



1. Content

| | | |
|-------|--|----|
| 1. | Content | 2 |
| 2. | Change history | 3 |
| 3. | Overview | 3 |
| 4. | Introduction | 4 |
| 5. | Getting started | 5 |
| 6. | Basic methods | 6 |
| 6.1 | Authentication and session IDs | 6 |
| 6.2 | Ping | 6 |
| 6.3 | Send message | 6 |
| 6.4 | Send simple SMS message | 6 |
| 6.5 | Query message | 7 |
| 7. | Published properties | 8 |
| 8. | Additional methods | 10 |
| 8.1 | Delete/Stop message | 10 |
| 8.2 | Query balance | 11 |
| 8.3 | Coverage query | 11 |
| 8.4 | Get message charge query | 11 |
| 8.5 | Token pay | 11 |
| 8.6 | Send an Over-The-Air configuration as XML | 11 |
| 8.7 | Send an internet bookmark Over-The-Air | 12 |
| 8.8 | Send settings Over-The-Air | 12 |
| 8.9 | Reset the COM object | 12 |
| 8.10 | Determining the com object version number | 13 |
| 8.11 | Encode to Unicode | 13 |
| 9. | Batch messaging | 14 |
| 9.1 | Start batch | 14 |
| 9.2 | Send messages to existing batch | 14 |
| 9.3 | Quick send to batch | 14 |
| 9.4 | End batch | 14 |
| 10. | 8-Bit messaging | 15 |
| 11. | Configuring internet connection | 15 |
| 11.1 | Configuring the proxy settings only | 15 |
| 11.2 | Configuring the proxy settings including testing | 16 |
| 11.3 | Setting the connection time-out value | 16 |
| 11.4 | Setting receive timeout value | 17 |
| 12. | Enums and their values | 17 |
| 12.1 | Message types (eSMSMsgType) | 17 |
| 12.2 | Concatentation (eSMSConcatType) | 17 |
| 12.3 | Callback type (eSMSCallbackType) | 17 |
| 12.4 | Callback URL | 18 |
| 12.5 | Communication type (eSMSCommunicationType) | 18 |
| 12.6 | Message ID type (eSMSAPI or CLI_ID) | 18 |
| 12.7 | eSMSInetConnection | 18 |
| 12.8 | Delivery queue (eSMSSendQueue) | 19 |
| 12.9 | Gateway escalation (eSMSEscalate) | 19 |
| 12.10 | Required features (eSMSReqFeatures) | 19 |
| 12.11 | Send Over-The-Air | 19 |
| 13. | Message examples | 21 |
| 13.1 | Simple example | 21 |
| 13.2 | Example 2 – login, send, and query balance | 23 |
| 13.3 | Send a Unicode message | 23 |

| | | |
|-----|--|----|
| 14. | Appendix A: Error codes | 24 |
| 15. | Appendix B: Message statuses (eSMSMsgStatus) | 25 |
| 16. | Terminology | 26 |
| 17. | Contact details | 27 |

2. Change history

Approximately six (6) months of changes are reflected.

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

| Version | Date | Section | Changes to Documentation |
|---------|-----------------|---------|---|
| 2.2.8 | 14 January 2009 | All | Update link added to document, terminology table moved. |
| 2.2.7 | 04 Sept. 2008 | 15 | Added new error codes 118, 120, 121,122, 123, 128, 129, 901 |

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 is available at <http://support.clickatell.com/guides/clickatell/routing.php>. Please note that we request delivery acknowledgement for **every** message we send although delivery acknowledgement status will only be delivered to you if requested.

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

- SMTP - enabling a server or client generated email 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.
- XML – posting to our gateway using XML over HTTP/S.
- COM Object – for Windows based development.
- SOAP – Submit SOAP packets over HTTP/S
- SMPP – customers requiring a high throughput binary socket connection.

In order to reduce testing costs, Clickatell offers a test number range. Messages sent to any number on this prefix will only be charged 1/3 of a credit. Use the number 279991xxxx (for South Africa) or 1999xxxxxx (for the U.S.) where “xxxx” represents any numeric string. Message statuses will be returned.

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

4. Introduction

The Clickatell HTTP specification describes how a user can interact with the Clickatell SMS Gateway via the HTTP or HTTPS communication protocols. Since not everybody is able or willing to implement the whole HTTP/S channel, we have made a COM Object available that takes the hassle out of this process.

The COM-API Object exports a rich set of methods and definitions that will make it easy for a user to integrate SMS sending into their programs or web pages. Various sections of this document will show you how to send SMSs in Visual Basic, both in early and late binding mode, and in VBScript (essentially late bound Visual Basic). Since the COM Object interface is completely documented, any COM-enabled programming language or framework such as .NET can be used.

The COM Object is defined by the following class name: **SMS_COMAPI.SMS**

The COM Object supports all interfaces described by the HTTP API specification with the exception of wap and mms push. Additionally it also provides properties for all parameters defined in the HTTP API specification. It is important that parameter lengths and allowed characters, as defined in the HTTP API Specifications, are followed.

When using the COM Object with early binding, e.g. referencing the SMS_COMAPI type library in VB, a range of predefined constants is also available. When using late binding the SDK supplies the SMS_COMAPI.bas file with the required constant definitions.

Important: Many of the parameters and methods defined in this document have more detailed descriptions, which are found in the HTTP API Technical Specification document:
http://www.clickatell.com/downloads/http/Clickatell_HTTP.pdf

Security

Since the Internet is an open communication medium, security might be an issue. If you do not want anybody to intercept your SMS messages as they are being sent to Clickatell, we offer you the means to send your SMS encrypted. The secure communication channel is established via a SSL (Secure Socket Layer) channel.

