



## 1. Content

---

1.	Content.....	2
2.	Change History.....	3
3.	Overview.....	3
4.	Introduction.....	3
5.	Getting started.....	4
5.1	Step 1 - register for an account.....	4
5.2	Step 2 - add an API sub-product (registered API connection).....	4
6.	Terminology.....	5
7.	Basic email structure.....	6
7.1	Authentication.....	6
7.2	Send a message.....	7
7.3	Text parameter versus the data parameter.....	8
8.	Message parameters.....	9
8.1	Table of parameters.....	9
8.2	Message parameters in detail.....	12
8.2.1	Destination address (to).....	12
8.2.2	Source address (from).....	12
8.2.3	Delivery acknowledgement (deliv_ack).....	13
8.2.4	Callback URL.....	13
8.2.5	Delivery time (deliv_time).....	14
8.2.6	Concatenation (concat).....	15
8.2.7	Maximum credits (max_credits).....	15
8.2.8	Required features (req_feat).....	15
8.2.9	Delivery queue (queue).....	16
8.2.10	Gateway escalation (escalate).....	17
8.2.11	Mobile originated (mo).....	17
8.2.12	Message types (msg_type).....	18
8.2.13	Validity period (val_period).....	19
8.2.14	Reply (reply).....	19
8.2.15	URL encoded text (urltext).....	19
9.	Batch messaging.....	21
9.1.1	CSV line parameter (csv).....	21
9.1.2	CSV template parameter (csvtemplate).....	22
10.	8-bit messaging.....	22
11.	Appendix A: Error codes.....	23
12.	Appendix B: Message Statuses.....	25
13.	Contact details.....	25

## 2. Change History

---

Approximately six (6) months of changes are reflected.

Version	Date	Section	Changes to Documentation
2.3	13/042007	All	Revised document branding

## 3. Overview

---

This document is intended for those users who wish to develop applications that make use of our SMS Gateway. It is recommended that you have an understanding of Profiles before reading this document. Information about Profiles is available from:

<http://support.clickatell.com/guides/wholesale/routing.php>.

There are a number of different ways of gaining access to the Gateway:

- SMTP - enabling a server or client generated e-mail to be delivered as an SMS.
- HTTP / HTTPS - submitting either a POST or GET to the API server.
- FTP – uploading a text file to our FTP Server.
- SMPP – Customers requiring a high throughput binary socket connection.

We will cover the SMTP method in this document. Additional documentation is available for the other methods.

## 4. Introduction

---

Quite often the only way of getting a server to send SMS messages via Clickatell Wholesale's gateway, is via a server-generated email. This API connection facilitates that operation. It supports most of the functionality of the HTTP API version 2.3.0 including 8-bit messaging and templating.

The recommended batch size is 10 000 messages with a maximum of 50 000 messages per email allowed.

**Note: It is important that the ENTIRE document is read before contacting support.**

## 5. Getting started

---

In order to use the Clickatell Wholesale gateway you need a Clickatell Wholesale account and at least one registered connection (API sub-product instance) between your application and our gateway. Each connection method (SMTP, HTTP, FTP, SMPP), is known as a sub-product (of our API product). Here's how:

### 5.1 Step 1 - register for an account

If you do not already have a Clickatell Wholesale account, you need to register for one as per below. Otherwise proceed to Step 2.

- Go to [http://www.clickatellwholesale.com/brochure/products/developer\\_solutions.php](http://www.clickatellwholesale.com/brochure/products/developer_solutions.php), and choose the appropriate API sub-product (connection method) you wish to use.
- Click on the registration hyperlink.
- Fill out the registration form.

After successfully submitting the form you will automatically be logged into your new account and taken to a page where you can add your chosen API sub-product.

### 5.2 Step 2 - add an API sub-product (registered API connection)

If you are not already logged into your account, then you must do so at <http://www.clickatell.com/login.php>

- Select "Manage my products" from the top menu.
- Select the API connection type you wish from the drop down menu in the "My Connections" page.
- Select "add sub-product" from the main page (it will default to this page if you have not yet added a sub-product).
- Complete the form.

After successfully submitting the form, your authentication details will be displayed, including the sub-product instant's own unique API ID (**api\_id**). These authentication details are required when connecting to the Clickatell Wholesale gateway to send a message.

Note: For more information on managing your API connections within your Clickatell Wholesale account see our API guide at [http://www.clickatellwholesale.com/brochure/api\\_guide.php](http://www.clickatellwholesale.com/brochure/api_guide.php).

