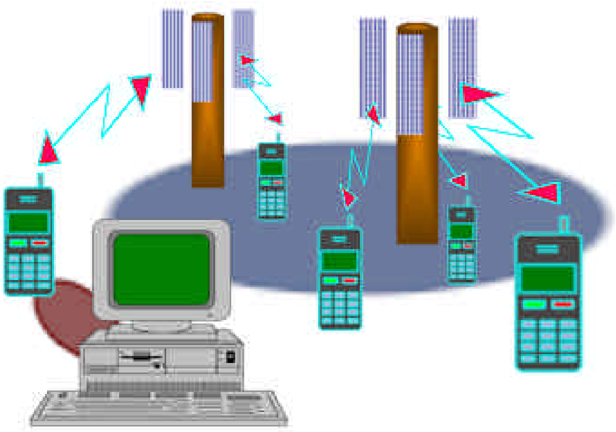
GPA Technology

SMS Gateway



Version

8.0

Installation, Users & Developers Manual

SMS Gateway by GPA Technology

Installation, Users & Developers Manual

PO Box 741
Frankston Victoria 3199
Australia

e-mail   support@winsms.com
www   http://www.winsms.com

Every effort has been made to ensure that the information in this manual is accurate.

The software described in this document is protected by copyright, and may not be copied on any medium except as specifically authorised in the licence or non-disclosure agreement

All company or product names mentioned are trademarks or registered trademarks of their respective owners.

Table Of Contents

Introduction

SMS Gateway is a unique Short Message Service (SMS) messaging tool. It facilitates complete two way messaging capabilities over GSM networks for Windows 95 / NT based applications through the use of DDE, OLE Automation, and the Command Line Interface (CLI).

SMS Gateway connects a PC to a GSM handset (mobile terminal), via a PCMCIA “Cellular Data Card”, or cable connected to a COM port. Through this connection all messages currently stored in the mobile terminal can be retrieved, new messages sent, and new incoming messages be sent straight through to the PC.

SMS Gateway comes with a powerful interactive two way messaging utility named SMS Messenger, that demonstrates the capabilities of SMS Gateway. This utility may be used for ad-hoc messaging alongside a corporate messaging or database access application.

For developers, SMS Gateway leverages the common Dynamic Data Exchange (DDE) and Object Linking and Embedding Automation (OLE Automation) specifications to allow the transmission and reception of SMS messages directly from any application supporting these standards. Applications supporting DDE & OLE Automation include; Delphi, Paradox, FoxPro, WordPerfect, Excel, Word, Access, Visual Basic, and many more.

SMS Gateway also supports the sending of messages to individual or groups of destinations directly from the Windows Command Line. This facility is useful for easy integration with Network Management utilities that are able issue Windows commands under certain prescribed circumstances, e.g. to send a message to a technician when a certain equipment alarm is raised.

Fully commented source code for the above mentioned SMS Messenger application, written in Delphi 3.0, is also included in the package, so you can easily adapt its functions for your particular needs.

Sample code for sending messages, receiving messages and controlling SMS Gateway using Excel 97 is also included with the installation files.

SMS Gateway is suitable for dispatch, field access to databases, telemetry, vehicle tracking, and many more such applications.

Before You Begin

Thankyou for choosing SMS Gateway by GPA Technology. Before you install your new software, please set aside some time to read this manual. By following the recommendations in this manual you will be ensured of making the most of your investment.

Registration

Please take a moment to fill out and return the registration card that came with your package.

As a registered user we will keep you informed of new products and updates, as they become available. You will be also eligible for discounts on software upgrades. Technical support (via E-mail & Fax) is only available to registered users.

Note: If you purchased your software directly from GPA Technology, you will automatically be registered, there is no need to fill in and return a registration card.

System Requirements

System Requirements	
Processor	486 or better
Operating System	Microsoft Windows 95 or NT
Memory	8MB Minimum
Display	VGA or Better
Mouse	Recommended
Mobile Terminal Interface	ETSI 07.05 “Block” or “PDU” Mode compliant (Check our web page at **www.winsms.com** for details of tested hardware equipment)

About This Manual

This manual is broken into the following main sections:

Basic Concepts	An overview of the SMS Gateway system, explaining fundamental terms and concepts.
Getting started	A step by step guide to installing, and configuring SMS Gateway and its partner application SMS Messenger.
Developers Reference	Comprehensive details for interworking SMS Gateway with other applications through DDE and OLE Automation.

Document Conventions

- **Hot Keys**

  Underlined letters that appear in bold type refer to "**H**ot Keys". Hot Keys are short-cuts to commonly used menu commands and functions.

- **Menu Commands**

  When you see **`File | Exit`** it means to choose the File Menu at the top of the screen, then select Exit from the list of commands displayed.

Basic Concepts

What is GSM

GSM stands for **G**lobal **S**ystem for **M**obiles. This is a word-wide standard for digital cellular telephony, or as most people know them Digital Mobile Telephones. GSM was created by the Europeans, and originally meant "Groupé Special Mobile", but this didn't translate well, so the now common more globally appealing name was adopted. GSM is a published standard by ETSI, and has now enjoys widespread implementation in Europe, Asia, and increasingly America.

Who are ETSI

ETSI stands for **E**uropean **T**elecommunications **S**tandards **I**nstitute. This is the body that created the specification for GSM, they publish their specifications to manufacturers and software developers who can then build equipment and software that can work together properly.

The ETSI standard for messaging on GSM is "Digital cellular telecommunications system (Phase 2); Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS) (GSM 07.05)". This is the protocol that SMS Gateway uses to talk between the PC and GSM Handset.

What is SMS

SMS stands for **S**hort **M**essage **S**ervice. It is the ability to send and receive "Short Messages" to and from GSM handsets, or as they are named by ETSI '**M**obile **T**erminals' (MT). A "Short Message" may contain up to 160 characters.

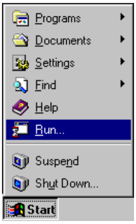
What is SMS Gateway

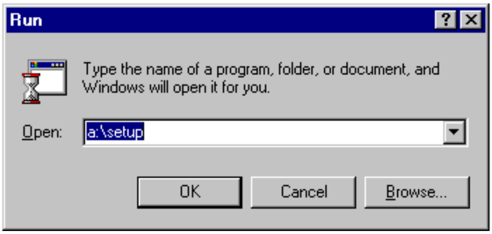
SMS Gateway presents developers with a simple and powerful programming interface to the complex ETSI 07.05 protocols, enabling faster time to market for new SMS messaging based products, and fast in-house development of corporate messaging solutions.

Installation

Software Installation

The SMS Gateway installation process is managed by the popular InstallShield™ program. To start the install program, select Run from the Windows “Start” Menu, and then enter the path to the SMS Gateway setup program (usually A:\setup).

