ActiveSMS V5.1

User Manual

IntelliSoftware Ltd

Contents

1 Introduction

ActiveSMS supports 2-way text messaging using various SMS transport methods. Messages can be sent and received in two ways:

- COM Component Interface - an easy to use COM interface allows messages to be sent and received using any language that supports COM (e.g. ASP, JScript, VBScript, C++, VB).
- 2-way Email to SMS gateway – Send and receive SMS text message as emails.

1.1 Email->SMS and SMS->Email Gateway

ActiveSMS can integrate with your current mail server to provide 2-way SMS messaging from your standard email clients (e.g. Microsoft Outlook).

SmartRouting

SmartRouting is technology that allows 2-way conversations between a mobile user and an email user within your organisation. ActiveSMS will attempt to route SMS replies back to the email user that initiated the conversation.

1.1.1 Email->SMS Support

ActiveSMS can be configured to forward Emails to SMS. The body of the email will be sent in the SMS text message and can optional include 'From' address and Subject field.

There are two ways to connect ActiveSMS (for Email->SMS) to your mail system:

- POP3 Mailbox Polling
- SMTP Server

POP3 Mailbox Polling

ActiveSMS can be configured to periodically check a POP3 mailbox/account for outgoing messages. By default, the destination number will be extracted from the subject field, separate numbers by commas or semi-colons (e.g. Subject: 447710123456, 447710654321). The destination phone number must be supplied in international format unless the international dialling rules have been set (see section 4.1.4)

SMTP Server

ActiveSMS can be configured to run as an SMTP Server, you will need to make sure your mail server is configured to forward Emails to the ActiveSMS Server. By default, the destination number will be extracted from the To address (e.g. To: 447710123456@mygateway.com). The destination phone number must be supplied in international format unless the international dialling rules have been set (see section 4.1.4)

1.1.2 SMS->Email Support

ActiveSMS can be configured to forward received SMS messages to email. By default, the subject will contain the sender's phone number and the email body will contain the SMS text message.

See section 4.1.4 for more information.

1.2 Architecture

ActiveSMS consist of three parts:

- ActiveSMS Server – Sends and receives SMS messages using the available SMS Transports. The ActiveSMS Server runs as a NT Service.
- Message Queues – Incoming and outgoing SMS Messages are stored here whilst waiting to be processed.
- Client Components – COM components used by your application to send and receive SMS text messages.

A typical installation will install all three parts. However, you can use ActiveSMS Server remotely from another PC (on the same LAN) by installing the Client Only Components.

2 Technical Support

For technical support on this product please send an email to support@intellisoftware.co.uk. Please provide information about your operating system and the devices you are using. We will make every effort to reply to your support questions within 24hrs.

3 Installation

3.1 Software Install

When you install ActiveSMS you choose from the following installation types.

Installation	Description
ActiveSMS Basic	Installs the ActiveSMS Server Basic version and the Client COM Components. The Basic version supports outgoing messages only on 1 transport.
ActiveSMS Professional	Installs the ActiveSMS Server Professional version and the Client COM Components. The Professional version supports incoming and outgoing messages on 1 transport.
ActiveSMS Advanced	Installs the ActiveSMS Server Advance version and the Client COM Components. The Advance version supports incoming and outgoing messages on 8 transports.
Client Only Components	Installs Client COM Components that allow connection to a remote ActiveSMS Server running on another computer. Does not require a Server Licence.

NOTE: Licence only works for the installation type you have purchased. The Client Only installation does not require a licence.

3.2 Licensing

Installing ActiveSMS starts the 30-day trial period. When the trail period expires the ActiveSMS Server will not send or receive any SMS messages.

To continue using the ActiveSMS you will need to purchase a licence from http://www.intellisoftware.co.uk. You must purchase a licence for the same product type (Basic/Profession/Advanced) as you have installed.

NOTE: A separate licence must be purchased for each server installation. The Client Only installation does not require a licence.

3.3 SMS Transports

To send and/or receive SMS messages you need to consider which transports you are going to use. The following transports are supported:

- Mobile Handset or GSM Modem (Send and Receive)
- HTTP Protocol, Internet->SMS Gateway (Sending only)
- TAP Protocol (Sending only)

Mobile Handset or GSM Modem

SMS messages are sent and received using a GSM device(s) directly connected to the PC. ActiveSMS is compatible with most mobile phones (with a built in modem) and most GSM Modems. Here is a list of devices with which ActiveSMS is known to be compatible:

Maestro GSM/GPRS
Wavecom WMOD2, Fastrack M1206B
Falcom A2D-3
Nokia 22, Nokia Premicell
Multi-Tech Systems GSM Modem
Siemens M20 Terminal, Siemens TC35, Siemens TC65
FirstFone Radio Card
XACOM Audiotel GSM Modem
DigiCom Pocket GMS modem
Psitek Fusion 1000

Nokia 6210/6250/6310/6310i/7110/7160/7190/8310 mobile phone with DLR-3 cable
Nokia 5110/5130/6110/6150 mobile phone with Nokia Data Suite 2.0/3.0 (NT only)
Ericsson R300/R320/R380/R520/T39/T60/T62/T65/T68/T68i with DRS-11 cable
Siemens M55 Handset
Motorola Timeport 260
Samsung S300 GSM

Any AT-Compatible device (e.g. GSM Modems, other makes of mobile phones)

HTTP Protocol

SMS Messages can be sent via the IntelliSoftware Internet->SMS Gateway. Connection to the SMS Gateway is made using the HTTP Protocol over an Internet connection.