## 6. Terminology

---

- **Mobile originated (MO):** A message sent (originating) from a mobile handset to an application via Clickatell.
- **Mobile terminated (MT):** A message sent from an application to (terminating on) a mobile handset via Clickatell.
- **Premium rated message (MO):** A mobile user is charged a premium for the message that they send to a particular number.
- **Premium rated message (MT):** A mobile user is charged a premium for a message that they receive from a particular number.
- **Revenue share:** This refers to the portion of the increased cost associated with a premium rated message, which is passed on to the content provider.
- **Content provider:** This is the Clickatell client offering service(s) based on a premium rated SMS system.
- **Customer:** A registered Clickatell customer utilising the Clickatell API for message delivery and receipt.
- **Sender ID:** The “from” address that appears on the user’s handset. Also known as the originating number, MSISDN or originator ID.
- **Destination address:** This is the number to which the mobile user sent the message.
- **Source address:** The number of the handset to which the message must be delivered.
- **Shortcode:** A short number which is common across all the operators for a specific region.
- **Subscriber:** The handset that will be paying for the premium rate MT service.
- **Upstream gateway:** A network operator, third party or our own short message service centre (SMSC).

## 7. Basic email structure

---

To send a message with the Clickatell Wholesale SMTP API, the system interprets variables that you pass in the body of the email. Each line represents a variable in the form of:

```
variablename:value  
variablename:value  
variablename:value
```

We endeavour to strip any superfluous white space that may occur either side of the colon. However, to avoid any erroneous characters from creeping into the email by mail clients, we suggest that you keep to the format shown above.

Note: We only accept plain-text emails. We do not process HTML-formatted emails. The “Content-Transfer-Encoding” header may only be set to 7-bit which is the default.

The emails themselves should be sent to:

**sms@messaging.clickatellwholesale.com**

The subject field of the message is ignored. All the following examples are of the body only, unless otherwise indicated. Please check to determine after how many characters your email client forces word-wrapping (inserts a line-break). Make sure the length of each line is less than this value. We read a maximum of 8192 bytes from each line, so avoid having too many comma delimited destination addresses. If uncertain, please test before sending large volumes.

### 7.1 Authentication

In order to deliver a message, the system needs to authenticate the request as coming from a valid source. We use a number of parameters to achieve this:

- **api\_id**: This is issued to you when you register for the API product.
- **user**: This is the username of your account.
- **password**: The current password you have set on your account.

You can send multiple emails on the same API account simultaneously.

#### **Example**

```
api_id:1234  
user:xxxxxxxx  
password:xxxxxxxxxx
```

**Note:** If you use the ‘reply:your email address’ parameter, you will receive feedback on your email. Use this parameter while you are testing. See the **reply** parameter for more information.

## 7.2 Send a message

Once you have set up the three authentication variables listed above, there are only another two parameters that are required before you can send an SMS via SMTP.

`to:xxxxxxxxxxxxxxxx`

This parameter sets the destination address. You may send the same SMS to one or more recipients in a single email. There are two ways to achieve this, either using comma-delimited destination addresses (ensure that there are no spaces before or after the comma), for example:

`to:xxxxxxxxxxxxxxxx,xxxxxxxxxxxxxxxx,xxxxxxxxxxxxxxxx,xxxxxxxxxxxxxxxx`

or multiple lines, e.g:

`to:xxxxxxxxxxxxxxxx`  
`to:xxxxxxxxxxxxxxxx`  
`to:xxxxxxxxxxxxxxxx`  
`to:xxxxxxxxxxxxxxxx`

The last variable is the text parameter. The text parameter can run over multiple lines. This enables you to insert new lines in your SMS. It also ensures that when your mail client is line wrapping, that the text delivered to the handset is not cut at the line wrap of your email, for example:

Email:	Resultant SMS:
<code>text:Line 1</code>	<code>Line 1</code>
<code>text:Line 2</code>	<code>Line 2</code>
<code>text:Line 3</code>	<code>Line 3</code>

### **Simple Example**

`api_id:1234`  
`user:xxxxxxxx`  
`password:xxxxxxxx`  
`to:1234567890,1234567890`  
`text:This is my first test SMS`

**Note:** It is suggested that the maximum length of each text line be less than 40 characters. However the maximum length is determined by the email client that you use. As such this could be more or less than 40 characters.

### 7.3 Text parameter versus the data parameter

An alternative to the **text** parameter is the **data** parameter.

This parameter is used to avoid mal-formatting of an SMS, such as when a mail client inserts an unwanted line break in a text string, due to automatic line wrapping. If this parameter is used, any line break automatically inserted by the mail client will be removed when delivering the SMS.

