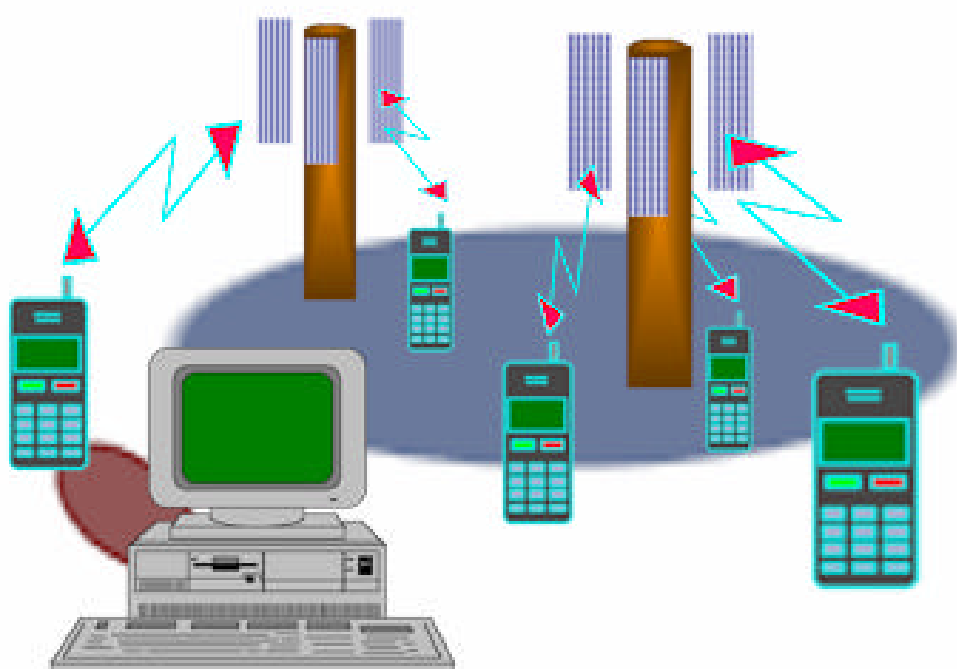


SMS Gateway

Powered By...
GPA
Technology

Version 11.4



Installation, Users & Developers Manual

SMS Gateway by GPA Technology

Installation, Users & Developers Manual¹

Copyright © 1997 - 2004 GPA Technology Pty. Ltd.
All Rights Reserved.
PO Box 567
Carnegie Victoria 3163
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.

¹ Cross Platform edition

Table Of Contents

TABLE OF CONTENTS	3
INTRODUCTION	5
BEFORE YOU BEGIN	6
SYSTEM REQUIREMENTS	6
ABOUT THIS MANUAL	6
DOCUMENT CONVENTIONS	6
BASIC CONCEPTS	7
WHAT IS GSM	7
WHO ARE ETSI	7
WHAT IS SMS	7
WHAT IS SMS GATEWAY	7
INSTALLATION	8
SOFTWARE INSTALLATION – MICROSOFT WINDOWS	8
SOFTWARE INSTALLATION – LINUX	9
GETTING STARTED	10
SETTING UP YOUR HARDWARE	10
FIRST TIME CONFIGURATION OF SMS GATEWAY	11
LAUNCHING SMS GATEWAY - WINDOWS	11
LAUNCHING SMS GATEWAY - LINUX	11
TAKING A LOOK AROUND SMS GATEWAY	12
<i>Application Control menu</i>	13
<i>Menu Bar</i>	13
<i>GSM Status Panel</i>	13
<i>Send Status Indicator</i>	14
<i>Message Retrieval Entity</i>	14
<i>Interactive Controls</i>	15
<i>Message Send Queue</i>	16
<i>Received Message Queue</i>	17
<i>Status Bar</i>	18
PROGRAM SETUP	19
<i>Program Options</i>	19
<i>Communications Settings</i>	21
<i>Message Settings</i>	25
<i>Default Message Attributes</i>	31
<i>POP3 Settings</i>	33
<i>SMTP Settings</i>	35
<i>HTTP Settings</i>	37
<i>SNMP Settings</i>	43
TESTING YOUR SMS GATEWAY CONFIGURATION	45
FIRST TIME CONFIGURATION OF SMS MESSENGER	46
TAKING A LOOK AROUND SMS MESSENGER	46
<u><i>Message Composition Area</i></u>	47
<u><i>Send Controls</i></u>	47
<u><i>Message Reception Area</i></u>	49
<u><i>Receive Controls</i></u>	49
<i>Message Settings</i>	51

START YOUR ENGINES!	53
SENDING A MESSAGE - WINDOWS	53
SENDING A MESSAGE - LINUX	53
RECEIVING A MESSAGE	54
CONCLUSION	54
TESTING AND DIAGNOSIS	55
COMMUNICATIONS BETWEEN THE PC AND GSM MOBILE TERMINAL	55
COMMUNICATIONS BETWEEN SMS GATEWAY AND EMAIL SERVERS	55
IF ALL ELSE FAILS	55
RUNNING MULTIPLE INSTANCES OF SMS GATEWAY	57
INTEGRATORS & DEVELOPERS REFERENCE	58
THE COMMAND LINE INTERFACE	58
<i>Control commands</i>	58
<i>Message Submission</i>	62
DYNAMIC DATA EXCHANGE (DDE)	63
OLE AUTOMATION	64
<i>Message handling flow diagrams.</i>	64
<i>Command summary</i>	65
<i>Detailed Command Reference</i>	66
OLE AUTOMATION SAMPLE CODE	73

Introduction

SMS Gateway is a unique Short Message Service (SMS) messaging tool. It facilitates complete two way messaging capabilities over GSM networks to/from applications through the use of DDE², OLE Automation¹, SMTP, POP3, HTTP, XML, and the Command Line Interface (CLI). SMS Gateway is available for the Windows³ and Linux⁴ Operating Systems

SMS Gateway connects a PC to a GSM handset (Mobile Terminal), via a PCMCIA "Cellular Data Card", specialised data cable connected to a COM (serial) port, and also via Infrared or Bluetooth wireless connections. 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 an interactive two way messaging utility named SMS Messenger¹, which demonstrates some of the capabilities of SMS Gateway when using the OLE¹ interface. 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 (DDE¹) and Object Linking and Embedding Automation (OLE Automation¹) specifications to allow the transmission and reception of SMS messages directly from any application supporting these standards. Applications supporting DDE¹ & OLE Automation¹ 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 Command Line Interface (CLI). This facility is useful for easy integration with Network Management utilities that are able issue commands under certain prescribed circumstances, e.g. to send a message to a technician when a certain equipment alarm is raised.

Additionally, SMS Gateway supports a powerful bi-directional HTTP based eXtensible Markup Language (XML) interface that enables easy integration with web enabled environments, such as Active Server Pages, Java, Perl and other IP environments.

Fully commented source code for the above mentioned SMS Messenger¹ application, written in Delphi 7.0, 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 Visual Basic in Excel¹ is also included with the installation files.

SMS Gateway includes a set of management utilities including a HTML based diagnostic facility, generation of SNMP traps for significant events, and comprehensive logging to text files (viewable via a HTML management page).

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

² Only available under the Windows Operating System

³ Windows is a trademark of Microsoft Corporation.

⁴ Linux is a trademark of Linus Torvalds.

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.

System Requirements

System Requirements	
Processor	486 or better
Operating Systems	Microsoft Windows 95 and Above & Red Hat ⁵ Linux 7.2 and Above
Memory	8MB Minimum
Display	VGA or Better
Mouse	Recommended
Network	IP Version 4
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.
Troubleshooting	Diagnostic tools and approaches to solving problems.
Developers Reference	Comprehensive details for interworking SMS Gateway with other applications through DDE, Command Line Interface 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.

⁵ RED HAT is a registered trademark of Red Hat, Inc.

Basic Concepts

What is GSM

GSM stands for **G**lobal **S**ystem for **M**obiles. This is a world-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) (TS 100 585 / 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 7 Bit 160 characters or 140 octets of binary data.

The SMS standard incorporates a number of advanced messaging options that can be leveraged via SMS Gateway. Some of these include the ability to:

- Send custom Ringtones and Graphics and Operator Logos
- Send WAP forms
- Send messages that appear directly on the screen, without the need for the user to navigate any menus
- Turn on and off specialised indicators such as; Message, Fax, and Email waiting
- Download SIM Toolkit applications "Over The Air"
- Use languages other than English via support of custom character mappings and double byte UCS2 characters
- Send messages that overwrite existing messages, very useful for time sensitive information and avoiding filling up a recipients memory capacity

The above capabilities may not be able to be achieved on all receiving handsets, this is dependent on what the handset manufacturer has decided to implement. It should be noted that the Mobile Terminal connected to SMS Gateway does not need to support these features, only the receiving handset does.

What is SMS Gateway

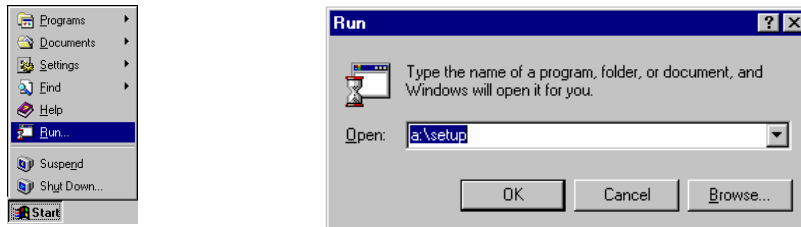
SMS Gateway presents developers and integrators with a simple and powerful interface to the complex ETSI GSM SMS protocols, enabling faster time to market for new SMS messaging based products, and fast in-house development of corporate messaging solutions.

