



1. Content

1.	Content	2
2.	Change history	2
3.	Overview	2
4.	Context.....	3
5.	Introduction	3
6.	Available short codes	3
7.	How outbound messaging works	3
8.	How inbound messaging works	4
8.1	Method 1: The mobile user initiates the conversation	4
8.2	Method 2: Your application initiates the conversation	4
9.	Stop command	5
9.1	Stop commands – automatic unsubscribe	5
9.1.1	STOP command implementation – short codes & MO numbers	5
10.	How to configure your application.....	5
10.1	How to set the MO parameter	5
10.2	How to configure the sender ID	5
10.3	How to receive inbound messages.....	6
10.3.1	HTTP.....	7
10.3.2	FTP log file	7
10.3.3	SMPP (for advanced users).....	8
10.3.4	XML (over HTTP/S)	8
10.3.5	SOAP	8
10.3.6	Online reports only	9
10.4	Important notice about testing:	9
11.	Examples.....	9
11.1	Read First.....	9
11.2	Example 1: Mobile users SMS in to a short code to receive News Alerts	10
11.3	Example 2: Mobile users reply to a message you have sent them	10
12.	Terminology.....	10

2. Change history

Approximately six (6) months of changes are reflected:

Visit http://www.clickatell.com/downloads/Clickatell_USA_shortcode_technical_guide.pdf to check for updates to this document.

Version	Date	Section	Changes to Documentation
2.0	18 July 2008	11.3.5	Added SOAP MO callback format

3. Overview

This technical document is intended for users who wish to implement two-way messaging within the U.S. and states what number types are available for two-way messaging. All two-way messaging in the U.S. must be implemented through the use of short codes.

An explanation of how inbound and outbound messaging works is given, as well as how the API you have chosen needs to be configured to enable two-way messaging.

4. Context

This document is designed to follow on from Clickatell's HTTP, SMTP, FTP, XML, SMPP, SOAP and COM API Specification documents. It is presumed that the reader has a thorough understanding of these documents (whichever may be appropriate) prior to reading this technical guide. These documents are linked from <http://www.clickatell.com/developers.php>.

As you will need to configure your numbers within Clickatell Central, it is suggested that you read the online help on configuring two-way messaging:
https://www.clickatell.com/downloads/Clickatell_two-way_technical_guide.pdf.

You can apply for a two-way number within Clickatell Central.

5. Introduction

There are various applications and services that are associated with two-way messaging. Clickatell's role in this regard is to act as a conduit between the mobile user and your application. We will also route any responses from your application back to mobile recipients and vice versa. There is a predefined set of variables that Clickatell will pass back to your application when a message is received. You will need to ensure that your application is capable of interpreting these and responding accordingly.

6. Available short codes

Only dedicated short codes are available in the U.S. You may either choose to have a short code number randomly assigned to you or you may choose a specific short code number, if available.

These are known as:

- **Random:** A random short code assigned to you; e.g. 76432
- **Selected:** A specific short code requested by you if available; e.g. 33333

Note: A **selected** short code incurs higher ongoing monthly rental fees.

Short codes and services are subject to U.S. carrier approval; typical approval time is around 90 days.

7. How outbound messaging works

These are messages delivered by your application to the user via the Clickatell gateway. There are a number of ways of routing these messages to the Clickatell gateway for delivery:

- SMTP (EMAIL to SMS) - enabling a server or customer-generated email to be delivered as an SMS.
- HTTP/S - submitting either a POST or GET to the API server.
- FTP Upload – uploading a text file to Clickatell's FTP Server.

- COM Object – windows based application development.
- XML – XML wrapper over HTTP.
- SOAP – submit SOAP packets over HTTP/S.
- SMPP - Short message peer to peer (SMPP) protocol.

The API you choose to send these outbound messages must be linked by Clickatell to the short code and long numbers you use. This can be done within Clickatell Central.

Step 1: Log into your account at: <http://www.clickatell.com/login.php>

Step 2: On the main screen, select **Manage My Products**

Step 3: On the left select **Two-Way Messaging**

Step 4: Click **Edit** on the number you want to configure and choose the API you want to link it to from the drop down box provided.

Please review your chosen API specification document for more information on how to implement outbound messaging.

8. How inbound messaging works

Inbound messages to your application can arise in two ways:

8.1 Method 1: The mobile user initiates the conversation

A mobile handset user will create and send a text message to your short code number, which will be routed to your application by Clickatell.