The actual communication between the COM Object and Clickatell is transparent to your programming, so your code does not have to change. We have provided the smsCommunicationType method to allow you select encrypted or non-encrypted SMS submittal.

The COM Object does not need to have the SSL libraries installed in order to send SMS. The CanSendViaSSL property indicates if the SSL libraries are loaded or not.

Calling the method with the SMS_COMMS_SSL parameter, switches the COM Object into encrypted communication mode. SMS_COMMS_HTTP is the default (non-encrypted) way of submitting your SMS.

5. Getting started

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

Step 1 - register for a Clickatell account

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

- Go to http://www.clickatell.com/products/sms_gateway.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 connection.

Step 2 - add a registered API connection (sub-product)

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 to use from the drop down menu (**Add Connection**).
- Complete the form.

If you register more than one API connection, the description name you enter for each must be unique - you cannot have multiple APIs with the same name.

After successfully submitting the form, your authentication details will be displayed, including each connection's unique API ID (api_id). These authentication details are required when connecting to the Clickatell gateway to send a message.

Note: For more information on managing your API connections within your Clickatell account see our API guide at http://support.clickatell.com/guides/clickatell/api_guide.php.

6. Basic methods

The following sections describe each of the methods published by the COM Object. Every section gives details about the method name, the input parameters list and return values.

6.1 Authentication and session IDs

| | |
|--------------------|--|
| Name: | Authenticate |
| Desc: | Used to authenticate. A Session ID is returned, which must be used with all future transactions |
| Parameters: | String API_ID String Username String Password Boolean Encode [True] |
| Returns: | OK: session_id or ERR: Error number |

6.2 Ping

| | |
|--------------------|--|
| Name: | Ping |
| Desc: | This will prevent a Session ID from expiring in periods of inactivity. The Session ID is set to expire after 15 minutes of inactivity. |
| Parameters: | |
| Returns: | OK or ERR: Error number |

6.3 Send message

| | |
|--------------------|--|
| Name: | SendMsg |
| Desc: | This method will send an SMS message. |
| Parameters: | Boolean UseSessionID [True] String TextOfSMS String ToMobileNumber |
| Returns: | ID: apiMsgId or ERR: Error number |

6.4 Send simple SMS message

| | |
|--------------------|--|
| Name: | SendSimpleSMS |
| Desc: | This method will send an SMS message without you having to authenticate first. |
| Parameters: | String API_ID String Username String Password String ToMobileNumber |

| | |
|----------|---|
| | String TheMessage |
| Returns: | ID: apiMsgId or ERR: Error number |

6.5 Query message

| | |
|--------------|--|
| Name: | QueryMsg |
| Desc: | This returns the status of a message. Either the apiMsgId or the Client Message ID can be used to identify a message. See Section 13: Enums and Their Values for status codes. |
| Parameters: | String ID Enum eIDType [SMS_API_ID] |
| Returns: | ID: xxxx Status: xxxx or ERR: Error number |

| | |
|--------------|--|
| Name: | QueryMsgAPIID |
| Desc: | Same as QueryMsg but only for querying with an apiMsgId. |
| Parameters: | String ApiMsgID |
| Returns: | ID: xxxx Status: xxxx or ERR: Error number |

| | |
|--------------|---|
| Name: | QueryMsgCLIID |
| Desc: | Same as QueryMsg but only for querying with a cliMsgId. |
| Parameters: | String CliMsgID |
| Returns: | ID: xxxx Status: xxxx or ERR: Error number |

7. Published properties

| Name | Data Type | Description | Valid Values | Direction |
|------------------|-------------------|--|---|---------------|
| smsAPI_ID | BSTR | API sub-product ID – The value for this mandatory parameter can be found or created by logging in online and going to Manage My Products. | | R/W |
| smsUSER | BSTR | The username you specified. | | R/W |
| smsPASSWORD | BSTR | Your account password. | | R (limited)/W |
| smsSESSION_ID | BSTR | The session ID from Authenticate. | | R |
| smsTO | BSTR | 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. | R/W |
| smsTEXT | BSTR | The text content of the message. | | R/W |
| smsFROM | BSTR | The source/sender address. | A valid international format number between 1 and 16 characters long, or an 11 character alphanumeric string. | R/W |
| smsDELIVERY_ACK | Boolean | 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 required features parameter. | Always set to True on the gateway. Setting to false will have no affect. | R/W |
| smsCALLBACK | eSMSCallback Type | Enables you to receive message delivery statuses via one of the following callback types: HTTP, SOAP or XML. This is posted to a URL of yours using the POST or GET method. This is done every time a message status is updated. | Given by enum | R/W |
| smsDELIVERY_TIME | long | Delays delivery of SMS to mobile device in minutes relative to the time at which the SMS was received by our gateway. This should be greater than 10 minutes for best effect. Smaller time frames | | R/W |