We will also strip out any white space before and after the message text lines, resulting in an unbroken text line in the resultant SMS. Therefore use the **data** parameter when sending binary or Unicode messages.

#### Example 1

Email sent	SMS received
api_id:1234 user:xxxxxxxxx password:xxxxxxxxxxx to:1234567890,1234567890 text:Test SMS with the email client inserting a line break here in this sentence text: . This is a new sentence	Test SMS with the email client inserting a line break . This is a new sentence.

In the above example the email client automatically inserts a line break before the word “here”. When the SMS is received, all text after this line break (for the first instance of the text parameter) is lost. A line break is then inserted at the end and the contents of the second text parameter added.

#### Example 2

Email sent	SMS received
api_id:1234 user:xxxxxxxxx password:xxxxxxxxxxx to:1234567890,1234567890 data:Test SMS with the Email client inserting a line break here data: . This is a new sentence.	Test SMS with the Email client inserting a line break. This is a new sentence.

In the above example the email client automatically inserts a line break before the word “here”. When the SMS, is received all text after the line break for the first data line is lost. The second data line is then appended to the remaining portion of the first data line.

**Note:** Another alternative to the **text** parameter is the **urldata** parameter. This allows you to use a URL encoded string which is useful when sending special characters such as “new line” or Greek text.

## 8. Message parameters

### 8.1 Table of parameters

There are a variety of messaging and SMS features supported by the gateway, which can be activated by including a number of additional parameters. These parameters include those in the table below. Parameters are case-sensitive.

Name	Parameter name	Short description	Default value	Restricted values
API product ID	api_id	The value for this <b>mandatory</b> parameter can be found or created by logging in online and going to Product Control.		
Username	user	The username you specified.		
Password	password	Your account password.		
Session ID	session_id	The session ID from the <i>auth</i> command. Not applicable to the FTP or SMTP APIs.		
Destination address	to	The number of the handset to which the message must be delivered. The number should be in international number format.		No '00' prefix or leading "+" symbol should be used.
Text	text	The text content of the message.		
Source address	from	The source/sender address that the message will appear to come from.	gateway assigned number	A valid international format number between 1 and 16 characters long, or an 11 character alphanumeric string.
Delivery acknowledgement	deliv_ack	Where possible, this will return a delivery acknowledgement for any message, upon delivery of the message to the mobile handset or upstream gateway. Also see <b>required features</b> parameter.	0	0 – Upstream gateway 1 - Handset acknowledgement.
Enable callback	callback	Enables you to receive message delivery statuses via an HTTP callback which is posted to a URL of yours using the GET method. This is done every time a message status is updated.	0	0,1,2,3 Read detailed description of parameter.
Delivery time	deliv_time	Delays delivery of SMS to mobile device in minutes relative to the time at which the SMS was received by our gateway. This		The upper limit is 7 days, or 10080 minutes.

Name	Parameter name	Short description	Default value	Restricted values
		should be greater than 10 minutes for best effect. Smaller time frames may be delivered too soon.		
Concatenation	concat	Enables you to send messages longer than a standard message.	1	1,2,3
Maximum credits	max_credits	Overrides the maximum charge specified online in “profiles”. It works within the bounds of the profiles. In other words a profile must exist for the maximum credit that you set.	As per profiles	0,8,1,1.5,2,2.5,3
Required features	req_feat	Some parameters and features are not set as “required” by default, and may be dropped if the least-cost route does not support them. This parameter allows you to ensure that the features set when an SMS is sent are supported by the gateway used. This could increase the cost per message if a more expensive gateway is used.		Read detailed description of parameter.
Delivery queue	queue	Delivers the message through one of three queues assigned to each client account. Messages in the highest priority queue will be delivered first.	3	1, 2,3 1 is highest priority.
Gateway escalation	escalate	Prompts an escalation to an alternative delivery gateway, should the message be delayed for a set length of time.	0	0 - off 1 - Escalate immediately to an alternative route if messages are queued on the least-cost route.
Mobile originated	mo	We route via a pre-defined carrier to enable the ability for a reply to be received back. This is only applicable to clients that have subscribed to a two-way messaging service.	0	0 – Off. We use our normal routing rules. 1 – Enable Reply.
Client message ID	cliMsgId	Client message ID defined by user for message tracking.		Up to 32 alphanumeric characters. No spaces.
Unicode message	Unicode	Two digit language code. Convert your text to Unicode [UCS-2 encoding]. See <a href="http://www.Unicode.org/">http://www.Unicode.org/</a> .	0	0 – No Unicode 1 – Send as Unicode.