Example: a mobile user sees an advertisement on TV stating that they should send an SMS to the number 36000 to get the latest News Alerts. They then send an SMS to 36000.

8.2 Method 2: Your application initiates the conversation

Your application sends a message via the Clickatell gateway to the user with the Sender ID set to your assigned short code. The user then replies to the message which is routed back to your application via the Clickatell gateway.

For this, Clickatell will route your messages through a specific carrier that will enable the ability for mobile users to reply to your allocated number.

Example: You send out a message to your users who have already opted-in to receive your service. It states "Reply now with the word NEWS to receive the latest news". The user replies to the message with the word "NEWS" and gets sent an SMS with the latest news.

9. Stop command

9.1 Stop commands – automatic unsubscribe

Clickatell has implemented a stop command system in line with the Mobile Marketing Association's Consumer Best Practice guidelines. The system will ensure that a mobile user's wish to unsubscribe is honoured in the event that your application or service fails to do so.

Note: Clickatell will always deliver all messages that were received from the handset to you, provided you have sufficient credits in your account.

9.1.1 STOP command implementation – short codes & MO numbers

Example: A handset sends in one of the following case-insensitive keywords (STOP, END, CANCEL, UNSUBSCRIBE or QUIT) to a short code.

Clickatell will pass this message on to you. The Clickatell gateway will expect a "confirmation of cancellation" message from you to the handset within a period of 6 hours. This must be sent either from the short code it was sent to.

If no confirmation of cancellation message is sent: You will no longer be able to send MT messages to the mobile number from the short code or MO number you sent it from. Clickatell will also send the message "You will no longer receive messages from the short code xxxxx" to the mobile number on your behalf. This will be billed to your account at normal message rates. This message will be sent from your short code. This MO number will be assigned and linked to your short code by Clickatell.

Note: If the handset sends in a new message that does not contain one of the following; **STOP, END, CANCEL, UNSUBSCRIBE, QUIT, STOP STOP or STOP ALL**, then you will once again be allowed to send messages to the mobile number.

If the handset sends in "STOP ALL" or "STOP STOP" to a short code then you will no longer be able to send messages from any of your MO numbers that are linked to your account.

10. How to configure your application

Note: It is assumed that you have already integrated your chosen API and are able to send messages.

10.1 How to set the MO parameter

When sending messages to mobile users that should be able to reply, the MO parameter must be set to 1. If you use SMPP the MO parameter is automatically set to 1. For more information on how to set this for your chosen API, see the API specification document.

When this parameter is set to 1, Clickatell will:

- Use a specific carrier to deliver the message.
- Set the correct Sender ID when sending to the handset.

10.2 How to configure the sender ID

Please note that when you send a message where you may need a reply, you will need to set the correct Sender ID (pre-configured short code) and set the MO parameter to 1. The pre-configured short code is the number that you have chosen to link to the API that you use to send outbound (MT) messages.

If the MO parameter is set to 1 and you specify a Sender ID that is different from the pre-configured Sender ID, then your setting will take preference. Clickatell will still use a specific carrier to try and enable the ability for the user to reply to the message. If however, your Sender ID is not supported by the carrier being used, the message will fail.

When Clickatell delivers the message the Sender ID may be changed as follows.

Your Sender ID	Sender ID set by Clickatell	
	If MO=1	If MO=0 or not set
If you set an alpha Sender ID (e.g. from=abc)	We will try to set the Sender ID you submitted. If we cannot, we will set a default Clickatell number.	We will try to set the Sender ID you submitted. If we cannot, we will set a default Clickatell number.
If you set a numeric Sender ID that is not a configured MO number.	We will try to set the Sender ID you submitted. If we cannot, we will set a default Clickatell number.	We will try to set the Sender ID you submitted. If we cannot, we will set a default Clickatell number.
If you set a configured MO number.	We will try to set the Sender ID you submitted.	We will try to set the Sender ID you submitted. If we cannot, we will set a default Clickatell Number.
If you set a short code as the Sender ID field.	We will set the Sender ID you submitted.	We will try to set the Sender ID you submitted. If we cannot, we will set a default Clickatell Number.
If you do not set anything as the Sender ID field.	We will set a default Clickatell number – you will not receive the replies.	We will set a default Clickatell number – you will not receive the replies.

10.3 How to receive inbound messages

In order to receive inbound messages, one of the methods below can be used. You can change which method you wish to use at any time. Configuration is done through Clickatell Central. Go to https://www.clickatell.com/central/twoway/twoway_central.php and select the number you wish to configure.

