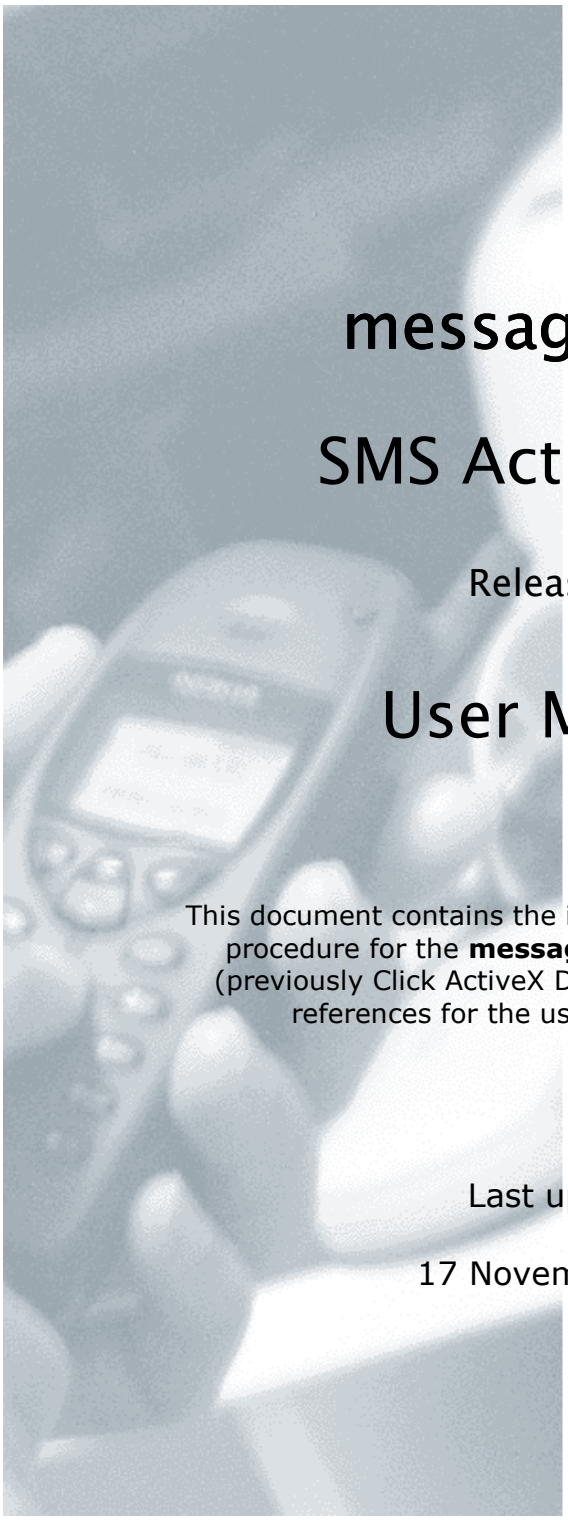

messagemedia

SMS ActiveX DLL

Release 2.8

User Manual

This document contains the installation requirements and procedure for the **message**media SMS ActiveX DLL (previously Click ActiveX DLL) and provides language references for the use of that component.

Last updated

17 November 2006

Table of Contents

Product Information

Product Name

messagemedia SMS ActiveX DLL Release 2.8
(previously Click ActiveX DLL)

Purpose

This document lays out the requirements and specifications for the integration of **message**media SMS ActiveX DLL 2-way messaging system and third party products.

Prepared by

Alex Barry
Software Development Manager
Message4U Pty Ltd trading as **message**media

Modifications

Created	14 August 2001
First update	27 August 2001
…	…
11th update	28 August 2002
12th update	19 September 2002
13th update	28 July 2003
14th Update	16 December 03
15th Update	17 November 06

Release

2.5.0.0	28 August 2002
2.5.0.1	9 September 2002
2.5.0.2	10 September 2002
2.5.0.3	26 September 2002
2.5.0.4	26 May 2003
2.6.X.X	Released with M4U SMS Client 2.5
2.7.0.0	28 July 2003
2.8.0.0	17 November 2006

Introduction

Messagemedia **SMS ActiveX DLL**

messagemedia SMS ActiveX DLL (previously Click ActiveX DLL) is created to help API users utilise 2-way SMS messaging through the **message**media SMS gateway. The ActiveX DLL is built around Microsoft's Component Object Model (COM) technology and is written in Visual Basic language to deliver greater flexibility and faster turn-around time.