Name	Parameter name	Short description	Default value	Restricted values
Message type	msg_type	Optional parameter which must be set to send specially formatted messages; e.g. logos and ringtones.	SMS_TEXT	
User data header	udh	Allows you to set your own message types. Do not use if you set the <b>message type</b> parameter.		Set UDH data manually.
Data	data	The data content of a message, if the UDH component is set manually.		
Validity period	<i>See detailed information on message parameter</i>	The validity period in minutes relative to the time at which the SMS was received by our gateway. The message will not be delivered if it is still queued on our gateway after this time period.	1440 minutes (24 hours)	Set value in X minutes from 1 – 1440 minutes.

The following additional parameters are also available to the SMTP API

Name	Parameter Name	Short Description	Default Value	Restricted Values
Reply	reply	One or more email addresses which will be sent either a unique identifier for each message or an error code		
CSV Line	csv	Used in conjunction with text parameter to handle personalised batch messaging.		The mobile number must be the first value in the CSV field.
Delimiter	delimiter	Specify a specific character to be used to delimit values in the CSV field. Useful if certain values contain a comma. If not set, defaults to a comma.	,	
CSV Template	csvtemplate	Used in conjunction with text parameter to handle personalised batch messaging where additional message parameters need to be set on a message by message basis		
URL encoded text	urltext	Useful for special Greek characters, etc.		

## 8.2 Message parameters in detail

### 8.2.1 Destination address (to)

SMS messages need to be sent in the standard international format, with country code followed by number. No leading zero to the number and no special characters such as "+" or spaces must be used. For example, a number in the UK being 07901231234 should be changed to 447901231234.

If have you set the preferred dial prefix preference within your client account after logging in on-line, any mobile numbers starting with zero will have the zero stripped and replaced with your default prefix. If the mobile number does not have a zero, the default prefix will not be changed.

#### **Parameter:**

[to:xxxxxxxxxx](#)

### 8.2.2 Source address (from)

The source address (**from**), also known as the sender ID, can be either a valid international format number between 1 and 16 characters long, or an 11 character alphanumeric string. Note that characters such as spaces, punctuation, Unicode and other special characters may not always be supported to all destinations and could interfere with your delivery.

We suggest that you refrain from using such characters on the source address. If this is set, then delivery acknowledgements may be unavailable. The use of an alphanumeric source address with 8-bit

messaging may cause message failure. This service is not guaranteed across all mobile networks and may interfere with delivery to certain handsets.

**Note:** To ensure that this feature is supported when delivering your message, the required features (**req\_feat**) parameter for this feature must be set.

**Parameter:**

from:xxxxxxxxxxx

**Note:** To ensure that this feature is supported when delivering your message, the required features (**req\_feat**) parameter for this feature must be set.

### 8.2.3 Delivery acknowledgement (deliv\_ack)

In order to determine whether an SMS has been received by a handset or not, we offer delivery acknowledgement. The ability to receive reliable delivery acknowledgements varies between mobile networks.

Values to set are:

Value	Status
0	Off - default status, no delivery to handset reported. However delivery to the upstream gateway will be reported.
1	Handset acknowledgment - handset has received SMS (delivery acknowledgment).

To ensure that this feature is supported when delivering your message, the required features (**req\_feat**) parameter for this feature must be set. This ensures that an upstream gateway that supports this feature is used to deliver the message. This gateway may be more expensive.

Handset acknowledgements are not guaranteed by all of our upstream gateways. Please test to a specific mobile network first, before assuming that you will receive handset acknowledgments for messages that are delivered.

GSM handsets that are off will only be reported once a user has switched their phone back on. If the validity period is exceeded, the Clickatell Wholesale gateway will return a message status of 7, "Error with message".

A delivery acknowledgment can be monitored via the callback URL or online reports.

**Parameter:**

deliv\_ack:1

### 8.2.4 Callback URL

Final or intermediary statuses are passed back by the API depending on the **callback** value set in the original post. This is done by means of an HTTP GET. The callback URL is set in the preferences section of the particular API product within your client account, after logging in online. The URL must begin with either *http://* (un-secure) or *https://* (secure).

The variables returned are **apiMsgId**, **cliMsgId**, **api\_id**, **to**, **timestamp**, **from**, **status** and **charge**.