MO callback methods:

- HTTP GET
- HTTP POST
- SMPP
- XML GET
- XML POST
- FTP
- SOAP GET
- SOAP POST
- Online Reports

All inbound messages are recorded on your Clickatell account, which can be viewed and downloaded through **Reports** within Clickatell Central. This is useful if you do not wish to receive inbound messages into your application.

Extended length messages will be sent to you as individual messages. The last two digits of the UDH header will indicate which message part it is. The remaining part of the UDH header will be identical for all messages that are part of the extended message.

Example UDH header for message part 1: 050003330201

Example UDH header for message part 2: 050003330202

10.3.1 HTTP

Callback URLs will be used to send messages back to applications via a standard HTTP GET or POST. The reply-path URL is set by you within Central. The URL must begin with http://. HTTPS is accepted but only encryption is supported. Variables are passed back by the API on message response.

The variables returned to the URL are:

- Api_id (api_id=)
- MO message ID (moMsgId)
- Originating ISDN (from=)
- Destination ISDN (to=)
- Date and Time [MySQL format, GMT + 0200] (timestamp=)
- DCS Character Coding (charset=) [when applicable]
- Header Data [e.g. UDH etc.] (udh=) [when applicable]
- Message Data (text=)

Example: If you provide this URL <http://www.yourdomain.com/sms/sms.asp> then we will do a POST or GET as follows:

https://www.yourdomain.com/sms/sms.asp?api_id=12345&from=279991235642&to=27123456789×tamp=2008-08-0609:43:50&text=Hereisthe%20messagetext&charset=ISO-8859-1&udh=&moMsgId=b2aee337abd962489b123fda9c3480fa

What happens when we are unable to connect to your server?

Clickatell provides retries of MO callbacks. We follow retry as follows:

1. 2 minutes after the original attempt
2. 4 minutes after last retry
3. 8 minutes after last retry
4. 16 minutes after last retry
5. 25 minutes after last retry (max retries reached)

After this, we do not retry again.

10.3.2 FTP log file

As an alternative to using a callback URL, replies can be logged to an FTP file. The FTP file will be a text file, which can be retrieved manually, or via a script. The fields in the text file will match those listed above. If necessary, an example of a text file will be provided on request

If you don't already have an FTP account, you can add an FTP product within your Central account under **My Connections**.

```
Text file name: mo.log
Example of text within text-file:
2005-01-06[space]12:26:18[space]handset_number_here[tab]mo_number_here[tab]text_here
2005-01-06[space]12:27:18[space]handset_number_here[tab]mo_number_here[tab]text_here2
2005-01-06[space]12:28:18[space]handset_number_here[tab]mo_number_here[tab]text_here3
```

10.3.3 SMPP (for advanced users)

We can also send the response back to you via SMPP. Please see our SMPP API specification document for more information.

10.3.4 XML (over HTTP/S)

Callback URLs will be used to post XML formatted messages back to applications via a standard HTTP/S GET or POST. The reply-path URL is set by you within Clickatell Central. The URL must begin with http://. HTTPS is accepted but only encryption is supported. Variables are passed back by the API on message response. The data contained in the XML elements is encoded using standard XML entities.

The variables returned to the in the xml <clickmo> packet are:

- Api_id (<api_id>)
- MO message ID (moMsgId)
- Originating ISDN (<from>)
- Destination ISDN (<to>)
- Date and Time [MySQL format, GMT + 0200] (<timestamp>)
- DCS Character Coding (<charset>) [when applicable]
- Header Data [e.g. UDH etc.] (<udh>) [when applicable]
- Message Data (<text>)

Example: If you provide this: <http://www.yourdomain.com/sms/sms.asp> then we will do a post via a standard HTTP POST as follow:

```
data=<?xml version="1.0"?>
<clickmo>
  <api_id>xxx</api_id>
  <moMsgId>fa6ba35b330ce1bc7e2008e5d92d57aa</moMsgId>
  <from>handset_number_here</from>
  <to>mo_number_here</to>
  <timestamp>2007-02-26 14:36:50</timestamp>
  <text>xxx</text>
  <charset>ISO-8859-1</charset>
  <udh></udh>
</clickmo>
```

10.3.5 SOAP

SOAP callbacks can be sent as an HTTP GET or POST with in the 'data' parameter (eg: [http://www.yourscript.com/callback.php?data=<?xml version="1.0"](http://www.yourscript.com/callback.php?data=<?xml version='1.0')