TAP Protocol

SMS messages can be sent using a standard Modem and telephone line (i.e. PSTN). ActiveSMS will dial-up a TAP service on demand to send SMS messages. TAP services are available in must countries and do not require registration, you normally only pay for the cost of the phone call.

For a list of TAP numbers please see:
http://www.intellisoftware.co.uk/redir?page=tapnumbers.

3.4 System Requirements

Intel Pentium I 100Mhz or equivalent
Windows NT4, Windows 2000 and Windows XP
10Mbytes of free Disk Space

4 Configuring ActiveSMS

To configure ActiveSMS open Start Menu > Program Files > ActiveSMS > ActiveSMS Console.

4.1 Transports Page

See section 3.3 for a description of SMS Transports.

This page allows configuration of the transports used to send and/or receive SMS messages. This page also shows the real-time status of each transport.

If you have purchase the Advanced version you will be able to configure 8 transports, otherwise only 1 transport will be available.

Each transport can be individual configured to be either:

- Mobile Handset
- GSM Modem
- HTTP Protocol (Internet->SMS Gateway)
- TAP Protocol (Sending only)

4.1.1 Mobile Handset or GSM Modem Transport

Each GSM Device can be individual configured to support text messaging in one or both directions (Basic version supports outgoing messages only).

Select the ‘Handset’ transport if you are using a mobile phone, or select ‘GSM Modem’ transport if you are using a GSM Modem. To set-up the transport click the ‘Configure’ button, you will be prompted for the following information:

Comms Port	Selected the comms port that the GSM device is connected to.
Comms Settings	You may need to change the comms settings if the default settings are not compatible with your GSM Device.
Concat. Msgs. Max.	Maximum number of concatenation messages that will be sent for long messages. Set to 1 to disable message concatenation.
PIN Number	(Optional) If you are using a GSM Modem then ActiveSMS can automatically unlock the Modem using this PIN number.
SMSC Number	(Optional) SMS Service Centre Number. Set this to override the default SMCS Number set in the SIM card. You will need to sent this item if you SIM card is not

	configured with an SMSC Number.
Log Errors To File	When checked, ActiveSMS will log comms activity to a log file. This feature should be normally disabled to prevent disk space usage.
Log Filename	Path and filename of the log file (e.g. 'C:\GSMLog.txt')

4.1.2 HTTP Protocol Transport

Select the 'HTTP' transport and then click 'Configure'. You will be prompted for the following information:

Primary Server	This is the Internet Address of the Internet->SMS Gateway. Default is 'www.intellisoftware.co.uk'
Backup Server	This is the backup Internet->SMS Gateway Default is 'www.intellisoftware2.co.uk'
Username	This is the Username for your account. You will need to register on the www.intellisoftware.co.uk website before you can send messages.
Password	This is the Password for your account.
Sender's ID	The receiver of the SMS message will see this as the sender of the SMS message. The sender ID can contain numbers and letters. TIP: If you are using a GSM Modem for the incoming path, you can specify the phone number of the GSM Modem to allow the user to reply to the SMS messages.
Use HTTP Proxy	If you require a proxy server to access the internet then select this checkbox.
Proxy Server	Address of your proxy server

4.1.3 TAP Protocol Transport

Select the 'TAP' transport and then click 'Configure'. You will be prompted for the following information:

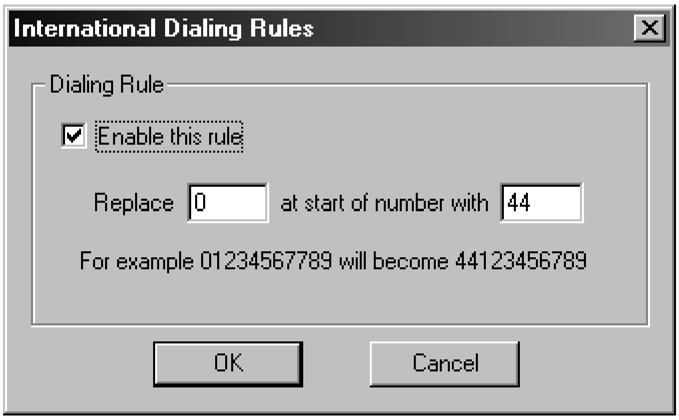
Modem	ActiveSMS will use this Modem to dial out to the TAP service.
Dial-up Number	This is the telephone number of the TAP service. For a list of TAP numbers please see: http://www.intellisoftware.co.uk/redir?page=tapnumbers.

Min Baudrate	Minimum baudrate for the connection
Max Baudrate	Maximum baudrate for the connection
Msgs Per Connection	Maximum number of messages that should be sent during one dial-up connection. Most TAP services will limit the number of messages that can be sent in one connection.
Retry Delay After Error	Time period (in secs) that ActiveSMS should wait before retrying after a connection error has occurred.
Log Errors To File	When checked ActiveSMS will log TAP activity to a log file. This feature should be normally disabled to prevent disk space usage.
Log Filename	Path and filename of the log file (e.g. ‘C:\TAPLog.txt’)

4.1.4 International Dialling Rules

By default, ActiveSMS will expect all phone numbers to start with the international dialling code. For example, a UK number should start with 44 (e.g. 447710123456).