Values to set with delivery acknowledgment (**deliv\_ack**) set to 0 (no delivery to phone reported) are:

Callback value	Message status types returned	Message status code returned
0	No message status returned.	
1	Returns only intermediate statuses.	002, 011
2	Returns only final statuses of a message.	003, 005, 006, 007, 008, 009, 010, 012
3	Returns both intermediate and final statuses of a message.	All except 001

Values to set with delivery acknowledgment (**deliv\_ack**) set to 1 (handset acknowledgment) are:

Callback value	Message status types returned	Message status code returned
0	No message status returned.	
1	Returns only intermediate statuses.	002, 003, 011
2	Returns only final statuses of a message.	004, 005, 006, 007, 008, 009, 010, 012
3	Returns both intermediate and final statuses of a message.	All except 001

Sample callback to your callback URL:

[http://www.yoururl.com/script.asp?api\\_id=12345&apiMsgId=96905854f5045354a1b36134acf81&cliMsgId=123456&status=003&timestamp=1055155528&to=2782123456&from=Sender&charge=2.5](http://www.yoururl.com/script.asp?api_id=12345&apiMsgId=96905854f5045354a1b36134acf81&cliMsgId=123456&status=003&timestamp=1055155528&to=2782123456&from=Sender&charge=2.5)

Delivery acknowledgment (**deliv\_ack**) and **callback** explained:

**Example 1:**

With **deliv\_ack=1** (acknowledgment of delivery to handset), you will receive a callback of messages reaching the SMSC (003), if you requested intermediate statuses, and a callback of messages reaching handsets (004), if you requested final statuses.

**Example 2:**

With **deliv\_ack = 0** (default), you will only receive a callback of messages reaching the SMSC (003), if you have set **callback = 2**.

**8.2.5 Delivery time (deliv\_time)**

The delivery of a SMS message may be delayed by setting an amount of time in **minutes** relative to the time at which it was received by our gateway. We will store the message until the required time frame has elapsed. The maximum delay time is 10080 minutes or 7 days.

**Parameter:**

**deliv\_time:120**

### 8.2.6 Concatenation (concat)

If this value is set to 1, 2 or 3 the message will span across 1, 2 or 3 SMS messages where applicable. One text SMS will be sent for every 160 characters or 140 bytes. If a message is concatenated, it reduces the number of characters contained in each message by 7. With 8-bit concatenated messages, each SMS can support up to 160 bytes including the UDH headers.

Please be aware that a single Unicode SMS can only contain a maximum of 70 characters. 8-Bit messages will be split over multiple messages, where necessary, irrespective of whether the flag for concatenated messages has been set. If a Unicode message is concatenated, it reduces the number of characters contained in each message part by 3.

Values set are:

Value	Status
1	Default - No concatenation: only 1 message.
2	Concatenate a maximum of 2 messages.
3	Concatenate a maximum of 3 messages.
N	Concatenate a maximum of N messages. (Delivery is dependant on mobile and gateway. A maximum of 3 is recommended).

**Parameter:**  
 concat:x

### 8.2.7 Maximum credits (max\_credits)

This parameter overrides the maximum charge associated with message delivery, as set by the profiles selected within your client account after logging in online. This parameter can be used to limit the cost of a message to a particular value and is bound by the maximum credit value specified in your profiles.

A valid API message ID can still be returned for messages that are not delivered as a result of the maximum credits value set. These messages will have a status of routing error (009).

The credit value in this parameter can be set to any amount of credits. Please see [http://www.clickatell.com/pricing/basic\\_coverage.php](http://www.clickatell.com/pricing/basic_coverage.php) to view relative destination costs.

To set your delivery profile, go to “Central>>My Settings>>Routing Profiles”

**Parameter:**  
 max\_credits:3

### 8.2.8 Required features (req\_feat)

This parameter specifies the features that must be present in order for message delivery to occur. If all features are not present, the message will not be delivered. This prevents SMS messages arriving at a destination via the least-cost gateway, without certain features. This would, for instance, prevent the dropping of an alphanumeric sender ID.

This means is that we will not route messages through a gateway that cannot support the required features you have set. For certain message types, we always set the required feature bitmask where relevant. These are FEAT\_8BIT, FEAT\_UDH, FEAT\_UCS2 and FEAT\_CONCAT.

This parameter is set using a combined decimal number to refer to the additional required features.

E.g.:  $32 + 512 = 544$  – Numeric sender ID and Flash SMS both required.

The value you would set to ensure that Flash and numeric sender ID are both supported, would therefore be **544**.

To ensure that delivery acknowledgment and alphanumeric IDs are supported you would use the value **8240** ( $16 + 32 + 8192$ ).