SMS Gateway is designed to be very flexible, offering numerous ways to interface to other environments, and access to all the advanced SMS messaging attributes so all capabilities can be leveraged.

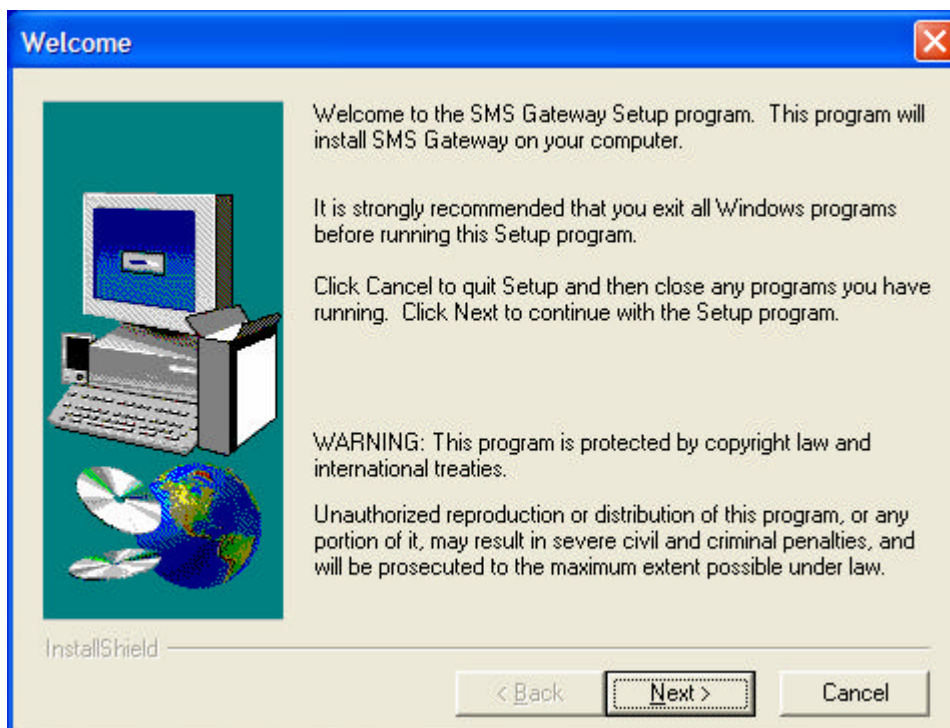
Installation

Software Installation – Microsoft Windows

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).



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

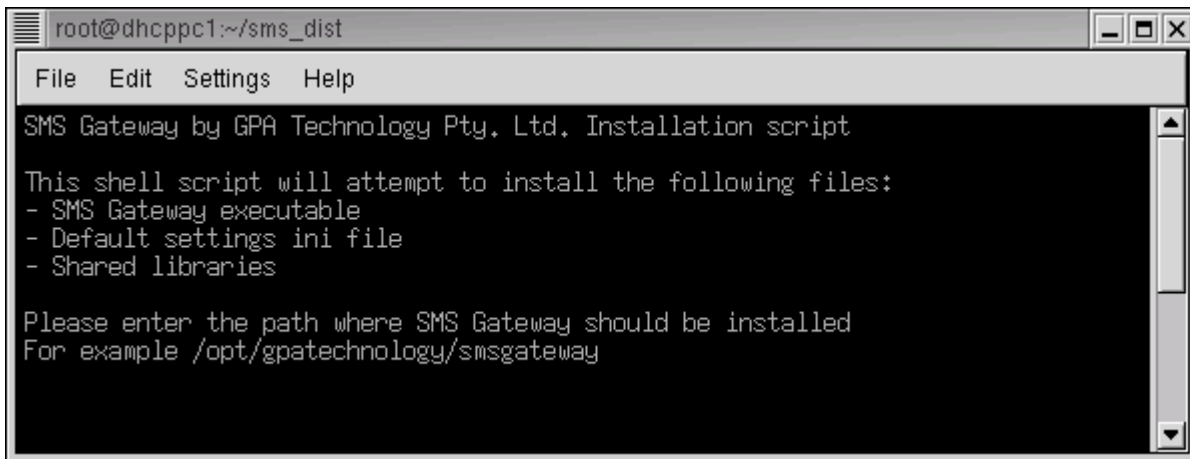


Follow the on screen instructions to complete the installation.

Software Installation – Linux

The SMS Gateway installation process is managed a terminal shell script. To begin the installation run the install script with following command "**sh ./msgwininstall**".

Note: the install script must be run from the **root** account.



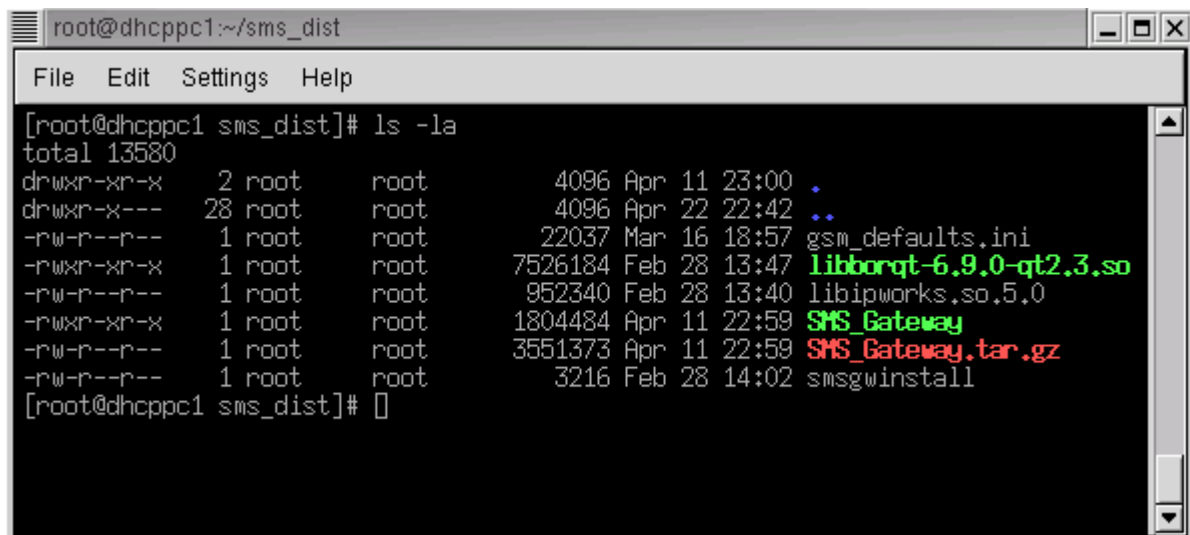
```
root@dhcpc1:~/sms_dist
File Edit Settings Help
SMS Gateway by GPA Technology Pty. Ltd. Installation script

This shell script will attempt to install the following files:
- SMS Gateway executable
- Default settings ini file
- Shared libraries

Please enter the path where SMS Gateway should be installed
For example /opt/gpatechnology/smsgateway
```

Follow the on screen instructions to complete the installation.

Once installed, change into the directory in which you installed the program, you should find the following files:



```
root@dhcpc1:~/sms_dist
File Edit Settings Help
[root@dhcpc1 sms_dist]# ls -la
total 13580
drwxr-xr-x  2 root  root    4096 Apr 11 23:00 .
drwxr-xr-x 28 root  root    4096 Apr 22 22:42 ..
-rw-r--r--  1 root  root   22037 Mar 16 18:57 gsm_defaults.ini
-rwxr-xr-x  1 root  root  7526184 Feb 28 13:47 libborqt-6.9.0-qt2.3.so
-rw-r--r--  1 root  root   952340 Feb 28 13:40 libipworks.so.5.0
-rwxr-xr-x  1 root  root  1804484 Apr 11 22:59 SMS_Gateway
-rw-r--r--  1 root  root  3551373 Apr 11 22:59 SMS_Gateway.tar.gz
-rw-r--r--  1 root  root    3216 Feb 28 14:02 msgwininstall
[root@dhcpc1 sms_dist]#
```

The remainder of this manual shows the Windows version of SMS Gateway, the Linux version is visually and functionally very similar, where there are differences these will be detailed.

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 a GSM network.

The Mobile Terminal connects to the PC using a PCMCIA card, serial cable, Infrared or Bluetooth. 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 a Communications program (E.g. Windows HyperTerminal) to access the COM (serial) 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. A common problem is the connection speed, most devices support either 9600 or 19200 Bps, so try both of these speeds before looking for other problems.

If you cannot get your Mobile Terminal to return an "OK" to you then SMS Gateway will not work.

· Positioning

SMS Gateway relies on the Mobile Terminal having reliable coverage from the GSM Network to which it is subscribed. If 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.

While SMS Gateway will operate with low signal strengths, the message throughput will be negatively affected, as the radio device and network may have to retransmit many times to successfully transfer a message. Under good conditions, a message submission should take approximately 4 seconds.