Click on the ‘International Dialling Rules…’ button to configure ActiveSMS to accept national format number (e.g. 07710123456).



☑ Enable this rule

Replace 0 at start of number with 44

For example 01234567789 will become 44123456789

Select ‘Enable this rule’ to enable dialling rule. Change ‘44’ with the international dialling code for your location. In the example above, the leading ‘0’ will be replaced with ‘44’, e.g. ActiveSMS will convert ‘07710448818’ to ‘447710123456’ before sending a message.

4.2 Email Page

You can ignore these settings if you are not intending to use the Email gateway features (i.e. if you are using only the COM components).

The two checkboxes at the top of this page control the Email gateway functionality:

Enable Email -> SMS	Select this checkbox for ActiveSMS to forward Emails via SMS.
Enable SMS -> Email	Select this checkbox for ActiveSMS to forward incoming

	SMS messages to Email.

4.3 Email -> ActiveSMS Settings

These settings effect how ActiveSMS will receive Emails.

There are two ways ActiveSMS can receive emails:

- POP3 Mailbox Polling
- SMTP Server

POP3 Mailbox Polling

ActiveSMS can be configured to periodically check a POP3 mailbox/account for emails.

POP3 Server	Address of your POP3 server (e.g. ‘mycompany.com’)
POP3 Username	Username for the POP3 mailbox (e.g. ‘smsout’)
POP3 Password	Password for the POP3 mailbox

You can use the ‘Test POP3’ button to check these settings.

SMTP Server

ActiveSMS can be configured to run as an SMTP Server.

SMTP Port	ActiveSMS will listen on this port for incoming SMTP emails. (Default: 25)

4.4 ActiveSMS -> Email

These settings effect how ActiveSMS will send Emails.

SMTP Server	Address of your SMTP server (e.g. ‘mycompany.com’)
From Address	This is the From address that will appear in emails from the gateway. You can include a * to have ActiveSMS insert the senders number in the email address. (e.g. ‘*@mygateway.com’ will become ‘447710123456@mygateway.com)
To Address	Default email address to which incoming SMS messages will be forwarded (can be override by the Routing Options, see below) (e.g. ‘smsin@mycompany.com’) Multiple email addresses can be specified, separated by semicolon (e.g. ‘smsin1@test.com; smsin2@test.com’).

You can use the ‘Test SMTP’ button to check these settings.

4.4.1 Routing Options

Click the ‘Advanced Options…’ button on the Email Page and then select the ‘Routing’ tab.

Outgoing (Email to SMS) message routing options:

Extract destination phone number from the e-mail’s Subject field. (Default)	The destination number is extracted from the e-mail’s subject field (all non-numeric characters will be ignored). Multiple numbers must be separated by a comma or semi-colon (e.g. 447710123456, 447710654321). The destination phone number must be in international format unless the international dialling rules have been set (see section 4.1.4)
Extract destination phone number from To addresses.	The destination number is extracted from the e-mail’s To addresses (e.g. 447710123456@mygateway.com). The destination phone number must be in international format unless the international dialling rules have been set (see section 4.1.4)
Always forward to fixed phone number.	All text messages will be sent to the specified Fixed phone number.

Incoming (SMS to Email) message routing options:

SmartRouting (Enabled by default)	SmartRouting is the technology that allows 2-way conversations between a mobile user and an email user within your organisation. ActiveSMS will attempt to route SMS replies back to the email user that initiated the conversation.
To email addresses found in the incoming text message	ActiveSMS will examine the text message content for email addresses. Message will be forwarded to all email addresses found in the text message.

If the email address can not be determined, ActiveSMS will use ‘To Address’ on the Email page.

4.4.2 Access Control (Email->SMS)

Click the ‘Advanced Options…’ button on the Email Page and then select the ‘Access Control’ tab.

Here you are able to restrict the email users that can use the ActiveSMS Email gateway to send SMS messages.

Access restrictions are based on the user’s email address. To restrict access, select the ‘Only email users listed below’ option and add allowed users to the list. You can add a full email addresses or email domains.

4.4.3 SMS Formats (Email->SMS)

Click the ‘SMS Format…’ button (on the Email Page) to configure the content of SMS messages sent by the Email to SMS gateway.

Here you are able edit the template used by ActiveSMS to generate the SMS Message content, for example, to add a signature to the SMS message. The template can contain free text and pre-defined Tag fields. Tag fields are placed in the template to indicate to ActiveSMS were certain details should be placed.

The following Tag fields are supported:

<EmailFrom>	From email address of the source email
<EmailSubject>	Subject field of the source email
<EmailBody>	Message body of the source email

The maximum field length can also be controlled, for example '<EmailSubject:40>' will restrict the length of the Subject inserted into the SMS message to 40 characters maximum.

4.4.4 Email Formats (SMS->Email and Email Reports)

Click the 'Email Formats…' button (on the Email Page) to configure emails sent by the email gateway.

Emails are generated when SMS messages are received (SMS->Email). Email messages are also generated to report the progress of outgoing messages (Email->SMS).