The DLL is designed to be as developer-friendly as possible. It reduces the complexity of dealing with:

- Internet protocol
- **message**media SMS gateway protocol

It can be used in nearly any Windows programming environment that supports ActiveX COM objects.

Tested or known-to-work environments are:

- ✓ ASP
- ✓ Visual Basic
- ✓ Visual C++
- ✓ Visual FoxPro
- ✓ Delphi
- ✓ Microsoft Access (VBA)
- ✓ Lotus Notes

Untested environments are:

- ✓ Cold Fusion
- ✓ Borland C++

Modes

messagemedia SMS ActiveX DLL can be utilised in one of two modes:

1. Server mode
   It allows proxy/firewall authentication and secure transfer of messages. Recommended to be used in server type environment, such as: IIS/ASP, Lotus Notes, ColdFusion.
2. Client mode
   It allows installation in any Windows operating system. Recommended to be used when the DLL has to be installed at multiple environment end-users. In client mode, you can also enable NTLM firewall authentication.

The mode determines which DLLs should be installed, and also how the ServerMode property should be set. Please check the **Requirements** and **Installation** section for more information on how the two modes differ.

*Please note: If the **servermode** property is set to true, and API is executed on non server machine, the API will automatically set this property to suit the environment.*

Requirements

To be able to use the DLL, you would need to have all of the following:

1. For Server mode:
   - Windows NT4 (SP6) or Windows 2000 (SP3) operating system
   - Internet connection (it uses HTTP port 80)
   - Microsoft Visual Basic 6.0 Run-time (msvbvm60.dll)
   - Microsoft XML Parser 4 (msxml4.dll)
   - Microsoft Windows HTTP Services (winhttp5.dll)
2. For Client mode:
   - Windows 95 (with Winsock2 update), Windows 98, Me, NT, 2000, or XP
   - Internet connection (it uses HTTP port 80)
   - Microsoft Visual Basic 6.0 Run-time (msvbvm60.dll)

- Coalesys HTTP client (cshttpclient.dll) for default
- Microsoft XML Parser 4 (msxml4.dll) for NTLM challenged firewall/proxy server

Constraints

The **message**media SMS ActiveX DLL is designed to be subject to the following constraints:

1. API-generated MessageId
   API-generated MessageIds are limited to 1,000 unique ids per second. If the AddMessage or AddMessageEx is called more than 1,000 times in a second, the API will deliberately slow down the API to accommodate only 1,000 ids per second.
2. Reply checking interval
   The reply checking interval is required to be more than a minute. If the CheckReply or CheckReplyEx is called faster than the predefined time, it will then return an error code (1011 – CheckReplyDenied). Please note that Messagemedia reserves the right to increase or decrease the interval length. ***A real-time inbound SMS solution is available. Please contact us about a customised or tailored service.***

Using the COM API

Installation

Please follow the instructions applying to the mode you are going to use:

1. Server mode

   - Installing Microsoft Visual Basic 6.0 Run-time.
     Note: If you already use or have installed Microsoft Visual Basic 6.0 on the target machine, you can skip this step.
     - Download Visual Basic 6.0 Run-time from the Microsoft download site http://www.microsoft.com/downloads/release.asp?releaseid=28337
     - Run the self-extracting file to install the run-time files

   - Installing Microsoft XML Parser 4.X with Windows HTTP Service.
     Note: Windows HTTP Service (WinHTTP5) can only be installed on WinNT or Win2000 or WinXP machines. If you are using Win9X or Win ME machines, please see client mode installation.
     - Download XML Parser 4.0 SP2 from the Microsoft download site http://www.microsoft.com/downloads/details.aspx?familyid=3144b72b-b4f2-46da-b4b6-c5d7485f2b42&displaylang=en
     - Run the self-extracting file to install the Microsoft XML Parser

