponse     (PreviewType,PreviewInfoIError)>
<!ATTLIST PreviewResponse     version     CDATA   #REQUIRED
                              request_id  CDATA   #REQUIRED>
```

```
<!-- Element represents lookup type as defined in request (only “Carrier” in
this version) -->
<!ELEMENT PreviewType (#PCDATA)>
<!ELEMENT PreviewInfo     (NumberString,CarrierInfo)>
<!-- Element represents phone number in international format -->
<!ELEMENT NumberString (#PCDATA)>
<!ELEMENT CarrierInfo
(CarrierId,CarrierName,NetworkType,CountryId,WapPushSupport)>
<!-- Element represents carrier id in CellTrust network -->
<!ELEMENT CarrierId (#PCDATA)>
<!-- Element represents short carrier name -->
<!ELEMENT CarrierName (#PCDATA)>
<!-- Element represents carrier network type ( gsm,cdma,tdma,iden ) -->
<!ELEMENT NetworkType (#PCDATA)>
<!-- Element represents country (two-letter standard abbreviation) -->
<!ELEMENT CountryId (#PCDATA)>
<!-- Element defines if carrier supports wap push ( truelfalse ) -->
<!ELEMENT WapPushSupport (#PCDATA)>

<!ELEMENT Error     (ErrorCode,ErrorId,ErrorMessage?)>
<!-- Element represents internal error code -->
<!ELEMENT ErrorCode (#PCDATA)>
<!-- Element represents internal error string id -->
<!ELEMENT ErrorId (#PCDATA)>
<!-- Element represents error message -->
<!ELEMENT ErrorMessage (#PCDATA)>
```

Notes

There is an additional charge for using the Preview API. Please contact your account manager prior to using this API.

All communication with handset is SMS based, and no data plan is required.

Encryption increases the length of the message, so your account can be charged more credits then if the message is delivered through regular gateway.

Based on regulations, application installed on cell phone cannot initiate any message without explicit confirmation from handset owner. As so, while agent will always try to send delivery and/or read confirmations as requested by customer application, the final result will totally depend on handset owner

To ensure complete security always use HTTPS for sending Secure SMS messages

CellTrust SMS Gateway supports both GET and POST requests

Error Codes

201	wrong or missing username and password
204	missing destination (phone number is required)
206	security violation
207	service disabled
216	secure agent not installed on handset (identified by provided cell phone number)
401	server error
202	preview_service_disabled_for_customer Service disabled for customer

Example

https://gateway.celltrust.net/preview/PreviewConnector?
Username=USERNAME&Password=PASSWORD&
PhoneNumber=14801234567&RequestType=Carrier

The above example requests all available carrier information about phone number: 14801234567.

Success Response Example

<?xml version='1.0'?> <PreviewResponse request_id="12AA34BB" version="1.0.0">
<RequestType>CARRIER</RequestType>
<PreviewInfo>
<NumberString>14801234567</NumberString>
<CarrierInfo>
<CarrierId>3</CarrierId>
<CarrierName>cingular</CarrierName>
<NetworkType>gsm</NetworkType>
<CountryId>US</CountryId>
<WapPushSupport>true</WapPushSupport>
<MMSSupport>true</MMSSupport>
</CarrierInfo>
</PreviewInfo>

```
 </PreviewResponse>
```

Error Response Example

```
<?xml version='1.0'?> <PreviewResponse request_id="12AA34BB"
version="1.0.0">
   <RequestType>CARRIER</RequestType>
   <Error>
    <ErrorCode>305</ErrorCode>
    <ErrorId>preview_carrier_not_found</ErrorId>
    <ErrorMessage>Carrier not found</ErrorMessage>
   </Error>
   <PreviewInfo>
    <NumberString>18005551212</NumberString>
   </PreviewInfo>
 </PreviewResponse>
```

NOTE: Please Refer to the Carier codes Table on page 27 for the list of the Carriers and their Carrier IDs

Sample Code

HTML Example

```
<html><body>
<form name="myform"
action="http://gateway.celltrust.net/preview/PreviewConnector?">
Enroll In Text Messaging services (Enter Your Phone Number) :
<input type='text' name='PhoneNumber'>
<input type='hidden' name='Username' value='USERNAME'>
<input type='hidden' name='Password' value='PASSWORD'>
<input type='hidden' name='RequestType' value='Carrier'>
<A href="javascript: submitform()">Submit</A>
</form>
<SCRIPT language="JavaScript">
function submitform()
{
document.myform.submit();
}
</SCRIPT>
</body></html>
```

Perl Example

```
use strict;
use LWP::UserAgent;
use HTTP::Request::Common;

my $message = '';

my $userAgent = LWP::UserAgent->new();

my $response = $userAgent->request(POST
http://gateway.celltrust.net/preview/PreviewConnector?PhoneNumber
=PHONE&Username=USERNAME&Password=PASSWORD&Requ
estType=carrier');

print $response->error_as_HTML unless $response->is_success;

print $response->as_string;
```

VB Example

```
<%@ Page Language="VB" src="httputils.cs"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>

<%

    'this example uses VB to interact with csharp object. it was originally
    'developed in c#, porting should not be difficult.  should run on basic
    '.net server.  If web service, an d smtp do not work, you can try http
    'post using this class

    'vars for message
    Dim strNickname = "YOURNICKNAME"
    Dim strPassword = "YOURPASSWORD"
    Dim strMessage = Session("Message")
    Dim strDestination = Session("OriginatorAddress")
    Dim strXMLResponse = "true"
    Dim strTemplate = "test"
    Dim strWAPPushURL = "http://google.com"
    Dim strRedirectURL = "http://google.com"
    Dim strAdminEmail = "admin@google.com"
```

```
    Dim strDestPort = "0"
    Dim strShortcode = Session("Shortcode")
    Dim strPricePoint = ""
    Dim strDoubleOptin = "false"

    'response returned by httppost
    Dim strResponse = ""

    'defined in c# source file
    Dim myhttputils As New httputils()
    strResponse =
httputils.HttpPost("https://gateway.celltrust.net/preview/PreviewConne
ctor?", _
    "PhoneNumber=" + strDestination + "&Username=" + strNickname
+ "&Password=" + strPassword + "&ResponseType=Carrier")

    'check the response
    If (strResponse <> "") Then
        Response.Write("Response: " + strResponse)
    Else
        Response.Write("XML Response null or empty!")
    End If
%>

</head>
<body></body>
</html>
```

Note: you can download httputils.cs from
http://www.celltrust.com/docs/httputils.zip

Chapter

3

Web Services

This section describes APIs that are available as Web Services.

A 'Web service' (also Web Service) is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network"[1]. Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

The W3C Web service definition encompasses many different systems, but in common usage the term refers to clients and servers that communicate over the HTTP protocol used on the Web. Such services tend to fall into one of two camps: Big Web Services and RESTful Web Services.

"Big Web Services" use XML messages that follow the SOAP standard and have been popular with traditional enterprise. In such systems, there is often machine-readable description of the operations offered by the service written in the Web Services Description Language (WSDL). The latter is not a requirement of a SOAP endpoint, but it is a prerequisite for automated client-side code generation in many Java and .NET SOAP frameworks (frameworks such as Spring, Apache Axis2 and Apache CXF being notable exceptions). Some industry organizations, such as the WS-I, mandate both SOAP and WSDL in their definition of a Web service. CellTrust Gateway deploys this type of Web Services.

More recently, RESTful Web services have been regaining popularity, particularly with Internet companies. These also meet the W3C definition, and are often better integrated with HTTP than SOAP-based services. They do not require XML messages or WSDL service-API definitions

Send Standard SMS, Email and Voice

Purpose

The purpose of this API is to send standard SMS and email messages from an application.

WSDL

http://gateway.celltrust.net/pmws/services/TxTMessageService?wsdl

Web Service URL

http://gateway.celltrust.net/pmws/services/TxTMessageService

Description

The CellTrust Gateway enables you to send SMS and email messages using CellTrust's platform. An external application prepares all the information needed and submits a request using one of procedure calls defined by the service. The CellTrust Gateway processes the request and sends the message to the required mobile/email destinations. Optionally, you can include an additional administrative email address that will also receive a copy of the message.

Any Web Service call to the CellTrust requires the setting of two additional parameters to the SOAP Header: "Username" and "Password". The values provided with these parameters must correspond to the username and password which are being used to login to CellTrust Gateway Web Portal and will be used for customer authentication. Sample JAVA code when using Axis could look like:

```
SOAPHeaderElement userElement = new SOAPHeaderElement(
XMLUtils.StringToElement(
                "", "Username", "myname") );
call.addHeader( userElement);
SOAPHeaderElement passElement = new SOAPHeaderElement(
XMLUtils.StringToElement(
                "", "Password", "mypassword") );
call.addHeader( passElement);
```

For .NET users, a project can be downloaded from this link:

http://celltrust.com/docs/pmconsole.zip

Note

- Please refer to WSDL for full documentation of each method
- It's very important that input parameters are used in the exact order they appear in the service section (below) and in the WSDL. If they are not provided in the specified order the call will fail.
- All custom objects are defined on the Custom Data Type. All other parameters are typically of type soapenc::string.
- For .Net users using Version 1 of our web services, you need to download and install Microsoft Web Service Enhancements 3.0 then add a reference to it in your project. The ServicePipeline module must also remain in the project. MS WSE 3 can be downloaded from http://www.celltrust.com/docs/MicrosoftWSE3.0.msi
- .Net has some issues with passing authentication parameters in the SOAP Headers, if you run into issues with the SOAP Headers, you can use version 2 of our webservices located at http://gateway.celltrust.net/pmws2/services/TxTMessageService Version 2 of our web services allows the passing of a "Login" object in the web service method call

Services

sendMessage()

Use this method to send SMS, Email, and Voice messages.

Input Parameters: String nickname, String subject, String message, Recipient[] recipients, String adminEmail, ExtraParams params

Output Object: Returns a "Reply" Custom Data Type

sendSMS()

Use this method to send a SMS message to one or more recipients.

Input Parameters: String nickname, String destination, String message, ExtraParams params

Output Object: Returns a "Reply" Custom Data Type

sendEmail()

Use this method to send an Email message to one or more recipients.

Input Parameters: String nickname, String destination, String message, String subject

Output Object: Returns a "Reply" Custom Data Type

scheduleMessage()

Use this method to schedule a message to be sent in the future or setup a recurring message.

Input Parameters: String nickname, String subject, String message, Recipient[] recipients, String adminEmail, SchedulerTask task

Output Object: Returns a "Reply" Custom Data Type

sendWAPPushMessage()

Use this method to send a WAP Push* to a mobile phone.

Input Parameters: String nickname, String text, String wapPushUrl, Recipient[] recipients, String adminEmail

Output Object: Returns a "Reply" Custom Data Type

NOTE: *Some carriers (such as Verizon Wireless in US and other Non-GSM carriers) do not allow sending WAP Push* messages. Please review the glossary at the end of this document for more details*

sendMessageAsTemplate()

Use this method to send a message based on a predefined template.

Input Parameters: String nickname, String subject, String template, TemplateParam[] tags, Recipient[] recipients, String adminEmail, ExtraParams params

Output Object: Returns a "Reply" Custom Data Type

scheduleMessageAsTemplate()

Use this method to schedule a message based on a predefined template.

Input Parameters: String nickname, String subject, String template, TemplateParam[] tags, Recipient[] recipients, String adminEmail, SchedulerTask task

Output Object: Returns a "Reply" Custom Data Type

***Note**: the below web service methods can only be used with Version 2 of our web services and require a "login" object. Please review the WSDL at http://gateway.celltrust.net/pmws2/services/TxTMessageService?wsdl

*cancelScheduledTask()

Use this method to cancel a scheduled message based on a task name provided to the web service when the scheduleMessage method is called using Version 2 of our Web Services.

Input Parameters: Login login, String taskName

Output Object: Returns a "Reply" Custom Data Type

*sendMessageList()

In some cases you may want to control where a message is split. Use this method to break up a large message into separate messages and send them to the gateway in an array.

Input Parameters: Login login, String subject, String[] messages, Recipient[] recipients, ExtraParams params

Output Object: Returns an array of "Reply" Custom Data Type(s)

Custom Data Types

The following custom objects are used by the Campaign Module service:

Login (login)

Represents customer credentials used for authentication and authorization of request. Please note that Login object can only be used with Version 2 of our web services. Contains these fields:

Username
Customer username (currently the same as used to login to your account through the web)

Password
Password for above username

Nickname
Keyword (identifies specific campaign, by default it is your user name)

Recipient (recipients)

This object represents the recipient of the message and contains the following fields:

firstName
Recipient first name (can be used to personalize the message);

lastName
Recipient last name (can be used to personalize the message);

Destination
Address for message delivery (phone number or email address) in string form;

deliveryType
Delivery channel. Only these int values are accepted: 0 – deliver as SMS, 1 – deliver as Email, 3 – deliver as Voice

TemplateParam (tags)

Represents value to be used as template tag replacement. Contains these fields:

Tag
tag name (as used in template);

Value
tag replacement

SchedulerTask (task)

Represents information required to correctly schedule a message to be sent in the future or setup a recurring message. The object contains these fields:

fromYear
The year of first task execution;

fromMonth
The month of first task execution (values in range 0 – 11);

fromDay
The day of first task execution;

fromHour
The hour of first task execution (values in range 0 – 23);

fromMin
The minute of first task execution (values in range 0 – 59);

fromSec
The second of first task execution (values in range 0 – 59);

toYear
The year of last task execution;

toMonth
The month of last task execution;

today
The day of last task execution;

toHour
The hour of last task execution;

toMin
The minute of last task execution;

toSec
The second of last task execution;

numberOfExecutions
Total number of executions of this task;

intervalValue
The value of interval between executions ;

intervalUnit
The unit for intervalValue field. Possible units are:

1 – second;

2 – minute;

3 – hour;

4 – day;

5 – week;

6 – 2 weeks;

7 – month;

8 – 2 months;

9 – half a year;

10 – year

NOTE: All times are local (based on time zone defined in customer profile).

ExtraParams (params)

Miscellaneous parameters for mainly used for Premium messaging. Contains these fields:

destPort

The destination port on cell phone. This is used to direct a message to a specific port on a mobile phone. This requires an application listening to that on the mobile phone and is not supported by Non-GSM devices and networks.

pricePointValue

The price point value (in cents) associated with this SMS message. A price point is an amount that a mobile phone customer will get charged after the content and/or service has been sent to their device successfully. Your short code has to be approved by the carrier for this price point. Once approved CellTrust will configure the appropriate price point for your account.

requireDoubleOptin

It defines if CellTrust should implement double optin when sending premium SMS message. Double opt will send a message to the end user informing them of a charge to their mobile phone bill. Once customer responds with OK, then the content is distributed and their account is charged. Double opt in is required by most US carriers.

short code
The specific short code to be used to deliver SMS message (relevant for dedicated short codes only);

Sender
supported for Emails only – will appear as “ReplyTo” in outgoing email;

NOTES: If using ExtraParams, fill only those parameters relevant for a specific message.

Reply

This object represents response returned by invoked call. Contains these fields:

returnCode
An integer value representing result of operation;

Message
The generated reply string;

smsMessageId
The message ID for SMS message as assigned by the server.

emailMessageId
The message ID for Email message as assigned by the server.

Sample Code

Simple Send SMS – PHP