· Power

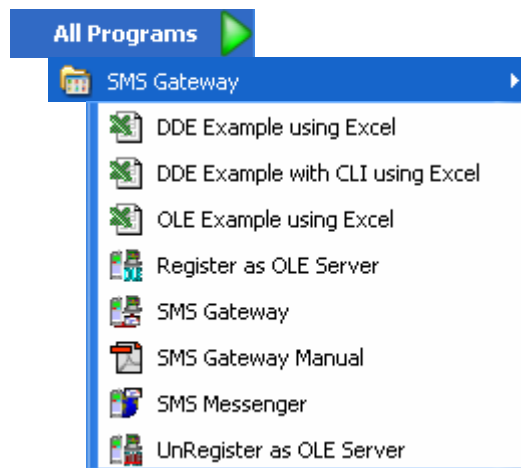
If SMS Gateway will be used as full time messaging gateway for your business, particularly if unattended, it is strongly recommended an Uninterruptible Power Supply (UPS) be used to enable continued operation during power outages.

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.

Launching SMS Gateway - Windows

You will now find three new entries on your Start Menu, these are:



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.

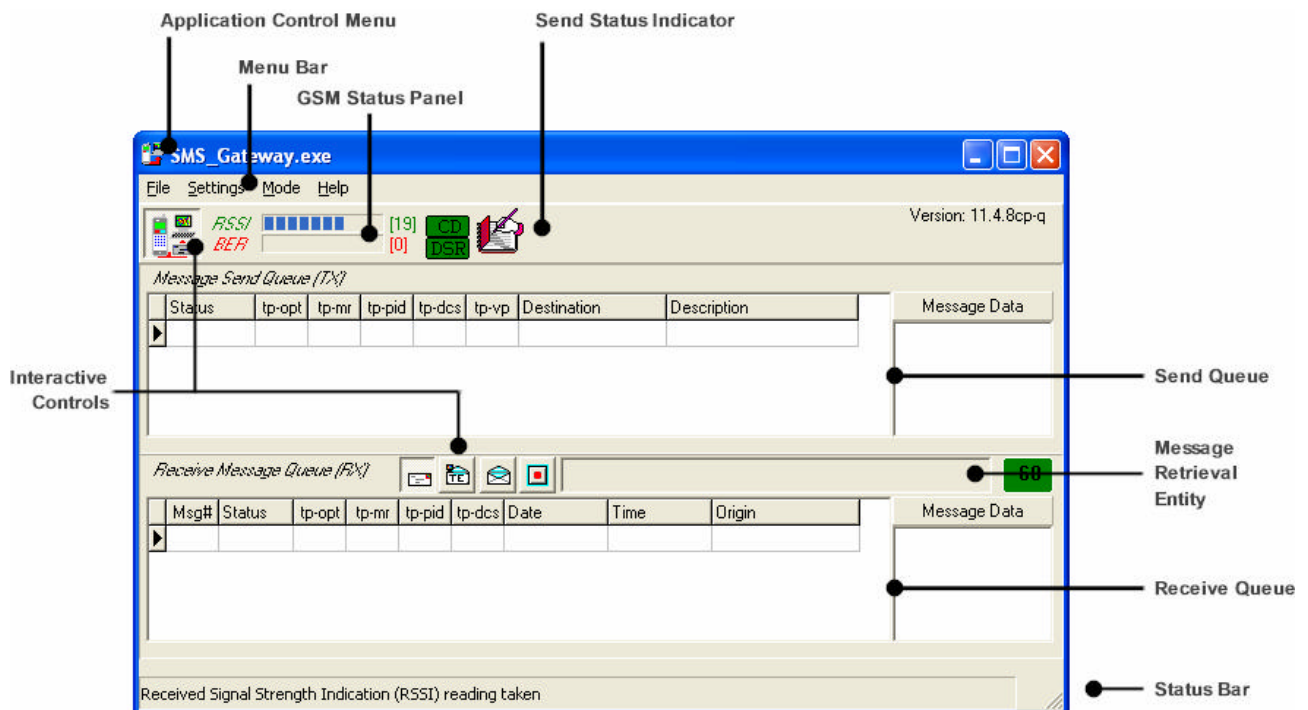
To start SMS Gateway, move the mouse pointer over its run icon and click the left mouse button.

Launching SMS Gateway - Linux

Change to the directory in which you installed SMS Gateway, and run the executable with the following command: **`./SMS_Gateway`**

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 activate 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.

GSM Status Panel



This is a visual indication of the status of the connection between the PC, Mobile Terminal, and the GSM Network.



These bar graphs show the Received Signal Strength Indication (RSSI) and Bit Error Rate (BER) reported by the Mobile Terminal.

The values are detailed below:

<i>Label</i>	<i>Range</i>	<i>Description</i>
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.



These two indicators show the state of the Carrier Detect (CD) and Data Set Ready (DSR) leads on the interface from the Mobile Terminal to the PC. Green indicates the signal is active (on) and red indicates inactive (off).

SMS Gateway instructs the Mobile Terminal to force both of these signals to on during its connection phase with the following commands:

AT&C0	Force CD on
AT&S0	Force DSR on

Some Mobile Terminals do not provide these signals through their serial interfaces, so if either one is off this may not present a problem.

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.

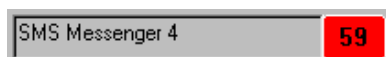


A message has been queued for sending and a request has been issued to the Mobile Terminal to send the message into the GSM network.



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.

Message Retrieval Entity



This field displays the name of the entity currently registered for receipt of incoming messages from SMS Gateway.

SMS Gateway supports message retrieval via OLE⁶, Extended Logging, HTTP(XML) and SMTP, this field displays the following for each retrieval method:

OLE	The name passed by the controlling OLE Automation application
Extended Logging	The directory in which log files of the received messages are stored
HTTP	The URL of the HTTP server to which the received SMS messages will be POSTed in XML format. In this mode, the field reflects the state of the connection to the HTTP server as either being idle, connected
SMTP	The hostname or IP address of the SMTP server to which received SMS messages are forwarded. In this mode, the field reflects the state of the connection to the SMTP server as either being idle, connected.

As incoming SMS messages contain no explicit destination information other than 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 single controlling entity 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 59 at the right hand side of the field is a timer that counts down the seconds since a poll was last received from the registered entity. If a poll has not been received from the entity within 60 seconds, its registration will be automatically revoked (this timeout is configurable from the "Program Options" menu of SMS Gateway).

⁶ Only available under the Windows Operating System

Interactive Controls

These buttons provide quick one-shot access to functions within the SMS Gateway application. As you move the mouse pointer over the 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 buttons is detailed below.



- **Connect to Mobile Terminal.** This button controls the connection and disconnection of the Mobile Terminal from SMS Gateway. When connected, an envelope is displayed between the handset and PC, and the button remains in the down position. If the connection fails for any reason the button will remain up, and you will need to determine the cause of the problem before you can successfully use SMS Gateway. Please refer to the troubleshooting section of this manual for information on resolving connection problems.



A special test facility may be entered by holding down Ctrl + Alt and Right clicking the mouse over this button. In this test mode, a valid connection to a GSM terminal is emulated, and all messages submitted to SMS Gateway are treated as being successfully sent. This facility is useful for testing your API calls without needing a GSM terminal, or sending actual messages.



Do not transfer any incoming messages from the Mobile Terminal



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.



Copy all messages stored in the Mobile Terminal (or SIM) to the SMS Gateway receive queue.

Note that the act of copying messages from the Mobile Terminal does not result in them being deleted from the Mobile Terminal. Messages must be “read” from SMS Gateway (via OLE/HTTP/SMTP or Extended Logging) before the command to delete the messages from the Mobile Terminal will be issued. Alternatively, you can configure SMS Gateway (in the Message Options form) to delete all messages from the Mobile Terminal immediately.

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.

The Send queue is a grid representation of a database that contains the messages to be sent. This "Transmit database" table is located in the "Working Directory" (discussed later) in which SMS Gateway is running, when you place the mouse pointer over the send queue, a popup text box appears detailing actual location of table.

SMS Gateway uses the Dbase file format, which you can access directly via any application supporting Dbase tables while SMS Gateway is not running. While SMS Gateway is running, exclusive locks are placed on the tables, so you cannot access them concurrently with SMS Gateway.

SMS Gateway automatically creates the send queue table when first run, should this table be deleted for any reason, it will be automatically recreated next time the application is run without need for user intervention.

The Send queue shows all the fields in the Transmit database, each of which is detailed below.

Status

The Status field tracks the progress of a message through the sending process, there are four distinct stages, these are

Pending A message is placed in the queue, but no action has been taken to send it

Processing SMS Gateway is preparing the command to pass to the Mobile Terminal requesting a message be sent

Spooled The Mobile Terminal has submitted the message to the GSM network for sending and is waiting a confirmation back

Sent OK The GSM network has accepted the message for sending, and it will be delivered to the destination as soon as possible.

Should a message not be able to be submitted to the GSM network for any reason (ie the Sort Message Service Centre rejected the message), then this field tracks the number of failed attempts, showing "*Fail #1*" then "*Fail #2*", upon the third failure, the message sending will be aborted, and the message will be purged from the Send queue.

-
- **tp-opt, tp-mr, tp-pid, tp-dcs, tp-vp**

These fields are discussed in the “Default Message Attributes” section of this manual.

- **Destination**

The GSM number to which the message is to be sent.

- **Description**

When using the CLI or POP3 interface, a description entry can be associated to the message, this allows the sending application to uniquely tag each message.