   - Installing the M4U SMS ActiveX DLL.
     Note: Please uninstall any previous M4U SMS ActiveX DLLs. However it can be installed side-by-side with the earlier version of Click ActiveX DLL.
     - Copy the m4usms.dll into your application directory or nominated directory and register it using the following command line syntax:

       regsvr32.exe "<drive>:\<path>\m4usms.dll"

Installation

2. Client mode

- Installing Microsoft Visual Basic 6.0 Run-time.
  Note: If you already use or have installed Microsoft Visual Basic 6.0 on the target machine, you can skip this step.
  - Download Visual Basic 6.0 Run-time from the Microsoft download site http://www.microsoft.com/downloads/release.asp?releaseid=28337
  - Run the self-extracting file to install the run-time files

- Installing the Coalesys HTTP Client DLL for default.
  Note: This DLL is included in the *Messagemedia* ***SMS COM API installation package.***
  - Copy the cshttpclient.dll into your application directory or nominated directory and register it using the following command line syntax:
    regsvr32.exe "<drive>:\<path>\cshttpclient.dll"

- Installing Microsoft XML Parser 4.X for NTLM challenged mode.
  - Download XML Parser 4.0 SP2 from the Microsoft download site http://www.microsoft.com/downloads/details.aspx?familyid=3144b72b-b4f2-46da-b4b6-c5d7485f2b42&displaylang=en
  - Run the self-extracting file to install the Microsoft XML Parser

- Installing the M4U SMS ActiveX DLL.
  Note: Please uninstall any previous M4U SMS ActiveX DLLs. However it can be installed side-by-side with the earlier version of Click ActiveX DLL.
  - Copy the m4usms.dll into your application directory or the nominated directory and register it using the following command line syntax:
    regsvr32.exe "<drive>:\<path>\m4usms.dll"

Examples

Examples in VB demonstrating the capabilities of this object are provided later in this document. VC++ or ASP examples are available upon request. Please contact us to request one.

Delphi and other programming language examples are in development. Please check with us in due course. Please contact us if you have a suggestion of another language example that should be made available.

Checklists

Before spending time on coding or testing the DLL, you may want to review the following checklist to ensure that the DLL can work in your particular environment.

- ✓ Check whether the programming language you use supports COM objects
- ✓ Check under which operating systems you are going to install the DLL
- ✓ Make sure you have chosen which mode you want to use based on the previous two criteria
- ✓ Follow the installation instructions as per mode you choose
- ✓ Make sure that the target machine is able to access the Internet from another program (such as: IE browser)
- ✓ Check out the examples provided in this document or contact **message**media for furher examples (see ***Examples*** section)

Examples in VB

Create an instance of the DLL (early binding)

Syntax

Dim iSMS As New M4USMS.SMS2

or

Dim iSMS As M4USMS.SMS2

Set iSMS = New M4USMS.SMS2

Description

This requires that you reference the DLL into your VB project/program. Click on menu Projects | References and select 'Click' to import the type library from the DLL.

Create an instance of the DLL (late binding)

Syntax

Dim iSMS As Object

Set iSMS = CreateObject("M4USMS.SMS2")

Description

This way you don't have to predefine the reference to the type library.

Set up particular service (only when required)

Syntax

iSMS.ServerFile = "sing-server.txt"

iSMS.MessageFile = "sing-message.txt"

Description

When required, you may be asked to change these properties by **message**media to allow Messagemedia to provide a customised service.

Set up a connection with the username and password

Syntax

Dim res As Long

iSMS.ServerMode = False

res = iSMS.SMSConnect("trial9999", "3kl0pm")

If res <> 0 Then 'Error found

Description

This will connect you to the **message**media SMS gateway with the given username and password in client mode (if you are using server mode, ignore the ***ServerMode*** property). Note: You need to call SMSConnect before sending and receiving messages and reconnect after you change password.