| | | | | |
|----------------------|----------------|--|-----------------------------------|-----|
| | | may be delivered too soon. Maximum value of 10080 or 7 days. | | |
| smsCONCAT | eSMSConcatType | Split long messages into smaller messages where needed. | Given by enum | R/W |
| smsMAX_CREDITS | Long | Max number of credits that can be sent (Please refer to the HTTP API Technical Specification document) | As per profiles 0.8,1,1.5,2,2.5,3 | R/W |
| smsREQUIRED_FEATURES | Long | A bit-field. 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. | See table eSMSreqFeatures | R/W |
| smsQUEUE | ESMSSendQueue | Delivers the message through one of three queues assigned to each client account. Messages in the highest priority queue will be delivered first. | Given by enum | R/W |
| smsESCALATE | ESMSEscalate | Prompts an escalation to an alternative delivery gateway, should the message be delayed for a set length of time. | Given by enum | R/W |
| smsMO | Boolean | 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. | True/False | R/W |
| smsCLI_MSG_ID | BSTR | Client Message ID defined by user for client message tracking. The value you set for this parameter should not contain spaces. | | R/W |
| smsUNICODE | Boolean | Two digit language code. Convert your text to Unicode [UCS-2 encoding]. See http://www.Unicode.org/ . | True/False | R/W |
| smsMSG_TYPE | SMSMsgType | Optional parameter which must be set to send specially formatted messages; e.g. logos and ringtones. | Given by enum | R/W |
| smsUDH | BSTR | Allows you to set your own message types. Do not use if you set the message type parameter. | | R/W |

| | | | | |
|----------------------|-----------------------|--|---------------|-----|
| smsVAL_PERIOD | long | The validity period, after which the message will not be delivered. Set in minutes. | | R/W |
| smsERROR | SMSErrors | The last Error number that occurred in the COM Object. | Given by enum | R |
| smsERROR_DESC | BSTR | A text description of the last error. | | R |
| ThrowError | Boolean | If set, causes COM Object to report errors via the COM interface. If not set, the programmer is responsible for checking each method call. | True/False | R/W |
| smsIsLoggedIn | Boolean | Returns true if the COM Object has a valid Session ID. | True/False | R |
| smsCommunicationType | eSMSCommunicationType | Send the SMS via HTTP or SSL. | Given by enum | R/W |
| smsBATCH_ID | BSTR | The ID of a newly created batch SMS. | | R |

8. Additional methods

8.1 Delete/Stop message

| | |
|--------------|--|
| Name: | StopMsg |
| Desc: | This method enables you to stop the message delivery. Only useful for deferred deliveries. If messages have already reached a delivered status, they can no longer be stopped. |
| Parameters: | String ID Enum eIDType [SMS_API_ID] |
| Returns: | ID: xxxx Status: xxxx or ERR: Error number |

| | |
|--------------|---|
| Name: | StopMsgAPIID |
| Desc: | Same as StopMsg but only for use with apiMsgId's. |
| Parameters: | String ApiMsgID |
| Returns: | ID: xxxx Status: xxxx or ERR: Error number |

| | |
|--------------|---|
| Name: | StopMsgCLIID |
| Desc: | Same as StopMsg but only for use with cliMsgId's. |
| Parameters: | String CliMsgID |
| Returns: | ID: xxxx Status: xxxx or ERR: Error number |

8.2 Query balance

| | |
|--------------------|--|
| Name: | QueryBalance |
| Desc: | This will return the number of credits available on the currently authenticated account. |
| Parameters: | |
| Returns: | Credits: xxxx or ERR: Error number |

8.3 Coverage query

| | |
|--------------------|--|
| Name: | QueryCoverage |
| Desc: | Use this function to find out if we support the sending of SMSs to the specified mobile number. Please refer to HTTP API Technical Specification document for further information. |
| Parameters: | String To |
| Returns: | String Informing whether covered or not |

8.4 Get message charge query

| | |
|--------------------|--|
| Name: | GetMsgCharge |
| Desc: | Use this function to find out how much a specific message has cost. All you need to do is make sure that you are logged into the server (authenticated), or that the API_ID, User and Password values are set correctly. Then call the GetMsgCharge function with the ApiMsgID of the message you want to get the cost of. |
| Parameters: | Bool UseSessionID String ApiMsgID |
| Returns: | String CostOfMessage |

NOTE: The return parameter can be a fractional number. Depending on the routing parameters that you have selected a message might have a cost of 0.8 credits.

8.5 Token pay

| | |
|--------------------|--|
| Name: | TokenPay |
| Desc: | Update the credits via a payment token. This redeems the credit value of the token or voucher, into the authenticated account. |
| Parameters: | String Token |
| Returns: | OK: 605 - if the payment was successful or ERR: Error number |

8.6 Send an Over-The-Air configuration as XML

| | |
|--------------|---------------------|
| Name: | SendOTAasXML |
|--------------|---------------------|

| | | |
|-------------|--|--------------|
| Desc: | This function sends bookmark or WAP settings to a mobile device. All settings need to have been XML encoded. | |
| Parameters: | Bool | UseSessionID |
| | String | To |
| | String | ConfigInXML |
| Returns: | OK: or ERR: Error message | |

8.7 Send an internet bookmark Over-The-Air

| | | |
|--------------|--|--------------------------|
| Name: | SendOTABookmark | |
| Desc: | This function sends a single bookmark to a mobile device | |
| Parameters: | String | To |
| | String | Name (max 20 characters) |
| | String | Url (max 255 characters) |
| Returns: | OK: or ERR: Error message | |

8.8 Send settings Over-The-Air

| | | |
|--------------|--|-----------------------------------|
| Name: | SendOTA | |
| Desc: | This function sends a bookmark or WAP settings to a mobile device. | |
| Parameters: | String | To |
| | ENUM | eOTAType Type |
| | String | Name |
| | String | Url |
| | ENUM | eOTABearer Bearer |
| | ENUM | eOTAPPPLoginType PPPLoginType |
| | ENUM | eOTAPPPAuthType PPPAuthType |
| | String | PPPAuthName |
| | String | PPPAuthSecret |
| | ENUM | eOTAProxyLoginType ProxyLoginType |
| | ENUM | eOTAProxyType ProxyType |
| | String | ProxyAuthName |
| | String | ProxyAuthSecret |
| | String | Proxy |
| | ENUM | eOTACSDCallType CSDCallType |
| | ENUM | eOTACSDCallSpeed CSDCallSpeed |
| | String | CSDDialString |
| | ENUM | eOTAPort Port |
| | String | ISPName |
| | String | SMSC |
| | String | GPRSAccess |
| Returns: | OK: or ERR: Error message | |