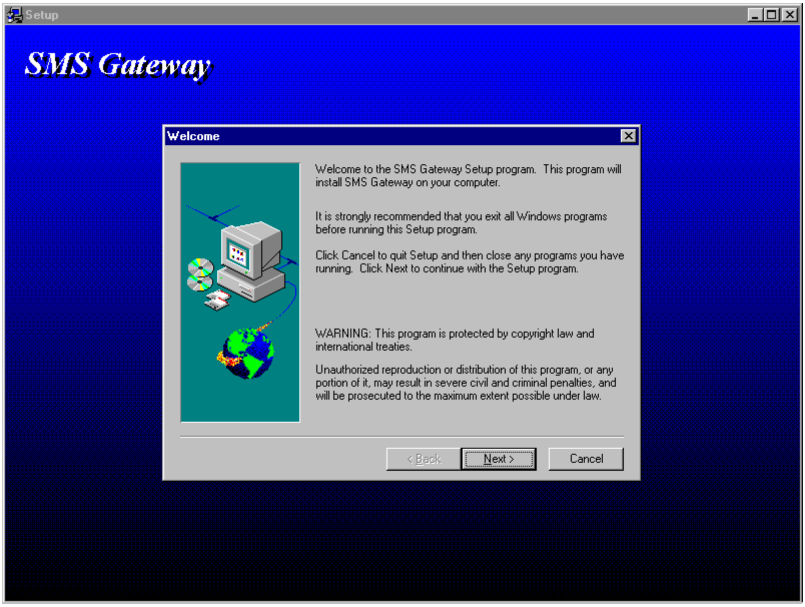




Type the name of a program, folder, or document, and Windows will open it for you.

Open: a:\setup

A short time after you enter the path to the setup program, the following screen will be displayed.



Welcome to the SMS Gateway Setup program. This program will install SMS Gateway on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Follow the on screen instructions to complete the installation.

Getting Started

Setting Up Your Hardware

- **Connecting the hardware**

  As mentioned in the Introduction, SMS Gateway uses a connection to a GSM Mobile Terminal (MT) to send and receive messages over the GSM network.

  The Mobile Terminal connects to the PC using a PCMCIA card, or a serial cable. These interfaces are generally supplied by the Mobile Terminal manufacturer, and come with their own installation instructions. For information on compatible hardware, please refer to our World Wide Web site at **www.winsms.com** .

  Once your Mobile Terminal is connected to a PC, you can test it is working by using Windows HyperTerminal to access the COM port to which it is connected and typing “AT” followed by enter, the interface device should respond with “OK”, if it does not something is wrong, and will need to be rectified before SMS Gateway will work.

- **Positioning**

  SMS Gateway relies on the Mobile Terminal having reliable coverage from the GSM Network to which it is subscribed. If a strong and stable signal strength is not received, it is recommended you move the equipment to a location where a high quality signal is available.

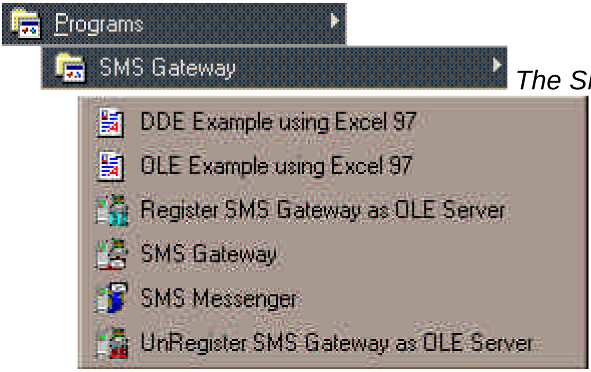
  If moving the PC and Mobile Terminal is not practical, the installation of an external antenna (as used when installed in vehicles), may achieve good results.

- **Power**

  If SMS Gateway will be used as full time messaging gateway for your business, particularly if unattended, it is strongly recommended an Uninterruptible Power Supply (UPS) be used to prevent possible message loss during power outages.

First Time Configuration of SMS Gateway

Once SMS Gateway has been installed on your hard-disk, there are a number of steps that need to be followed to setup the system correctly. You will now find three new entries on your Start Menu, these are:


Programs
SMS Gateway
DDE Example using Excel 97
OLE Example using Excel 97
Register SMS Gateway as OLE Server
SMS Gateway
SMS Messenger
UnRegister SMS Gateway as OLE Server

The SMS Gateway Folder

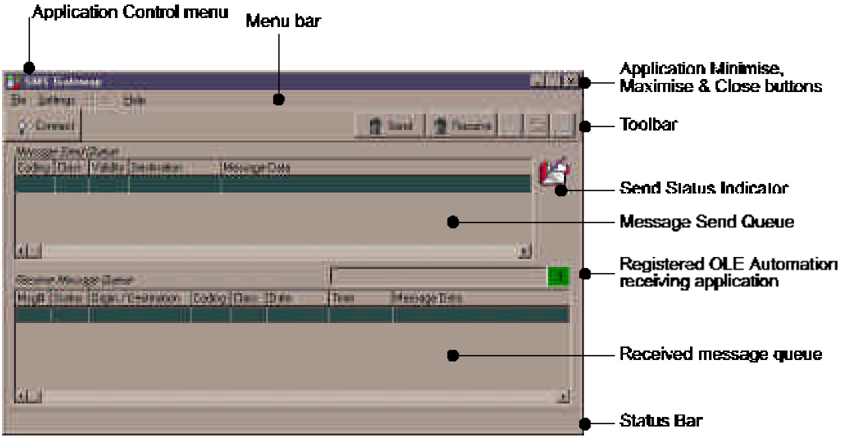
SMS Gateway run icon

The SMS Gateway folder icon will reveal a number of further items when selected. To open this folder, move the mouse pointer over it and pause for a few moments or click the left mouse button. The items in this folder are discussed later in this manual.

To start SMS Gateway, move the mouse pointer over its run icon and click the left mouse button.

Taking a look around SMS Gateway

Now you have run SMS Gateway successfully, take a moment to get familiar with the SMS Gateway environment. The first thing you see is the Main Form, this contains a number of elements, each of which is detailed below:



Application Control menu

The Application Control Menu enables you to control the behaviour of the entire SMS Gateway application. Clicking the mouse pointer over this box will display a pop up menu offering a number of choices, click the mouse pointer over the choice you wish to select. You may also activate this menu by pressing Alt + Spacebar.

Menu Bar

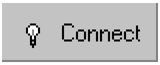
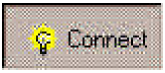
The Menu Bar provides access to all SMS Gateway functions. Clicking the mouse pointer over a menu option will display a pop up menu offering a number of choices, click the mouse pointer over the choice you wish to select. You may also active menus by pressing the Alt key, the first menu option will now be highlighted and you can use the arrow keys to move around the choices, pressing the Enter key on the desired menu selection will enable the required action.

Application Minimise, Maximise & Close buttons

Clicking on these boxes provides quick one-shot selection of the choices available in the Application Control Menu. You may also use the keyboard to perform these actions by remembering the Hot-Key combinations displayed within the Application Control Menu.

Toolbar

The Toolbar provides quick one-shot access to functions within the SMS Gateway application. As you move the mouse pointer over the Toolbar buttons you will see a brief description of the purpose of the button. Simply click on the desired button to activate the choice. Each of the Toolbar buttons is detailed below.

 Connect  Connect	Attempt to establish a link to the Mobile Terminal. If successful, the lightglobe icon will change to a bright yellow “on” state and the button will remain down. If unsuccessful, the lightglobe will remain unlit, and the button will pop back up. For failed connection attempts, you will be informed of the reason the connection failed, and be able to view a communications log.