```
<?php
$URN = "urn:notify.soap.primemessage.com";
$WSDL="http://gateway.celltrust.net/pmws/services/TxTMessageService?wsdl";
//$WSDL="https://gateway.celltrust.net/pmws/services/TxTMessageService?wsdl";

//SOAP elements (don’t edit, and case sensitive!)
$SOAP_ACTION = "sendSMS";
$CTUSERNAME = "Username";
$CTPASSWORD = "Password";
$CTNICKNAME = "nickname";
$DESTINATION = "destination";
```

```
$CARRIERID="carrierId";

$MESSAGE = "message";
$RECIPIENT= "recipients";
$ADMINEMAIL= "adminEmail";
$SUBJECT= "subject";
$PARAMS= "params";



$USER_ID = "USERNAME"; //your username at CellTrust
$NICKNAME = "NICKNAME";          //your nickname at Celltrust
$PASSWORD = "PASSWORD"; //your password at Celltrust
$Message= "Test Message using webservice";

//create user and password SOAP header elements
$UserHdr = new SoapHeader( $URN, $CTUSERNAME, $USER_ID,
false);
$PassHdr = new SoapHeader( $URN, $CTPASSWORD,
$PASSWORD, false);


//create SOAP body elements, add your destination phone number,
and message here

$dialstring = array(array("deliveryType" => 0, "destination" =>
"16024038599", "firstName" => "John", "lastName" => "Doe",
"carrierId" => "5" ));

$body = array($CTNICKNAME => $NICKNAME, $SUBJECT=> "",
$MESSAGE=> $Message,  $RECIPIENT=> $dialstring,
$ADMINEMAIL=> "",  $PARAMS => $params);

//create SOAP client
$client = new SoapClient($WSDL, array('trace' => 1, 'exceptions'
=>0));

//set SOAP headers
$client->__setSoapHeaders(array($UserHdr, $PassHdr));

//call web service
$result = $client->__call($SOAP_ACTION, $body, NULL);
```

```
//check for SOAP error
if (is_soap_fault($result))
{
trigger_error("SOAP Fault: (faultcode: {$result->faultcode}, faultstring:
{$result->faultstring})", E_ERROR);
echo "ERROR\n";
echo $result->faultstring."\n";
}

//very useful for debug purposes
//show what was requested
var_dump($client->__getLastRequest());
//show gateways response
var_dump($client->__getLastResponse());
?>
```

Send Message List – PHP

When a message is larger than the carrier specified length, typically 160 characters, CellTrust Gateway breaks the message up into multiple confirming messages. In some cases you may want to control where a message is split. This example illustrates how to break up a large message into separate message and send them to the gateway in an array. Please note you must use Version 2 of our web services to utilize this method and example.

```
<?php
$URN = "urn:notify.soap.primemessage.com";
$WSDL="http://gateway.celltrust.net/pmws2/services/TxTMessageSer
vice?wsdl";

//SOAP elements (don't edit, and case sensitive!)
$SOAP_ACTION = "sendMessageList";
$CTUSERNAME = "username";
$CTPASSWORD = "password";
$CTNICKNAME = "nickname";
$DESTINATION = "destination";
$MESSAGES = "messages";
$RECIPIENT = "recipients";
$SUBJECT = "subject";

$USERID = "USERNAME"; //your username at CellTrust
$NICKNAME = "NICKNAME"; //your nickname at Celltrust
```

```
$PASSWORD = "PASSWORD"; //your password at Celltrust

$MESSAGES_DATA= array(
"1/5 CellTrust customers using the SMS Gateway can easily take
advantage of the added security that comes with the Secure SMS
Module. Once the parameter has",
"2/5 added, CellTrust can manage and exchange all customer
messages via the highly secure Advanced Encryption Standard (AES)
protocol. To achieve full Secure",
"3/5 SMS Module functionality, a small micro-client application is
installed on the cell phone of the end user. CellTrusts patent-pending
micro-client tech-",
"4/5 nology is the impetus for the Secure SMS Module allowing you to
send and receive communication to and from your target audiences
via the mobile phone with",
"5/5 unprecedented reliability.");

//create SOAP body elements, add your destination phone number,
and message here
$dialstring = array(array("deliveryType" => 0, "destination" =>
"14805551212", "firstName" => "John", "lastName" => "Doe",
"carrierId" => "19"));

$bodymulti = array("Login" => array($CTNICKNAME =>
$NICKNAME, $CTPASSWORD => $PASSWORD, $CTUSERNAME
=> $USERID),
                $SUBJECT => "Test MultiPart Message", $MESSAGES
=>$MESSAGES_DATA, $RECIPIENT => $dialstring);

//create SOAP client
$client = new SoapClient($WSDL, array('trace' => 1, 'exceptions'
=>0));

//call web service single sms
$result = $client->__call($SOAP_ACTION, $bodymulti, NULL);

//very usefull for debug purposes
//show what was requested
var_dump($client->__getLastRequest());
//show gateways response
var_dump($client->__getLastResponse());

?>
```

Send Message to multiple recipients - PHP

This example sends a message to multiple recipients.

```
<?php
$URN = "urn:notify.soap.primemessage.com";
$WSDL="http://gateway.celltrust.net/pmws/services/TxTMessageServ
ice?wsdl";
$encoding = SOAP_DEFAULT_ENCODING;

//SOAP elements (don't edit, and case sensitive!)
$SOAP_ACTION = "sendMessage";
$CTUSERNAME = "Username";
$CTPASSWORD = "Password";
$CTNICKNAME = "nickname";
$DESTINATION = "destination";
$MESSAGE = "message";
$RECIPIENT = "recipients";
$ADMINEMAIL = "adminEmail";
$SUBJECT = "subject";

$USER_ID = "USERNAME"; //your username at CellTrust
$NICKNAME = "NICKNAME"; //your nickname at Celltrust
$PASSWORD = "PASSWORD"; //your password at Celltrust

//create user and password SOAP header elements
$UserHdr = new SoapHeader( $URN, $CTUSERNAME, $USER_ID,
false);
$PassHdr = new SoapHeader( $URN, $CTPASSWORD,
$PASSWORD, false);

//create SOAP body elements, add your destination phone number,
and message here
$dialstring = array(array("deliveryType" => 0, "destination" =>
"14805551212", "firstName" => "John", "lastName" => "Doe",
"carrierId" => "19"),
array("deliveryType" => 0, "destination" => "14805551213",
"firstName" => "Judy", "lastName" => "Doe", "carrierId" => "19"),
array("deliveryType" => 0, "destination" => "14805551214",
"firstName" => "Test", "lastName" => "Doe", "carrierId" => "19"));
```

```
$bodymulti = array($CTNICKNAME => $NICKNAME, $SUBJECT =>
"Test SMS Message", $MESSAGE => "Test SMS Message Multiple
recipients", $RECIPIENT => $dialstring, $ADMINEMAIL = "");

//create SOAP client
$client = new SoapClient($WSDL, array('trace' => 1, 'exceptions'
=>0));

//set SOAP headers
$client->__setSoapHeaders(array($UserHdr, $PassHdr));

//call web service single sms
$result = $client->__call("sendMessage", $bodymulti, NULL);

//very usefull for debug purposes
//show what was requested
var_dump($client->__getLastRequest());
//show gateways response
var_dump($client->__getLastResponse());


?>
```

Send a Scheduled Message – PHP

```
<?php
$URN = "urn:notify.soap.primemessage.com";
$WSDL="http://gateway.celltrust.net/pmws/services/TxTMessageServ
ice?wsdl";
$encoding = SOAP_DEFAULT_ENCODING;

//SOAP elements (don't edit, and case sensitive!)
$SOAP_ACTION = "scheduleMessage";
$CTUSERNAME = "Username";
$CTPASSWORD = "Password";
$CTNICKNAME = "nickname";
$DESTINATION = "destination";
$MESSAGE = "message";
$RECIPIENT = "recipients";
$ADMINEMAIL = "adminEmail";
$SUBJECT = "subject";

$USER_ID = "USERNAME"; //your username at CellTrust
$NICKNAME = "NICKNAME"; //your nickname at Celltrust
$PASSWORD = "PASSWORD"; //your password at Celltrust
```

```
//create user and password SOAP header elements
$UserHdr = new SoapHeader( $URN, $CTUSERNAME, $USER_ID,
false);
$PassHdr = new SoapHeader( $URN, $CTPASSWORD,
$PASSWORD, false);

//create SOAP body elements, add your destination phone number,
and message here
$dialstring = array(array("deliveryType" => 0, "destination" =>
"14805551212", "firstName" => "Jason", "lastName" => "Doe",
"carrierId" => "19"));