Examples in VB

Quickly send a message (without error handling)

Syntax

```
iSMS.AddMessageEx "0412345678", "test message", 0
iSMS.SendMessages
```

Description

This will send "test message" text to mobile number "0412345678" without delay.

Quickly send a message (with error handling)

Syntax

```
If iSMS.AddMessageEx("0412345678", "test message", 0) <> 0 Then
    `Error handling
    …
End if
If iSMS.SendMessages <> 0 Then
    `Error handling
    …
End if
```

Description

This will send "test message" text to mobile number "0412345678" without delay.

Batch send a message to n recipients (with AddMessage)

Syntax

```
iSMS.MessageDelay = 120
iSMS.MessageText = "test message"
iSMS.RecipientNumber = "0412345678"
iSMS.AddMessage
iSMS.RecipientNumber = "0411234567"
iSMS.AddMessage
If iSMS.SendMessages <> 0 Then
    `Error handling
    …
End if
```

Description

This will send "test message" text to mobile number "0412345678" and "0411234567" with 120 seconds (2 minutes) delay.

Examples in VB

Batch send a message to n recipients (with AddMessageEx)

Syntax

```
If iSMS.AddMessageEx("0412345678", "test message", 120) <> 0 Then
    `Error handling
    …
End if
If iSMS.AddMessageEx("0411234567") Then
    `Error handling
    …
End if
If iSMS.SendMessages <> 0 Then
    `Error handling
    …
End if
```

Description

This will send "test message" text to mobile number "0412345678" and "0411234567" with 120 seconds (2 minutes) delay.

Tag the outgoing message

Syntax

```
iSMS.IDMode = idUserDefined
iSMS.MessageId = 1
iSMS.MessageDelay = 0
iSMS.MessageText = "test message"
iSMS.RecipientNumber = "0412345678"
If iSMS.AddMessage <> 0 Then
    `Error handling
    …
End if
If iSMS.SendMessages <> 0 Then
    `Error handling
    …
End if
```

Description

This will send "test message" text to mobile number "0412345678" without delay with MessageId set to 1. You can then match any reply with the outgoing message based on its MessageId.

Examples in VB

Checking for reply messages (with CheckReply)

Syntax

```
Dim I As Long
If iSMS.CheckReply <> 0 Then
        `Error handling
        …
Else
    For I = 1 To iSMS.Replies.Count
        ...
        ... = iSMS.Replies(I).MessageId
        ... = iSMS.Replies(I).PhoneNumber
        ... = iSMS.Replies(I).MessageText
        ...
    Next I
End If
```

Description

This will check the server for any reply messages and download it to the API. The replies are accessible through the Replies() property only. The Replies() property returns a Reply object which consists of MessageId, PhoneNum and MessageText properties.

Checking for reply messages (with CheckReplyEx)

Syntax

```
iSMS.ReplyFile = "replies.dat"
If iSMS.CheckReplyEx <> 0 Then
        `Error handling
        …
Else
    ...
    <codes that read the tab delimited file: replies.dat>
    ...
End If
```

Description

This will check the server for any reply messages and download it to the API. The replies are accessible through both the Replies() property and the ReplyFile specified. The ReplyFile file is written as a tab delimited file. Note that the ReplyFile file permission should be set to "RW" for the API user logon.

Language Reference

Architecture

The ActiveX DLL is written in Visual Basic using the standard VB and XML library available through Microsoft. It allows as little as four lines of code to send an SMS message.

There is one main object class exposed, SMS2, with the ClassId: M4USMS.SMS2.

M4USMS.SMS2

Syntax

Dim iSMS As M4USMS.SMS2

Set iSMS = New M4USMS.SMS2

Description

The SMS2 object is used for sending and receiving SMS messages by setting its properties and calling its methods.

The SMS2 object allows you to send a message, check for replies, and change your password. The properties and methods for this object are listed below.