Purge Message Queues	
🗑 Send	Immediately empty the Send Queue
🗑 Receive	Immediately empty the Receive Queue

Incoming Message Transfer Mode	
✉	Do not transfer any incoming messages from the Mobile Terminal
TE	Transfer only incoming messages marked TE Specific from the Mobile Terminal. Refer to the SMS Messenger section for further detail of this facility. Also the ETSI 07.05 specification contains a concise definition of TE Specific.
✉	Transfer all incoming messages from the Mobile Terminal to the PC. A major advantage of this mode is that an unlimited number of messages may be received without the Mobile Terminal running out of memory and therefore rejecting further incoming messages from the network.

Send Status Indicator

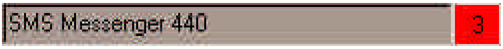
This is a visual indication of the progress of sending messages into the GSM network. The indicator has three states, each of which is discussed below.

📝 No messages are queued for sending.

📫 A message had been queued for sending and a request has been issued to the Mobile Terminal to send the message into the GSM network.

📡 The message has been sent to the GSM network, and the program is waiting for an acknowledgment from the network that the message was sent OK.

Registered OLE Automation receiving application



SMS Messenger 440

3

This field, pictured above, displays the name of the OLE application currently registered for receipt of incoming messages from SMS Gateway.

As incoming SMS messages contain no destination information other that the phone number of the Mobile Terminal, there is no ability to intelligently route messages from SMS Gateway to more than one receiving application. To ensure only one application can retrieve messages from SMS Gateway at any one time, a 'registration' is held by a controlling OLE application that ensures exclusive message retrieval rights.

The concept of registration is also discussed in the Programmer Reference section of this manual.

The Red box containing the number 3 at the right hand side of the field is a timer that counts down the minutes since a poll was last received from the registered OLE application. If a poll has not been received from the OLE application within 3 minutes, its registration will be automatically revoked.

Message Send Queue

Messages that are ready to be handed to the Mobile Terminal for sending are stored in the message send queue until they have been successfully sent into the GSM network. Once a message has been sent, it is removed from the send queue, and if logging is turned on (discussed later), a copy of the message will be stored on the PC in the Transmit Log file. This log file may be viewed by selecting **`File | Transmit Log`** from the menu bar.

Received Message Queue

All messages received from the Mobile Terminal are displayed in a grid forming the Received Message Queue. This grid details a number of message parameters, each of which is detailed below:

- *Msg#*

  This is a decimal number between 0 and 255 sequentially assigned to each incoming message by the Mobile Terminal.

- *Status*

  This indicates if the message has been previously read while it was stored in the Mobile Terminal, or if the message is a copy of a previously sent message. Unread messages are labelled “New”, read messages are labelled “Read”, and previously sent messages are labelled “Sent”.

- *Origin / Destination*

  For incoming messages, this is the GSM Network phone number of the original sender of the message. This number may be a normal mobile phone number, or, a special number used by network operators for services such as Voicemail and E-Mail notification.
  For messages that have been sent directly from the Mobile Terminal interface (ie via the built in menus), this field shows the destination number.

- *Coding*

SMS Messages may contain either text or binary data. The Coding field indicates what type of data the message contains. Possible values are “7 Bit” for text, and “8 Bit” for binary data.

- *Class*

  SMS Messages may be designated a particular class. The message class defines how the Mobile Terminal should handle the message. The available classes and their meaning are:

Class	***Type***	***Meaning***
0	Default	Display the message immediately if possible.
1	ME Specific	Store the message in the Mobile Equipment
2	SIM Specific	Store the message in the SIM Card
3	TE Specific	Send the message directly through the attached Terminal Equipment if possible.

- *Date*

  This is the local date the message was sent into the GSM network.

- *Time*

  This is the local time the message was sent into the GSM network.

- *Message Data*

  The actual text or hex representation of the binary data contained within the message.

If logging is turned on (discussed later), a copy of the message will be stored on the PC in the Receive Log file. This log file may be viewed by selecting **`File | Receive Log`** from the menu bar.

Status Bar

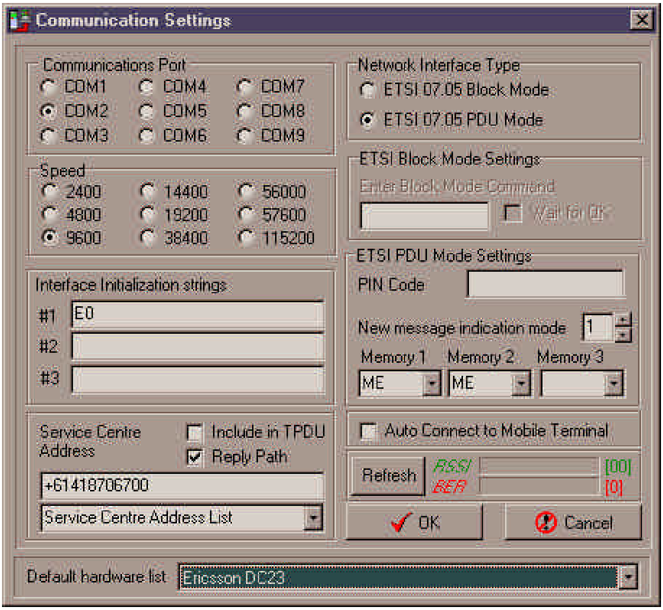
The Status Bar gives you information about what’s going on "under the hood" of SMS Gateway. All communications activity between SMS Gateway and the Mobile Terminal is either described or shown verbatim in the Status Bar.

Program Setup

Before you can send and receive messages using SMS Gateway you need to tell the program how to communicate to the Mobile Terminal.

Communications

From the Menu Bar select **`Settings | Communications`** to load the SMS Gateway Communications setup form. This will display the form (shown below) that contains information that tells the program how to communicate with the Mobile Terminal.



Communication Settings

Communications Port
○ COM1 ○ COM4 ○ COM7
● COM2 ○ COM5 ○ COM8
○ COM3 ○ COM6 ○ COM9

Speed
○ 2400 ○ 14400 ○ 56000
○ 4800 ○ 19200 ○ 57600
● 9600 ○ 38400 ○ 115200

Interface Initialization strings
#1 E0
#2
#3

Service Centre Address □ Include in TPDU
☑ Reply Path
+61418706700
Service Centre Address List

Network Interface Type
○ ETSI 07.05 Block Mode
● ETSI 07.05 PDU Mode

ETSI Block Mode Settings
Enter Block Mode Command
□ Wait for OK

ETSI PDU Mode Settings
PIN Code
New message indication mode 1
Memory 1 ME
Memory 2 ME
Memory 3

□ Auto Connect to Mobile Terminal

Default hardware list Ericsson DC23

The Communications Setup Form.

Each field within this form is detailed below:

- **Communications Port**

  This is the PC “COM” port the interface to the Mobile Terminal is connected to. The COM port assignment depends on the configuration of the PC equipment to which the Mobile Terminal Interface is connected, refer to the Windows Control Panel for details of which COM port your equipment is connected. If you have selected the incorrect port, you will receive an error message when SMS Gateway attempts to connect to the Mobile Terminal.