$TASK = array("fromDay" => 29, "fromMonth" => 0, "fromYear" =>
2008, "fromHour" => 17, "fromMin" => 59, "fromSec" => 1, "toDay" =>
30, "toMonth" => 0, "toYear" => 2008, "toHour" => 17, "toMin" => 59,
"toSec" => 1, "numberOfExecutions" => 2, "intervalUnit" =>
1,"intervalValue" => 1);

//schedule message arg's: nickname subject message recipients
adminEmail task
$bodymulti = array($CTNICKNAME => $NICKNAME, $SUBJECT =>
"Test SMS Message", $MESSAGE => "Test Schedule SMS Message
1 execution", $RECIPIENT => $dialstring, $ADMINEMAIL = "",
$TASK);

//create SOAP client
$client = new SoapClient($WSDL, array('trace' => 1, 'exceptions'
=>0));

//set SOAP headers
$client->__setSoapHeaders(array($UserHdr, $PassHdr));

//call web service single sms
$result = $client->__call($SOAP_ACTION, $bodymulti, NULL);

//very usefull for debug purposes
//show what was requested
var_dump($client->__getLastRequest());
//show gateways response
var_dump($client->__getLastResponse());

?>
```

Send a Voice Message – PHP

```
<?php
$URN = "urn:notify.soap.primemessage.com";
$WSDL="http://gateway.celltrust.net/pmws/services/TxTMessageServ
ice?wsdl";
$encoding = SOAP_DEFAULT_ENCODING;

//SOAP elements tag names (don't edit, and case sensitive!)
$SOAP_ACTION = "sendMessage";
$CTUSERNAME = "Username";
$CTPASSWORD = "Password";
$CTNICKNAME = "nickname";
$DESTINATION = "destination";
$MESSAGE = "message";
$TEMPLATE = "template";
$RECIPIENT = "recipients";
$ADMINEMAIL = "adminEmail";
$SUBJECT = "subject";

$USER_ID = "YOURUSERNAME"; //your username at CellTrust
//your nickname at Celltrust, this is generally the same as username,
but
//if you have more than one keyword, and are using a template
parameter, it must
//match the keyword the template was attached to
$NICKNAME = "YOURNICKNAME";
$PASSWORD = "YOURPASSWORD"; //your password at Celltrust

//email address where you would like a summary sent
$EMAILADDRESS = "";

//create user and password SOAP header elements
//these have to be present for proper authentication
$UserHdr = new SoapHeader( $URN, $CTUSERNAME, $USER_ID,
false);
$PassHdr = new SoapHeader( $URN, $CTPASSWORD,
$PASSWORD, false);

//create SOAP body elements, this is one or more destination phone
number(s)
//delivery type, specifies type of transport method, 0 = SMS, 1 = email,
2 = not used, 3 = voice
```

```
$recipientArray= array(array("deliveryType" => 3, "destination" =>
"14805551212", "firstName" => "JOHN", "lastName" => "DOE",
"carrierId" => "19"));

//sends an Voice Message using specified template and email
notification to email address
//$bodymulti = array($CTNICKNAME => $NICKNAME, $MESSAGE
=> "Another Test", $SUBJECT => "Test3", $TEMPLATE => "Test3",
$RECIPIENT => $recipientArray, $ADMINEMAIL =
$EMAILADDRESS);

//sends an Voice Message using specified MESSAGE
$bodymulti = array($CTNICKNAME => $NICKNAME, $SUBJECT =>
"", $MESSAGE => "Another Test", $RECIPIENT => $recipientArray);

//create SOAP client
$client = new SoapClient($WSDL, array('trace' => 1, 'exceptions'
=>0));

//set SOAP headers
$client->__setSoapHeaders(array($UserHdr, $PassHdr));

//call web service single sms
$result = $client->__call($SOAP_ACTION, $bodymulti, NULL);

//very usefull for debug purposes
//show what was requested
var_dump($client->__getLastRequest());
//show gateways response
var_dump($client->__getLastResponse());
?>
```

Get Status of a Message

Purpose

The purpose of this API is to get the status of a Message.

WSDL

http://reports.celltrust.net/pmws/services/TxTReportService?wsdl

Web Service URL

http://reports.celltrust.net/pmws/services/TxtReportService

Description

The CellTrust Gateway enables you to track the status of message using the message id returned after a message has been accepted. It may take up to 24 hours for Carriers to update the status of a message.

Note

- Please refer to WSDL for full documentation of each method
- It's very important that input parameters are used in the exact order they appear in the service section (below) and in the WSDL. If they are not provided in the specified order the call will fail.
- All custom objects are defined on the Custom Data Type. All other parameters are typically of type soapenc::string.

Services

getMessageStatusReport()

Use this method to get the status of one or more messages.

Input Parameters: String[ ] messageIDs

Every message excepted by the CellTrust Gateway is assigned a unique message id. These ids are used in getMessageStatusReport API to obtain the current status of the message.

Output Object: Message [ ]

Custom Data Types

The following custom objects are used by this service:

Message

This object represents the message and contains the following fields:

messageId
message ID as assigned by CellTrust Gateway;

text
full message text;

subject
message subject (relevant only for emails);;

deliveryType
delivery channel. Only three possible int values are accepted: 0 – deliver as SMS, 1 – deliver as Email, 3 Voice

parts
the list of Message Part objects representing information for each send event. Note that in case of SMS delivery, messages that exceed max message length will be split into a number of shorter messages

MessagePart

This object represents information for each delivery event. Note that during a SMS delivery, messages which exceed max message length will be split into a number of shorter messages. Also, each split message will be sent independently to each recipient. As a result, one send request to two recipients containing text that is too big to be sent as one message, will be represented in the report as a message with four message parts, where each part contains information about one part of the original message sent to the individual recipient:

partNumber
The sequential number of this message part (starts with 1);

text
The part of the original message sent in this delivery.

destination
Cell phone number or email address of recipient.

status
Current delivery status:

1 – Accepted
3 – Success
4 – Failure (reflects network problems, like wrong cell phone number)
5 – Enroute
6 – Undeliverable (reflects specific cell phone problem, like missing credit for prepaid plan)

reason
Short description of failure reason.

Sample Code

Get Message Status – PHP

```
<?php
$URN = "urn:report.soap.primemessage.com";
$WSDL="http://reports.celltrust.net/pmws/services/TxTReportService
?wsdl";

//SOAP elements tag names (don't edit, and case sensitive!)
$SOAP_ACTION = "getMessageStatusReport";
$CTUSERNAME = "Username";
$CTPASSWORD = "Password";
$CTNICKNAME = "nickname";
$MESSAGETAG = "ids";

$USER_ID = "YOURUSERNAME"; //your username at CellTrust
//your nickname at CellTrust, this is generally the same as username,
but
//if you have more than one keyword, and are using a template
parameter, it must
//match the keyword the template was attached/assigned to
$NICKNAME = "YOURNICKNAME";
$PASSWORD = "YOURPASSWORD"; //your password at CellTrust

//create user and password SOAP header elements
//these have to be present for proper authentication
$UserHdr = new SoapHeader( $URN, $CTUSERNAME, $USER_ID,
false);
$PassHdr = new SoapHeader( $URN, $CTPASSWORD,
$PASSWORD, false);

$MessageIDsArray= array("ids" => "AAXF155120849CT5169");

//sends an Voice Message using specified template
$bodymulti = array($MESSAGETAG => $MessageIDsArray);

//create SOAP client
$client = new SoapClient($WSDL, array('trace' => 1, 'exceptions'
=>0));

//set SOAP headers
$client->__setSoapHeaders(array($UserHdr, $PassHdr));
```

```
//call web service single sms
$result = $client->__call($SOAP_ACTION, $bodymulti, NULL);

print_r($result);
echo("<br/>");
echo($client->__getLastRequest());
echo("<br/>");
echo($client->__getLastResponse());

//very usefull for debug purposes
//show what was requested
//var_dump($client->__getLastRequest());
//show gateways response
//var_dump($client->__getLastResponse());
```

Manage Contacts

Purpose

The purpose of this API is to manage your contacts without using the CellTrust Gateway Web Portal.

WSDL

http://gateway.celltrust.net/pmws/services/ConfigService?wsdl

Web Service URL

http://gateway.celltrust.net/pmws/services/ConfigService

Description

The CellTrust Gateway enables you to manage a list of your contacts from the server. Contacts can be created manually using the CellTrust Gateway Web Portal. Every message that is sent or received using the CellTrust Gateway using API or Web, creates a contact for the recipient or sender a message. These entries will only have the information used to send/receive the message, either email address or phone number.

Contacts can be categorized into groups. Workflows can be configured to put contacts in different groups based on the keyword used in the MO message.

Note

- Please refer to WSDL for full documentation of each method
- It's very important that input parameters are used in the exact order they appear in the service section (below) and in the WSDL. If they are not provided in the specified order the call will fail.
- All custom objects are defined on the Custom Data Type. All other parameters are typically of type soapenc::string.
- CellTrust recommends that you backup/export your contacts and groups prior to utilizing these API's.

Services

getContact()

Use this method to get a list of all contacts or contacts in a group.

Input Parameters: Login login, String groupName

Output Object: ContactList

addContact()

Use this method to add one or more contacts.

Input Parameters: Login login, ContactList contactList, String groupname

Output Object: void

emptyGroup()

Use this method to remove all contacts from a group and optionally permanently delete these contacts from the contacts. The group will remain after its contacts are removed.

Note: if you do not specify a group, all groups are emptied. If deleteContacts is true, then contacts will also be deleted from your contact list.

Tip: To remove a contact from a group, you need to retrieve the current list of contacts in the group using getContact(), then empty the group using emptyGroup(**remember to specify a group**), edit the list of contacts previously retrieved to contain only the contacts desired, then

use addContact() with a contact list to re-populate the group with the correct contacts.

Input Parameters: Login login, String groupName, Boolean deleteContacts

Output Object: void

Custom Data Types

The following custom objects are used by the Campaign Module service:

Contact (contact)

This object represents the contact and contains the following fields:

firstName
Contact's first name;

lastName
Contact's last name;

email
Email address of the contact;

phone
Phone number of the contact

Login (login)

This object holds the authentication information and contains these fields:

username
This is the username that was selected during registration;

password
This is the password set during registration. Your password can be changed using the CellTrust Gateway Web Portal.

nickname
This is the default keyword. Typically nickname is same as the username unless it has been changed.

ContactList (contactlist)

This is object is an array of contact objects.

Sample Code

Adding multiple contacts – PHP

```
<?php
//import statement, for service class
require_once 'PMConfigServiceService.php';

//the login object
class Login {
  public $nickname = "YOURNICKNAME"; // string
  public $password = "YOURPASSWORD"; // string
  public $username = "YOURUSERNAME"; // string
}

//a contact object example
class Contact {
  public $email = "julieRay@gmail.com";     // string
  public $firstName = "julie"; // string
  public $lastName = "Ray";  // string
  public $phone = "";     // string
}

//another contact object example
class Contact2 {
  public $email = "tulieDehoolie@gmail.com";     // string
  public $firstName = "tulie"; // string
  public $lastName = "Dehoolie";  // string
  public $phone = "14801234567";     // string
}

//init objects, only one required is "Login"
$_login = new Login();
$_contact = new Contact();
$_contact2 = new Contact2();

//an array containing just one recipients data
$recipient = array("email" => "george5@gmail.com","firstName" =>
"george","lastName" => "long","phone" => "13434567890");
$recipient2 = array("email" => "jimmy5@gmail.com","firstName" =>
"jam","lastName" => "short","phone" => "13436547890");

//an array containing 2 recipients
$recipients = array($recipient, $recipient2) ;
```

```
// create a new service object, service object does all dirty work of
communicating with server
$service = new PMConfigServiceService();

try
{
  //set the headers for the service/soap call, this is required!
  $service->setHeaders();


  echo("Adding  Two Contacts:\n");
  print("Status (0 success, 1 failure, 2 partial failure):" .$service-
>addContacts($_login, $recipients, "test-3"));