The description field is stored in the Extended Log file entries associated with the message. By looking for this identifier, log files can be uniquely matched against messages sent from other applications.

- **Message Data**

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

Once a message has been sent successfully, 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

The Receive queue is a grid representation of a database that contains the received messages. This “Receive database” table is located in the “Working Directory” in which SMS Gateway is running, when you place the mouse pointer over the Receive queue, a popup text box appears detailing actual location of table.

As with the Transmit Database, SMS Gateway automatically creates the receive queue table when first run, should this table be deleted for any reason, it will be automatically recreated next time the application is run without need for user intervention. The Receive queue shows all the fields in the Receive database, 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. If a message has not been stored by the Mobile Terminal, as in the case where messages are directly transferred to SMS Gateway, then this field is always 0.

- **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”.

There is also a special case where a test “Loopback” message (discussed later in the “Testing and Diagnosis” section) has been sent through SMS Gateway. In this case, the field will indicate “Loop”.

-
- **tp-opt, tp-mr, tp-pid, tp-dcs**

These fields are discussed in the "Default Message Attributes" section of this manual.

- ***Date & Time***

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

- ***Origin***

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.

- ***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

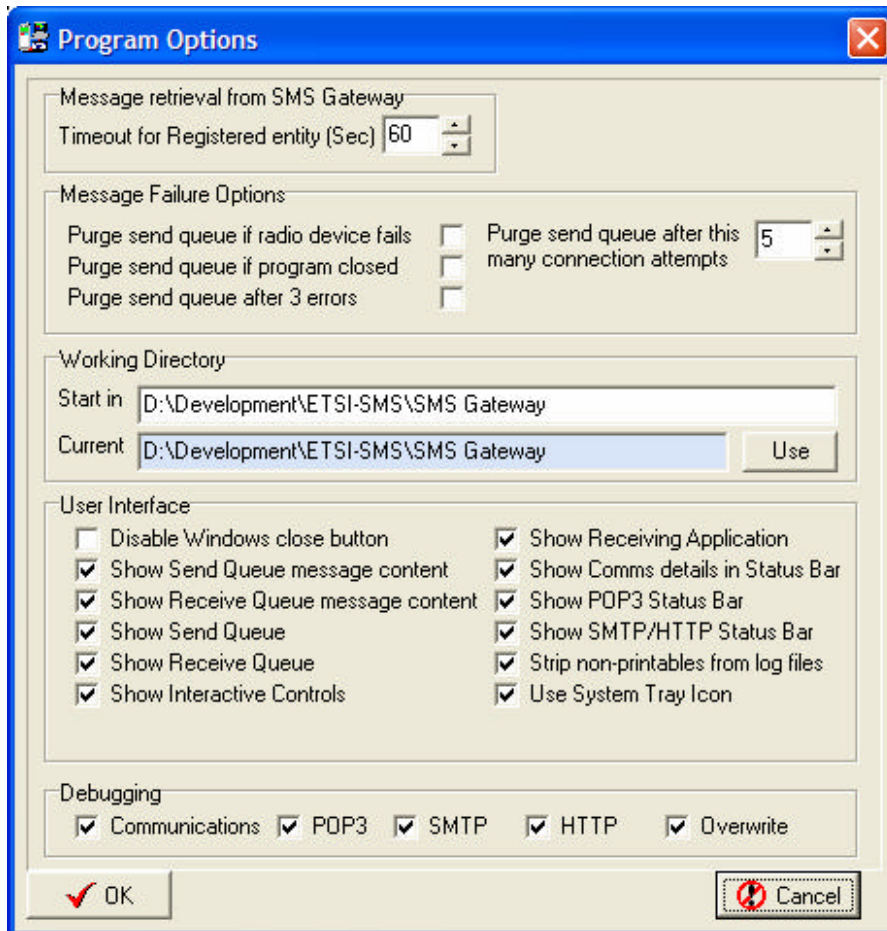
The Status Bar gives you information about what's going on "under the hood" of SMS Gateway. All communications activity between SMS Gateway and the Mobile Terminal is either described or shown verbatim in the Status Bar. The status bar also shows details of POP3 and HTTP/SMTP activity, when these features are activated.

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.

Program Options

From the Menu Bar select **Settings | Program Options** to load the SMS Program Options setup form. This will display the form (shown below) that contains general SMS Gateway settings.



The Program Options Setup Form.

- **Timeout for Registered entity**

SMS Gateway only permits reception of SMS messages by one entity at any time. The receiving entity may be an OLE client, or an internal driver such as SMTP or Extended Logging.

In the case of an OLE client, the client must first register for message reception, then “poll” for new messages regularly, if a poll is not received within this configured timeout period (in seconds), the registration will be revoked, and another entity may be registered.

For SMTP and Extended Logging, the internal drivers will automatically register and then issue polls. Should the polls not be issued, then this is an indication that the process has failed for some reason, and the gateway may need restarting.

- **Message Failure Options**

Under certain conditions you may want all the unsent messages cleared (purged) from the send queue. When the send queue is purged, the unsent messages in it are deleted and written to the Failed log(s), by looking for entries in the failed log(s), you can easily determine when a problem has occurred, and take any required actions.

- **Working Directory**

SMS Gateway uses the Working Directory to store its database tables, configuration files, log files, and as the parent directory for Extended Logging. By default the Working Directory is the directory where the SMS Gateway executable is located.

- **User Interface**

These controls restrict the SMS Gateway user interface. System administrators may wish to use these options when SMS Gateway is installed on individual users PCs.

The “strip non-printables from log files” facility can be useful when performing automated processing of the log files, where the presence of non-printable characters might cause problems. When selected, the only characters that will be written to the log files are; Carriage Return, Line Feed and ASCII values 20 through 128.

Note: The “Use System Tray Icon” item is only available under the Windows O/S

- **Debugging**

SMS Gateway supports comprehensive debugging capabilities for all message exchange functions. Activation of debugging results in a log file being created for the selected function, which can be displayed in the default windows text file editor (usually “Notepad”) by selecting the log under the **File** menu. Also, the debugging logs can be accessed via the HTML Management facility, discussed later in this manual.

For more information on debugging, please refer to the Troubleshooting section of this manual.

Communications Settings

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.

The Communications Setup Form.

Each field within this form is detailed below:

- **Default hardware settings**

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 refer to the Troubleshooting section of this manual, and the FAQ page of our web site.

- **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.

- **Flow Control**

Serial communication links are capable of a variety of methods of controlling the flow of information between two devices such that data is not lost due to data being sent when the receiving device is not ready. As SMS does not result in large amounts of data being transferred between the PC and Mobile Terminal, flow control is generally not required. If your mobile terminal does support flow control, then SMS Gateway can implement a matching system.

- **Retransmission Attempts**

By default, SMS Gateway will attempt to send a command three times without getting the expected result, after which time it will be assumed a communications error has occurred, and SMS Gateway will enter the "Disconnected" state. By lowering this setting, you can shorten the time it takes for SMS Gateway to detect and report a problem, alternatively, by raising it you can increase the tolerance to temporary failure conditions in the Mobile Terminal.

- **Timeout**

This value defines how long (in seconds) SMS Gateway will wait for a response from the Mobile Terminal or GSM network before it retransmits a command. The default value of 45 seconds should be suitable for most circumstances.

- **Transmit Pacing**

Some Mobile Terminals are not able to sustain a high message throughput when sending multiple messages. This value enables you to set a delay (in seconds) between the sending of each queued message.

- **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 string contains "Hayes" commands, which are executed 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, sorted by country 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.

- **Network Interface Type**

There are two variants of the protocol standard used for exchanging Short Messages between SMS Gateway and GSM Mobile Terminals. The choice of protocol required would depend on which mode is implemented by your GSM Mobile Equipment manufacturer. Most GSM Terminals support PDU mode.

- **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 when entering Block Mode, others do not send an OK until Block Mode is exited. In most instances this field should be checked. If required, refer to your equipment documentation for the behaviour of your device.

- **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. NOTE, this is generally not recommended as some GSM terminals perform a reset when the PIN code is entered, this can upset the initialisation sequence used by SMS Gateway.

- 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:

<mode>	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.

In addition to the above <mode> setting, the behaviour of SMS Gateway for each individual message type may also be defined.

<mt>	0	No SMS-DELIVER indications are routed to the TE
	1	Indication of SMS-DELIVER is routed to the TE
	2	SMS_DELIVERs (except class 2 messages) are routed directly to the TE
	3	Class 3 SMS-DELIVERS are routed directly to the TE
<bm>	0	No Cell Broadcast Messages (CBM) are routed to the TE
	1	Indication of new CBMs are routed directly to the TE **
	2	New CBMs are routed directly to the TE **
	3	Class 3 CBMs are routed directly to the TE **
<ds>	0	No SMS-STATUS-REPORTs are routed to the TE
	1	SMS-STATUS-REPORTs are routed directly to the TE**
<bfr>	0	TA buffer of unsolicited result codes defined within this command is flushed to TE when <mode> 1..3 is entered
	1	TA buffer of unsolicited result codes defined within this command is cleared when <mode> 1..3 is entered

** Not supported by SMS Gateway and will be ignored.

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 Adaptor message storage
IM & OM	Proprietary Motorola memory settings

- **Check Received Signal Strength Indicator interval**

Most Mobile Terminals are able to report their Received Signal Strength Indication (RSSI) and Bit Error Rates (BER) via their serial connection, the command to solicit this report is "AT+CSQ".