Properties

SMS2.SMSServer

Syntax

Msgbox iSMS.SMSServer

Return Value

String

Description

Returns the last used domain name or IP address of the SMS gateway. This property is no longer used and should not be retrieved except for checking the last used SMS gateway.

SMS2.Username

SMS2.Password

Syntax

iSMS.Username = "trial9999"

iSMS.Password = "3kl0pm"

Return Value

String

String

Description

Returns or sets the username and password used to connect to the SMS gateway. This username and password is supplied by **message**media when the user signed up for a product trial or commercial use.

These properties are no longer used. You should set these properties when calling the SMSConnect method. (See also: ***SMS2.SMSConnect*** method)

SMS2.ServerMode

Syntax

iSMS.ServerMode = False

Allowed Value

Boolean (True/False)

Description

Returns or sets whether to use server or client mode. This property has to be set properly based on the operating system and DLL installed (please check the *Installation* section). The default is True (for server mode).

Properties

SMS2.NTLMChallenged

Syntax

iSMS.NTLMChallenged = True

Allowed Value

Boolean (True/False)

Description

Returns or sets whether to use XML4 or CSHTTPClient in client mode. This property has to be set properly based on the operating system and DLL installed (please check the *Installation* section). The default is False (use CSHTTPClient).

SMS2.IDMode

Syntax

iSMS.IDMode = idUserDefined

Allowed Value

1 = idDefault (API generated)

2 = idUserDefined (user defined)

Description

Indicates whether the message Id is generated by the API or is user defined. The default is API generated message Id (idDefault). (See also: ***SMS2.MessageId*** property)

SMS2.ProxyName

SMS2.ProxyPort

Syntax

iSMS.ProxyName = "proxy.yourisp.com"

iSMS.ProxyPort = "8080"

Return Value

String

Description

Returns or sets the proxy name and proxy port when a proxy server or firewall is used. When proxy name and proxy port are not specified, a direct connection to the Internet is expected. Please consult your system administrator on whether a proxy server or firewall is being used.

Properties

SMS2.ProxyUsername

SMS2.ProxyPassword

Syntax

iSMS.ProxyUsername = "abc"

iSMS.ProxyPassword = "123"

Return Value

String

Description

Returns or sets the proxy/firewall username and password when a proxy server or firewall is used. Please consult your system administrator on whether a proxy server or firewall is being used and an authentication is required. This property is ignored when using client mode. (See also: ***SMS2.ServerMode*** property)

SMS2.SecureTransfer

Syntax

iSMS.SecureTransfer = True

Allowed Value

Boolean (True/False)

Description

Returns or sets whether to securely transfer the messages. When enabled, it uses SSL port (443) rather than HTTP port (80) and all message transfer is encrypted with 128-bit key encryption. The default is False (for normal/unencrypted HTTP transfer).

SMS2.IntlPrefix

Syntax

iSMS.IntlPrefix = "+65"

Return Value

String

Description

Returns or sets the international prefix number to be used when sending messages without a leading prefix. The string should always start with a "+" symbol. The default is Australian prefix (+61).

Properties

SMS2.AutoSplit

Syntax

iSMS.AutoSplit = True

Allowed Value

Boolean (True/False)

Description

Returns or sets whether the object automatically splits the message if it is longer than 160 characters. The components of the message will be sent at 30 second intervals and will be tagged as linked. The default is False (do not split message). Note that at most a message can only be split into up to nine (9) messages.

SMS2.RecipientNumber

Syntax

iSMS.RecipientNumber = "0412345678"

Return Value

String

Description

Returns or sets the mobile phone number of the next message to be added into the batch.

SMS2.MessageText

Syntax

iSMS.MessageText = "Test message…"

Return Value

String

Description

Returns or sets the message text of the next message to be added into the batch.

SMS2.MessageDelay

Syntax

iSMS.MessageDelay = 24 * 60 *60

Return Value

Long

Description

Returns or sets the delay (in seconds) for the next message to be added into the batch.