  //called in any of above (group add) scenarios, will not get here if
soap exception is thrown
  echo("\nSuccessfully Added Contacts\n");


}
catch(Exception $e)
{
  echo("\Exception is:\n" .$e);
}
?>
```

Get a list of Contacts – PHP

```
<?php
//import statement, for service class
require_once 'PMConfigServiceService.php';

//the login object
class Login {
  public $nickname = "YOURNICKNAME"; // string
  public $password = "YOURPASSWORD"; // string
  public $username = "YOURUSERNAME"; // string
}

//a contact object example
class Contact {
  public $email = "julieRay@gmail.com";     // string
  public $firstName = "julie"; // string
  public $lastName = "Ray";  // string
```

```
  public $phone = "";     // string
}

//another contact object example
class Contact2 {
  public $email = "tulieDehoolie@gmail.com";     // string
  public $firstName = "tulie"; // string
  public $lastName = "Dehoolie";  // string
  public $phone = "14801234567";     // string
}

//init objects, only one required is "Login"
$_login = new Login();
$_contact = new Contact();
$_contact2 = new Contact2();

//an array containing just one recipients data
$recipient = array("email" => "george5@gmail.com","firstName" =>
"george","lastName" => "long","phone" => "13434567890");
$recipient2 = array("email" => "jimmy5@gmail.com","firstName" =>
"jam","lastName" => "short","phone" => "13436547890");

//an array containing 2 recipients
$recipients = array($recipient, $recipient2) ;

// create a new service object, service object does all dirty work of
communicating with server
$service = new PMConfigServiceService();

try
{
  //set the headers for the service/soap call, this is required!
  $service->setHeaders();

   /*
   /Uncomment, to list all contacts in group, in this example, "OPT-iN"
is a group, login is login credential object
   /returns a contactlist array, which is printed to system out using
"print_r"
  */
  echo("Group Contacts:\n");
  print_r($service->getContacts($_login, "OPT-iN"));
```

```
}
catch(Exception $e)
{
  echo("\Exception is:\n" .$e);
}
?>
```

Remove Contacts from a Group

This example demonstrates how to remove all contacts from a group and delete all contacts permanently. Please note that if you set the group name to NULL, then all groups and contacts are deleted. In the example below, contacts in group “test-3” is removed permanently.

```
<?php
//import statement, for service class
require_once 'PMConfigServiceService.php';

//the login object
class Login {
  public $nickname = "YOURNICKNAME"; // string
  public $password = "YOURPASSWORD"; // string
  public $username = "YOURUSERNAME"; // string
}

//a contact object example
class Contact {
  public $email = "julieRay@gmail.com";     // string
  public $firstName = "julie"; // string
  public $lastName = "Ray";  // string
  public $phone = "";     // string
}

//another contact object example
class Contact2 {
  public $email = "tulieDehoolie@gmail.com";     // string
  public $firstName = "tulie"; // string
  public $lastName = "Dehoolie";  // string
  public $phone = "14801234567";     // string
}

//init objects, only one required is "Login"
$_login = new Login();
$_contact = new Contact();
```

```
$_contact2 = new Contact2();

//an array containing just one recipients data
$recipient = array("email" => "george5@gmail.com","firstName" =>
"george","lastName" => "long","phone" => "13434567890");
$recipient2 = array("email" => "jimmy5@gmail.com","firstName" =>
"jam","lastName" => "short","phone" => "13436547890");

//an array containing 2 recipients
$recipients = array($recipient, $recipient2) ;

// create a new service object, service object does all dirty work of
communicating with server
$service = new PMConfigServiceService();

try
{
  //set the headers for the service/soap call, this is required!
  $service->setHeaders();

  /*
   /Uncomment, to , to empty a specific group of it's contacts, while
removing said contacts from contacts list
  */
  echo("Empty Group, removing from contacts:\n");
  print("Status (0 success, 1 failure, 2 partial failure):" .$service-
>emptyGroup($_login, "test-3", 1));
}
catch(Exception $e)
{
  echo("\Exception is:\n" .$e);
}
?>
```

Carrier Preview

Purpose

This API can be used to determine if a phone number is a valid mobile phone number. It can also be used to determine the carrier servicing a given phone number and also get details about the carrier, i.e. network

type(GSM/CDMA), if it supports MMS, WAP push. This API is currently only available in United States.

Description

This API can be used to obtain the carrier of a phone number.

Mandatory Parameters

PhoneNumber

The phone number you want to investigate. Note that it must include country code and only digits are accepted, for example:

14805551212 for US and Canada

342475551212 for international

Username

The user name is used to authenticate your account. The same username as created during registration.

Password

The password is used to authenticate your account. The same password as created during registration.

Nickname

The Nickname is used to authenticate your account. Any Nickname associated with your account can be used.

Notes

There is an additional charge for using the Preview API. Please contact your account manager prior to using this API.

All communication with handset is SMS based, and no data plan is required.

Based on regulations, application installed on cell phone cannot initiate any message without explicit confirmation from handset owner. As so, while agent will always try to send delivery and/or read confirmations as requested by customer application, the final result will totally depend on handset owner

CellTrust SMS Gateway supports both GET and POST requests

Error Codes

201	wrong or missing username and password
204	missing destination (phone number is required)
206	security violation
207	service disabled
216	secure agent not installed on handset (identified by provided cell phone number)
401	server error
202	preview_service_disabled_for_customer Service disabled for customer

Sample PHP Example

```
<?php

class Login
{
  public $nickname = ""; // string
  public $password = ""; // string
  public $username = ""; // string
}


//init objects, only one required is "Login"
$_login = new Login();

//login credentials: registered with CellTrust Corporation
$_login->username = "Your CellTrust Username";
$_login->nickname = "Your CellTrust Nickname";
$_login->password = "Your CellTrust password";

// enter here the phone number to preview
$destination = "PHONENUMBER";

//$URN = "urn:notify.soap.primemessage.com";
$URN =
"urn:PreviewServlet.preview.communication.primemessage.eclerisy.com";
```

```
$WSDL="http://gateway.celltrust.net/pmws2/services/CarrierPreview?
wsdl";

//SOAP elements
$SOAP_ACTION = "previewNumber";
$DESTINATION = "number";
$LOGIN = "login";

//ready body for lookup
$bodymulti = array($LOGIN => $_login, $DESTINATION =>
$destination);

//create SOAP client
$client = new SoapClient($WSDL, array('trace' => 1, 'exceptions'
=>0));

//call web service single sms
$result = $client->__call($SOAP_ACTION, $bodymulti, NULL);

echo $client->__getLastResponse();

?>
```

Response Format

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"xmlns:x
sd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body><ns1:previewNumberResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://pmws.communication.primemessage.eclerisy.com"
>
<previewNumberReturn
href="#id0"/></ns1:previewNumberResponse>
   <multiRef id="id0" soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns2:NumberPreviewResponse"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns2="urn:bean.notify.soap.primemessage.com">
<carrierId xsi:type="xsd:int">5</carrierId>
<carrierName xsi:type="xsd:string">sprint</carrierName>
<countryId xsi:type="xsd:string">US</countryId>
```

```
<mmsSupport xsi:type="xsd:boolean">false</mmsSupport>
<networkType xsi:type="xsd:string">cdma</networkType>
<numberString xsi:type="xsd:string">16024038599</numberString>
<wapPushSupport xsi:type="xsd:boolean">false</wapPushSupport>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

Track a Flight

Purpose

The purpose of this API is to subscribe one or more users to get flight status information sent to them via SMS.

WSDL

http://fm.celltrust.com/flight/services/FlightSubscriptionService?wsdl

Web Service URL

http://fm.celltrust.com/flight/services/FlightSubscriptionService

Description

The CellTrust Gateway provides flight tracking APIs. You can subscribe a phone number to receive real time flight status updates. A message is sent to a subscriber, if a flight is delayed, cancelled, and re-routed. Subscriber is also informed when the flight departs and arrives. In addition, reminders can be setup to send a message to the subscriber to remind them of their upcoming flight.

Note

- Please refer to WSDL for full documentation of each method
- It's very important that input parameters are used in the exact order they appear in the service section (below) and in the WSDL. If they are not provided in the specified order the call will fail.
- All custom objects are defined on the Custom Data Type. All other parameters are typically of type soapenc::string.

Services

subscribe()

Use this method to monitor a flight.

Input Parameters: FlightSubscriptionRequest

Output Object: FlightSubscriptionResponse

cancel()

Use this method to cancel a subscription

Input Parameters: Login login, ContactList contactList, String groupname

Output Object: void

Custom Data Types

The following custom objects are used by the Campaign Module service:

RequestMetaData

This object represents the request id. Request id is used to pass a request id to the Subscribe web service. The request id can be used later to cancel a scheduled flight. It contains the following fields:

requestId
string - Customer-generated ID - will be returned in response;

version
string - Current version "1.0";

Login (login)

This object holds the authentication information and contains these fields:

username
This is the username that was selected during registration;

password
This is the password set during registration. Your password can be changed using the CellTrust Gateway Web Portal.

accountId

This is the account id assigned by CellTrust, once flight management account has been setup. Please note this is different than your CellTrust Gateway account used for sending messages.

QuietWindow

This defines the time window during which reminder messages should not be sent.

allowCritical

boolean - if "true" - notifications will be sent even in quiet period;

endDate

dateTime - end of the time window.

startDate

dateTime - beginning of the time window

threshold

int - change in departure/arrival time that should be considered critical event (optional parameter)

Reminder

Reminders are messages that are sent to the subscriber to remind them of their flight and is not generated based flight status or gate change.

delta

int - time interval before departure/arrival to trigger notification;

notifDate

dateTime - absolute time to trigger notification - has precedence over "delta".

watchFor

int - 0 - delta relative to departure / 1 - delta relative to arrival

NotificationConfig

This object includes Quiet Window and reminder configuration configurations.

quietWindows

Quiet Window configuration, see above;

reminders

Reminder configuration, see above;

timezoneOffset
int - Timezone offset, like "-5" for EST"

Error

This object defines the error and is included FlightSubscriptionResponse.

errorCode
int – error code;

message
String – error message;

stringId
String – error id

FlightSubscriptionRequest

This object defines the flight to be monitored. All information has to be validated before filling in this object.

airlineCode
String – two letter airline code. For example AA for American Airlines;

arrivalAirportId
String – three letter airport code. For example SFO for San Francisco International Airport;

arrivalScheduledDate
dateTime – Scheduled arrival date and time. For example, 2008-02-02T17:40:00. Please note the “T” separating date from time.

departAirportId
String – two letter airport code for the departing city. For example SFO for San Francisco International Airport;

departDate
String – three letter airport code. For example SFO for San Francisco International Airport;

departScheduledDate
dateTime – Scheduled departure date and time. For example, 2008-02-02T17:40:00. Please note the “T” separating date from time.

Destination
String – phone number of the subscriber where all messages are sent to;

flightNumber
String – Flight number of the flight to be monitored;

login
Login – Authentication Object, see above

metaData
RequestMetaData – Request object, see above

msgNumber
int – Maximum number of messages sent for a flight

notifConfig
NotificationConfig – Notification object, see above

watchFor
int – 0 for departure or 1 for arrival

FlightSubscriptionResponse

This object is returned in response to a flight subscription request.

airlineCode
String – two letter airline code. For example AA for American Airlines;

arrivalAirportId
String – three letter airport code. For example SFO for San Francisco International Airport;

departAirportId
String – two letter airport code for the departing city. For example SFO for San Francisco International Airport;

departDate
String – three letter airport code. For example SFO for San Francisco International Airport;

Destination
String – phone number of the subscriber where all messages are sent to;

flightNumber
String – Flight number of the flight to be monitored;

errors
Error – list of errors returned by subscribe method

metaData
RequestMetaData – Request object, see above

flightRequestID
String – Request Id assigned to this flight

FlightCancelRequest

This object is used to cancel a subscription.

flightRequestId
String – this id is returned by the subscribe method when a subscription is successful Request id can be set in the object to cancel a subscription.

metaData
RequestMetaData – Request object, see above

login
Login – Authentication Object, see above

FlightCancelResponse

This object is returned by the cancel subscription method.

flightRequestId
String – this id is returned by the subscribe method when a subscription is successful Request id can be set in the object to cancel a subscription.

metaData
RequestMetaData – Request object, see above

errors
Error – list of errors returned by subscribe method

Sample Code

Subscribe for a flight – PHP