The following email types can be configured:-

SMS Message Received (SMS->Email)	Generated when an SMS message is received and forwarded as an email.
SMS Message Sent Report	Generated when an SMS message has been successfully sent through the Email->SMS Gateway.
SMS Message Send Failure Report	Generated when an a failure occurred sending an SMS message through the Email->SMS Gateway
SMS Message Delivery Report	Generated when an SMS message (sent through the Email->SMS Gateway) has been successfully delivered.
SMS Message Non-Delivery Report	Generated when an SMS message (sent through the Email->SMS Gateway) has failed delivered.
Email Submission Error Report	Generated when an email could not be processed (e.g. because destination number is missing).
Email Rejected (Due to Access Control)	Generated when a request was rejected due to access control (see section 4.4.2)

To edit an email, first select the email type you wish to edit and then click the 'Edit…' button.

Uncheck the 'Generate this message' checkbox if you do not want the email to be generated.

You can configure the email's Subject and Body with free text of your choice. You can also add the following pre-defined fields:-

<PhoneNumber>	Outgoing SMS: The phone number message was sent on. Incoming SMS: The phone number of the message's sender.
<TextMessage>	Content of the SMS message
<Transport>	Transport(1-8) used to received/sent message.
<ErrorMessage>	Error message (Applicable to SMS Message Send Failure Report only).

4.4.5 Advanced POP3 Settings (Email->SMS)

Click the ‘Advanced Options…’ button on the Email Page and then select the ‘POP3’ tab.

NOTE: The ‘POP3’ tab will only be available when ‘POP3 Polling’ protocol is selected

Here you are able to configure the POP3 Polling Period (Default is 1 minute). ActiveSMS connects to your POP3 mailbox periodically to check for new emails.

4.4.6 Advanced SMTP Settings (Email->SMS)

Click the ‘Advanced Options…’ button on the Email Page and then select the ‘SMTP Server’ tab.

NOTE: The ‘SMTP Server’ tab will only be available when ‘SMTP Server’ protocol is selected

Here you are able to control the IP Addresses that the ActiveSMS SMTP Server will attempt to bind to. By default, the ActiveSMS SMTP Server will attempt to bind to all available IP Addresses.

Restricting the IP Address bindings for the ActiveSMS SMTP Server will allow ActiveSMS to co-exist with another SMTP Server on the same machine.

4.5 Logging Page

NOTE: This feature is not available in the ActiveSMS Basic version.

ActiveSMS can log details of all incoming and outgoing messages to either a text file (CSV) or database table.

The following items can be logged:

- Message ID
- Text of the message
- Message Type (1-Text, 2-Unicode, 3-Data)
- Phone Number
- Transport Number (1-8)
- Timestamp
- Message Sent Timestamp
- Status

To set-up logging, check the relevant checkbox on the Logging Page. Click the Setup button to change logged items and destination (e.g. text file or database).

By default, a logging database (C:\ Program Files\ IntelliSoftware\ ActiveSMS\ Database\ ActiveSMSLog.mdb) is automatically set-up when ActiveSMS is installed.

You can also configure ActiveSMS to log to a Microsoft SQL Server or MySQL database if you prefer by performing the following steps:

- Create a new SQL Server or MySQL database
- Create a new System DSN and set the default database.
- In the SMS Logging settings enter the DSN and Username/Password (if required)
- Click Test Connection
- Click Create Tables

4.6 Queuing Page

NOTE: To access this feature click on the ‘Advanced’ button (bottom-left of console window).

ActiveSMS supports both incoming and outgoing message queuing. The queued messages are stored in either a Database or Memory. By default, when ActiveSMS is installed, an Access database in configured.

NOTE: If Memory storage is used, then queued messages will be lost when the ActiveSMS service is re-started.

After making changes to the queuing settings you will need to Stop and re-Start the SMS Engine using the ActiveSMS Console.

You can reconfigure ActiveSMS to use a Microsoft SQL Server or MySQL database if you prefer by performing the following steps:

- Create a new SQL Server or MySQL database
- Create a new System DSN and set the default database.
- Open the ActiveSMS Console and select the Config Page
- Enter the DSN and Username/Password (if required)
- Click Test Connection
- Click Create Tables

4.7 Scripts Page

NOTE: To access this feature click on the ‘Advanced’ button (bottom-left of console window).

This page allows you to configure a script or webpage to run whenever a message is received. See ActiveSMS Scripts.

4.8 Advanced Page

NOTE: Only available in the Professional and Advanced versions.
NOTE: To access this feature click on the ‘Advanced’ button (bottom-left of console window).

This page allows you to configure ActiveSMS to use delivery reports.

4.8.1 Delivery Reports

When ‘Use delivery reports’ is checked, ActiveSMS will request a delivery report for each message sent.

4.8.2 Notifications

You can configure ActiveSMS to notify your application when events occur, such as receiving a delivery report or sending a message. Notifications are received in your application using the ActiveSMS.GetNotification method or the ActiveSMSListener.OnNotificationReceived event.

5 ActiveSMS System Health Monitoring

NOTE: This feature is only available in the ActiveSMS Advanced version.
NOTE: To access this feature click on the ‘Advanced’ button (bottom-left of console window).