- **Speed**

  This is the maximum speed the interface to the Mobile Terminal can handle. This is generally 19200 Bits Per Second (BPS), although little value is gained from selecting a speed above 9600 Bps, as SMS is a store and forward system, and the local interface speed will be much faster than actual message throughput capability.

- **Interface Initialisation Strings**

  When SMS Gateway first attempts to connect to the Mobile Terminal, it sends the strings defined in these fields to the Mobile Terminal interface device. This strings contains “Hayes” commands, which are executes sequentially. These commands instruct the interface device how to behave. Refer to the interface device documentation for further details.

- **Service Centre Address**

  The Service Centre address is the number of the message store and forward facility operated by your GSM service provider. Contact your service provider for this number. You will not be able to send messages without entering this address. A pull down list of commonly known Service Centre numbers, sorter by county and service provider is located just below this field.

Include in TPDU

Some Mobile Terminal interfaces (such as Nokia) require that the SMS Service Centre Address is included at beginning of every message submission request. This field enables you to cater both types of equipment.

Reply Path

This setting requests that the SMS Service Centre makes a commitment to deliver a reply from the destination mobile terminal should one be sent. This setting is not supported on certain Mobile Terminals, such as the Siemens M1.

- **Auto Connect to Mobile Terminal**

  In some instances you may want SMS Gateway to connect to the Mobile Terminal as soon as the program loads, this removes the need for a user to manually press the Connect button, or an OLE application to issue the connect instruction. Should SMS Gateway become disconnected from the Mobile Terminal for any reason it will attempt to periodically re-connect.

- **Network Interface Type**

  There are two variants of the protocol standard used for exchanging Short Messages between SMS Gateway and GSM terminals. The choice of protocol required would depend on which mode is implemented by your GSM Mobile Equipment manufacturer. A listing of which manufacturers support which standard is supplied on the Hardware our web site (www.winsms.com).

- **ETSI Block Mode Settings**

  Enter Block Mode Command
  This is the command string sent to the Mobile Terminal interface to switch it from the standard Hayes "AT" command set to the ETSI 07.05 Block Mode protocol.

  Wait for OK
  Some implementations of Block Mode reply immediately with an "OK" response, others do not send an OK until Block Mode is exited. In most instances this field should be checked. If required, refer to your interface documentation for the behaviour of your equipment.

- **ETSI PDU Mode Settings**

  PIN Code

  GSM terminals are generally able to be locked through the use of a PIN code. If your GSM terminal requires a PIN code to be entered before it can send/receive messages then the code can be entered here.

  New message indication mode

  This setting controls how the mobile terminal will handle informing SMS Gateway of new incoming messages. There are four possible values for this field, as detailed below:

0 Buffer unsolicited result codes in TA.
1 Discard indication and reject new received message unsolicited result codes when the TA-TE link is reserved.
2 Buffer unsolicited result codes in the TA when the TA-TE link is reserved (e.g. in on-line data mode) and flush them to the TE after reservation. Otherwise forward them directly to the TE.
3 Forward unsolicited result codes directly to the TE. TA-TE link specific inband technique used to embed result codes and data when TA is in on-line data mode.

In the above cases the TA is the Mobile Terminal or Mobile Terminal Interface Device. TE refers to the PC on which SMS Gateway is running.

Memory 1, Memory 2, Memory 3

This field defines where short messages should be stored and retrieved from for the hardware combination you are using. The possible storage locations are:

Memory 1	Message Reading and Deleting.
Memory 2	Message Writing and Sending.
Memory 3	Received Message Storage (unless forwarded directly to the PC).

Valid entries for these fields are:

BM	Broadcast Message storage
ME	Mobile Equipment storage
MT	Any of the storage's associated with the Mobile Equipment
SM	SIM message storage
TA	Terminal Adaptation message storage

Signal Quality indicators

By pressing the Refresh button, the command *"AT+CSQ"* is sent to the mobile terminal. If this command is supported by the device, it will return two figures that are shown as readings on the bar graphs. The values are detailed below:

Label	*Range*	*Description*
RSSI	0 - 31	Received Signal Strength Indicator, the higher the reading the better
BER	0 - 7	Bit Error Rate, the lower the reading the better.
	99	Not known or detectable for either RSSI or BER.

- **Default hardware list**

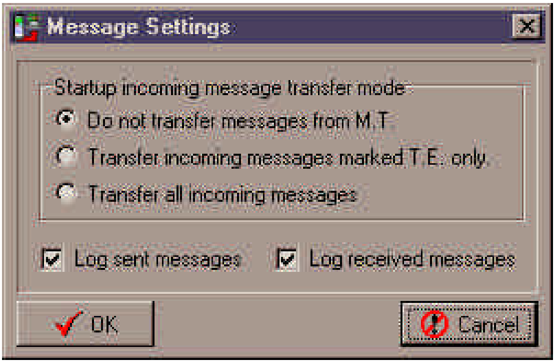
  To simplify the configuration process, a list of tested equipment types is provided in a pull down selection box. If your equipment type is listed, click on its entry to load SMS Gateway with the default communication settings for that device.

  Note: The default equipment settings may not always be correct. It is possible to re-configure Mobile Terminal interface devices to behave differently, and therefore the SMS Gateway default settings may not be appropriate.

  If you are having trouble connecting to your equipment please send us an e-mail to support@winsms.com and we will endeavour to assist wherever possible.

Message Settings

From the Menu Bar select **`Settings | Messages`** to load the SMS Gateway Message Settings form. This will display the form (shown below) that contains information that tells the program how to behave in terms of message sending and receiving.



Message Settings

Startup incoming message transfer mode

● Do not transfer messages from M.T

○ Transfer incoming messages marked T.E. only

○ Transfer all incoming messages

☑ Log sent messages ☑ Log received messages

The Message Settings Form.

Each field within this form is detailed below:

- **Startup incoming message transfer mode**

  Please refer to the previously discussed Toolbar section for an explanation of the available transfer modes.

  If SMS Gateway is being used in a fixed installation (i.e. it is generally left permanently running), it will probably be desirable to have SMS Gateway transfer all new incoming messages from the Mobile Terminal memory directly to the PC by selecting the ***Transfer all incoming messages*** mode. This way there is reduced chance of messages being missed due to the Mobile Terminal memory being full, and this also enables OLE Automation connections to receive messages without operator intervention.

  It may however be more desirable to select Transfer All mode though an OLE Automation call (discussed later in this manual), as there is no point in receiving messages into SMS Gateway if there is no associated application to retrieve and process the incoming messages.

- **Log sent messages**

  When this setting is selected, SMS Gateway will copy all successfully sent messages to a text file on the hard disk.

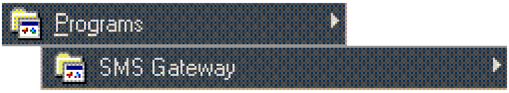
- **Log received messages**

  When this setting is selected, SMS Gateway will copy all incoming message to a text file on the hard disk.