Hex value	Decimal	Feature	Description
0x0001	1	FEAT_TEXT	Text – set by default.
0x0002	2	FEAT_8BIT	8-bit messaging – set by default.
0x0004	4	FEAT_UDH	UDH (Binary) - set by default.
0x0008	8	FEAT_UCS2	UCS2 / Unicode – set by default.
0x0010	16	FEAT_ALPHA	Alpha source address (from parameter).
0x0020	32	FEAT_NUMER	Numeric source address (from parameter).
0x0200	512	FEAT_FLASH	Flash messaging.
0x2000	8192	FEAT_DELIVACK	Delivery acknowledgments.
0x4000	16384	FEAT_CONCAT	Concatenation – set by default.

**Parameter:**

req\_feat:####

### 8.2.9 Delivery queue (queue)

Setting this parameter will assign the message to one of three queues assigned to each user account. This sets the priority of a message sent to us, relative to other messages sent from the same user account. Messages in queue number 1, will always be delivered before messages in queue number 2 and 3, while messages in the 3rd queue, will have the lowest priority (relative to queues 1 and 2).

This is useful when delivering, for example, a single high priority message while you have a large batch going through that same account. The large batch will be queued through queue number 3 (default), and urgent alerts (sent through queue 1), will be delivered ahead of those messages in the batch (queue 3), regardless of when they are actually sent to us.

Values set are:

Value	Status
1	Use first / primary user queue (highest priority).
2	Use second user queue.
3	Use third user queue (lowest priority) - Default status.

**Parameter:**  
 queue:2

### 8.2.10 Gateway escalation (escalate)

By default, the message router will select the lowest cost route (matching features and reliability) that is available for a given destination.

This parameter ensures that, should a message be delayed due to gateway congestion or some other reason on the initial gateway selected by our router, then alternative routes that match the required features will be sought. This is done by moving through the available gateways in order of increasing cost, up to the maximum charge set by the user either using the parameter that defines the maximum credits or based on the profiles selected.

When urgent and high priority messages are sent, they should be posted with escalate set to 1 (on), combined with a high maximum credit value to ensure that the greatest number of gateways are available.

Values set are:

Value	Status
0	Off – Default value.
1	On - If the least-cost route has messages queued then escalate immediately to an alternative route.

**Parameter:**  
 escalate:1

### 8.2.11 Mobile originated (mo)

This parameter is only used when a message is sent to a handset and a reply is expected.

**PLEASE NOTE: This parameter is only valid for clients that have signed up and paid for our two-way messaging service. An alternative to our least-cost gateway may be used, which could result in a higher cost per message. Please email Clickatell Wholesale support for pricing or view online.**

When sending a normal MT message to a handset and you expect a reply to your registered MO number, please set the **mo** parameter to “1”.

Values to set are:

Value	Status
0	Off - Default status. Clickatell Wholesale uses the normal routing feature.
1	Enables reply ability. Clickatell Wholesale routes via a pre-defined carrier to enable the ability to reply.

It is important that the user specifies the correct **from** parameter together with this parameter. If no **from** parameter is specified, we will use a default originator number as set by Clickatell Wholesale. You will NOT receive these replies.

If you specify the originator (the purchased mo number), then we will route the message such that it can be replied to by the recipient. This reply will be sent to you.

**Parameter:**

mo:1

If this value is set to 1, the text field must contain two-byte UTF-16 Unicode. Each SMS can handle a maximum of 70 characters. Each Unicode character must be hex-encoded. More information is available at <http://www.Unicode.org/>.

**Note:** When using the batch send facility for delivering Unicode messages, it is not possible to substitute variables into the message content. This is only possible with Germanic characters.

Values set are:

Value	Status
0	Off - default status.
1	On - delivers the text as two-byte Unicode.

### 8.2.12 Message types (msg\_type)

A wide variety of messages can be sent through our gateway. We have pre-defined a number of SMS message-types in the API, so that you do not have to set the UDH (user data header) manually. You may optionally set the UDH rather than using one of the message types set below. Message types are case sensitive.

For non-Nokia message types (EMS, etc.), please generate your own UDH and data according to the manufacturers specifications of the message type you wish to send.

**This parameter need not be included if the SMS is a standard text message.**

Values set are:

Value	Description
SMS_TEXT	This is the default message type. It is optional to specify this parameter.
SMS_FLASH	To send an SMS that displays immediately upon arrival at the phone.
SMS_NOKIA_OLOGO	Send an operator logo to a Nokia handset.
SMS_NOKIA_GLOGO	Send a group logo to a Nokia handset.
SMS_NOKIA_PICTURE	Send a picture message to certain Nokia handsets.
SMS_NOKIA_RINGTONE	Send a ringtone to a Nokia handset.
SMS_NOKIA_RTTL	Send an RTTTL format ringtone to Nokia handsets.
SMS_NOKIA_CLEAN	Remove operator logo from a Nokia handset.