ActiveSMS is able to monitor its internal systems status and alert you of any problems by email. To configure system health monitoring please click the ‘System Monitoring…’ button

The following items can be monitored:

ActiveSMS Server: Overall Status	Generated when there is a general failure of the ActiveSMS Server
ActiveSMS Server: SMS Transport Status	Generated when an SMS Transport has failed
ActiveSMS Server: Send SMS Message Failure	Generated when ActiveSMS Server is unable to send an SMS message
ActiveSMS Email Gateway: Email->SMS Processor Status	Generated when there is a general failure of the Email to SMS Gateway
ActiveSMS Email Gateway: SMS->Email Processor Status	Generated when there is a general failure of the SMS to Email Gateway
ActiveSMS Email Gateway: Send Email Failure	Generated when the ActiveSMS Email Gateway is unable to send an Email message

6 Getting Started

6.1 Testing the installation

The installation can be tested using the ActiveSMS Test, open Start Menu > Program Files > ActiveSMS > ActiveSMS Test.

To send a message, enter the phone number in international format, enter the text message and click Send. Received messages will appear in the Incoming messages list.

To send messages without entering the international dialling code, see section 4.1.4.

6.2 Samples

Sample Visual Basic Project, Visual C++ Project and Active Server Pages are installed with ActiveSMS. These projects directories are accessible from the Start menu: Start Menu > Program Files > ActiveSMS > Samples.

For more information about using the COM components see Appendix A – COM Component Reference.

7 SendSMS Command Line Utility

A command line utility is installed with ActiveSMS for sending messages. The command line utility can be used from batch files, schedules job, performance monitor alarms etc.

SendSMS <PhoneNumbers> "<Text Message>" [<Timeout in secs>] [<Transport No>]

PhoneNumbers in international format (e.g. '447710123456', comma separate multiple numbers).

TextMessage to be sent in quotes (use '\n' to generate a new line).

Timeout in milliseconds, default is 15 seconds.

Transport No to be used to send the message (1-8), default is any transport.

To send messages without entering the international dialling code, see section 4.1.4.

8 ActiveSMS Scripts

8.1 Incoming SMS Script

You can configure ActiveSMS to run a script each time a message is received. For example, you could query a database and send some information back to the sender. Scripts can be a VBScript (.vbs), JScript (.js), Webpage or Executable (.exe).

An example script file is installed with ActiveSMS, open Start Menu > Program Files > ActiveSMS > Samples > VBS -> VBSSample.vbs

To configure a script file to run:

- Open the ActiveSMS Console
- Select the Scripts Page
- Check Run Script When Message Received
- Click **…** and select script file to be run
- Click OK

8.1.1 Calling a web page

On message reception ActiveSMS can make a HTTP request to a web page, e.g. 'http://localhost/smsreceive.asp'

ActiveSMS will call the web page with the following parameters:

smsreceive.asp?PhoneNumber=447710448818&Message=The+Message+Text&Handset=1

PhoneNumber	Phone number of the message's sender.
Message	Text of the SMS message.
Handset	Transport (1-8) on which the message was received

Generating an SMS response message

Your processing web page can optionally pass back an SMS response message to ActiveSMS within the HTTP response. The HTTP response should be format as below:

```
<ActiveSMS>
  <Actions>
    <SendMessage>
      <To>44771012345</To>
      <Text>Hello</Text>
      <Handset>1</Handset>
    </SendMessage>
  <Actions>
<ActiveSMS>
```

The XML above, when supplied in the HTTP response, tells ActiveSMS to send an SMS message to '44771012345', with text 'Hello' and using Transport 1.

8.2 SMS Notification Script

NOTE: Option only available when Notifications are selected on the Advanced Page.

You can configure ActiveSMS to run a script when an event occurs, such receiving a delivery report or send a message. Scripts can be VBScript (.vbs) or JScript (.js), you could even specify an executable (.exe).

An example script file is installed with ActiveSMS, open Start Menu > Program Files > ActiveSMS > Samples > VBS -> NotificationSample.vbs

To configure a script file to run:

- Select the Advance Page
- Select Use Delivery Reports (if required)
- Select the type of notifications you are interested in
- Click OK
- Open the ActiveSMS Console
- Select the Scripts Page
- Check Run Script When Event Occurs
- Click **…** and select script file to be run
- Click OK

8.2.1 Calling a web page

ActiveSMS can direct notifications to a web page, e.g. 'http://localhost/smsnotify.asp'

ActiveSMS will call the web page with the following parameters:

smsnotify.asp?OriginalId=111&Type=1&Message=The+Message+Text&PhoneNumber=447710448818&Handset=1&ErrorCode=55

OriginalId	MessageId as returned by SendMessage
Type	Notification type, possible values: 0 - Delivery report from network 1 - Delivery failure report from network 2 - Message sent 3 - Send Failure
Message	Text of the SMS message.
PhoneNumber	Destination Phone number.
Handset	Transport (1-8) on which the message was sent.
ErrorCode	Error code returned from the cellular network (0=no error).

8.3 Notes about using scripts

Do not use MsgBox or alert within your scripts, this will cause your scripts to hang forever. This is because script runs as a different User, so prompts will not be visible.