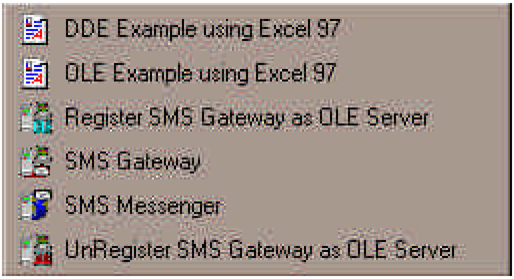
First Time Configuration of SMS Messenger

SMS Messenger is a partner application to SMS Gateway. It provides a friendly user interface for the transmission and reception SMS messages using SMS Gateway.

The icon for SMS Messenger is located in the SMS Gateway folder on your Start Menu.



Click on this folder to reveal the following items:

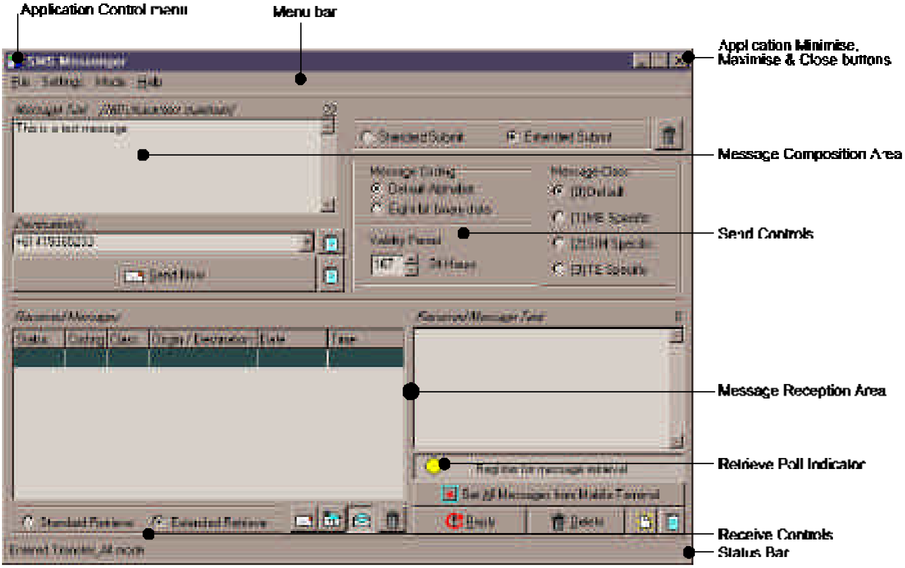

DDE Example using Excel 97
OLE Example using Excel 97
Register SMS Gateway as OLE Server
SMS Gateway
SMS Messenger
UnRegister SMS Gateway as OLE Server

Before you can use SMS Messenger, you must first register SMS Gateway as an OLE Automation server. To do this double click the left mouse button over the “Register SMS Gateway as OLE Server” icon, you will receive no visual feedback that this action has been performed successfully, but you will receive an error message if it fails for any reason.

Once SMS Gateway is registered an OLE Automation server, you may run SMS Messenger. To do so move the mouse pointer over the SMS Messenger icon and click the left mouse button once.

Taking a look around SMS Messenger

Now you have run SMS Messenger successfully, take a moment to get familiar with the SMS Messenger environment. The first thing you see is the Main Form, this contains a number of elements, each of which is detailed below:



The ***Application Control menu***, ***Menu Bar***, ***Application Minimise, Maximise & Close buttons***, have been discussed in the previous section. The function of these controls is no different in SMS Messenger to SMS Gateway.

Message Composition Area

This area is used for the interactive creation and sending of messages. It comprises four separate components, each of which is discussed below:

- **Destination**

  Enter the phone number of the Mobile Terminal you wish to send the message to. You can enter multiple destinations for the same message by separating the numbers by a comma, e.g. destination1, destination2, destination3 ; and so on.

  You can select from a list of pre-stored destinations by pressing the down arrow symbol at the far right of the Destination field. This action will read all the entries in your Address Book (discussed next) and display them in alphabetical order. You can select an entry by clicking the mouse pointer over it.

- **Address Book button**

  SMS Gateway uses a simple text file to store a list of addresses, you can view and edit the file by pressing this button. When you press this button, SMS Gateway will launch your standard Windows text file editor (usually "WordPad").

  The syntax of address book entries is:

  > *destination1* **,** *destination2* **,** *destination3* **;** *Comments*
  >
  > Each line starts with one or more destination phone numbers, separated by commas, followed by a semicolon and a description of the entry (which can be anything you like).

  A number of sample address book entries are supplied for you.

- **Message Text**

  Type the text you wish to send here. A single message can be up to 160 characters in length (less for unlicensed users, due to the presence of the signature message), if you exceed the 160 character length limit you will be warned prior to sending.

  Any special characters you enter that are unable to be understood by the GSM network will be sent as spaces.

- **"Send Now" button**

  Once you have selected the destination and entered your desired message text, press the **Send** Now button to place the message(s) into the send queue for transmission into the GSM Network.

Send Controls

SMS Gateway supports two modes of message submission; these are Standard and Extended.

The Standard submit mode does not allow the sending of binary data, nor the setting of message attributes such as message coding, class and validity period. The standard message submission mode will work with older copies of SMS Gateway.

By selecting Extended submit, the disabled message sending controls become active. The meaning of these setting is discussed below:

- *Coding, and Class*

  For details of these fields, please refer to the "Received Message Queue" area of the "Taking a look around SMS Gateway" section found earlier in this manual.

- Validity Period

  The GSM network will attempt to deliver an SMS message to a destination for a set period of time, after which it will discard the message. The duration the GSM network will hold onto the message for is known as the Validity Period (VP).

  The default VP used by SMS Gateway is one day, you can change this setting up or down by clicking the small arrows at the right hand side of this field.

- **Purge SMS Gateway send queue**

  🗑

  Pressing this button will immediately empty the SMS Gateway send queue, messages that have not been sent to the GSM network will be lost.

Message Reception Area

A summary of messages received from the Mobile Terminal is displayed in the grid at the left hand side of the message reception area. This grid details a number of message parameters, each of which is detailed below:

RECEIVED MESSAGE GRID

- *Status, Coding, Class, Origin/Destination, Date, and Time*

  For details of these fields, please refer to the “Received Message Queue” area of the “Taking a look around SMS Gateway” section found earlier in this manual.

To view the message text associated with each message, click the mouse pointer over the message summary, the associated text will be displayed in the field at the right of the grid.

Receive Controls

SMS Gateway supports two modes of message retrieval from SMS Gateway; these are Standard and Extended.

The Standard receive mode does not allow the native retrieval of binary data (binary data is received as a text hexadecimal representation), nor the recording of message attributes such as message coding and class. The standard message retrieval mode will work with older copies of SMS Gateway.

- **SMS Gateway Incoming Message Transfer Mode**

  For details of these buttons, please refer to Toolbar" area of the "Taking a look around SMS Gateway" section found earlier in this manual.

- **Purge SMS Gateway receive queue**

  Pressing this button will immediately empty the SMS Gateway receive queue, messages that have not been retrieved by SMS Messenger, but are still stored in the Mobile Terminal will <u>not</u> be lost.