Value	Description
SMS_NOKIA_VCARD	Send a business card to a Nokia handset.
SMS_NOKIA_VCAL	Send an event calendar to a Nokia handset.

**Parameter:**

Please see the messaging examples at the end of this document.

### 8.2.13 Validity period (val\_period)

A message may be given a time frame for which it is valid. After this period the message will expire. This parameter takes an amount of time in **minutes** relative to the time at which it was received by our gateway. If the message is queued on our gateway for a period exceeding the validity period set then a routing error of 115 will be returned. The default validity period is 1440 minutes (24 hours).

**Note:** The validity period is not passed on to the upstream gateway.

**Parameter:**

[val\\_period:120](#)

### 8.2.14 Reply (reply)

Unique message identifiers (API message IDs) are issued for each message submitted to Clickatell Wholesale. In order to have API message IDs for each message returned to an email address, set this email address using the reply parameter. Multiple addresses may be comma delimited.

Each message returns a unique identifier in the form of an API Message ID. This can be used to track and monitor any given message. The **apimsgid** is returned after each post. Alternatively an error code will be returned if the message is not accepted for delivery.

Note: A message status is not returned but may be viewed using the online reports or via the HTTP callback.

**Parameter:**

[reply:email@yourdomain.com](#)

**Response:**

ID: xxxx To: xxxx

or

ERR: Error number

### 8.2.15 URL encoded text (urltext)

Used instead of the **data** or **text** parameter, the **urltext** parameter can be used to send URL encoded text, which will be decoded back to normal text before the message is delivered to the phone.

It is possible that text could consist of characters that will confuse email-clients. To prevent possible errors, the URL encoding scheme translates all "special" characters to their corresponding hexadecimal codes. These special characters include control characters (carriage returns, line feeds, etc.), certain

alphanumeric symbols (% , ' , " , # , & , ? , = , / , :), and other characters (Greek Bulgarian and Cyrillic characters, etc).

For example, the string "Your URL encoded text!" could be sent as below.

***Parameter:***

`urldata:Your%20URL%20encoded%20text%21`

## 9. Batch messaging

This facility enables one to do high volume delivery and server-side message merging. It offers the end-user the ability to define all elements common to a batch, and then send only the parameters that change on a message by message basis. The following parameters are used for batch messaging.

Name	Parameter name	Short description	Default value	Restricted values
CSV Line	csv	Each csv line has a mobile number followed by a list of field values that are used to customise a text message. The text parameter value acts as a template with placeholders inserted into the text which are replaced by these field values when the message is sent.		The mobile number must be the first value in the CSV field.
CSV template	csvtemplate	Used in conjunction with the <b>text</b> and <b>csv</b> parameters to handle personalised batch messaging where additional message parameters (such as delivery time) need to be set on a message by message basis. These parameter values are added after the mobile number in the CSV Line.		
Delimiter	delimiter	Specifies a specific character to be used to delimit values in the CSV parameter field values. Useful if certain values contain a comma.	, (comma)	If you use the default delimiter you will not be able to use a comma in your field values. If you are using a TAB character, there must be no white space either before or after the TAB.

### 9.1.1 CSV line parameter (csv)

`csv:mobile_number[delimiter]field-value[delimiter]field-value[delimiter]field-value` etc.

Where `field-value` is the data to be inserted into the template, and `[delimiter]` is the value as determined by the delimiter parameter.

Placeholders may be inserted within the message body itself. These will take the form of `#field1#` through to `#fieldn#`. See example below. If you wish to customise the whole message on each `csv` line, you would then use:

```
text:#field1#
```

**Example:**

```
api_id:1234
user:xxxxxxxxx
password:xxxxxxxxx
text:Hi #field1#, your balance is #field2#, please come to the office on #field3#
delimiter:|
csv:1234567890|Fred|€15.50|Mondays,Wednesdays,Fridays
csv:1234567890|Jane|€299.45|Tuesdays,Thursdays,Saturdays
```

**Note:** If a ',' (comma) was used as the default delimiter, then it would only have passed through 'Mondays' as the value of field3 for Fred and 'Tuesdays' for Jane.

### 9.1.2 CSV template parameter (csvtemplate)

```
csvtemplate:[parameter-1],[ parameter-2],[parameter-n]
```

**Example:**

```
csvtemplate:deliv_time,cliMsgId,val_period etc.
```

These values would then appear on the **csv** line immediately after the mobile number. The **field-values** for each of the placeholders that you have inserted in your text (text parameter) are then appended to the csv line.