If you have a virus checker with a script-blocking we recommend that this feature is disabled as this may cause you scripts to run improperly. For example, script blocking will probably prevent your scripts from writing a file to disk.

9 SMS Delivery Reports

NOTE: Only available in the Professional and Advanced versions.

Delivery reports are enabled in the ActiveSMS Console (see Advanced Page). Once enabled, ActiveSMS will request a delivery report for every message sent.

Reception of a delivery report can be detected in two ways:

1) By checking the message status (using the ActiveSMS.GetSendStatus method). Message status will change from jsSentNotConfirmed to jsSent when a delivery report is been received.

2) By using Notifications, notifications are enabled in the ActiveSMS Console (see Advanced Page). Notifications are received in your application using the ActiveSMS.GetNotification method or the ActiveSMSListener.OnNotificationReceived event.

10 Advanced Topics

10.1 Email->SMS Gateway - Forcing Transport

By default, the email gateway will send an SMS message on a transport selected at random.

The email gateway can be force to use a particular transport by using the 'Transport:[1-8]' syntax in the email subject. If a phone number is also specified in the subject, then it must be prefixed by 'To:', for example:-

Subject: **To:447710123456 Transport:2**

This will send a message to 447710123456 using transport 2.

11 Appendix A – COM Component Reference

COM Objects

ActiveSMS Object	Used to send and receive SMS messages.
SMSMessage Object	SMS text message.
SMSNotification Object	Notification of message delivery (i.e. Delivery report) or message sent.
ActiveSMSListener Object	Design-time control for VB and MSVC. Provides event based SMS message reception on VB Forms and MSVC Dialog boxes.

ActiveSMS Object

Properties:
ActiveHandset

Methods:
SendMessage
SendUnicodeMessage
SendUnicodeMessageHex
SendDataMessage
GetSendStatus
GetMessage
GetNotification

ActiveHandset Property

Direction	Read and Write

Valid Values	hsAll 1-8
Default	hsAll (=0)
Type	Integer

This property selects the transport that will be used in subsequent calls to SendMessage and GetMessage.

If ActiveHandset is hsAll then SendMessage will use any available transport, otherwise SendMessage will only use the transport specified.

If ActiveHandset is hsAll then GetMessage will return a message received on any transport, otherwise GetMessage will only return a message received on the transport specified.

SendMessage Method

SendMessage (PhoneNumber As String, Message As String, WaitTimeout As Integer) As Interger

Parameters:

Name	Type	Dir	Description
PhoneNumber	String	IN	Destination phone number for the message, this must be in international format (e.g. '447710123456'). To send messages without entering the international dialling code, see section 4.1.4.
Message	String	IN	Text message to be sent.
WaitTimeout	Integer	IN	Time in mS function will wait for message to be sent. wtNoWait (=0) – Function returns immediately without waiting. wtInfiniteWait (=-1) – Function does not return until the message is sent.

Return Value:
Returns a MessageId that can be used to obtain status information for the message (see GetSendStatus for details)

Remarks:
This function inserts a message in the outgoing message queue and optional waits for the message to be sent.

Example:
Id = SendMessage ("447710448818", "Message Text", 15000)

SendUnicodeMessage Method

SendUnicodeMessage (PhoneNumber As String, Message As String, WaitTimeout As Integer) As Interger

Parameters:

Name	Type	Dir	Description
PhoneNumber	String	IN	Destination phone number for the message, this must be in international format (e.g. '447710123456'). To send messages without entering the international dialling code, see section 4.1.4.

Message	String	IN	Unicode text message to be sent.
WaitTimeout	Integer	IN	Time in mS function will wait for message to be sent. wtNoWait (=0) – Function returns immediately without waiting. wtInfiniteWait (=-1) – Function does not return until the message is sent.

Return Value:
Returns a MessageId that can be used to obtain status information for the message (see GetSendStatus for details)

Remarks:
This function inserts a Unicode message in the outgoing message queue and optional waits for the message to be sent.

Example:
Id = SendUnicodeMessage (“447710448818”, “Message Text”, 15000)

SendUnicodeMessageHex Method

SendUnicodeMessageHex (PhoneNumber As String, MessageHex As String, WaitTimeout As Integer) As Interger

Parameters:

Name	Type	Dir	Description
PhoneNumber	String	IN	Destination phone number for the message, this must be in international format (e.g. ‘447710123456’). To send messages without entering the international dialling code, see section 4.1.4.
MessageHex	String	IN	Unicode data in AsciiHex (4 characters per Unicode char)
WaitTimeout	Integer	IN	Time in mS function will wait for message to be sent. wtNoWait (=0) – Function returns immediately without waiting. wtInfiniteWait (=-1) – Function does not return until the message is sent.

Return Value:
Returns a MessageId that can be used to obtain status information for the message (see GetSendStatus for details)

Remarks:
This function inserts a Unicode message in the outgoing message queue and optional waits for the message to be sent.