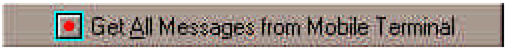
At the bottom right hand side of the message reception area is a group of buttons used to manage the reception of messages. The function of each of these buttons is detailed below:

- **Register for message retrieval**

  Attempt to register SMS Messenger as the recipient of incoming messages from SMS Gateway. If successful, the button will remain down, if unsuccessful, the button will pop back up. If another application is already registered for message retrieval from SMS Gateway, this request will fail. You will still however be able to send messages without difficulty.

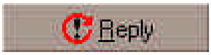
When registered, the yellow indicator at the left hand side of the button flashes red periodically, as SMS Messenger checks with SMS Gateway if any new messages have arrived.

- **Get All Messages from Mobile Terminal**



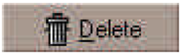
Tell SMS Gateway to request a transfer of all received message currently stored in the Mobile Terminal. The messages will be automatically deleted from the Mobile Terminals’ memory once successfully transferred to SMS Messenger.

- **Reply**



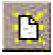
Copy the origin number of the currently selected received message into the Destination field of the Message Composition Area, and move the cursor to the *Send Message Text* area.

- **Delete**



Delete the currently selected received message. If logging is enabled (discussed later), a copy of the message will have previously been stored in the Receive Log file when it was first transferred to SMS Messenger.

- **Clear Grid button**



This button will delete all entries in the Received Message Grid. If logging is enabled (discussed later), copies of the messages will have been previously stored in the Receive Log file when they were first transferred to SMS Gateway.

- **Receive Log button**

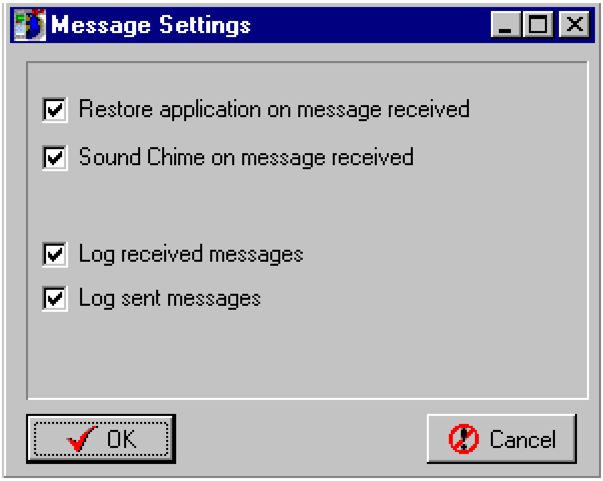
  If logging is enabled (discussed later), copies of all messages appearing in the Received message grid are stored on your PC in the Receive Log file. This log file may be view and edited by pressing the small button with the notepad icon on it at the bottom right hand side of the message reception area.

Status Bar

The Status Bar gives you information about what's going on "under the hood" of SMS Messenger. All communication between SMS Messenger and SMS Gateway is detailed in the Status Bar.

Message Settings

From the Menu Bar select **`Settings | Messages`** to load the SMS Messenger Message Settings form. This will display the form (shown below) that contains information that tells the program how to behave in terms of message sending and receiving.



☑ Restore application on message received

☑ Sound Chime on message received

☑ Log received messages

☑ Log sent messages

The Message Settings Form.

Each field within this form is detailed below:

- **Restore application on message received**

  This setting tells SMS Messenger to jump to the Windows foreground when a new message is received from SMS Gateway.

- **Sound Chime on message received**

  This facility provides an audible notification (beep) of the reception of a new incoming message.

- **Log received messages**

  When this setting is selected, SMS Messenger will copy all incoming message to a text file on the hard disk. This log is separate and additional to the logging function within SMS Gateway.

- **Log sent messages**

  When this setting is selected, SMS Messenger will copy all successfully sent messages to a text file on the hard disk. This log is separate and additional to the logging function within SMS Gateway.

Start your engines!

Now that you are familiar with SMS Gateway and SMS Messenger you are ready to send and receive messages from the PC. The following section steps you through sending and receiving your first message.

Sending a message

The first message we send will be to yourself, this is called a “Loopback” message. Each of the required steps is listed below:

1. Run the SMS Messenger application, the SMS Gateway will automatically be started, and a connection to the Mobile Terminal attempted.

2. Move to the *Destination* field and enter your own mobile phone number

3. Enter some text in the *Message Text* field, e.g. “My first test”.

4. Press the **S**end Now button.

5. If everything is working properly, the *Message Text* field should be cleared, and the message ‘1 Message(s) queued for sending’ appear in the Status Bar.

   Immediately within SMS Gateway a copy of your message should appear in the Message Send Queue. The Send Status Indicator will now be displaying a letterbox, and after a few seconds should move to a satellite dish, then return to the writing in a book symbol.

A few moments after the message has been sent, if should arrive in back in your Mobile Terminal. We will now look at receiving this message.

Receiving a message

The following assumes that SMS Gateway is in the default “Do not transfer messages from the M.T.” mode. If this is true, the incoming message will have been stored in the Mobile Terminal, and it probably beeped to alert you of the message arrival.

To retrieve the message, and any existing messages stored in the mobile terminal. Perform the following steps.

1. Press the “Register for message retrieval” button. If successful the button will remain down, and the Retrieve Poll Indicator begin to flash.

2. Press the “Get **A**ll Messages from Mobile Terminal” button. A few moments later you message should appear in the Received message grid.

From this point on, any new messages arriving in the Mobile Terminal will be directly transferred to SMS Messenger via SMS Gateway.

Conclusion

Congratulations, hopefully you have now successfully sent and received an SMS Message using SMS Gateway with SMS Messenger

If you experience trouble connecting to your Mobile Terminal, or sending messages, please send an e-mail to support@winsms.com and we will endeavour to solve your problem.

The next section will take you through advanced programming topics.

The Command Line Interface

As stated previously, SMS Gateway supports the sending of messages directly from the Windows Command Line.

The syntax for sending a message is as follows:

"C:\Program Files\GPA_Technology\SMS Gateway\SMS_Messenger.exe" SEND 0419000000, 0419000001 “Hello, this is a test message to two destinations”

Note: The above is entered as one continuous line with no line breaks

The Command Line Interface may be used at any time, is SMS Gateway is not already running it will be automatically started. If SMS Gateway is currently running, the send request will be passed to the existing instance, and will not result in two copies of SMS Gateway running.

Developers Reference

This section details interworking between SMS Gateway and other Windows applications. It is assumed the reader has a reasonable understanding of Windows programming concepts and is familiar with DDE and OLE Automation.

Dynamic Data Exchange (DDE)

DDE provides a simple way to submit messages to SMS Gateway for sending over the GSM network, from most common Windows applications.

SMS Gateway is a DDE server application, with the following DDE attributes:

Application name	**= “SMS_Gateway”**
DDE Topic	**= “DDE_Send”**

SMS Gateway receives the send instruction from a DDE client application in the form of a macro. The following example syntax is Excel 97 based using Visual Basic:

channelNumber = Application.DDEInitiate("SMS_Gateway", "DDE_Send")

Attempt to establish a DDE conversation with the DDE_Send topic of SMS Gateway. For this command to work SMS Gateway must already be running on your PC.