8.9 Reset the COM object

| | |
|--------------|--|
| Name: | ResetSMS |
| Desc: | Reset the COM Object, into a well defined state. |
| Parameters: | |
| Returns: | |

8.10 Determining the com object version number

| | |
|--------------|---|
| Name: | Version |
| Desc: | Query the version number of the COM Object, or find the version number of the latest available COM Object from the server |
| Parameters: | Bool FromServer |
| Returns: | String VersionNumber |

Note: This function returns the version number of the installed SMS COM Object or queries number of the latest available SMS COM Object from the server. The numbers are in x.y.z.w format.

8.11 Encode to Unicode

| | |
|--------------|--|
| Name: | EncodeToUnicode |
| Desc: | Converts the passed string into a hex encoded Unicode representation. See 5.3, Example 3 – Send a Unicode Message. |
| Parameters: | String StringToConvert |
| Returns: | Converted string |

9. Batch messaging

9.1 Start batch

| | |
|--------------------|---|
| Name: | StartBatch |
| Desc: | This method starts a batch of SMS messages. A BatchID is returned, which needs to be used in all subsequent batch transactions. |
| Parameters: | String Template |
| Returns: | ID: xxxx or ERR: Error number |

9.2 Send messages to existing batch

| | |
|--------------------|--|
| Name: | SendBatchItem |
| Desc: | Send a message to a batch passing the destination mobile number and optional field replacement values. |
| Parameters: | String To String Field1 [""] String Field2 [""] String Field3 [""] String Field4 [""] String Field5 [""] String Field6 [""] String Field7 [""] String Field8 [""] String Field9 [""] |
| Returns: | ID: apiMsgId or ERR: Error number |

9.3 Quick send to batch

| | |
|--------------------|---|
| Name: | QuickBatchSend |
| Desc: | Send the same batch (template) message to comma-separated list of mobile numbers. |
| Parameters: | String To |
| Returns: | ID: xxxx To: xxxx or ERR: Error number To: xxxx |

9.4 End batch

| | |
|--------------------|--|
| Name: | EndBatch |
| Desc: | Terminate the started Batch. Batches will expire automatically after 24 hours. |
| Parameters: | |
| Returns: | OK |

or
 ERR: Error number

10. 8-Bit messaging

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. If you are comfortable with the creation of your own UDH, we also enable you to set it directly using the **udh** parameter. To simplify the process, we have provided a number of pre-defined message types (see the **msg_type** parameter).