Example:
Id = SendUnicodeMessageHex (“447710448818”, “03A903A80398”, 15000)

SendDataMessage Method

SendDataMessage (PhoneNumber As String, DataCodingScheme As Interger, UserData As String, UserDataHeader As String, WaitTimeout As Integer) As Interger

Parameters:

Name	Type	Dir	Description
PhoneNumber	String	IN	Destination phone number for the message, this must be in international format (e.g. ‘447710123456’). To send messages without entering the international dialling code, see section 4.1.4.
DataCodingScheme	Interger	IN	PDU DCS Field (See GSM 03.40 specification)
UserData	String	IN	PDU User Data in AsciiHex (2 chars per byte)
UserDataHeader	String	IN	PDU User Data Header in AsciiHex (2 chars per byte)
WaitTimeout	Integer	IN	Time in mS function will wait for message to be sent. wtNoWait (=0) – Function returns immediately without waiting. wtInfiniteWait (=-1) – Function does not return until the message is sent.

Return Value:
Returns a MessageId that can be used to obtain status information for the message (see GetSendStatus for details)

Remarks:
This function inserts a Data message in the outgoing message queue and optional waits for the message to be sent.

Example (Send Ringtone):
Id = SendDataMessage (“447710448818”, 245,
“024A3A5585E195B198040042D9049741A69761781B6176156174288B525D85E0A26C24C
49A617628930BB125E055856049865885D200”, “06050415810000”, 15000)

Example (Send Operator Logo):
Id = SendDataMessage (“447710448818”, 245,
“32F40100480E0100000000000000000000000000000000000000000000000000000000000
1247803000000E0080120CC06400000040080120CC06C00000479E7124F0EFFADCF64448
892479B6DAC6CD4448F9241DB6DACECF44488124CDB6CD96CC44488924CDB6CD96C
DE446712478E66D9EC6000000000000000000000000000000000000”, “06050415820000”,
15000)

GetSendStatus Method

GetSendMessage (MessageId As Integer) As EnumJobStatus

Parameters:

Name	Type	Dir	Description
MessageId	Integer	IN	MessageId as returned by SendMessage.

Return Value:
Status of the message sent with SendMessage, possible values:

jsPending	Message is waiting in the outgoing message queue
jsProcessing	Message is in the progress of being sent
jsSent	Message has been sent successfully
jsDeliveryFailed	Delivery Reported received with error status
jsErrNumberInvalid	Message not sent - Supplied number was invalid
jsErrNetworkFailure	Message not sent - Cellular Network reported error
jsErrPhoneCommsError	Message not sent - Error communicating with device

jsErrSystemFailure	Message not sent - General system failure

The following states occur when Delivery Reports are enabled

jsSentNotConfirmed	The message has been sent. Delivery report is still pending.
jsSent	Delivery Report received with success status
jsDeliveryFailed	Delivery Report received with error status

Remarks:
Used to track messages sent with SendMessage.

GetMessage Method

GetMessage (WaitTimeout As Integer) As SMSMessage

Parameters

Name	Type	Dir	Description
WaitTimeout	Integer	IN	Time in mS for function to wait for a message. wtNoWait (=0) – Function returns immediately without waiting. wtInfiniteWait (=-1) – Function does not return until a message is received.

Return Value:
If a message is received then an SMSMessage object is returned. Otherwise if function times out then NULL object (Nothing in Visual Basic) is returned.

Remarks:
Attempts to read a message from the incoming message queue. Function returns when message received or the timeout has expired.

GetNotification Method

NOTE: Only available in Professional and Advance versions.

GetNotification (WaitTimeout As Integer) As SMSNotification

Parameters

Name	Type	Dir	Description
WaitTimeout	Integer	IN	Time in mS for function to wait for a notification. wtNoWait (=0) – Function returns immediately without waiting. wtInfiniteWait (=-1) – Function does not return until a notification is received.

Return Value:
If a notification is received then an SMSNotification object is returned. Otherwise if function times out then NULL object (Nothing in Visual Basic) is returned.

Remarks:
Waits for a notification of message delivery (i.e. Delivery Report) or message sent. Function returns when notification received or the timeout has expired.

The operation of this method is dependant on ActiveSMS settings, see Advanced Page.

SMSMessage Object

Properties:
PhoneNumber
Message
UserData
UserDataHeader
DataCodingScheme
Handset

PhoneNumber Property

Direction	Read and Write
Default	N/A
Type	String

Phone number message was sent from.

Message Property

Direction	Read and Write
Default	N/A
Type	String

Body of the text message.

MessageType Property

Direction	Read and Write
Default	N/A
Type	Integer

Message type, possible values: -

mtTextMessage	- 7bit Text Message
mtUnicodeMessage	- 16bit Unicode Message
mtDataMessage	- PDU Data Message

TimeStamp Property

Direction	Read and Write
Default	N/A
Type	String

Time message was sent/received, format ‘dd/mm/yy hh:mm:ss’.

UserData Property

Direction	Read and Write
Default	N/A
Type	String

PDU User Data in AsciiHex (2 chars. per byte).

UserDataHeader Property

Direction	Read and Write
Default	N/A
Type	String

PDU User Data Header in AsciiHex (2 chars. per byte).

DataCodingScheme Property

Direction	Read and Write
Default	N/A
Type	Interger

PDU DCS Field (See GSM 03.40 specification).

Handset Property

Direction	Read and Write
Valid Values	hsAll 1-8
Default	hsAll (=0)
Type	Integer

Transport message was received on.

SMSNotification Object

Properties:
MessageId
NotificationType
ErrorCode
Message

MessageId Property

Direction	Read and Write
Default	N/A
Type	Integer

MessageId as returned by SendMessage.