DDEExecute channelNumber, "[destination1, destination2 ; Message text]”

Use the “channelNumber” assigned to the DDE conversation established in the last command to send a macro command to SMS Gateway. The command is in the form of a string that contains one or more destination mobile terminal numbers, separated by commas, to which to send the message, followed by a semicolon and then the actual message text to send.

Application.DDETerminate (channelNumber)

Close the DDE Conversation.

Note : SMS gateway <u>MUST</u> be running prior to issuing the above commands, otherwise your message will simply be lost. Unfortunately, due to the limited nature of DDE, there is no explicit verification that messages have been successfully submitted to SMS Gateway, so you must manually check the SMS Gateway Transmit Log (discussed previously in this manual) to verify messages have been sent.

Included with SMS Gateway is an example Microsoft Excel 97 for Win95 spreadsheet file named *EXCELDDE_97.XLS* that demonstrates sending messages with SMS Gateway using DDE.

OLE Automation

SMS Gateway is written in Delphi 3.0, and compiled as an OLE Automation server in the form of an out-of-process executable. For an application to interwork with SMS Gateway, it must support references to OLE Objects of this type.

Before SMS Gateway can be called as an OLE Object, it must be registered in Windows. An icon is provided in the SMS Gateway folder that performs this function, click on the icon to “Register SMS Gateway as OLE Server”, you will receive no visual feedback that this action has been performed successfully, but you will receive an error message if it fails for any reason.

As SMS Gateway is an “out-of-process” executable (EXE), only one copy of SMS Gateway will ever be loaded, this single instance can be accessed concurrently by as many different applications as you wish.

Message handling flow diagrams.

When implementing solutions with SMS Gateway, it is important to understand the flow of messages between the various entities. Because a number of systems, each capable of storing messages in one form or another are inter-operating, care must be taken to avoid message duplication,

The following flow diagrams should assist you to ensure messages are not lost, or improperly duplicated.

Message Transmission

OLE Application	SMS Gateway	Mobile Terminal
Send_Message => *(Log Message to “m_tx.txt”)*		
	Submit Message =>	
		<= Confirm Request (wait for acknowledge from network) <= Message Sent
	Delete message from Send Queue. *Log Message to “transmit.txt”*	

Message Reception

OLE Application	SMS Gateway	Mobile Terminal
		<= Incoming message notification
	Read Message =>	
		<= Message Content
	Log Message to “receive.txt”	
Retrieve_Message Poll =>		
	<= Message Content	
(Log Message to “m_rx.txt”)		
	Delete Message =>	
	Delete message from Receive Queue.	<= Confirm Delete

Command summary

```
function Is_Connected(const appName: WideString): WordBool;
function Is_Auto_Connect(const appName: WideString): WordBool;

procedure Connect(const appName: WideString);
procedure Disconnect(const appName: WideString);

procedure Toggle_Connection(const appName: WideString);

function Receive_Mode(const appName: WideString): WideString;

procedure No_Transfer_Mode(const appName: WideString);
procedure Transfer_All_Mode(const appName: WideString);
procedure Transfer_TE_Mode(const appName: WideString);

procedure Get_All(const appName: WideString);

function Register_Retrieve(const appName: WideString): WordBool;

function Retrieve_Message(const appName: WideString): WideString;
function Retrieve_Message_Ex(const appName: WideString): OleVariant;

function deRegister_Retrieve(const appName: WideString): WordBool;

function Send_Message(const appName, destinations,message_text:
WideString): Integer;
function Send_Message_Ex(const appName, destinations: WideString;
binary: WordBool; message_class, validity_period, data_length: Smallint;
message_data: OleVariant): Integer;

function Purge_Receive(const appName: WideString): Integer
function Purge_Send(const appName: WideString): Integer;

procedure SMS_Minimize(const appName: WideString);
procedure SMS_Restore(const appName: WideString);
```

Detailed Command Reference

The following is a list of all procedures and functions surfaced by SMS Gateway.

Note: the parameter **“appName”** is present in all OLE function and procedure calls surfaced by SMS Gateway. This parameter is a String that contains the name of the application issuing the OLE command. This field is arbitrary (however mandatory) and should be employed consistently within an application.

function Is_Connected(const appName: WideString): WordBool;

This command tells your application if SMS Gateway is currently connected to the Mobile Terminal

Return:

If SMS Gateway is connected to the Mobile Terminal the return value is True, otherwise False.

function Is_Auto_Connect(const appName: WideString): WordBool;

This command tells your application if SMS Gateway is configured for automatic connection to the Mobile Terminal at startup.

Return:

If SMS Gateway is configured for automatic connection the return value is True, otherwise False.

procedure Connect(const appName: WideString);

This command tells SMS Gateway to attempt connection to the Mobile Terminal immediately.

Return:

This is a procedure, no values are returned. You can check the results of this operation by waiting a few moments, and then calling the Is_Connected() function.

procedure Disconnect(const appName: WideString);

This command tells SMS Gateway to disconnect from the Mobile Terminal immediately.

Return:

This is a procedure, no values are returned. You can check the results of this operation by waiting a few moments, and then calling the Is_Connected() function.

procedure Toggle_Connection(const appName: WideString);

This command tells SMS Gateway to toggle its connection to the Mobile Terminal. This command is retained for backward compatibility only, it is recommended the more deterministic “connect” and “disconnect” commands be used.

Return:

This is a procedure, no values are returned. You can check the results of this operation by waiting a few moments, and then calling the Is_Connected() function.

function Receive_Mode(const appName: WideString): WideString;

This command returns the current receive mode of SMS Gateway.

Return:

A string containing one of the following:

Return Value	Meaning
Disconnected	SMS Gateway is not connected to the Mobile Terminal.
No_Transfer	Do not transfer any incoming messages from the Mobile Terminal.
Transfer_TE	Transfer only incoming messages marked TE Specific from the Mobile Terminal to SMS Gateway.
Transfer_All	Transfer all incoming messages from the Mobile Terminal to SMS Gateway.

procedure No_Transfer_Mode(const appName: WideString);

This command tells SMS Gateway to enter No_Transfer receive mode.

Return:

This is a procedure, no values are returned. You can check the results of this operation by waiting a few moments, and then calling the Receive_Mode() function.

procedure Transfer_TE_Mode(const appName: WideString);

This command tells SMS Gateway to enter Transfer_TE receive mode.

Return:

This is a procedure, no values are returned. You can check the results of this operation by waiting a few moments, and then calling the Receive_Mode() function.

procedure Transfer_All_Mode(const appName: WideString);

This command tells SMS Gateway to enter Transfer_All receive mode.

Return:

This is a procedure, no values are returned. You can check the results of this operation by waiting a few moments, and then calling the Receive_Mode() function.

procedure Get_All(const appName: WideString);

This command tells SMS Gateway to transfer all messages currently stored in the Mobile Terminal to the SMS Gateway Receive Message Queue. Note: Messages are not deleted from the Mobile Terminal as they are received by SMS Gateway. The messages must be retrieved from SMS Gateway by an OLE application to trigger a message deletion from the Mobile Terminals' memory.