```
<?php

//populate login object
$_Login = new Login();
$_Login -> accountId = "YOURACCOUNTID";
```

```
$_Login -> password = "YOURPASSWORD";
$_Login -> username = "YOURUSERNAME";

//populate RequestMetaData object
$_RequestMetaData = new RequestMetaData();
$_RequestMetaData -> requestId = "ARequestXXXXXX";
$_RequestMetaData -> version = "1.0";

//populate QuietWindow object
$_QuietWindow = new QuietWindow();
$_QuietWindow -> allowCritical = true; //bool
$_QuietWindow -> endDate = ""; //date/time
$_QuietWindow -> startDate = ""; //dateTime
$_QuietWindow -> threshold = ""; //int

//populate Reminder object
$_Reminder = new Reminder();
$_Reminder -> delta = 30;//int - minutes - in example
//$_Reminder -> notifDate = ""; //date time = null in example!!!
$_Reminder -> watchFor = 2; //int - delta relative to departure

//populate NotificationConfig object using QuietWindow and Reminder
Objects
$_NotificationConfig = new NotificationConfig();
//$_NotificationConfig -> quietWindows = $_QuietWindow;//object -
null in example!!!
$_NotificationConfig -> reminders = $_Reminder;//object
$_NotificationConfig -> timezoneOffset = -7;//int

//populate FlightSubscriptionRequest object, using preceeding objects
$_FlightSubscriptionRequest = new FlightSubscriptionRequest();
$_FlightSubscriptionRequest -> airlineCode = "AA"; // string - in
example
$_FlightSubscriptionRequest -> arrivalAirportId = "DFW"; // string - in
example
$_FlightSubscriptionRequest -> arrivalScheduledDate = "2008-12-
06T16:55:00";// dateTime
$_FlightSubscriptionRequest -> departAirportId = "PHX"; // string  - in
example
$_FlightSubscriptionRequest -> departDate = "2008-12-
06T14:00:00";// dateTime
$_FlightSubscriptionRequest -> departScheduledDate = "2008-12-
06T13:40:00";// dateTime
```

```
$_FlightSubscriptionRequest -> destination = "14805551212"; // string
- end user phone
$_FlightSubscriptionRequest -> flightNumber = "2094"; // string - in
example
$_FlightSubscriptionRequest -> login = $_Login; //object
$_FlightSubscriptionRequest -> metaData =
$_RequestMetaData;//object - null in example!!!
$_FlightSubscriptionRequest -> msgNumber = 6; //int  - per sample
$_FlightSubscriptionRequest -> notifConfig =
$_NotificationConfig;//object
$_FlightSubscriptionRequest -> watchFor = 2;//int - per sample - 0
departure / 1 arrival
//!!!SHOULD BE GOOD

try
{
  // create a new service object, service object does all dirty work of
communicating with server
  $service = new FlightSubscriptionServiceService();
  echo("Created Service\n");
  //ready response object
  $response = new FlightSubscriptionResponse();
  echo("Created Response:\n");

  echo("Invoking Subscriber:\n");
  $response = $service->subscribe($_FlightSubscriptionRequest);

  //uncomment to dump full response object for debug
  //print_r ($response);

  //check for result errors
  $_errors = $response->errors;
  if(is_array($response->errors))
  {
          print ("Error returned by web service API call!!!"."\n");
          print ("Servers Error Code: ".$_errors[0]->errorCode."\n");
          print ("Servers Error Message: ".$_errors[0]->message."\n");
          print ("Servers String Id: ".$_errors[0]->stringId."\n");
          print ("Error with your request: "."\n");
  }
  else
  {
          print ("No errors returned by web service API call."."\n");
```

```
  }
  print("Tracking airline: ".$response->airlineCode."\n");
  print("Arriving at: ".$response->arrivalAirportId."\n");
  print("Departing from: ".$response->departAirportId."\n");
  print("Departing on: ".$response->departDate."\n");
  print("Alerts will be sent to phone: ".$response->destination."\n");
  print("Tracking flight: ".$response->flightNumber."\n");

}
catch(Exception $e)
{
  echo("\Exception is:\n" .$e);
}

//////////////////  Code generated from WSDL
////////////////////////////////////////////////
class Login {
  public $accountId; // string - account ID assigned to you by CellTrust
  public $password; // string - password assigned to you by CellTrust
  public $username; // string - username as assigned to you by
CellTrust
}

class RequestMetaData {
  public $requestId; // string - Customer-generated ID - will be returned
in response
  public $version; // string - Current version "1.0"
}

//window during which alerts should not be sent, unless they are
deemed critical
class QuietWindow {
  public $allowCritical; // boolean - if "true" - notifications will be sent
even in quiet period
  public $endDate; // dateTime - end of the window
  public $startDate; // dateTime - beginning of the window
  public $threshold; // int -  change in departure/arrival time that should
be considered critical event (optional parameter)
}

//i.e. if set for 1/2 hour, user would receive alerts 1/2 an hour before
arrival/departure
class Reminder {
```

```
  public $delta; // int - time interval before departure/arrival to trigger
notification
  public $notifDate; // dateTime - absolute time to trigger notification -
has precedence over "delta"
  public $watchFor; // int - 0 - delta relative to departure / 1 - delta
relative to arrival
}

//populate with QuietWindow and Reminder Objects
class NotificationConfig {
  public $quietWindows; // QuietWindowList
  public $reminders; // ReminderList
  public $timezoneOffset; // int - Timezone offset, like "-5" for EST"
}

class FlightSubscriptionRequest {
  public $airlineCode; // string
  public $arrivalAirportId; // string
  public $arrivalScheduledDate; // dateTime
  public $departAirportId; // string
  public $departDate; // dateTime
  public $departScheduledDate; // dateTime
  public $destination; // string
  public $flightNumber; // string
  public $login; // Login
  public $metaData; // RequestMetaData
  public $msgNumber; // int
  public $notifConfig; // NotificationConfig
  public $watchFor; // int - 0 departure / 1 arrival
}

//Possible errors that can currently be returned by Flight Monitor in
response are listed below:
//106 (missing_destination_error) - Cell phone not provided
//107 (depart_date_in_the_past_error) - Departure Date is more then
one day in the past
//108 (missing_quiet_window_date_error) - Start or End time missing
for Quiet Window configuration
//109 (missing_user_timezone_error) - Missing Subscriber timezone
offset
//201 (data_not_found_error) - Flight not found
//302 (no_credit_error) - No credits for prepaid account
//303 (unknown_service_provider_error) - Wrong Account ID
```

```
//501 (flight_security_error) - Wrong Username/Password
class Error {
  public $errorCode; // int - 'unique error code'
  public $message; // string - error message
  public $stringId; // string - unique string id (can be used for
localization)
}

class FlightSubscriptionResponse {
  public $airlineCode; // string
  public $arrivalAirportId; // string
  public $departAirportId; // string
  public $departDate; // dateTime
  public $destination; // string
  public $errors; // ErrorList
  public $flightNumber; // string
  public $metaData; // RequestMetaData
}


/**
 * FlightSubscriptionServiceService class
*
 * @author    Jason Glass
 * @copyright 2008 - CellTrust Corporation
 */
class FlightSubscriptionServiceService extends SoapClient {

  private static $classmap = array(
                                'Login' => 'Login',
                                'RequestMetaData' => 'RequestMetaData',
                                'QuietWindow' => 'QuietWindow',
                                'Reminder' => 'Reminder',
                                'NotificationConfig' => 'NotificationConfig',
                                'FlightSubscriptionRequest' =>
'FlightSubscriptionRequest',
                                'Error' => 'Error',
                                'FlightSubscriptionResponse' =>
'FlightSubscriptionResponse',
                               );
```

```
  public function FlightSubscriptionServiceService($wsdl =
"http://fm.celltrust.com/flight/services/FlightSubscriptionService?wsdl",
$options = array()) {
    foreach(self::$classmap as $key => $value) {
      if(!isset($options['classmap'][$key])) {
        $options['classmap'][$key] = $value;
      }
    }
    parent::__construct($wsdl, $options);
  }

  /**
   * @param FlightSubscriptionRequest $request
   * @return FlightSubscriptionResponse
   */
  public function subscribe(FlightSubscriptionRequest $request) {
    return $this->__soapCall('subscribe', array($request),       array(
            'uri' =>
'http://fm.celltrust.com/flight/services/FlightSubscriptionService',
            'soapaction' => ''
           )
      );
  }