Example

```
Data=<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="clickatell_mo">
<SOAP-ENV:Body>
<tns:mo_callback xmlns:tns="clickatell_mo">
<api_id xsi:type="xsd:int">xyz</api_id>
<from xsi:type="xsd:string">xyz</from>
<to xsi:type="xsd:string">xyz</to>
<timestamp xsi:type="xsd:string">xyz</timestamp>
<charset xsi:type="xsd:string">xyz</charset>
<udh xsi:type="xsd:string">xyz</udh>
<text xsi:type="xsd:string">xyz</text>
<momsgid xsi:type="xsd:string">xyz</momsgid>
</tns:mo_callback>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.3.6 Online reports only

All inbound messages are recorded on your Clickatell account, which can be viewed within Clickatell Central. Go to **Message Reports>>Two-Way** and generate a report using the parameters supplied. Choose this option if you do not wish inbound messages to be sent to your application.

10.4 Important notice about testing:

Do not try and send to any of your assigned numbers via the Clickatell system. This may cause your messages to go into a loop.

11. Examples

11.1 Read First

The following is required for the examples below to work as shown:

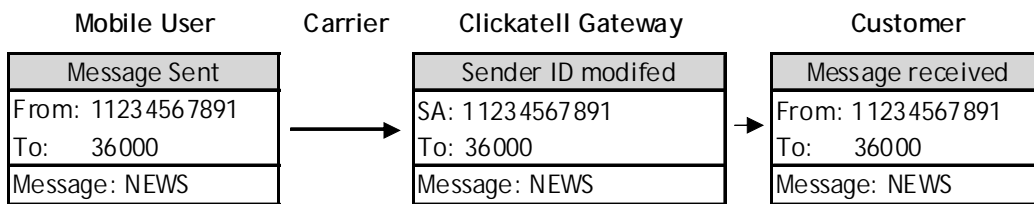
- The MO parameter is set to 1
- You have configured how Clickatell must send messages to you (through Clickatell Central)

Definitions:

- "Mobile User" is interchangeable with the word "handset".
- A "customer" is a customer of Clickatell's.
- SA = Source Address, the handset/mobile number the message comes from.

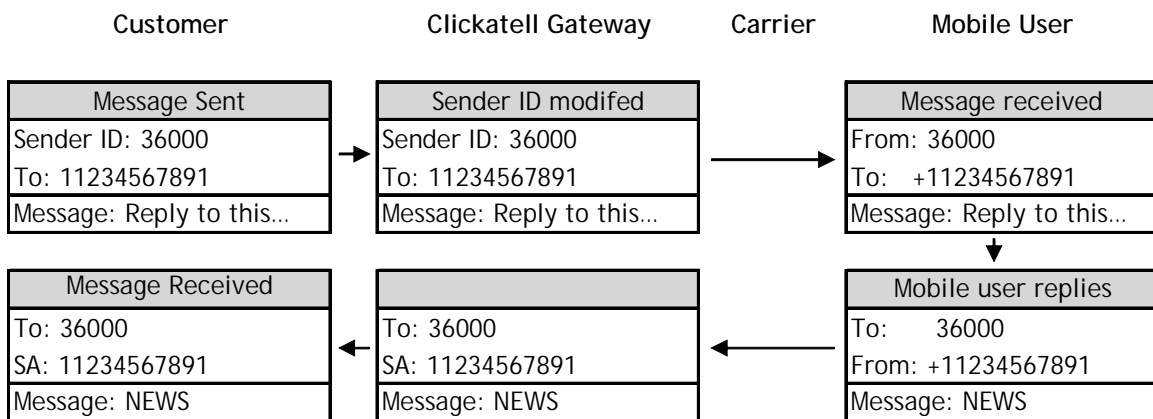
11.2 Example 1: Mobile users SMS in to a short code to receive News Alerts

In this example, the mobile user sees an advertisement in a newspaper that states “SMS 36000 starting with the word NEWS to receive the latest News Alerts”. The mobile user sends the requested word via SMS to 36000. The message is received from the handset by the Clickatell gateway (via the network operator) and delivered to the application.



11.3 Example 2: Mobile users reply to a message you have sent them

In this example you send out a message to an opt-in user “Reply to this message in order to receive the latest News Alerts. Start with the word NEWS”. The user then replies to the message.



12. 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.
- **Customer:** A registered Clickatell customer utilizing 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/MDN 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.

- **Short code:** A short number which is common across all the operators for a specific region.
 - **Upstream gateway:** A network operator, third party or Clickatell's message service centre (SMSC).
-