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:024A3A5585E195B198040042D9049741A69761781B6176156174288B525D85E0A26C24C49A61762
8930BB125E055856049865885D200
```

11. Configuring internet connection

11.1 Configuring the proxy settings only

| Name: | ConfigureProxy |
|-------------|--|
| Desc: | Configure the use of an Internet proxy when using the HTTP communication method. |
| Parameters: | Enum eSMSInetConnection [SMS_INET_PRECONFIG] String ProxyAddressOrIP String Port |
| Returns: | OK: or ERR: Error message |

The first parameter specifies the Internet access required.

SMS_INET_DIRECT - Resolves all host names locally, thus not using any proxies.

SMS_INET_PRECONFIG - Retrieves the proxy or direct configuration from the registry. This uses the same settings as Internet Explorer.

SMS_INET_PRECONFIG_NO_RUN - Retrieves the proxy or direct configuration from the registry and prevents the use of a startup JScript or Internet Setup (INS) file.

SMS_INET_PROXY - Passes requests to the proxy.

The second parameter is the name or IP address of the proxy server. This parameter is only required when SMS_INET_PROXY is used.

By default the configuration values are retrieved from the registry [SMS_INET_PRECONFIG] and use the same settings as IE.

11.2 Configuring the proxy settings including testing

| Name: | ConfigureProxyEx |
|-------------|---|
| Desc: | Configure the use of an Internet proxy when using the HTTP communication method. Also testing if the proxy server can be reached and if the configuration is correct and the SMS server can be reached. |
| Parameters: | Bool TestForOnline Enum eSMSInetConnection [SMS_INET_PRECONFIG] String ProxyAddressOrIP String Port String Username String Password |
| Returns: | OK: or ERR: Error message |

The first parameter can be used to test the communication channel that is being configured. There are two stages in this test. First a connection to the proxy server is established, if this fails an error with code 95 (ERR: 95, ...) is returned. If the first connection was successful then the proxy is configured and a test connection to the SMS server is attempted. If the second connection fails an error with code 96 (ERR: 96, ...) is returned.

The second parameter specifies the internet access required.

SMS_INET_DIRECT - Resolves all host names locally, thus not using any proxies, in this case no connection testing is done, even when the *TestForOnline* parameter is set to *true*.

SMS_INET_PRECONFIG - Retrieves the proxy or direct configuration from the registry. This uses the same settings as Internet Explorer.

SMS_INET_PRECONFIG_NO_RUN - Retrieves the proxy or direct configuration from the registry and prevents the use of a startup JScript or Internet Setup (INS) file.

SMS_INET_PROXY - Passes requests to the proxy.

The third parameter is the name or IP address of the proxy server. This parameter is only required when SMS_INET_PROXY is used.

The fourth and fifth parameters give the proxy's Username and Password combination. Some proxy servers require a user to authenticate before a connection via the proxy is allowed, these parameters represent the authentication credentials.

By default the configuration values are retrieved from the registry [SMS_INET_PRECONFIG] and use the same settings as IE.

11.3 Setting the connection time-out value

| Name: | SetConnectTimeout |
|-------------|--|
| Desc: | Use this function to set the number of milliseconds to wait before a HTTP connection-attempt times out. This can be used to increase the default HTTP timeout when you are on a slow connection. |
| Parameters: | Long TimeoutValueinMilliseconds |
| Returns: | Long PreviousTimeoutValueinMilliseconds |

11.4 Setting receive timeout value

| | | |
|--------------------|--|------------------------------------|
| Name: | SetReceiveTimeout | |
| Desc: | Use this function to set the number of milliseconds to wait before a HTTP read-attempt times out. This can be used to increase the default HTTP timeout when you are on a slow connection. | |
| Parameters: | Long | TimeoutValueinMilliseconds |
| Returns: | Long | PreviousTimeoutValueinMilliseconds |

12. Enums and their values

12.1 Message types (eSMSMsgType)

| Definition | Comment | Value |
|---------------------------|---|-------|
| SMS_TYPE_TEXT | TEXT | 0 |
| SMS_TYPE_FLASH | FLASH | 1 |
| SMS_TYPE_NOKIA_OLOGO | Nokia Operator Logo | 3 |
| SMS_TYPE_NOKIA_GLOGO | Nokia Group Logo | 4 |
| SMS_TYPE_NOKIA_PICMSG | Nokia Picture Message | 5 |
| SMS_TYPE_NOKIA_RINGTONE | Nokia Ringtone | 6 |
| SMS_TYPE_NOKIA_CLEAN_LOGO | Nokia Clean Logo | 7 |
| SMS_TYPE_NOKIA_VCARD | Nokia Business Card | 8 |
| SMS_TYPE_NOKIA_EVENT_CAL | Nokia Event Calendar | 9 |
| SMS_TYPE_NOKIA_RTTL | Send a RTTL format ringtone to Nokia handsets | 10 |

12.2 Concatentation (eSMSConcatType)

| Definition | Comment | Value |
|------------------|-------------------------|-------|
| SMS_CONCAT_NONE | NO concatenation | 0 |
| SMS_CONCAT_1_MSG | Concat to max 1 message | 1 |
| SMS_CONCAT_2_MSG | Concat to max 2 message | 2 |
| SMS_CONCAT_3_MSG | Concat to max 3 message | 3 |

12.3 Callback type (eSMSCallbackType)

| Definition | Comment | Value |
|-------------------|---|-------|
| SMS_CALLBACK_NONE | NO Callback [default] | 0 |
| SMS_CALLBACK_BOTH | Return SMSC and Delivery acknowledgements | 3 |

We now request delivery acknowledgements for all messages. The callback URL is set in the preferences section of the particular API product within your client account after logging in on-line. The URL must begin with `http://`.

Variables are passed back by the API on either message delivery or failure, providing the callback option is set in the original post. The variables returned are **to**, **apimsgid** and **climsgid**, **api_id**, **destaddr**, **timestamp**, **from**, **status** and **charge**. The possible statuses which are returned are 003, 004, 005, 006, 007 & 010 (as per appendix B).

Sample Callback to your callback URL:

http://www.yourURL.com/script.asp?api_id=12345&apiMsgId=96905854f5045354a1b36134acf81&cliMsgId=123456&status=003×tamp=1055155528&to=2782123456&from=Sender&charge=2.5

12.4 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:

- HTTP GET
- HTTP POST
- XML GET
- XML POST
- SOAP GET
- SOAP POST

The callback URL and optional “Username” and “Password” authentication parameters can be 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://` (non-encrypted) or `https://` (encrypted). These are NOT your Clickatell username and password but are a username and password of your choice to add additional security.

The variables returned are **apiMsgId**, **cliMsgId**, **to**, **timestamp**, **from**, **status** and **charge**

12.5 Communication type (eSMSCommunicationType)

| Definition | Comment | Value |
|----------------|----------------------------|-------|
| SMS_COMMS_HTTP | Using HTTP [default] | 0 |
| SMS_COMMS_SSL | Using HTTPS (SSL security) | 1 |

12.6 Message ID type (eSMSAPI or CLI_ID)

| Definition | Comment | Value |
|------------|---|-------|
| SMS_API_ID | apiMsgId (Gateway-specified message tracking code) | 1 |
| SMS_CLI_ID | cliMsgId (customer specified message tracking code) | 2 |

12.7 eSMSInetConnection

| Definition | Comment | Value |
|---------------------------|--|-------|
| SMS_INET_DIRECT | Directly connected to Internet | 1 |
| SMS_INET_PRECONFIG | Proxy config from registry | 2 |
| SMS_INET_PRECONFIG_NO_RUN | Proxy config from registry, no scripts | 3 |
| SMS_INET_PROXY | Use proxy list | 4 |

12.8 Delivery queue (eSMSSendQueue)

| Definition | Comment | Value |
|---------------|---|-------|
| SMS_QUEUE_1ST | 1 st Queue – Highest Priority | 1 |
| SMS_QUEUE_2ND | 2 nd Queue – Medium Priority | 2 |
| SMS_QUEUE_3RD | 3 rd Queue – Lowest Priority [default] | 3 |

12.9 Gateway escalation (eSMSEscalate)

| Definition | Comment | Value |
|----------------------------|---|-------|
| SMS_ESCALATE_OFF | Off [default] | 0 |
| SMS_ESCALATE_WITHIN_60SECS | Escalate immediately to an alternative route if messages are queued on the least-cost route | 1 |

12.10 Required features (eSMSReqFeatures)

| Definition | Comment | Value |
|-----------------------|---|-------|
| SMS_REQ_FEAT_TEXT | Text [default] | 1 |
| SMS_REQ_FEAT_8BIT | 8-bit [default] | 2 |
| SMS_REQ_FEAT_UDH | UDH (Binary) (default) | 4 |
| SMS_REQ_FEAT_UCS2 | UCS2 / Unicode (default) | 8 |
| SMS_REQ_FEAT_ALPHA | Alpha Originator | 16 |
| SMS_REQ_FEAT_NUMBER | Numeric Originator | 32 |
| SMS_REQ_FEAT_REPLY | Enable reply to a MT message with numeric Sender ID | 64 |
| SMS_REQ_FEAT_FLASH | Flash Messaging | 512 |
| SMS_REQ_FEAT_DELIVACK | Delivery Acknowledgements | 8192 |
| SMS_REQ_FEAT_CONCAT | Concatenation (default) | 16384 |

12.11 Send Over-The-Air

12.11.1 eOTAType

The next set of definitions all relate to the Send Over-The-Air (SendOTA) method. The method allows you to send WAP settings or Internet bookmarks to a mobile device.

| Definition | Comment | Value |
|-------------------|-----------------------|-------|
| OTA_TYPE_SETTINGS | Set settings | 1 |
| OTA_TYPE_BOOKMARK | Set a single bookmark | 2 |

12.11.2 eOTABearer - OTA Bearer

| Definition | Comment | Value |
|----------------------|--|-------|
| OTA_BEARER_GSM_CSD | Circuit Switched Data | 0 |
| OTA_BEARER_GSM_SMS | Short Message Service | 1 |
| OTA_BEARER_GSM_USSD | Unstructured Supplementary Services Data | 2 |
| OTA_BEARER_IS136_CSD | Circuit Switched Bearer | 3 |
| OTA_BEARER_GPRS | General Packet Radio Service | 4 |

These parameters select how the Over-The-Air (OTA) configuration data is to be delivered to the mobile device.

12.11.3 eOTAPPPLoginType - OTA PPP Login Type

| Definition | Comment | Value |
|-----------------------|--|-------|
| OTA_PPP_LOGIN_NOT_SET | Not Set | 0 |
| OTA_PPP_LOGIN_AUTO | Automatic login | 1 |
| OTA_PPP_LOGIN_MANUAL | Manual login, need username and password | 2 |

12.11.4 eOTAPPPAuthType- OTA PPP Auth Type

| Definition | Comment | Value |
|----------------------|------------------------------------|-------|
| OTA_PPP_AUTH_NOT_SET | Not Set | 0 |
| OTA_PPP_AUTH_PAP | Password Authentication Protocol | 1 |
| OTA_PPP_AUTH_CHAP | Challenge Handshake Authentication | 2 |
| OTA_PPP_AUTH_MSCHAP | Microsoft(TM)-CHAP | 3 |

12.11.5 eOTAProxyType- OTA Proxy Type

| Definition | Comment | Value |
|------------------------|---------------|-------|
| OTA_PROXY_TYPE_NOT_SET | Not Set | 0 |
| OTA_PROXY_TYPE_MSISDN | MSISDN number | 1 |
| OTA_PROXY_TYPE_IPV4 | IP Address | 2 |

12.11.6 eOTAProxyLoginType - OTA Proxy Login Type

| Definition | Comment | Value |
|------------|---------|-------|
|------------|---------|-------|

| | | |
|-------------------------|--|---|
| OTA_PROXY_LOGIN_NOT_SET | Not Set | 0 |
| OTA_PROXY_LOGIN_AUTO | Automatic login to proxy | 1 |
| OTA_PROXY_LOGIN_MANUAL | Manual login, need username and password | 2 |

12.11.7 eOTAPort

| Definition | Comment | Value |
|---------------|----------------------------------|-------|
| OTA_PORT_9200 | 9200 [Default] (connection less) | 0 |
| OTA_PORT_9201 | 9201 (connection oriented) | 1 |
| OTA_PORT_9202 | 9202 (connection less) | 2 |
| OTA_PORT_9203 | 9203 (connection oriented) | 3 |

12.11.8 eOTACSDCallType - OTA Circuit Switched Call Type

| Definition | Comment | Value |
|-----------------------|----------|-------|
| OTA_CSD_CALL_ANALOGUE | Analogue | 0 |
| OTA_CSD_CALL_ISDN | ISDN | 1 |

12.11.9 eOTACSDCallSpeed - OTA Circuit Switched Call Speed

| Definition | Comment | Value |
|-----------------------|------------|-------|
| OTA_CSD_SPEED_NOT_SET | Not Set | 0 |
| OTA_CSD_SPEED_AUTO | AUTO | 0 |
| OTA_CSD_SPEED_9600 | 9600 baud | 1 |
| OTA_CSD_SPEED_14400 | 1440 baud | 2 |
| OTA_CSD_SPEED_19200 | 19200 baud | 3 |
| OTA_CSD_SPEED_28800 | 28800 baud | 4 |
| OTA_CSD_SPEED_38400 | 38400 baud | 5 |
| OTA_CSD_SPEED_43200 | 43200 baud | 6 |
| OTA_CSD_SPEED_57600 | 57600 baud | 7 |

13. Message examples

The following sections give some programming examples on how the SMS COM Object can be used in various languages.

13.1 Simple example

13.1.1 VISUAL BASIC – Early Binding

To enable your VB programming environment to show the possible COM interfaces and defined constants, you need to reference the SMS_COMAPI Type Library. Click on Projects-References... and select the 'SMS_COMAPI 1.0 Type Library'. This will import the type library.

Step 1 – Create SMS object

To create an SMS object, type in the following:

```
Dim oSMS as SMS_COMAPILib.SMS  
Set oSMS = New SMS_COMAPILib.SMS
```

The first line will define a placeholder with which to reference the SMS object. The second line will create the COM Object and assign it to the placeholder.

Step 2 - Send your first SMS

To send your first SMS, make sure that you have a valid API ID, username and password. If you do, the next step will be easy.

```
oSMS.SendSimpleSMS API_ID, "username", "password", "278...", "My first SMS"
```

If you replace API_ID with your own assigned API_ID, and enter your username, password and your cell phone number, you should receive the SMS shortly.

This is the simplest way to send SMS messages.

The source code can be found in the Examples\Ex1\ Example1_EarlyBinding.vbp project.

13.1.2 ASP/VISUAL BASIC – Late binding

VB late binding is very similar to ASP page programming, hence the examples should work in both environments.

Step 1 – Create SMS object

To create an object that will reference the SMS COM API, type the following:

```
Dim oSMS As Object  
Set oSMS = CreateObject("SMS_COMAPI.SMS")
```

The first line will define a placeholder to reference the SMS object with. The second line will create the COM Object and assign it to the placeholder.

Step 2 - Send your first SMS

To send your first SMS, make sure that you have a valid API ID, username and password. If you do, the next step will be easy.

```
oSMS.SendSimpleSMS API_ID, "username", "password", "278...", "My first SMS"
```

If you replace API_ID with your own assigned API_ID and enter your username and password, and enter your cell phone number you should receive the SMS shortly.

This is the simplest way to send SMS messages.

NOTE: Because you use late binding, the VB editing environment will not help you in the construction of the command line.

The source code can be found in the Examples\Ex1\Example1_LateBinding.vbp project.

13.2 Example 2 – login, send, and query balance

For the second example, we will do a bit more. We will not use the SendSimpleSMS call, but we will rather follow the recommendation and first login and then send an SMS.

13.2.1 VISUAL BASIC – Early Binding

NOTE: Make sure that you reference the 'SMS_COMAPI Type Library' for all early binding examples.

For all other examples we assume that you know that you need to create an SMS object placeholder (Step 1 in Example 1).

Step 1 – Login

To get to the login point, type the following:

```
Dim oSMS as SMS_COMAPILib.SMS  
Set oSMS = New SMS_COMAPILib.SMS
```

```
Dim SessionID as string  
SessionID = oSMS.Authenticate(API_ID, "Username", "Password")
```

Line 3 defines a container to accept the Session ID that is returned when the authentication has succeeded. Line 4 will call the actual authentication method. Pass the supplied API_ID, your username and password.

Your SMS COM Object should now enable you to log into your account on-line.

Step 2 – Send an SMS

To send an SMS, we use the SendMsg function. This method takes two parameters. The first is a Boolean value indicating that we want to use the session ID, and the next is the message text. One thing that is missing is the destination number. The example will show how to set this.

```
oSMS.SendMsg True, "Some text for the SMS" , "278..."
```

The SMS text is set via the smsTEXT property of the SMS COM Object. There are many other properties that you can play with.

Step 3 – How many SMS are left?

To check how many more SMS messages can be sent via your account, you can use the following piece of code.

```
MsgBox oSMS.QueryBalance(), vbInformation + vbOKOnly, "SMS Query Balance"
```

The source code can be found in the Examples\Ex2\Example2_EarlyBind.vbp project.

13.3 Send a Unicode message

This command sends the 'Hello World' string as hex encoded Unicode.