}
////////////////// Code generated from WSDL
////////////////////////////////////////////////////
?>
```

Chapter

4

Java SDK

Add SMS messaging to your JAVA application in minutes using CellTrust Gateway Java SDK.

The CellTrust Gateway Java SDK enables customer to integrate with CellTrust Gateway using Java technology. Java SDK is aimed at Java developers that wish to send SMS messages or email from their existing Java code. CellTrust Gateway Java SDK supports both Java version 4 and 5. You can download the appropriate files from the CellTrust Gateway Web Portal under help.

The functionality of sending MT messages through the CellTrust Gateway is integrated inside a single CellTrust class. This class provides the interface to the underlying infrastructure and hides the communication details.

The CellTrust Java SDK is distributed as single pmclient.jar file. Submission of every message request requires user authentication which requires a valid account with CellTrust in order to use this package.

You can send a message to any number of recipients or recipient groups. However, you can use only one delivery type (SMS or email) at one time. By default, the delivery type is set to SMS. You can modify this delivery type by passing PMDeliveryType.SMS or PMDeliveryType.EMAIL constants to the setDeliveryType(...) method.

You can reuse the same PrimeMessage object to send another message (or to send the same message using different delivery types). If you need to configure different recipients, use clearGroupList() and clearRecipientList() prior to adding new recipients

The CellTrust core system keeps user passwords in an encrypted form. The password provided with sendMessage(...) method will be encrypted before it is sent to the server.

Instead of using simple text message, you can use a predefined template (the template must be defined and saved using the CellTrust Gateway Web portal application before the real message is sent). You can override the template using internal template ID or template name. Using setTemplateParams(...) method you can also override a set of values for the tags defined inside the template. The Hashtable passed has to include tag-value pairs, where the tag represents a specific tag defined inside the template and the value is the substitute for this tag

You can send a "WAP Push"* message and it will arrive as a link to specific web site (not all cell phones and carriers support this feature). To set the message as a "WAP Push"* message, simply add a URL using setWapPushUrl(..) method. The message will then become the link pointing to the provided URL.

NOTE: *Some carriers (such as Verizon Wireless in US and other Non-GSM carriers) do not allow sending WAP Push* messages. Please review the glossary at the end of this document for more details*

If debug mode is set to true using setDebugMode(true) method, some output will be printed to stdout.

You can send a message to a predefined port on the mobile phone by using the setDestPort(..) method. This feature is not supported by all carriers.

You can also send premium messages using setPricePointValue(..) to define price point. NOTE: Price points must be preconfigured in the customer account.

You can check the status of the last request that was sent using getOperationStatus(). Possible values are PMDeliveryStatus.ACCEPTED (1) or PMDeliveryStatus.FAILURE (4). In case of failure you can read the message returned by CT Gateway using getOperationErrorMessage()

Send Message

Description

This method is the main method in this class. All the relevant parameters have to be set using the methods described below. This method sends a message. User name and password have to be set. The message will be delivered only if authentication was successful. Username and password are assigned during registration.

Syntax

public String **sendMessage**(String userName, String userPasswd) throws PMException

Parameters

`userName` - User Name (used for authentication)
`userPasswd` - User Password (used for authentication)

Returns

Message Id as assigned by server

Throws

PMException

Get Operation Status

Description

This method is used to determine the status of the message. This is the initial status and should not be confused with message delivery status. Operation status will indicate if CellTrust Gateway accepted the message for delivery or if it was refused.

Syntax

public int getOperationStatus()

NOTE: The ACCEPTED status only means that the request was successfully validated and message is accepted for delivery. It doesn't promise that message is successfully delivered to recipient. To get the final message status use reporting tools (CellTrust Gateway Web Protocol or Reporting Module).

NOTE: It can take up to 24 hours to get final confirmation from the carrier.

Returns:

status of the last send operation. Operation status of the send operation can be or PMDeliveryStatus.ACCEPTED (1), or PMDeliveryStatus.FAILURE (4).

Get Operation Error Message

Description

If the getOperationStatus() returns PMDeiveryStatus.FAILURE, then this method can be used to obtain further information from CellTrust Gateway.

Syntax

public String getOperationErrorMessage()

Returns

operation error message as returned by CellTrust Gateway

Add Group Name

Description

Add a contact group by name. Group must be defined using CellTrust Gateway Web Portal. A message can be sent to one or more groups of contacts.

Syntax

public void **addGroupName**(String groupName)

Parameters

groupName – the name of predefined contact group

Add Recipient

Description

This method adds a recipient. All recipients will receive the message sent using sendMessage() method.

Syntax

public void **addRecipient**(String firstName, String lastName, int mobileCountryCode, int mobileAreaCode, int mobileNumber)

public void **addRecipient**(String firstName, String lastName, String destination)

Usage:
addRecipient("John", "Doe", 1, 480, 5551212)
addRecipient("John", "Doe", "14805551212")
addRecipient("John", "Doe", "John.Doe@ADomain.Com")

Parameters

`firstName` - First Name
`lastName` - Last Name
`mobileCountryCode` - Mobile phone country code
`mobileAreaCode` - Mobile phone area code
`mobileNumber` - Mobile phone number
destination – Email address or full phone number including country and area code

Clear Group List

Description

This method clears all previously added groups..

Syntax

public void **clearGroupList**()

Parameters

None

Clear Recipient List

Description

This method clears all previously added recipients.

Syntax

public void clearRecipientList()

Parameters

None

Customer Abbreviation

Description

These methods set/get the customer abbreviation. Customer Abbreviation (Nickname) is assigned during registration and it's a required parameter.

Syntax

public void **setCustomerAbbrev**(String abbrev)
public String **getCustomerAbbrev**( )

Parameters

abbrev – Nickname, usually the same as your username

Delivery Type

Description

These methods set/get the delivery type. Currently supported type are:

- PMDeliveryType.SMS (0)
- PMDeliveryType.EMAIL (1)

Syntax

public void **setDeliveryTypeId**(int delTypeId)
public int getDeliveryTypeId( )

Parameters

delTypeId – Delivery Type

Debug Mode

Description

These methods set/get the debug mode. If set to true, then additional debug information will be sent to stdout.

Syntax

public void **setDebugMode**(Boolean debugmode)
public boolean **getDebugMode**( )

Parameters

debugmode – enable/disable debug mode

Destination Port

Description

This parameter specifies the destination port on the mobile phone. Typically this is used to "wake up" an application listening on a specific port on a mobile device. This is referred to as PushRegistry on J2ME phones. Not all mobile phones support sending a SMS message to a port.

Syntax

public void **setDestPort**(int port)
public int **getDestPort**( )

Parameters

port – the port of the mobile phone where the receiving application is bound to or listening on.

Message Body

Description

These methods set/get the message to be sent to the recipient or group list.

Syntax

public void **setMessageBody**(String msg)
public String **getMessageBody**( )

Parameters

Msg – message to be delivered to one or more recipients

Message Subject

Description

These methods set/get the subject of an email. This is ignored for a SMS message.

Syntax

public void **setMessageSubject**(String subject)
public String **getMessageSubject**( )

Parameters

subject – subject of the email

Price Point Value

Description

These methods set/get the price point value for premium messages in cents (like 199 for $1.99 price point). Your account must be enabled for price points for this to work. The short code specified must be approved for a premium program for this feature to be active.

Syntax

public **setPricePointValue**(int value)
public int getPricePointValue( )

Parameters

value – dollar amount in cents

Sender

Description

These methods set/get the sender of the email message. This method does not apply to SMS.

Syntax

public void setSender(String sender)
public String getSender( )

Parameters

sender – email address of the sender

Shortcode

Description

These methods set/get the originating shortcode to be used for delivery of the message. This method is only relevant to customers that are using a dedicated shortcode.

Syntax

public void **setShortcode**(String shortcode)
public String **setShortcode**( )

Parameters

shortcode – shortcode for message delivery

Template Name

Description

These methods set/get the name of template to be used for this message. For a description of templates refer to the template section.

Syntax

public void **setTemplateName**(String name)
public String **getTemplateName**( )

Parameters:

`name` – Template Name

Template Params

Description

These methods are used to provide and obtain the parameters and values to be used in the message. Please refer to Template section for more information about templates.

Syntax

public void **setTemplateParams**(Hashtable params)
public Hashtable **getTemplateParams**( )

Parameters

`params` – Hashtable containing attribute-value pairs

WAP Push Url

Description

These methods are used to set/get the URL to be delivered to phone in a WAP push*.

NOTE: *Some carriers (such as Verizon Wireless in US and other Non-GSM carriers) do not allow sending WAP Push* messages. Please review the glossary at the end of this document for more details*

Syntax

public void **setWapPushUrl**(String url)
public String **getWapPushUrl**( )

Parameters

`url` – URL for WAP Push

Java SDK Sample Code

This example sends a SMS message (“Hello”) to one recipient (John Doe 14805551212).

```
/*
 * Main.java
 *
 * Created on December 4, 2008, 5:10 PM
 *
 */

package pmsdk;
import com.eclerisy.primemessage.sdk.PrimeMessage;
import com.eclerisy.primemessage.common.PMDeliveryType;
import com.eclerisy.primemessage.common.PMDeliveryStatus;
import com.eclerisy.primemessage.common.PMConstant;

/**
 *
 * @author user
 */
public class Main {

    /** Creates a new instance of Main */
    public Main() {
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        try{
            PrimeMessage pm = new PrimeMessage(“gateway.celltrust.net");
            pm.setCustomerAbbrev( "YOURNICKNAME");
            //!!Note, generally CustomerAbbrev Username and
NickName/Keyword are all the same, use this value for sUser
```

```
            pm.setMessageBody( "Hello");
            //sends to phone destination, specifying country code, area code
and phone
    //pm.addRecipient( "Joe", "Doe", 1, 480, 5551212);
   //sends to an email destination
              //pm.addRecipient( "Joe", "Doe", “Joe.Doe@MyDomain.Com”);
              //sends to a phone destination by specifying the full phone number
            pm.addRecipient( "Joe", "Doe", “14805551212”);

            pm.sendMessage( "YOURUSERNAME", "YOURPASSWORD");
            int status = pm.getOperationStatus();
            if( PMConstant.STATUS_SUCCESS != status)
            {
               String msg = pm.getOperationErrorMessage ();
            }
        }
        catch(Exception e){
            System.out.println("Error");
        }
    }