Properties

SMS2.MessageID

Syntax

iSMS.MessageID = 9999

Return Value

Long

Description

Returns or sets the message Id for the next message to be added into the batch. This property is ignored when IDMode is 1 (Default). (See also: ***SMS2.IDMode*** property)

SMS2.ValidityPeriod

Syntax

iSMS.ValidityPeriod = vpDefault

Return Value

Long

Description

Returns or sets the validity period for the next message to be added into the batch.

Validity period is defined through the following criteria:

0 (vpMinimum) – 5 minutes validity

11 (vpOneHour) – 1 hour validity

167 (vpOneDay) – 1 day validity

168 (vpDefault) – 2 days validity

173 (vpOneWeek) – 1 week validity

255 (vpMaximum) – 63 weeks validity

SMS2.DeliveryReport

Syntax

iSMS.DeliveryReport = True

Return Value

Boolean (True/False)

Description

Returns or sets whether a delivery report is required for the next message to be added into the batch.

Properties

SMS2.Messages(long index)

Syntax

Msgbox iSMS.Messages(1).MessageText

Return Value

Message object, consists of:

string MessageId

string PhoneNumber

string MessageText

long Delay

byte ValidityPeriod

boolean DeliveryReport

Description

Returns the message details from the messages batch. This property is read-only.

SMS2.ReplyFile

Syntax

iSMS.ReplyFile = "C:\TEMP\REPLY.DAT"

Return Value

String

Description

Full path and filename of the tab delimited file to store the replies. Only used by CheckReplyEx method. (See also: ***SMS2.CheckReplyEx*** method)

SMS2.ReplyFound

Syntax

Msgbox iSMS.ReplyFound

Return Value

Boolean (True/False)

Description

Indicates whether the last check found any reply. (See also: ***SMS2.CheckReply*** and ***SMS2.CheckReplyEx*** methods)

Properties

SMS2.Replies(long index)

Syntax

Msgbox iSMS.Replies(1).MessageText

Return Value

Reply object, consists of:

byte ReportStatus (0-3)

string MessageId

string PhoneNumber

string MessageText

Description

Returns the reply details from the replies batch after downloading replies and/or reports from the server. This property is read-only.

The ReportStatus defines whether the reply is a reply message (0) or a report (1-3). A report status is:

0 – Not a report, but a reply SMS message

1 – Pending report (acknowledged by SMSC)

2 – Delivered report (received by the mobile phone)

3 – Failed report (message expired)

SMS2.CreditsLeft

Syntax

Msgbox iSMS.CreditsLeft

Return Value

Long

Description

Returns the number of credits left in the account. Note that this is for prepaid account only. It returns (–1) if the account is a non-prepaid account (standard account) or returns (–2) if the credit information is not available.

Properties

SMS2.LastResponseTime

Syntax

Msgbox iSMS.LastResponseTime

Return Value

Long

Description

Returns the elapsed time for the last sending or receiving.

SMS2.LastStatus

Syntax

Msgbox iSMS.LastStatus

Return Value

String

Description

Returns the last sending or receiving status, or the last error description.

SMS2.MaxSend

Syntax

Msgbox iSMS.MaxSend

Allowed Value

Byte (1-100)

Description

Sets the number of messages to be transferred at any one time. The default is 100. Note that changing this to a lower number will reduce the risk of failure when using a slow Internet connection.

SMS2.AutoCheck

Syntax

iSMS.AutoCheck = False

Allowed Value

Boolean (True/False)

Description

Returns or sets whether the object automatically checks the Internet connection whenever there is a failure. The default is True (auto-check Internet connection).

Properties

SMS2.TimeOut

Syntax

iSMS.TimeOut = 30

Return Value

Long

Description

Returns or sets how long before the attempt to connect to server times out. The default is 60 (seconds – 1 minute). Note that this property is ignored if running client mode with NTLMChallenged property is set to True.

SMS2.EmailAddress

SMS2.AppVersion

Syntax

iSMS.EmailAddress = "myname@company.com"

iSMS.AppVersion = "yourappvX.X"

Return Value

String

Description

Sets the email address and app version of the API user. These properties are optional. Setting these property will provide additional identification when required.