SMS Gateway can issue this command at regular intervals, and display the results on the main form, and report it in status reports (discussed later).

This command can take a fairly long time on certain Mobile Terminals, which may cause problems with message throughput, in this case it may be desirable to turn this function off (set the field to zero), or make the interval very long.

- **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.

- **Auto ReConnect on loss of DSR**

Checking this field will instruct SMS Gateway to automatically attempt to reconnect to the Mobile Terminal if it detects loss of the Data Set Ready (DSR) from the Mobile Terminal while in a connected state.

- **Auto ReConnect on loss of CD**

Checking this field will instruct SMS Gateway to automatically attempt to reconnect to the Mobile Terminal if it detects loss of the Carrier Detect (CD) from the Mobile Terminal while in a connected state.

Message Settings

From the Menu Bar select **Settings | Message Options** 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.

The Message Settings Setup Form.

- **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.

When using Extended Logging with the “Delete rx’d message once logged” option (discussed later), then you should select ***Transfer all incoming messages*** mode.

- **Message Segmentation**

This facility enables you to send messages greater than the standard 160 characters in length.

Discard excess

Characters after the first 160 are discarded

Multiple messages

The submitted message will be sent across multiple standard length (160 character) messages.

Concatenated message

The submitted message will be sent as a single "Concatenated message". Concatenation is a method defined within the SMS protocol specification that enables messages of up to 39015 (255*153) characters to be sent as a "single" message. The actual message is sent in multiple 153 character segments, and the receiving terminal waits until it receives the last segment before reassembling them into a single message for display to the end user.

Notes:

- SMS Gateway does not reassemble concatenated messages it receives. Message concatenation is only supported for transmitted (sent) messages, not received.
- If your GSM service provider charges you per message sent, you will be charged for each segment of a concatenated message.
- Some Mobile Terminals do not support the sending of concatenated messages via the serial interface. The Ericsson T68 is known not to support this facility.

Max Segments

This field specifies the maximum number of segments you wish to permit SMS Gateway to send when Multiple or Concatenated message segmentation is enabled. Eg. If a message 500 characters long is submitted this would normally result in 4 segments (3 x 160 character segments and 20 characters in the last segment), however you may want only a maximum of 3 segments to be sent for any one message, this will result in the last 20 characters being discarded.

Nokia Length Calculation

When sending concatenated messages SMS Gateway must calculate the length of the each message segment and tell the Mobile Terminal. Nokia handsets require an offset to be added to this value, checking this field tells SMS Gateway to add the required offset (without which the message sending would fail).

- **Parse destination SMS addresses**

By default SMS Gateway looks for any numeric digits in the destination submitted to it, with individual addresses separated by commas. Sometimes you may not want SMS Gateway to interpret all numeric digits as being part of the destination addresses. This facility enables you to explicitly define where SMS Gateway should parse destination addresses from and to within the submitted destination text.

Begin After

Parse the submitted destination only after the specified character.

End Before

Parse the submitted destination only before the specified character.

If you specify both the Begin After and End Before fields, then SMS Gateway will look through the submitted destination text for multiple occurrences of the delimiter pairs. Eg "Destination #1 <+000000000000> Destination #2 <+000000000000> Destination #3 <+000000000002>" will result in all three addresses being parsed.

- **Removal of Messages from MT**

When messages are received by SMS Gateway, they can be either deleted from the Mobile Terminal immediately, or only after they have been retrieved from SMS Gateway via an external application (alternatively, internally via SMTP forwarding or Extended Logging with the Delete once rx'd message logged option).

Delete Immediately

This is the default behaviour. In this mode, SMS Gateway will delete the received message from the storage in the Mobile Terminal (or SIM) immediately. This mode results in the greatest message reception throughput, as there is no danger of the memory in the MT or SIM being filled.

Delete when retrieved

In this mode, the received SMS message is left in the MT or SIM memory until it is removed from the SMS Gateway receive queue by either an external application or internal processing. This mode results in lower message throughput, but does mean that there can be no loss of received messages, even if the SMS Gateway database tables are inadvertently deleted.

- **SMS Message Logging**

Standard Logging

Standard logs are text files which provide a basic record of messages handled by the gateway. The fields within the text files are fixed column width, suitable for importing into utilities such as Microsoft Excel.

Log sent messages

When this setting is selected, SMS Gateway will copy all successfully sent messages to the "transmit.txt" text file in the Working Directory.

Log received messages

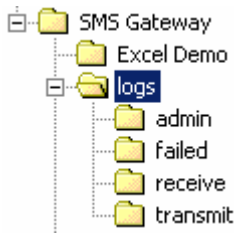
When this setting is selected, SMS Gateway will copy all incoming message to the "receive.txt" text file in the Working Directory.

Log failed messages

When this setting is selected, SMS Gateway will copy all messages that are failed to be sent to the "failed.txt" text file in the Working Directory.

- **Extended Logging**

Extended logging provides a powerful interface to SMS Gateway for message reception and monitoring program status.



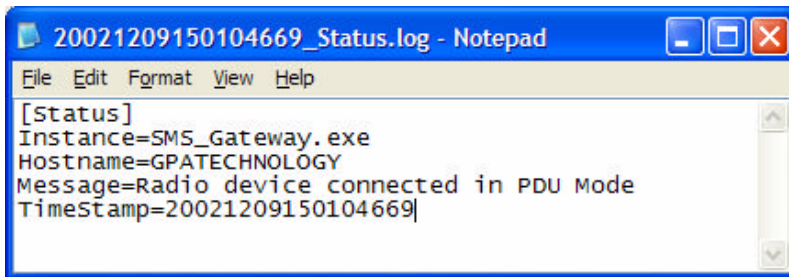
The log files are created under the “logs” directory, which is created in the Working Directory when SMS Gateway is first run with the extended logging option selected.

The extended logs are written in the “.ini” file format, this makes them easy read and access from common programming languages.

The log filenames all begin with a timestamp in the format: `yyyymmddhhnnsszzz`, where; `yyyy` = year, `mm` = month, `dd` = day, `hh` = hour, `nn` = minutes, `ss` = seconds, `zzz` = milliseconds. The timestamp is generated by the PC from its local clock.

There are four types of log file, these are; Admin, Failed, Receive and Transmit, each of these are detailed below.

Admin

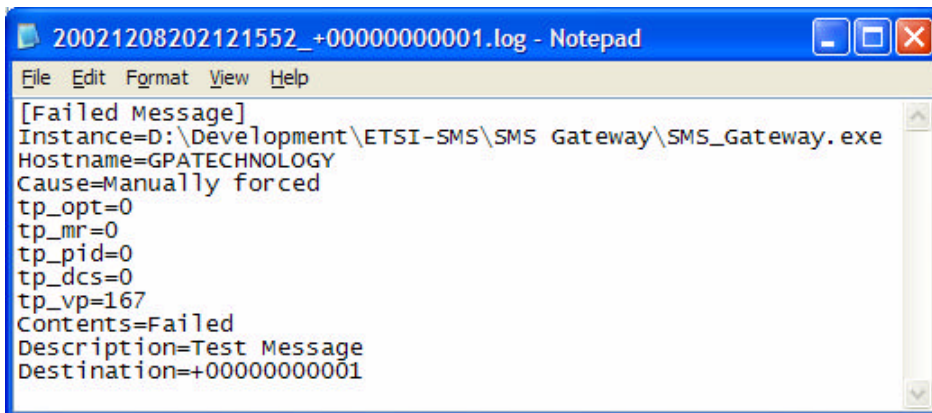


Each time a significant action of event occurs within SMS Gateway, a log file is created that details the event and when it occurred, and example is shown at right.

Events which are recorded to the admin log are;

1. Program startup and shutdown
2. Connection and disconnection from Mobile Terminal
3. Responses to command passed through to the Mobile Terminal

Failed

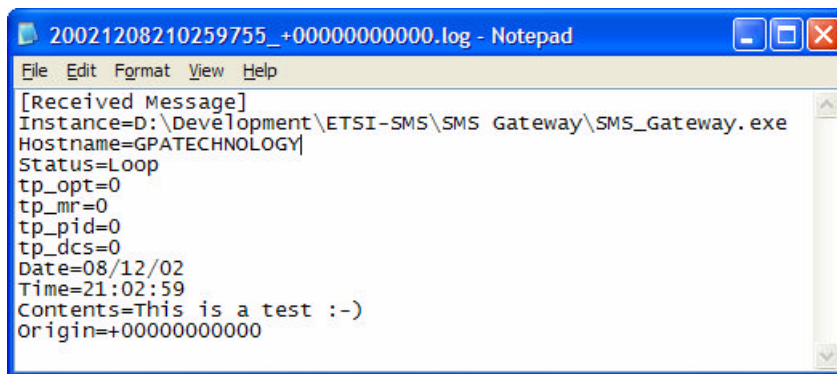


Messages that are failed to be sent are copied to the failed log in the format shown at left.

The filename consists of a timestamp, followed by an underscore then the destination GSM number.

The Cause field details the either the problem response from the Mobile Terminal, or the reason for SMS Gateway discarding the message, eg, purging the send queue.