}
```

Chapter

5

SMTP

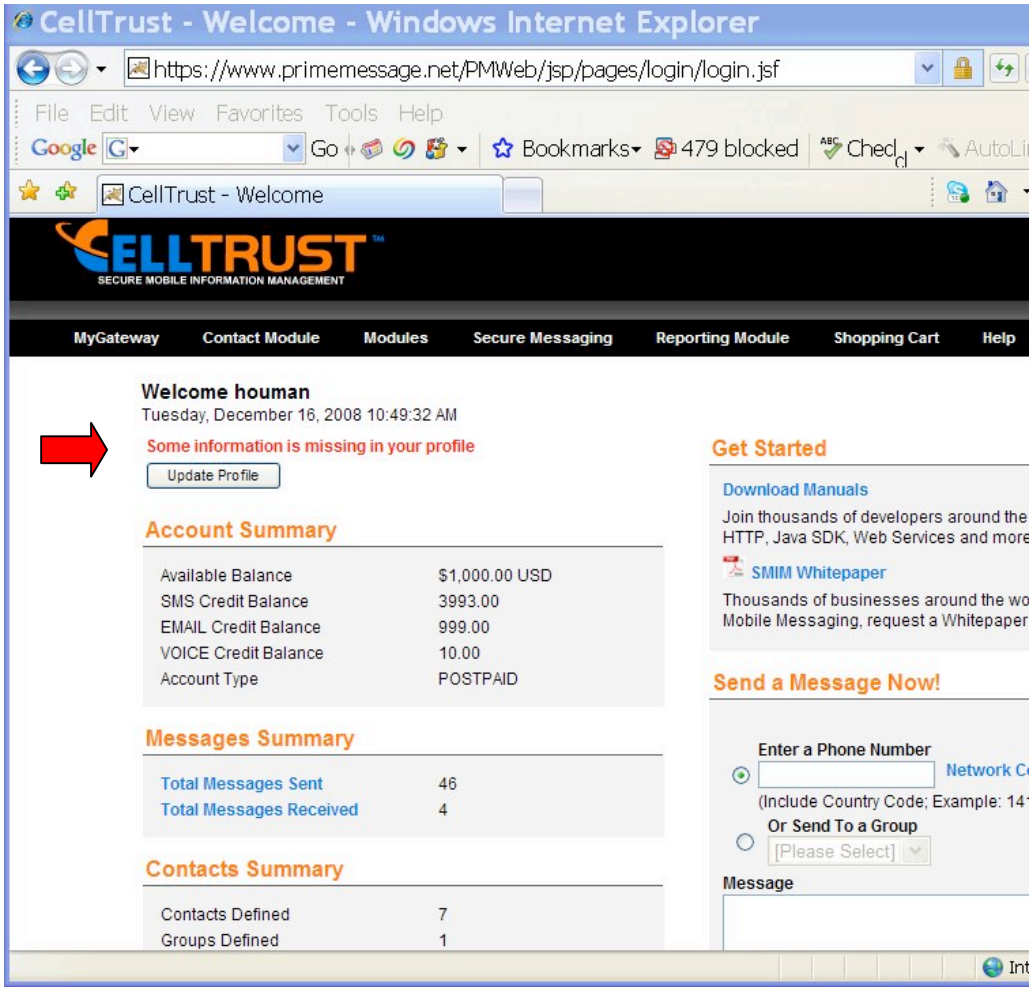
You can send an email to CellTrust Gateway and it will convert it to SMS and send it to one or more cell phones.

The simplest communication method with the CellTrust Gateway is to simply send a formatted email to gateway@celltrust.com. Based on the parameters specified in the email body, CellTrust Gateway will convert your email message to a SMS and send it to one or more recipients.

Prerequisites

1) The email must be initiated from the email address specified during registration. This to prevent unauthorized customers from sending messages without your permission.

2) Your profile has to be complete. CellTrust Gateway Web Portal will display an error message in red on the dashboard if your profile is incomplete (see picture below).



Notes

- The CellTrust Gateway will poll its mailbox within a predefined period of time that will average to 2-3 minutes. Thus, there could be a significant delay in message delivery as compared with the other CellTrust gateway APIs.
- Due to the insecure nature of email, customer authentication will be based on the email address used to send the SMTP request. This email address will be compared against the existing verified customer email address (NOTE: The comparison is case-sensitive) defined previously using the MyGateway application.
- Results of operation will be optionally delivered to the sender asynchronously through email

Send a Message

Purpose

This API can be used to send a text message, and/or email to one or more recipients or a group of recipients. The group must already exist in CellTrust Gateway before it can be used in this API.

Syntax

Fieldname1: <DATA>
Fieldname2: <DATA>

Description

This API can be used to convert properly formatted email to a text message, or email.

Mandatory Parameters

Destination

Comma separated list of phone numbers and/or email addresses to receive the message as SMS or Email (delivery type will be detected automatically), for example:

14805551212 for US and Canada

342475551212 for international

Nickname

The user name is used to authenticate your account. The same username as created during registration.

Message

Simple text message sent to recipients.

Optional Parameters

GroupName

Comma separated list of groups to receive the provided message (groups must be previously defined through CellTrust Gateway Web Portal).

DeliveryType

currently SMS or Email will be accepted (required only if "GroupName" parameter is used).

TemplateName

The name of the template to be used (template must be predefined using CellTrust Gateway Web Portal).

Subject

The subject of an email message. This only applies to emails and not SMS messages.

WAPPushURL

The URL for WAP Push* messages. If provided, message will be automatically sent as WAP Push* message, and the message text will appear as a label for "push" button on recipient handset.

NOTE: *Some carriers (such as Verizon Wireless in US and other Non-GSM carriers) do not allow sending WAP Push* messages. Please review the glossary at the end of this document for more details*

Other Parameters

Any other parameters will be considered as Template Tags with provided values, and will be passed "as is" for template processing.

Syntax Rules

- Email subject will be ignored and only email body will be parsed;
- Parameters can come in any order;
- Parameter names are not case-sensitive (both NICKNAME and nickName will work);
- Values for each parameter must be provided on single line (line length is unlimited);
- If the same parameter is provided twice, only one will be used and CellTrust SMTP Gateway does not guarantee which parameter it will be;
- Any extra lines which do not follow a predefined format and/or do not start with correct parameter names will be ignored.

Sample Code

Simple Message to three contacts

Nickname: USER
Destination: 16477654321, myemail@my.com,14161234567
Message: Hello, World!!!
Subject: test message

Template message sent to a group

GroupName: MyContacts,MusicFans
DeliveryType: Email
TemplateName: NewsLetter_5
MeetingPlace: 4950 Yonge St, 2200
MeetingDate: Nov 30, 2005 17:00

SMS message to a group and two additional numbers

Nickname: USER
GroupName: Friends
DeliveryType: SMS
Destination: 16477654321,12121234567
Message: Country Club tomorrow @ 8pm

WAP Push Message

Nickname: USER
GroupName: Ringtones
DeliveryType: SMS
WAPPushURL: http://myringtones.com
Message: Push Me

Glossary

Definition of terms used in Mobile Messaging.

Term	Definition
Gateway	One single point of integration providing access to carriers (nationally or globally) for two way communication.
Standard Rates	Standard Rate Messaging is the cost the end user incurs from their carrier for each SMS received or sent from their cell phone.
Opt-in, Opt-out	End user the opportunity to explicitly engage (opt-in) or disengage (opt-out) in a SMS program - or to have their cell phone number placed on an opt-out list.
Premium SMS	Charging the end user through the carriers billing system, the end user's cost is reflected on their monthly cell phone bill.
Double Opt-in	When an end user agrees to engage in a Premium SMS transaction, the content provider sends a confirmation requesting them to authorize the transaction before the content is delivered.
Short Code	When two persons send a text message to each other, they will see each other's number as the sender. Short Code also known as CSC or Common Short Code are 5-6 digit strings of numbers used to send SMS messages to cell phones from an application. The end user can also send a message to a short code to be delivered to an application. It is much easier to remember and identify a short code than a telephone number. A short code can also be what is called Vanity Short Code whereby it reflects the mobile program's brand or a concept on the alpha-numeric keypad (such as short code #466453 for GOOGLE).
Long Code	Long Codes are 10 digit strings of numbers previously used by carriers to send SMS messages in North America.

	International carriers often continue to use Long Code.
WAP	WAP, or Wireless Application Protocol, is the internet protocol used by mobile phones to create web pages and applications to be displayed on mobile phones. Wapsite A WAP Site is a website that can be displayed on a mobile device. It is widely used to send a link to the handset to download rich content (such as ringtones, wallpaper, video clips) or engage users in completing a form (lead generation) and browsing data.
Keyword	A keyword is the first word in an incoming message which defines a program or campaign on a short code. Using text processing techniques, mobile applications look at the first word and forward the message to the respective URL in a third party application, or save it under an account and execute a series of predefined actions (such sending an auto-response message, generating a coupon, etc.) As shared short codes are used by multiple customers, keywords are used to uniquely identify each customer's mobile initiative. Each customer selects one or more unique keywords during registration. For example, if you are using the shared short code 32075 and own keyword WKRP, you would instruct your customers to "Text WKRP to 32075 to register for our U2 ticket giveaway". The CellTrust Gateway will identify any incoming message that starts with WKRP and perform the required routing based on your configuration. With outgoing messages, the keyword you have specified is added to every message. This will help your customers identify the message is from you.
MO (Mobile originated)	A message that was originated on a mobile phone and was sent to CellTrust SMS Gateway via a carrier.
MT (Mobile Terminated)	A message that was sent through CellTrust SMS Gateway to a mobile phone via a carrier.
WAP Push*	A WAP Push is basically a specially encoded message which includes a link to a WAP address. WAP Push is specified on top of WDP; as such, it can be delivered over any WDP-supported bearer, such as GPRS or SMS.

	WAP push is not allowed by default in Canada based on Canadian wireless regulations, is subject to approval for every specific program and is also not supported by Non-GSM network types such as CDMA/TDMA/IDEN. Instead of using WAP push, you can put the URL in the body of message and send it as a regular text. The carriers gateway may also handle the messages in different ways, possibly causing latency with regards to delivery of message.
Originator	Originator of a message – may be the handset user, or an application.

Terms and Conditions

By using this manuals and services described in this manual you bound to these terms and conditions.

ATTENTION: PLEASE READ THESE TERMS CAREFULLY BEFORE USING THIS WEBSITE AND RELATED WEB PAGES, SOFTWARE, APPLICATIONS, AND OTHER SERVICES. USING THESE SERVICES INDICATES THAT YOU ACCEPT THESE TERMS OF USE. IF YOU DO NOT ACCEPT THESE TERMS OF USE, DO NOT USE THESE SERVICES.

Use of Site. CellTrust Corporation ("CELLTRUST") authorizes you to view and/or download the materials at this Website and other CELLTRUST Websites that are linked to this site or affiliated with this site (collectively, the "Site"), under the condition that all the information, programs, processes, methodologies, communications, software, scripting, photos, text, video, graphics, sounds, images and other materials and services found on the Site (collectively "Content") may not be copied or distributed, or republished, uploaded, posted, publicly displayed, performed, distributed or transmitted in any way, without the prior written consent of CELLTRUST EXCEPT: only for your personal, non-commercial use, including the evaluation of our software products, and provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials. For purposes of these Terms, any use of this Content on any other Web site or networked computer environment for any other purpose is prohibited.

Use of Software. If you download any applications, including software applications, or other software from this Site ("Software"), the Software, including all code, files, images, contained in or generated by the Software, and accompanying data, are deemed to be licensed to you by CELLTRUST for the purposes of evaluation and/or use as set forth in the Terms of Use and any other applicable agreements, for example, a Software Sales Agreement . Neither title nor intellectual property rights are transferred to you, but remain with CELLTRUST, who owns full and complete title. You may not resell, decompile, reverse engineer, disassemble, or otherwise convert the Software to a perceivable form. You may not download or install the Software until you have read and accepted these Terms.

Copyrights, Trademarks And Service Marks. Unless otherwise noted, all Content and other materials on the Site and in the Software (Site and Software sometimes collectively referred to as "Services") are protected as the copyrights, trade dress, trademarks and/or other intellectual properties owned by CELLTRUST or by other parties that have licensed their material to CELLTRUST. The Content of the Services is copyrighted and any unauthorized use of the Content of the Services may violate copyright, trademark, and other laws, in addition to being a material breach of the Terms of Use.

There are a number of proprietary logos, service marks, trademarks, slogans and product designations ("Marks") found on the Site and in the Software. By making these Marks available on the Site and in the Software, CELLTRUST is not granting you a license to use them in any fashion. Access to the Services does not confer upon you any license under any of CELLTRUST's or any third party's intellectual property rights. Use of CELLTRUST's proprietary logos, service marks, trademarks, slogans and product designations found on this Site and in the Software by users is restricted as set forth in the Terms of Use.

The following trademarks and other marks (not listed here) are used in connection with the Services, and are the proprietary trademarks of CELLTRUST:

CELLTRUST™ (U.S. Serial No. 78/940,065)

CELLTRUST & Design™

CELLTRUST WALLET™

CELLTRUST FLIGHTASSIST™

CELLTRUST MOBILE MESSAGING™

CELLTRUST MOBILE MESSAGING OEM™

CELLTRUST ENGAGE™

CELLTRUST ENGAGE OEM™

PRIMEMESSAGE™ (Canadian Application No. 1258479)

POWER OF MOBILITY! ™ (Canadian Application No. 1258480)

PRIME MESSAGE POWER OF MOBILITY! & Design ™ (Canadian Application No. 1258478)

SECURE MOBILE INFORMATION MANAGEMENT™

TXT AVENUE™ (Canadian Application No. 1258481)

TXT FLIGHT™ (Canadian Application No. 1306133)

CELLTRUST's trademarks may be used publicly only with prior written permission from CELLTRUST. Fair use of CELLTRUST's trademarks in advertising and promotion of CELLTRUST products requires proper acknowledgment. No CELLTRUST trademark or service mark may be used as a hyperlink without CELLTRUST's prior written permission. Other trademarks that may appear in connection with the Services may be owned by third-parties and used with the permission of the third-parties. You also agree not to use those trademarks without the permission of their respective owners.

The various marks used in connection with the Services represent some of the marks currently owned or controlled by CELLTRUST or under license to CELLTRUST. The display of these marks and of notices associated with these marks is not intended to be a comprehensive compilation of all CELLTRUST worldwide proprietary ownership rights, and CELLTRUST may own or control other proprietary rights in one or more countries outside of the United States.

User Submissions. CELLTRUST does not solicit, but may accept, any product or process ideas, innovations suggestions, improvements or other user submissions, with the understanding that all remarks, suggestions, ideas, innovations, graphics, materials, information, data, concepts, submissions or other communications you transmit or post to the Site and/or to CELLTRUST using the Services (together, the "Communications") are assigned to, and will forever be the property of, CELLTRUST without any further compensation or other benefit to the user submitting such Communications. Other than personally identifiable information, which is covered under the CELLTRUST Privacy Policy any Communications will be considered non-confidential and non-proprietary. CELLTRUST will not be liable for any ideas for its business (including without limitation, product, or advertising ideas) and will not incur any liability as a result of any similarities that may appear in future CELLTRUST operations. CELLTRUST will have exclusive ownership of all present and future existing rights to the Communications of every kind and nature everywhere.