SMS2.ServerFile

SMS2.MessageFile

Syntax

iSMS.ServerFile = "<if supplied by M4U>"

iSMS.MessageFile = "<if supplied by M4U>"

Return Value

String

Description

Returns or sets the server and message settings file as required. These properties are optional and should only be changed when you have a customised product from **message**media.

Methods

All the following methods return 0 (zero) when they are successful and an error code when they fail. For error codes explanation, see next section: Error codes.

SMS2.SMSConnect

Syntax

iSMS.SMSConnect (string UserName, string Password)

iSMS.SMSConnect "trial9999", "3kl0pm"

Parameters

UserName – as supplied by **message**media

Password – as supplied by **message**media

Possible error code(s)

cLoginError (1001)

cLibraryNotFound (2000)

cInetError (2001)

cInvalidResponse (2002)

cServerDown (2004)

cNoInternet (2005)

cWrongUserPass (3000)

cUnrecognizedError (9999)

Description

Connect to the next available SMS gateway using the username and password. This method needs to be called before sending and receiving messages.

SMS2.DownloadMessage

Syntax

iSMS.DownloadMessage

Parameters

None

Possible error code(s)

cLibraryNotFound (2000)

cUnrecognizedError (9999)

Description

Downloads the server-broadcasted message. The message is returned in HTML form in a file called message.htm under the same directory where the DLL resides. Note: Server-broadcasted messages are used by **message**media to advise users of any server-based maintenace and/or upgrades.

Methods

SMS2.AddMessage

Syntax

iSMS.AddMessage

Parameters

None

Possible error code(s)

cLoginError (1001)

cPhoneNumError (1002)

cEmptyMessage (1003)

cMessageTooLong (1004)

cDuplicateNumber (1005)

cInvalidMessageId (1008)

cBatchFull (1010)

Description

Add the current message (as described earlier in RecipientNumber, MessageText, and MessageDelay properties) into the batch.

SMS2.AddMessageEx

Syntax

iSMS.AddMessageEx (string Number, string MessageText, long Delay, byte Valid, boolean Report)

Parameters

Name – depreciated (not need to specify)

Number – recipient's mobile phone number

MessageText – the message text itself

Delay – the message delay in seconds

Valid – validity period of the message

Report – set whether delivery report is required

Possible error code(s)

Same as AddMessage (see above)

Description

Adds the current message into the batch. Note that all of the above parameters are optional. When they are not specified, they will be taken from corresponding properties.

Methods

SMS2.SendMessages

Syntax

iSMS.SendMessages

Parameters

None

Possible error code(s)

cSMSServerNotFound (1000)

cLoginError (1001)

cEmptyMessage (1003)

cLibraryNotFound (2000)

cInetError (2001)

cInvalidResponse (2002)

cServerDown (2004)

cNoInternet (2005)

cWrongUserPass (3000)

cInsufficientDailyCredit (3001)

cInsufficientCredit (3002)

cUnrecognizedError (9999)

Description

Sends all the messages already in the batch. This method will also clear all the messages from the batch when successfully sent. Note that if the transfer fails, all unsent messages remain in the batch.

SMS2.ClearMessages

Syntax

iSMS.ClearMessages

Parameters

None

Possible error code(s)

None

Description

Deletes all the messages in the batch.

Methods

SMS2.ChangePassword

Syntax

iSMS.ChangePassword (string NewPassword)

Parameters

NewPassword – the new password (more than 6 characters)

Possible error code(s)

cSMSServerNotFound (1000)

cLoginError (1001)

cNoNewPassword (1006)

cLibraryNotFound (2000)

cInetError (2001)

cInvalidResponse (2002)

cServerDown (2004)

cNoInternet (2005)

cWrongUserPass (3000)

cUnrecognizedError (9999)

Description

Replaces the user's existing password with a new password.

SMS2.IsLibraryPresent

Syntax

Msgbox iSMS.IsLibraryPresent

Parameters

None

Possible error code(s)

None

Description

Indicates whether the required library file is present.