```
oSMS.smsUNICODE = True
```

oSMS.SendMsg True, oSMS.EncodeToUnicode("hello world") , "278..."

14. Appendix A: Error codes

| Definition | Comment | Value |
|-------------------------------|---|-------|
| SMS_ERR_NONE | No error | 0 |
| SMS_ERR_AUTH_FAILED | Authentication failed | 1 |
| SMS_ERR_INV_USER_PASSWD | Unknown username or password | 2 |
| SMS_ERR_SESSION_EXP | Session ID expired | 3 |
| SMS_ERR_ACC_FROZEN | Account frozen | 4 |
| SMS_ERR_MISS_SESSION_ID | Missing Session ID | 5 |
| SMS_ERR_TALK_TO_PROXY_FAILED | Failed to communicate with the proxy server | 95 |
| SMS_ERR_TALK_VIA_PROXY | Failed to communicate via the proxy, to the SMS server | 96 |
| SMS_ERR_BASE_URL_INV_URL | Invalid URL | 97 |
| SMS_ERR_BASE_URL_INV_HASH_KEY | Invalid Hash Key | 98 |
| SMS_ERR_COMMUNICATION | Communication error | 99 |
| SMS_ERR_INV_PARAMETER | Invalid or missing parameter | 101 |
| SMS_ERR_INV_UDH | Invalid UDH. (User Data Header) | 102 |
| SMS_ERR_UNKNOWN_APIMSGID | Unknown apiMsgId (API Message ID) | 103 |
| SMS_ERR_UNKNOWN_CLIMSGID | Unknown cliMsgId (Client Message ID) | 104 |
| SMS_ERR_INV_DEST_ADDR | Invalid Destination Address | 105 |
| SMS_ERR_INV_SRC_ADDR | Invalid Source Address | 106 |
| SMS_ERR_EMPTY_MESSAGE | Empty message | 107 |
| SMS_ERR_INV_APIID | Invalid or missing api_id | 108 |
| SMS_ERR_MISS_MSG_ID | This can be either a Client Message ID or API Message ID. For example when using the <i>del_msg</i> command. | 109 |
| SMS_ERR_EMAIL_MSG | Error with e-mail message | 110 |
| SMS_ERR_INVALID_PROTOCOL | Invalid Protocol | 111 |
| SMS_ERR_INVALID_MSG_TYPE | Invalid msg_type | 112 |
| SMS_ERR_MAX_PARTS_EXCEEDED | The text message component of the message is greater than the permitted 160 characters (70 Unicode characters). Select Concat=1,2,3-N to overcome this by splitting the message across multiple messages. | 113 |
| SMS_ERR_CANNOT_ROUTE_MSG | This implies that the gateway is not currently routing messages to this network prefix. Please email support@clickatell.com with the mobile number in question. | 114 |
| SMS_ERR_MESSAGE_EXPIRED | Message Expired | 115 |

| | | |
|-----------------------------------|--|-----|
| SMS_ERR_THROTTLE_LIMIT_EXCEEDED | Throttle Limit Exceeded | 117 |
| SMS_ERR_MSG_BLOCKED | Number blocked | 118 |
| SMS_ERR_MSG_CIDWS | cliMsgId (Client Message ID) contains space (s) | 120 |
| SMS_ERR_MSG_MOBILE_BLOCKED | Destination mobile number blocked | 121 |
| SMS_ERR_MSG_MOBILE_OPT_OUT | Destination mobile number opted out | 122 |
| SMS_ERR_MSG_INVALID_SENDER | Unregistered sender ID | 123 |
| SMS_ERR_MSG_DELIST | Number delisted | 128 |
| SMS_ERR_MSG_FEATURE | Feature not available | 129 |
| SMS_ERR_INV_BATCH_ID | Invalid batch ID | 201 |
| SMS_ERR_NO_BATCH_TEMPLATE | No batch template | 202 |
| SMS_ERR_INVALID_SENDER | A sender ID needs to be registered and validated before it can be successfully used in message sending | 123 |
| SMS_ERR_NO_CREDIT_LEFT | No credit left | 301 |
| SMS_ERR_MAX_ALLOWED_CREDIT | Max allowed credit | 302 |
| SMS_ERR_INVALID_TOKEN | Invalid Token | 606 |
| SMS_ERR_EXPIRED_TOKEN | Expired Token | 607 |
| SMS_ERR_ACCOUNT_ALREADY_ACTIVATED | Account Already Activated | 610 |
| SMS_ERR_ACTIVATION_FAILED | Account Activation Failed | 611 |
| SMS_ERR_INVALID_PARAMETERS | Invalid Parameters | 700 |
| SMS_ERR_DUPLICATE_USERNAME | Duplicate username | 701 |
| SMS_ERR_INTERNAL_RETRY | Internal error – please retry | 901 |

15. Appendix B: Message statuses (eSMSMsgStatus)

| Definition | Comment | Value |
|----------------------------|--|-------|
| SMS_MSG_UNKNOWN | The message ID is incorrect or reporting is delayed | 1 |
| SMS_MSG_QUEUED | The message could not be delivered and has been queued for attempted redelivery. | 2 |
| SMS_MSG_DELIVERED | Delivered to the upstream gateway or network (delivered to the recipient). | 3 |
| SMS_MSG_RECEIVED | Confirmation of receipt on the handset of the recipient. | 4 |
| SMS_MSG_ERR_WITH_MSG | There was an error with the message, probably caused by the content of the message itself. | 5 |
| SMS_MSG_CANCELLED | User cancelled message delivery | 6 |
| SMS_MSG_ERR_DELIVERING_MSG | An error occurred delivering the message to the handset. | 7 |
| SMS_MSG_OK | Message received by gateway. | 8 |

| | | |
|-------------------------|---|----|
| SMS_MSG_ROUTING_ERROR | The routing gateway or network has had an error routing the message. | 9 |
| SMS_MSG_MESSAGE_EXPIRED | Message has expired at the network due to the handset being off, or out of reach. | 10 |
| SMS_MSG_QUEUED_DELIVERY | Message has been queued at the gateway for delivery at a later time (delayed delivery). | 11 |

16. 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 short or long code. This service is not available in all regions; please contact an Account Manager for more information.
- **Premium rated message (MT):** A mobile user is charged a premium for a message that they receive from a particular short or long code. This service is not available in all regions; please contact an Account Manager for more information.
- **Revenue share:** This refers to the portion of the premium charge associated with a premium rated message, which is passed on to the content provider.
- **Content provider:** This is the Clickatell customer who is offering one or more services that are usually 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. This is also known as the message originator or source address. A Sender ID must be registered within your account and approved by us before it may be used.
- **Destination address:** The mobile number/MSISDN of the handset to which the message must be delivered. The number should be in international number format, e.g. country code + local mobile number, excluding the leading zero (0).
- **Source address:** See ‘Sender ID’ above.
- **Short code:** A short number which is common across all the operators for a specific region.
- **Subscriber:** The mobile network subscriber who owns the mobile number (MSISDN) which will send or receive SMSs, or be billed for premium rated services.
- **Upstream gateway:** A network operator, third party or our own short message service centre (SMSC).

17. Contact details

Phone: +27 21 910 7700
Fax: +27 21 910 7701
Website: www.clickatell.com
Help URL: <http://support.clickatell.com/index.php>
Support: support@clickatell.com
Info: info@clickatell.com
Sales: sales@clickatell.com

---0---