Return:

This is a procedure, no values are returned. You can check the results of this operation by waiting a few moments, and then calling the Retrieve_Message() function in a loop until it returns an empty string when all messages have been retrieved from SMS Gateway.

function Register_Retrieve(const appName: WideString): WordBool;

This function attempts to register the calling application 'appName' as the recipient of messages from the SMS Gateway Receive Queue.

Return:

If successful the return value is True, otherwise False. This function will fail under the following conditions:

- SMS Gateway is not connected to the Mobile Terminal.
- Another application is already registered for message retrieval.

Once successfully registered a timer is started within SMS Gateway, if a Retrieve_Message poll (discussed next) is not received from the registered application within 3 minutes, its registration will be automatically revoked.

function Retrieve_Message(const appName: WideString): WideString;

This command directs SMS Gateway to copy the contents of the first available message in its Receive Message Queue to the requesting application, and then delete it from the queue.

Return:

If the calling application 'appName' is currently registered for message retrieval, and SMS Gateway is holding messages within its Received Message Queue (discussed earlier in this manual) it will return the first available message as a string containing the following fields :

Field Position	Field Length	Field Name
1	4	Status
6	15	Origin
22	8	Date
31	8	Time
40	0 to 160 (Variable)	Message Text

The meaning of the fields has been discussed previously in this manual.

If there are no messages to be retrieved, or the calling application is not registered for message retrieval, an empty string will be returned. To retrieve all messages from the SMS Gateway Received Message Queue, simply loop this command until it returns an empty string.

function Retrieve_Message_Ex(const appName: WideString): OleVariant;

This command directs SMS Gateway to copy the contents of the first available message in its Receive Message Queue to the requesting application, and then delete it from the queue.

This is the extended version of the previous **Retrieve_Message** command. This command returns binary data in the form of an OLE Variant

Return:

The command returns binary data in the form of an OLE Variant. The returned data is contained in a 300 byte single dimension array. Refer to the SMS Messenger source code for full details of how to exchange OLE Variant types with SMS Gateway.

If the calling application 'appName' is currently registered for message retrieval, and SMS Gateway is holding messages within its Received Message Queue (discussed earlier in this manual) it will return the first available message as a string containing the following fields :

Array Position[1]	Length	Field Name
1	4	Status[2]
5	5	Coding
10	1	Class
11	15	Origin
26	8	Date
34	8	Time
42	1	Data Length
43	0 to 160 (Variable)	Message Data[3]

1. The array position is enumerated starting from one, not zero.

2. The Status field has three possible meanings, depending on its value, these are:

0x00	Not registered for message retrieval
0xFF	No messages found
Otherwise	A message found, array position 1-4 contains the message Status

3. The message data may contain text or binary data, depending on the value of the Coding field. The Message data is represented as follows:

Coding	*Maximum Length*	*Data Type*
7 Bit	160	Ordinal representation of ASCII chars
8 Bit	140	Raw binary data

The meaning of the fields not listed here have been discussed previously in this manual.

To retrieve all messages from the SMS Gateway Received Message Queue, simply loop this command until it returns 0xFF in the first byte of the returned array.

function deRegister_Retrieve(const appName: WideString): WordBool;

This function attempts to de-register the calling application ‘appName’ from being the registered recipient of messages from the SMS Gateway Receive Queue.

Return:

If successful the return value is True, otherwise False. This function will fail under the following condition:

- If ‘appName’ is not currently registered as the message retrieval application.

function Send_Message(const appName, destinations,message_text: WideString): Integer;

This command will submit a message to the SMS Gateway send queue for transmission over the GSM network. If SMS Gateway is not “Connected “ to the Mobile Terminal when this command is received, it will automatically attempt to connect to the device.

Parameters:

destinations is a String that contains one or more destination mobile terminal numbers, separated by commas, to which to send the message.

message_text is a String that contains a verbatim copy of the message you wish to send to the specified destinations

Return:

If successful, the return value will be a positive integer equal to the number of destinations found in the **destinations** string

If unsuccessful, the return value will be negative integer, with the following possible values:

Return Code	*Description*
-1	Appname not provided
-2	No destination addresses were found

Note: A non-zero return value does not imply that a message has been successfully sent over the GSM network, merely that it has been added to the send queue for transmission when possible. To verify that a message has been successfully sent, the Transmit Log file may be used, only messages actually sent over the GSM network are added to this file.

function Send_Message_Ex(const appName, destinations: WideString; binary: WordBool; message_class, validity_period, data_length: Smallint; message_data: OleVariant): Integer;

This command will submit a message to the SMS Gateway send queue for transmission over the GSM network. If SMS Gateway is not “Connected “ to the Mobile Terminal when this command is received, it will automatically attempt to connect to the device.

This is the extended version of the previous **Send_Message** command.

Parameters:

destinations is a String that contains one or more destination mobile terminal numbers, separated by commas, to which to send the message.

binary is a Boolean flag that identifies if the type of message data being sent, this field is to be set as follows

Value	*Data Type*
False	7 Bit Text
True	8 Bit Binary data

message_class is a Small Integer that specifies the SMS Message Class as defined in ETSI Specification 07.05. Possible values are:

Value	*Meaning*
0	Default
1	ME Specific
2	SIM Specific
3	TE Specific

validity_period is a Small Integer that specifies the SMS Validity Period as defined in ETSI Specification 07.05. Possible values range between 0 and 255. Refer to the SMS Messenger source code for full details of how to convert real time message expiry times to this format.

message_data contains binary data in the form of an OLE Variant. The sent data is contained in a 300 byte single dimension array. Refer to the SMS Messenger source code for full details of how to exchange OLE Variant types with SMS Gateway.

Return:

If successful, the return value will be a positive integer equal to the number of destinations found in the **destinations** string

If unsuccessful, the return value will be negative integer, with the following possible values:

Return Code	*Description*
-1	Appname not provided
-2	No destination addresses were found
-3	An invalid message class was specified
-4	An invalid validity period was specified
-5	The binary data was too long
-6	'The message text was too long

Note: A non-zero return value does not imply that a message has been successfully sent over the GSM network, merely that it has been added to the send queue for transmission when possible. To verify that a message has been successfully sent, the Transmit Log file may be used, only messages actually sent over the GSM network are added to this file.

function Purge_Receive(const appName: WideString): Integer

This command immediately empties the SMS Gateway receive queue.

Return: This is a procedure, no values are returned.

function Purge_Send(const appName: WideString): Integer;

This command immediately empties the SMS Gateway send queue.

Return: This is a procedure, no values are returned.

procedure SMS_Minimize(const appName: WideString);

This command minimises SMS Gateway onto the Windows Taskbar.

Return:

This is a procedure, no values are returned.

procedure SMS_Restore(const appName: WideString);

This command restores SMS Gateway to its normal screen position.

Return:

This is a procedure, no values are returned.

OLE Automation Sample Code

Sample source code is provided with SMS Gateway in the form of the SMS Messenger application. SMS Messenger is written in Delphi 3.0 professional, if you do not have access to a Delphi compiler, you can view the source code in a standard text editor by opening the “Unit1.pas” file.

Also included with SMS Gateway is an example Microsoft Excel for Win95 spreadsheet file named *EXCELOLE_97.XLS* that demonstrates full bidirectional interworking with SMS Gateway.

Thankyou for choosing SMS Gateway by GPA Technology