Receive



```
20021208210259755_+000000000000.log - Notepad
File Edit Format View Help
[Received Message]
Instance=D:\Development\ETSI-SMS\SMS Gateway\SMS_Gateway.exe
Hostname=GPATECHNOLOGY
Status=Loop
tp_opt=0
tp_mr=0
tp_pid=0
tp_dcs=0
Date=08/12/02
Time=21:02:59
Contents=This is a test :-)
Origin=+000000000000
```

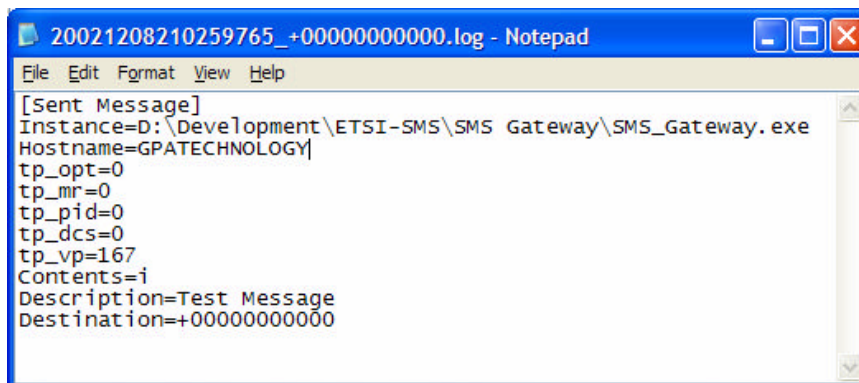
Received messages are written to the Receive log in the format shown at left.

The filename consists of a timestamp, followed by an underscore then the origin GSM number.

Delete rx'd message once logged

If this option is selected, the message will be deleted from the SMS Gateway receive queue once it is written to a log file. This facility can be used as the exclusive means of message reception by SMS Gateway, it is particularly well suited for use with the CLI commands for message sending and reception under program control. The combination of CLI or DDE and this Extended logging, means full two way messaging may be achieved without the need for OLE or SMTP and can be used when running multiple instances.

Transmit



```
20021208210259765_+000000000000.log - Notepad
File Edit Format View Help
[Sent Message]
Instance=D:\Development\ETSI-SMS\SMS Gateway\SMS_Gateway.exe
Hostname=GPATECHNOLOGY
tp_opt=0
tp_mr=0
tp_pid=0
tp_dcs=0
tp_vp=167
Contents=i
Description=Test Message
Destination=+000000000000
```

Messages that are successfully sent written to the Transmit log in the format shown at left.

The filename consists of a timestamp, followed by an underscore then the destination GSM number.

Callback operations

For each of the extended log types, a callback facility exists where each time a log file is written an external program may be called, and the name of the newly written log file passed to it as a command line parameter.

This facility may be used to integrate SMS Gateway into an environment where it needs to interact directly with other programs that do not support OLE, SMTP or HTTP(XML).

To enable callback, simply enter the fully qualified name of the program to be called for the desired log type into the space provided, browse buttons are provided to enable easy selection of the target programs.

Note: SMS Gateway supports the sending of SNMP traps for Failed messages and other events. This is detailed later in this manual.

- **Default Alphabet Character Mappings**

When SMS Gateway sends and receives SMS messages, it needs to convert the character set used on the host PC to/from the GSM Default alphabet defined in ETSI specification 03.38. The ETSI 07.05 specification defines a number of mapping tables for common PC code pages to the 03.38 alphabet, you can load these tables by selecting them from the pull down lists, note that you can use different mappings for transmit and receive.

When you load a mapping table, it copies the mappings to the corresponding “SMS Gateway Default” table, which is used when sending or receiving. Once you have loaded the mapping table you require, you can then manually modify the mappings by pressing the “Edit Active Mappings” button, this will display the form below.

The above form allows you to modify the mappings currently defined in the “SMS Gateway Default” mapping tables. By default the Transmit mapping table is displayed, the receive table can be selected from the pull down list at the top left of the form.

The left hand grid contains 256 cells, each of which represents an individual mapping, when you select a particular cell (by clicking the mouse over it), the input value, and corresponding output value will be displayed at the bottom of the form. To modify a mapping, select the desired input location, select a character from the grid at the right hand side, and then press the “Select” button, you will then see the left hand grid change to the new mapping.

When happy with the mapping table, press the “Save” button to write the entries to the selected SMS Gateway Default table.

Default Message Attributes

From the Menu Bar select **Settings | Default Message Attributes** to load the SMS Gateway Message Attributes form. This will display the form (shown below) that contains information that tells the program how to set the ETSI defined parameters for sent messages.

The Default Message Attributes Form.

- **Options**

- UDHI

- The SMS standard allows for the carriage of higher layer protocols within the SMS message payload. Setting the User Data Header Indicator bit indicates that a higher layer protocol is present, an example of where this facility is used is in the sending of Graphics and Ringtones to handsets, by setting UDHI, the handset looks in the message data and rather than simply displaying the message, it processes it and performs the required actions.

- 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.

- Reject Duplicates

- The setting instructs the SMSC to reject messages it receives from SMS Gateway that have the same Message Reference and Destination Address values.

- **Protocol ID (PID)**

The Protocol Identifier is used to specify what type of message is being transferred. The ETSI SMS standard defines many message types, and particularly interworking with other messaging systems such as email and fax, gateways to which may be provided within the GSM network by the operator.

There are many PID settings, as you use the up and down buttons to move through the various values a brief descriptor of the current value will be displayed in the centre box.

A full description of each value can be found in the ETSI 03.40 specification.

- **Message Reference (MR)**

The Message Reference serves as an identifier for each message and is used by the SMSC for reporting such things as Delivery Confirmation. SMS Gateway allows you to set MR, but does not act upon it.

- **Validity Period (VP)**

The Validity Period defines how long the SMSC will hold the message prior to delivery to the destination party. If the message cannot be delivered prior to expiration of the VP then the message will be discarded by the SMSC.

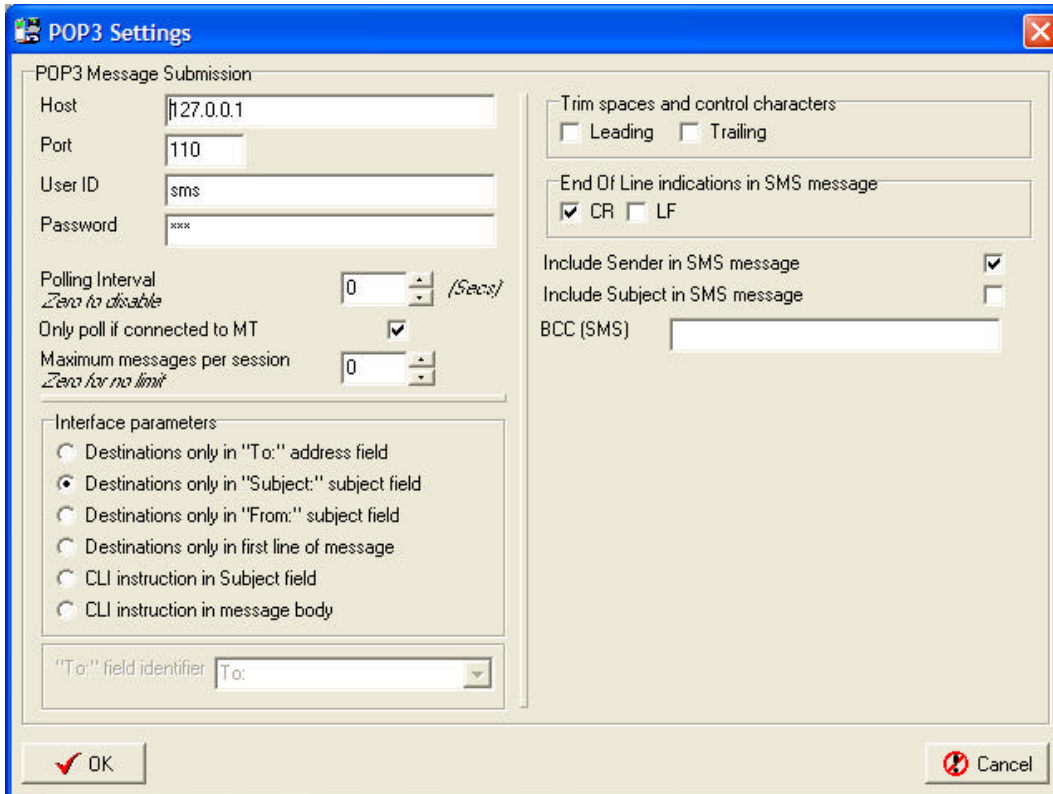
- **Data Coding Scheme (DCS)**

The Data Coding Scheme is used to access special SMS attributes and capabilities of handsets. There are many DCS settings, as you use the up and down buttons to move through the various values the corresponding settings are selected in the various DCS sub-groups. You may also select the desired DCS settings with the mouse, and the DCS numeric value will be updated to reflect it.

A full description of each value can be found in the ETSI 03.40 specification.

POP3 Settings

From the Menu Bar select **Settings** | **POP3 Settings** to load the SMS Gateway "Post Office Protocol 3" (POP3) form. This will display the form (shown below) that contains information that tells SMS Gateway how messages received via email will be forwarded via SMS.



The POP3 Settings Form.

- **POP3 Message Submission**

SMS Gateway is able to "Poll" an email mailbox via POP3, and collect all messages in it for forwarding via SMS.