NotificationType Property

Direction	Read and Write

Default	N/A
Type	Integer

Notification type, possible values: -

ntDeliveryReport	- Delivery report from network
ntDeliveryFailure	- Delivery failure report from network
ntMessageSent	- Message sent on transport.
ntSendFailure	- Message could not be sent.

ErrrorCode Property

Direction	Read and Write
Default	N/A
Type	Integer

Error code returned from the cellular network (0=no error).

Message Property

Direction	Read and Write
Default	N/A
Type	SMSMessage Object

The message that caused this notification.

ActiveSMSListener Object

Properties:
Enabled
ActiveHandset

Events:
OnMessageReceived
OnNotificationReceived

Enabled Property

Direction	Read and Write
Valid Values	True / False
Default	False
Type	Boolean

Enables or disables message reception.

ActiveHandset Property

Direction	Read and Write
Valid Values	hsAll 1-8
Default	hsAll (=0)
Type	Integer

If ActiveHandset is hsAll then ActiveSMSListener will receive messages on any transport, otherwise ActiveSMSListener will only receive messages on the transport specified.

OnMessageReceived Event

OnMessageReceived (Message As Object)

Parameters:

Name	Type	Dir	Description
Message	SMSMessage Object	IN	Incoming message.

Remarks:
Fired when a message is received on the incoming message queue.

NOTE: The message is removed from the incoming message queue when this event is fired. Another instance of the ActiveSMSListener is guaranteed not read the same message.

OnNotificationReceived Event

NOTE: Only available in Professional and Advance versions.

OnNotificationReceived (Notification As Object)

Parameters:

Name	Type	Dir	Description
Notification	SMSNotification Object	IN	Notification object containing the details of the notification.

Remarks:
Fired when a delivery report has been received or a message has been sent. The operation of this event is dependant on ActiveSMS settings, see Advanced Page.

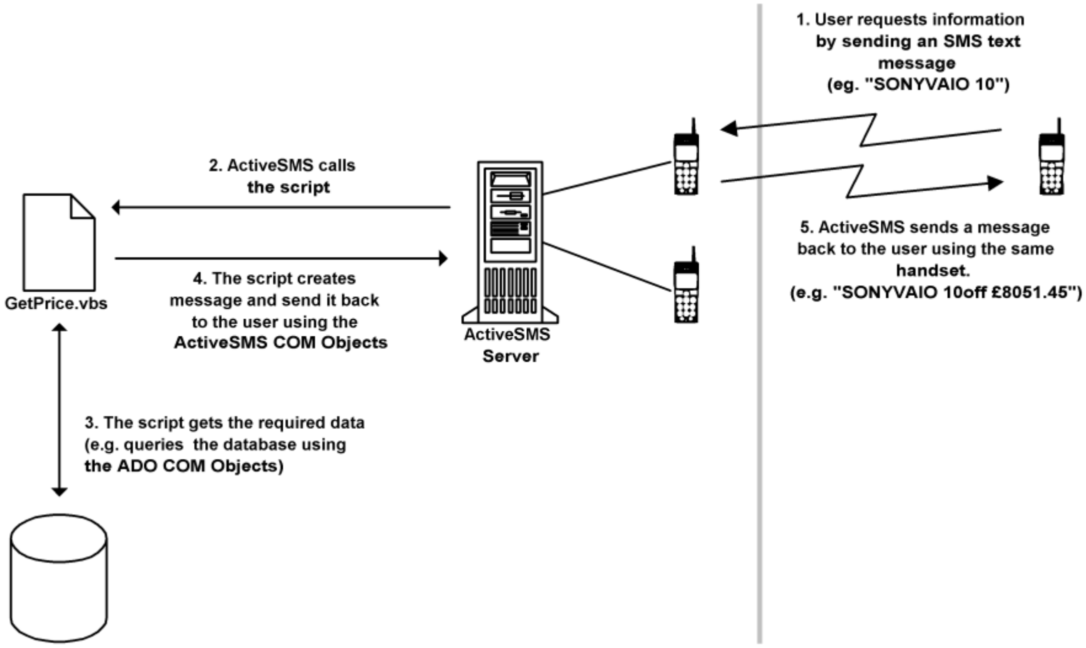
NOTE: Another instance of the ActiveSMSListener is guaranteed not read the same notification.

Appendix B – Info-on-demand Example

ActiveSMS Professional and Advanced versions are ideally suited for creating info-on-demand services.

In a typical info-on-demand application the user will send an SMS text message containing a request. The system will process the request and return an SMS text message containing the requested information.

With support for scripting languages, creating an info-on-demand service does not request extensive programming knowledge. The example below shows a field sales engineer requesting the latest product pricing information, which is queried from the corporate database.


1. User requests information by sending an SMS text message (eg. "SONYVAIO 10")
2. ActiveSMS calls the script
GetPrice.vbs
4. The script creates message and send it back to the user using the ActiveSMS COM Objects
ActiveSMS Server
5. ActiveSMS sends a message back to the user using the same handset. (e.g. "SONYVAIO 10off £8051.45")
3. The script gets the required data (e.g. queries the database using the ADO COM Objects)

For more information see ActiveSMS Scripts