SMS2.IsInternetConnected

Syntax

Msgbox iSMS.IsInternetConnected

Parameters

None

Possible error code(s)

None

Description

Indicates whether the object can access the Internet or not. Use this to determine whether an extra proxy server or firewall setting is required.

Methods

SMS2.CheckReply

Syntax

iSMS.CheckReply

Parameters

None

Possible error code(s)

cSMSServerNotFound (1000)

cLoginError (1001)

cCheckReplyDenied (1011)

cLibraryNotFound (2000)

cInetError (2001)

cInvalidResponse (2002)

cServerDown (2004)

cNoInternet (2005)

cWrongUserPass (3000)

cUnrecognizedError (9999)

Description

Checks for any reply messages. Reply messages are accessible through the Replies property. (See also: ***SMS2.Replies*** and ***SMS2.ReplyFound*** properties)

SMS2.CheckReplyEx

Syntax

iSMS.CheckReplyEx

Parameters

None

Possible error code(s)

Same as CheckReply (see above)

cNoReplyFile (1009)

Description

Checks for any reply messages. Reply messages are accessible through both the Replies property and the RepyFile nominated. (See also: ***SMS2.Replies***, ***SMS2.ReplyFound***, and ***SMS2.ReplyFile*** properties)

Troubleshooting

Error codes

One of the following error codes may be returned as a return code by the above methods.

cSMSServerNotFound

Value

1000

Description

Not connected to SMS gateway. Please connect using SMSConnect method.

cLoginError

Value

1001

Description

Empty username and/or password is given.

cPhoneNumError

Value

1002

Description

Invalid mobile phone number supplied.

cEmptyMessage

Value

1003

Description

Cannot send an empty message.

cMessageTooLong

Value

1004

Description

Message is longer than 160 characters.

cDuplicateNumber

Value

1005

Description

Duplicate phone number is found in the batch.

Error codes

cNoNewPassword

Value

1006

Description

No new password is provided. Please specify a new password of 6 or more characters.

cNoSenderName

Value

1007

Description

No longer used.

cInvalidMessageId

Value

1008

Description

Invalid Message Id. Please specify a number between 1 to 999,999,999.

cNoReplyFile

Value

1009

Description

No reply filename supplied. Please specify a valid filename in which to put replies.

cBatchFull

Value

1010

Description

You have reached the maximum limit of messages in the batch before sending. Please send all the messages before proceeding.

Error codes

cCheckReplyDenied

Value

1011

Description

You have reached the minimum time limit of checking for replies. Please try again later.

cLibraryNotFound

Value

2000

Description

Required library file is either not found or not properly registered.

cInetError

Value

2001

Description

Failed to send through to the Internet. Please check your Internet connection.

cInvalidResponse

Value

2002

Description

Invalid response received from HTTP transfer. Please check your firewall/proxy settings.

cSendTimeOut

Value

2003

Description

Timed out while trying to send. Please check your Internet connection and firewall/proxy settings.

Error codes

cServerDown

Value

2004

Description

All **message**media SMS gateways are unreachable. Please contact **message**media.

cNoInternet

Value

2005

Description

There is no Internet connection at present. Please check your Internet connection.

cWrongUserPass

Value

3000

Description

Wrong user name and/or password given.

cInsufficientDailyCredit

Value

3001

Description

Insufficient daily credit to send the message. Please contact **message**media to increase your daily credit limit.

cInsufficientCredit

Value

3002

Description

Insufficient credit to send the message. Please contact **message**media to top up your credit.

Contact Us

Feedback is always welcome. If anything in the document is unclear, or is not working as described, please do not hesitate to contact us:

AUSTRALIA

Tel: 1800 155 228
Email: support@message-media.com.au

NZ

Tel: 0800 68 69 64
Email: support@message-media.co.nz

USA

Tel: 1 866 884 8611
Email: support@message-media.com

UK

Tel: 0808 234 4874
Email: support@message-media.com