To activate POP3 Submission, set your Host, Port, User ID, and Password as defined by your email administrator, then set the Polling Interval to a value greater than zero. The Polling interval defines how often SMS Gateway will check the mailbox for any messages to be sent, it is best not to set this value too low as this will place unnecessary load on SMS Gateway and the email server, a value around 10-20 seconds should be fine.

By default, each time SMS Gateway connects to the nominated mailbox it will retrieve ALL messages in the mailbox, and enter them into the SMS Gateway send queue, as per the settings defined in the "Interface Parameters", which are discussed below. You may use the "Maximum messages per session" entry to limit how many messages SMS Gateway downloads from the POP3 server for each connection.

Only poll if connected to MT

It may be desirable to only have SMS Gateway retrieve messages from the POP3 mailbox when it is actually connected to the Mobile Terminal, this will avoid large quantities of messages building up in the send queue when there is a problem with the Mobile Terminal.

Interface Parameters

When submitting messages to be sent via this POP3 method, it is necessary to tell SMS Gateway where to find parameters it needs such as the destination address, and optionally set advanced attributes such as DCS and PID (discussed earlier).

As a minimum, the destination addresses need to be supplied, and these can be placed in any one of the; To, Subject, From, and First line of message fields. SMS Gateway will parse out the destination addresses from standard email formats, eg, if the supplied information is 0419000000@sms.yourdomain.com, 0419000001@sms.yourdomain.com , the "@sms.yourdomain.com" part of the addresses will be removed, and just the GSM numbers be used for message submission. The first four settings simply tell SMS Gateway where to look for destinations in this manner.

If you have specified any delimiters in the "Parse destination SMS addresses" (discussed earlier), these will be also be applied to the submitted destination information.

The last two options leverage the powerful Command Line Interface of SMS Gateway. By selecting one of these options, SMS Gateway will take the entire nominated field (Subject or First line of message body) and pass this to the same handler used by the CLI API, the syntax of which is discussed later in this manual. You can send any CLI instruction (ie sending messages or issuing commands to SMS Gateway) via POP3, please refer to the CLI section of this manual for detail of what can be achieved.

"To:" field identifier

Sometimes it may be desirable to use a field other than the standard "To:" as the location from which to parse destination addresses. This field enables you to choose from a pull down list of commonly used identifiers, or you can enter any string here as required by your local environment.

Trim spaces and control characters

This facility enables you to remove any leading and trailing spaces or non-printable characters (eg. CR and LF) from the POP3 message body.

End Of Line indications in SMS message

As the number of characters in an SMS message is very limited, it may be desirable to either remove or reduce the presence of Carriage Return (CR) and Line Feed (LF) characters. Here, these characters can be either removed, or limited to just one or the other (rather than both, which is the default in email).

Include Sender in SMS Message

If desired, the email address of the sender of the original message may be included at the beginning of the SMS message, check this field to do so.

Include Subject in SMS Message

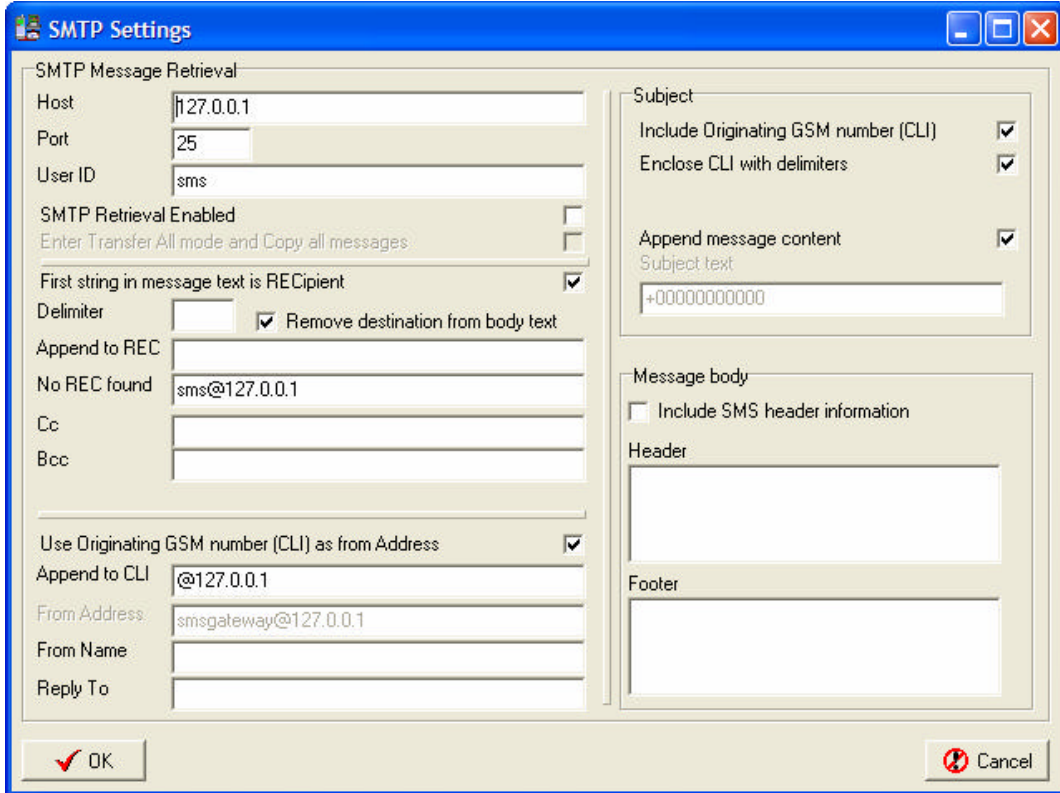
If desired, the Subject line of the original message may be included at the beginning of the SMS message, check this field to do so.

BCC (SMS)

SMS Gateway can optionally send an additional copy of each SMS message submitted via POP3 to a pre-defined address. The initial recipient of the SMS message is not made aware that a copy of the message has been Blind Carbon Copied (BCCd) to another recipient.

SMTP Settings

From the Menu Bar select **Settings** | **SMTP Settings** to load the SMS Gateway “Simple Message Transfer Protocol” (SMTP) form. This will display the form (shown below) that contains information that tells SMS Gateway how messages received via SMS will be forwarded via Email.



The SMTP Settings Form.

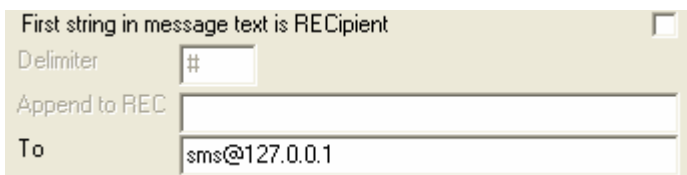
SMTP Message Retrieval

SMS Gateway is able to forward received SMS messages via SMTP to external mailboxes. To activate SMTP message forwarding, set your Host, Port, and User ID, as defined by your email administrator, then check the SMTP Retrieval Enabled box.

When using SMTP message retrieval, it may be desirable to automatically enter Transfer All mode, and Copy all messages currently stored in the Mobile Terminal when first connected, this will enable SMS Gateway to operate in a “Hands-off” mode, without need to any user intervention at program startup.

When messages are in the SMS Gateway receive queue, an SMTP connection will be made to the nominated host, and all messages forwarded as per the settings which are discussed next. Once all messages are forwarded they will be deleted from the SMS Gateway receive queue and the SMTP connection closed down.

When forwarding messages via SMTP, SMS Gateway needs to know what email address each message should be forwarded to. There are two options available; either send all messages to the same address, or to parse the first line of the SMS message for a destination email address.



The simplest option is to forward all messages to the same address. To do this, enter the desired email address in the “To” field.

First string in message text is RECIPIENT	<input checked="" type="checkbox"/>
Delimiter	#
Append to REC	
No REC found	sms@127.0.0.1

A more advanced and flexible option is to use the first string (ie up to the first CR or LF) in the SMS message as an email address.

When using this option, the sender should append a delimiter symbol to the email address, this tells SMS Gateway where the end of the address is, eg, "winsms@127.0.0.1#" or "winsms#". In the latter case, a domain name can be appended by filling in the "Append to REC" field. If no "#" symbol is found in the first line of the SMS message, then the "No REC found" entry will be used as the destination address, this enables incorrectly formatted messages to be captured and possibly forwarded to a help desk so the sender may be notified. The default delimiter symbol is "#", but may be changed to any character through the "Delimiter" field.

Cc	
Bcc	

The Cc and Bcc fields allow you to copy the incoming message to other recipients, these act as per standard email behaviour.

The From address used when forwarding SMS messages can either be the Calling Line Identification (CLI) field of the incoming SMS message, or can be statically defined.

Use Originating GSM number (CLI) as from Address	<input checked="" type="checkbox"/>
Append to CLI	@127.0.0.1
From Address	msggateway@127.0.0.1

By checking the "Use Originating GSM number (CLI) as from Address" field, the CLI will be placed in the email From address, and you can optionally append a domain name if required.

Use Originating GSM number (CLI) as from Address	<input type="checkbox"/>
Append to CLI	@127.0.0.1
From Address	msggateway@127.0.0.1

Alternatively, the From address can be set to always be the same.

From Name	
Reply To	

The From and Reply To fields act as per standard email behaviour.

Subject Field

The Subject field may be set to either use the SMS message content as the email subject, or it may be statically defined. Also, you may prepend the subject with the originating GSM number and optionally enclose the originating number with the destination parsing delimiters specified in the Message Options form.