Format: csv:mobile\_number[optional csvtemplate parameters] template\_placeholder\_parameters

**Example:**

```
api_id:1234
user:xxxxxxxxx
password:xxxxxxxxx
text:Hi #field1#, your balance is #field2#, please come to the office on #field3#
delimiter:|
csvtemplate:deliv_time|cliMsgId|val_period
csv:1234567890|10|id_1|60|Fred|€15.50|Mondays,Wednesdays,Fridays
csv:1234567890|40|id_2|60|Jane|€299.45|Tuesdays,Thursdays,Saturdays
```

## 10. 8-bit messaging

---

Through the SMTP interface, one is also able to send 8-bit messages. These are most often used for ringtones and logos, but one can also send vCards, vCalendar appointments and EMS messages. When sending 8-bit messages, you need to set the user data header (UDH) of the SMS as well as sending the data. To simplify the process, we have provided a number of pre-defined message types.

With the standard **text** parameter, line breaks are automatically inserted. The parameter **data**, is thus used for 8-bit messaging.

**Example**

```
api_id:1234
user:xxxxxxxxx
password:xxxxxxxxx
to:xxxxxxxxxxxxxxxxx
```

[msg\\_type:SMS\\_NOKIA\\_RINGTONE](#)  
[data:024A3A5585E195B198040042D9049741A69761781B6176156174288](#)  
[data:B525D85E0A26C24C49A617628930BB125E055856049865885D200](#)

## 11. Appendix A: Error codes

The following list of error messages are generated by our gateway. There will be no message charge if these errors are generated when sending a message.

Number	Description	Detail
001	Authentication failed	
002	Unknown username or password	
003	Session ID expired	
004	Account frozen	
005	Missing session ID	
007	IP Lockdown violation	You have locked down the API instance to a specific IP address and then sent from an IP address different to the one you set.
101	Invalid or missing parameters	
102	Invalid user data header	
103	Unknown API message ID	
104	Unknown client message ID	
105	Invalid destination address	
106	Invalid source address	
107	Empty message	
108	Invalid or missing API ID	
109	Missing message ID	This can be either a client message ID or API message ID. For example when using the <i>del_msg</i> command.
110	Error with email message	
111	Invalid protocol	
112	Invalid message type	
113	Maximum message parts exceeded	The text message component of the message is greater than the permitted 160 characters (70 Unicode characters). Select <b>concat</b> equal to 1,2,3-N to overcome this by splitting the message across multiple messages.

Number	Description	Detail
114	Cannot route message	This implies that the gateway is not currently routing messages to this network prefix. Please email <a href="mailto:support@clickatell.com">support@clickatell.com</a> with the mobile number in question.
115	Message expired	
116	Invalid Unicode data	
120	Invalid delivery time	
201	Invalid batch ID	
202	No batch template	
301	No credit left	
302	Max allowed credit	

## 12. Appendix B: Message Statuses

---

Number	Hex	Description	Detail
001	0x001	Message unknown	The message ID is incorrect or reporting is delayed.
002	0x002	Message queued	The message could not be delivered and has been queued for attempted redelivery.
003	0x003	Delivered to gateway	Delivered to the upstream gateway or network (delivered to the recipient).
004	0x004	Received by recipient	Confirmation of receipt on the handset of the recipient.
005	0x005	Error with message	There was an error with the message, probably caused by the content of the message itself.
006	0x006	User cancelled message delivery	The message was terminated by an internal mechanism.
007	0x007	Error delivering message	An error occurred delivering the message to the handset.
008	0x008	OK	Message received by gateway.
009	0x009	Routing error	The routing gateway or network has had an error routing the message.
010	0x00A	Message expired	Message has expired before we were able to deliver it to the upstream gateway. No charge applies.
011	0x00B	Message queued for later delivery	Message has been queued at the gateway for delivery at a later time (delayed delivery).
012	0x00C	Out of credit	The message cannot be delivered due to a lack of funds in your account. Please re-purchase credits.

## 13. Contact details

---

Phone: +27 21 910 7700  
 Fax: +27 21 910 7701  
 Website: [www.clickatell.com](http://www.clickatell.com)  
 Help URL: <http://www.clickatellwholesale.com/brochure/help.php>  
 Support: [support@clickatell.com](mailto:support@clickatell.com)  
 Sales: [sales@clickatell.com](mailto:sales@clickatell.com)

--- 0 ---