For any Communications that cannot be legally assigned to CELLTRUST, you hereby grant CELLTRUST and its designees an unrestricted, perpetual, royalty-free, irrevocable license to use, reproduce, display, perform, modify, transmit and distribute the Communications for any and all commercial or non-commercial purposes, and agree that CELLTRUST is free to use any ideas, concepts, know-how or techniques that you send CELLTRUST for any purpose whatsoever without compensation to you or any other person sending the Communication. In addition, you warrant that all so-called "moral rights" with respect to the Communication have been waived.

You are prohibited from using the Services to post, transmit, communicate, and/or deliver, to CELLTRUST or other users, any unlawful, threatening, libelous, defamatory, obscene, pornographic, or any other material that would violate any law, or that could give rise to any civil or criminal liability under the law.

User Forums or other Communication Networks. CELLTRUST may, but is not obligated to, monitor or review any areas on the Site where users transmit or post Communications or communicate solely with each other, including but not limited to chat rooms, bulletin boards, communication networks, text messaging (SMS, MMS, and the like) networks, or other user forums, and the content of any such Communications. CELLTRUST also may, but is not obligated to, monitor communications between users who employ the Services to communicate with other users. CELLTRUST, is free in its sole discretion to remove, edit, delete or modify any Communications deemed undesirable without prior notice to the user submitting such Communications; however, CELLTRUST will have no liability related to the content of any such Communications, whether or not arising under the laws of copyright, libel, privacy, obscenity, or otherwise, and retains the right to remove messages that include any material deemed abusive, defamatory, obscene or otherwise unacceptable. CELLTRUST may deny access to the Services to users who repeatedly violate these Terms of Use.

Your use of the Services results in your sending of various types of information to CELLTRUST. CELLTRUST's handling of such information is governed by the CELLTRUST Privacy Policy to which you agree when you accept these Terms of Use. If you do not accept the Privacy Policy or these Terms of Use, do not use the Services. CELLTRUST reserves the right to disclose, read, access, and preserve any information received from you by your using the Services as CELLTRUST reasonably believes is necessary to (i) enforce these Terms of Use; (ii) prevent, deter, or defend against any type of fraud or misrepresentation; (iii) comply with applicable law; (iv) provide customer support; and (v) protect the property and other rights of CELLTRUST, its employees, and its consumers.

Links To Third Party Web Sites. The Services may provide links to other third party websites or resources. Such links to third party websites in the Services are provided solely as a convenience to you, and do not constitute or imply an endorsement, sponsorship or recommendation of, or affiliation with the third party or its products and services. CELLTRUST has not reviewed all of these third party sites and does not control and is not responsible for any of these sites or their content. Thus, CELLTRUST makes no representations whatsoever about any other website, which you may access through the Services, or any information, software or other products or materials, found there, or any results that may be obtained from using them. If you decide to access any of the third party sites linked to the Services, you acknowledge and agree that CELLTRUST is not responsible for the availability of such external sites or resources and is not responsible or liable for any content, advertising, products, services or other materials on or available from such sites or resources. It is up to you to take precautions to ensure that whatever you select for your use is free of such items as viruses, worms, Trojan horses and other items of a destructive nature, and you do so entirely at your own risk. Accordingly, you agree that CELLTRUST shall not be responsible or liable, directly or indirectly, for any damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such third-party materials, content, products or services available on or through any such site or resource.

Fees. Certain aspects of the Services may require or involve sales fees, transaction fees, transmission fees, and other types of fees ("Fees"). By using these aspects of the Services, you consent to the Fees involved and agree that you decided to use the Services with full knowledge of the Fees. A conspicuous notice will be posted near each location on the Site or in the Software where you may incur a Fee, or, at a minimum, such a notice will be posted in a separate agreement to which you agree when you sign up for an aspect of the Services that may cause you to incur a Fee.

By sending and receiving text messages, emails, and other communications using the Services, you agree that CELLTRUST is not responsible for the cost of making the communications as may be charged by your cellular telephone carrier, internet service provider, and the like. Any aspect of the Services subject to a Fee is announced prior to your usage of the Services, and by your use of the Services, you take sole responsibility for the Fees involved.

CELLTRUST adheres to the Mobile Marketing Association's Code of Conduct for Mobile Marketing http://mmaglobal.com/modules/content/index.php?id=5 and Consumer Best Practices Guildelines for Cross-Carrier Content Programs

http://www.mmaglobal.com/bestpractices.pdf Therefore, CELLTRUST will not subscribe you to any text messaging program without your choice to opt in to the program. CELLTRUST will also provide clear instructions on how to cancel a subscription to which you have opted in.

Disclaimer. THE MATERIALS, INFORMATION AND SERVICES PROVIDED IN THE SERVICES ARE PROVIDED "AS IS" WITHOUT ANY WARRANTIES OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF INTELLECTUAL PROPERTY.

CELLTRUST DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE AVAILABILITY, USE, TIMELINESS, SECURITY, VALIDITY, ACCURACY, OR RELIABILITY OF, OR THE RESULTS OF THE USE OF, OR OTHERWISE RESPECTING, THE SITE, THE SOFTWARE, THE CONTENT OF THE SITE OR THE SOFTWARE OR ANY OTHER WEBSITES LINKED TO OR FROM THE SITE.

CELLTRUST DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE MATERIALS, DATA OR INFORMATION IN THE SERVICES IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. WHILE CELLTRUST MAY MAKE REASONABLE EFFORTS TO PROVIDE ACCURATE AND TIMELY INFORMATION ABOUT CELLTRUST ON THE SITE, YOU SHOULD NOT ASSUME THAT THE INFORMATION PROVIDED IS ALWAYS UP TO DATE OR THAT THE SITE CONTAINS ALL THE RELEVANT INFORMATION AVAILABLE ABOUT CELLTRUST. CELLTRUST UNDERTAKES NO OBLIGATION TO VERIFY OR MAINTAIN THE CURRENCY OF SUCH INFORMATION.

ANY MATERIAL DOWNLOADED OR OTHERWISE OBTAINED THROUGH THE USE OF THE SERVICES IS DONE AT YOUR OWN DISCRETION AND RISK AND YOU ARE SOLELY RESPONSIBLE FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT RESULTS FROM THE DOWNLOAD OF ANY SUCH MATERIAL, OR USE OF THE SERVICES. INFORMATION PUBLISHED AT THE SITE MAY REFER TO PRODUCTS, PROGRAMS OR SERVICES THAT ARE NOT AVAILABLE IN YOUR GEOGRAPHIC LOCATION.

THE INFORMATION PRESENTED ON THE SITE OR BROADCAST FROM THE SITE WITH WEBSITE SOFTWARE BY CELLTRUST FROM CELLTRUST PUBLICATIONS, WRITINGS AND/OR THIRD PARTY

BOOKS OR WRITINGS IS FOR INFORMATION PURPOSES ONLY AND IS NOT MEANT TO SERVE AS A SUBSTITUTE FOR PROFESSIONAL LEGAL OR FINANCIAL ADVICE WHICH SHOULD BE OBTAINED THROUGH CONSULTATION WITH APPROPRIATE PROFESSIONALS IN YOUR STATE.

Limitation of Liability. IN NO EVENT WILL CELLTRUST, ITS SUPPLIERS, OR OTHER THIRD PARTIES MENTIONED IN THE SERVICES BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOST PROFITS, LOST DATA OR BUSINESS INTERRUPTION, OR CAUSED BY OR RELATED TO ERRORS, OMISSIONS, INTERRUPTIONS, DEFECTS, DELAY IN OPERATION OR TRANSMISSION, COMPUTER VIRUS, LINE FAILURE, AND ALL OTHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES) ARISING OUT OF THE USE, INABILITY TO USE, OR THE RESULTS OF USE OF THE SERVICES, ANY WEB SITES LINKED TO THE SITE, OR THE MATERIALS OR INFORMATION OR SERVICES CONTAINED AT ANY OR ALL SUCH SITES, WHETHER BASED ON WARRANTY, CONTRACT, TORT OR ANY OTHER LEGAL THEORY AND WHETHER OR NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IF YOUR USE OF THE MATERIALS, INFORMATION OR SERVICES FROM THE SITE OR THE SOFTWARE RESULTS IN THE NEED FOR SERVICING, REPAIR OR CORRECTION OF EQUIPMENT OR DATA, YOU ASSUME ALL COSTS THEREOF. APPLICABLE LAW MAY NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU. NOTWITHSTANDING THE FOREGOING, CELLTRUST'S TOTAL LIABILITY TO YOU FOR ALL LOSSES, DAMAGES, AND CAUSES OF ACTION, INCLUDING BUT NOT LIMITED TO THOSE BASED ON CONTRACT, TORT OR OTHERWISE, ARISING OUT OF YOUR USE OF THE SERVICES, ITS CONTENT OR LINKS, SHALL NOT EXCEED THE AMOUNT YOU PAID TO ACCESS THIS SITE.

BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Procedure for Making Claims of Copyright Infringement. CELLTRUST respects the intellectual property rights of others, and expects its users to do the same. CELLTRUST will, upon receiving proper notice, act to remove or disable access to any such material as set forth in the Digital Millennium Copyright Act (17 USC § 512) www.copyright.gov/legislation/dmca.pdf. Any

notifications of claimed copyright infringement must be sent to CELLTRUST at the following address: Legal Department, 20701 N. Scottsdale Road, Suite #107-451, Scottsdale, Arizona 85255 , USA. When notifying CELLTRUST of the alleged copyright infringement, please provide complete and sufficient information, including identification of the copyrighted work alleged to have been infringed, the alleged infringing material, the address and contact information for the owner of the alleged copyright material, and a statement that the information in the notification is accurate, and, under the penalty of perjury, that the complaining party is authorized to act on behalf of the owner of the alleged copyright.

GENERAL TERMS

Applicable Laws. CELLTRUST from its offices within the United States of America controls the Services. CELLTRUST makes no representation that the Content in the Services is appropriate or available for use in locations other than the United States, and access to them from territories where their content is illegal is prohibited. Those who choose to access the Services from other locations do so on their own initiative and are responsible for compliance with applicable local laws. You may not use or export the Content in violation of U.S. export laws and regulations.

These Terms of Use and any disputes arising under or related to these Terms of Use (whether for breach of contract, tortuous conduct or otherwise) and any claim relating to the Content shall be governed by the internal substantive laws of the State of Arizona. By using the Services, and agreeing to the Terms of Use, you agree that any dispute to enforce, defend or interpret any right or remedies under, or arising in connection with or relating to, these Terms of Use, shall be settled through binding arbitration under the AAA Rules of Arbitration, located at a mutually convenient forum in Phoenix, Arizona.

Termination. CELLTRUST, in its sole discretion, may terminate or restrict your use or access to this Site (or any part thereof) for any reason, including, without limitation, if CELLTRUST believes you have violated or acted inconsistently with the letter or spirit of these Terms of Use, or if you are in breach of the terms of the Terms of Use. Upon termination, you will immediately destroy any copies of Content of the Services, whether in printed or software format.

Notices. Notices to you may be made via either email or regular mail. CELLTRUST may also provide notices of changes to the Terms of Use or other matters by displaying notices or links to notices to you generally on the Site.

Privacy. CELLTRUST recognizes the need to protect the privacy of users of this Site, and to provide additional privacy protection to children. Children under the age of 13 may visit this Site only with parental permission. CELLTRUST does not require children under the age of 13 to disclose any personally identifiable information. Please see our **Privacy Policy** for more information.

Revisions. CELLTRUST may make changes to the materials and services at this Site, or to the products and prices described in them, at any time without notice. You should visit this page from time to time to review the then-current Terms of Use because they are binding on you. Certain provisions of these Terms of Use may be superseded by expressly designated legal notices or terms located on particular pages at this Site. The materials and services at the Site may be out of date, and CELLTRUST makes no commitment to update the materials and services at this Site.

Additional Terms. Certain items or programs offered by the Site, whether by CELLTRUST or its partners, and certain areas within this Site may be governed by additional terms of use and/or other agreements ("Additional Terms") presented in conjunction with those items or programs. You must agree to these Additional Terms before using those areas. The Additional Terms and this Terms of Use shall apply equally. In the event of an irreconcilable inconsistency between the Additional Terms and this Terms of Use, the Additional Terms shall control.

Waiver. CELLTRUST's failure to enforce any part of these Terms of Use shall not constitute a waiver of any of CELLTRUST's rights under these Terms of Use, whether for past or future actions on the part of any person. Neither the receipt of any funds by CELLTRUST nor the reliance of any person on CELLTRUST's actions shall be deemed to constitute a waiver of any part of these Terms of Use. A specific, written waiver signed by an authorized representative of CELLTRUST may only provide a legal waiver.

Severability. If any provision of these Terms of Use shall be found to be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from these terms and conditions and shall not affect the validity and enforceability of any remaining provisions.

Acceptance of Terms. You acknowledge you have read, and agree to be bound by these Terms and to comply with all applicable laws and regulations. You further agree to comply with all local laws, regulations and rules regarding online conduct and acceptable Content. You represent you have the legal authority to accept these Terms on behalf of yourself or any

party you represent. **IF YOU DO NOT AGREE TO THESE TERMS, PLEASE DO NOT USE THE SERVICES**.