By prepending the originating GSM number to the Subject field, this provides an easy way to enable email recipients to Reply to an SMS message and have the replay find its way back to the original sender. To achieve this, you will need to also set the POP3 submission facility to use the Subject field as the source from which to parse destination addresses.

HTTP Settings

From the Menu Bar select **Settings** | **HTTP Settings** to load the SMS Gateway “Hyper Text Transfer Protocol” (HTTP) form.

The HTTP Settings Form.

HTTP Interface

SMS Gateway supports a number of functions via a HTTP based interface, each of which is discussed in the following sections.

The first step in configuring the HTTP interface is to set the local Host and Port addresses. The Host address is the IP address of the local interface that you wish the HTTP services to bind to. The Port number is the TCP Port that the inbuilt HTTP server will use. When setting the HTTP Port, be sure to select a Port that is not already in use on the machine, the default Port is 8080, which is generally unused. Please consult your local Network Administrator for the correct Host and Port addresses for your environment.

For both XML POST and Interactive HTML Submission, you will need to specify the IP Addresses that are able to access the facility in the “Allowed IP Addresses” field. This field accepts the “*” wildcard, for example to allow access for the entire RFC 1918 Class C subnet, enter the address in the form 192.168.0.*. To allow access from all addresses, just enter a single “*” character. It is also possible to explicitly block access from IP Addresses in the same manner in the “Blocked IP Addresses” field.

ALLOW XML POST SUBMISSION

SMS Gateway supports the submission of SMS Messages for sending via SMS through an XML formatted HTTP based interface in the following format:

```
<!-- Test of the XML Interface -->
<msggateway>
  <cli cid="0">
    <command> -send -dest +000000000000 -msg "Hello World" </command>
```

```
    <command> -send -dest +000000000000 -msg "Second Message" </command>
  </cli>
</msggateway>
```

The <cli> element encapsulates a set of SMS Gateway Command Line Interface instructions, each bounded by the <command> identifier. The format of the instructions is defined in the Command Line Interface section of the “Integrators & Developers Reference” section of this manual.

When sending the POST, you must specify the following content type:

```
<?xml version="1.0" encoding="ISO-8859-1"?">
```

Special Characters

As detailed in the Command Line Interface section, the CLI interface requires special characters to be replaced as per below:

<CR> Carriage Return <LF> Line Feed <DQ> Double Quote (“)

The XML specification requires the use of escape characters as per below;

```
>      &gt;
<      &lt;
&      &amp;
"      &quot;
'      &apos;
```

Therefore, when submitting CLI commands via XML, you must obey the XML Escape requirements, an example of which is below;

```
<command> -send -dest +000000000000 -msg "Hello World &lt;CR&gt; Line 1
&lt;CR&gt; Line 2" </command>
```

Reply to XML POST

The “cid=” attribute of the <cli> element is used to provide a unique numeric identifier for the submission, SMS Gateway will include this identifier in its acknowledgement reply to the HTTP POST.

Once enabled, POST XML submissions to the following URL: <http://hostipaddress:port/xml>

The reply message from SMS Gateway is in the form:

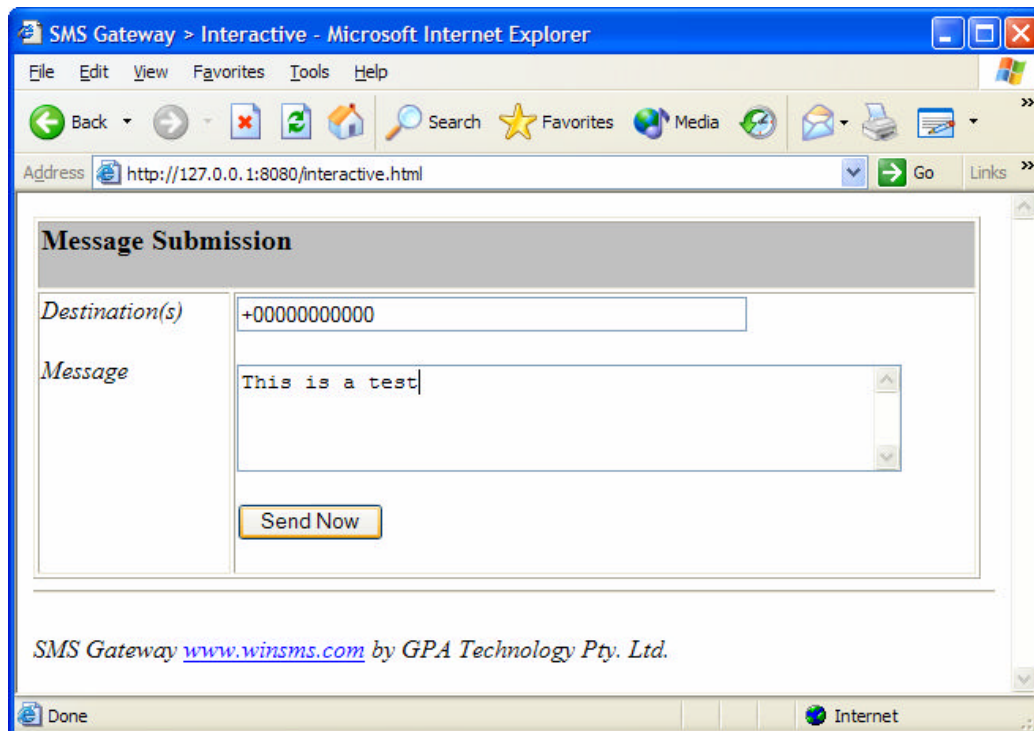
```
<!-- SMS Gateway XML Interface -->
<msggateway instance="SMS_Gateway.exe">
  <cli cid="0">
    <ack status="ok"></ack>
  </cli>
</msggateway>
```

The <ack> element indicated the success or otherwise of the POST submission. If the submission was successful, a status=“ok” attribute will be returned, otherwise a status=“failed” will be returned.

ALLOW INTERACTIVE HTML SUBMISSION

SMS Gateway supports a limited ability to submit messages for sending via SMS using a HTML web form. This interface is only intended for lightweight testing purposes, for high volume applications please use the XML POST facility.

Once enabled, point your browser to the following URL: <http://hostipaddress:port/interactive.html> , and you should be presented with the web page shown below.



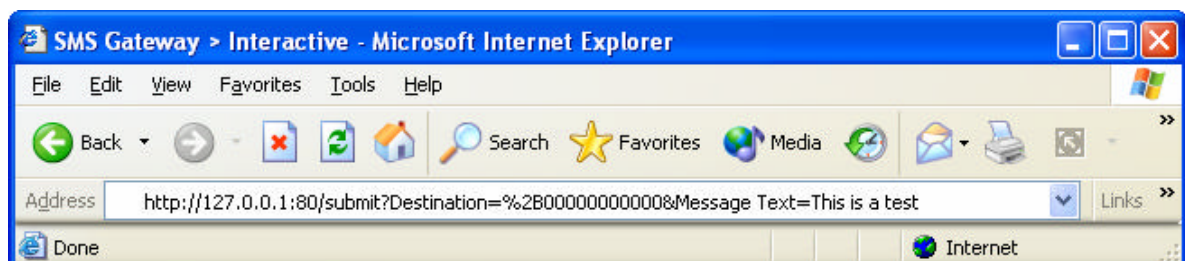
Additionally, when Interactive HTML Submission is enabled, you may submit messages for sending directly within a URL, in either a simple Destination & Message Text format, or a more advanced CLI formatted instruction, each of which is discussed below:

Simple Destination & Message Text within HTTP URL

In this mode you submit a message in the following format:

<http://hostipaddress:port/submit?Destination=xxxxxxxxxxx&Message Text=xxxxxxxxxxl>

An example submission is below:



You will note in the above example that the "+" character has been replaced by "%2B", this is because certain characters have a special meaning inside URLs. The following table shows the characters that must be substituted when submitting messages in URLs:

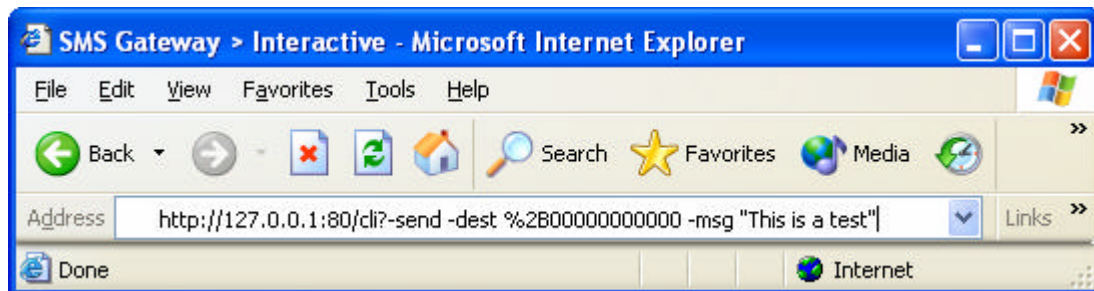
Character	Substitute with
&	%26
%	%25
\	%5C
"	%22
'	%27
#	%23
*	%2A
!	%21
<	%3C
>	%3E
,	%2C
+	%2B
Space	%20

CLI formatted instruction URL

In this mode you submit a message in the following format:

<http://hostipaddress:port/cli?xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

The actual CLI syntax is detailed later in this manual. An example CLI based submission is below:



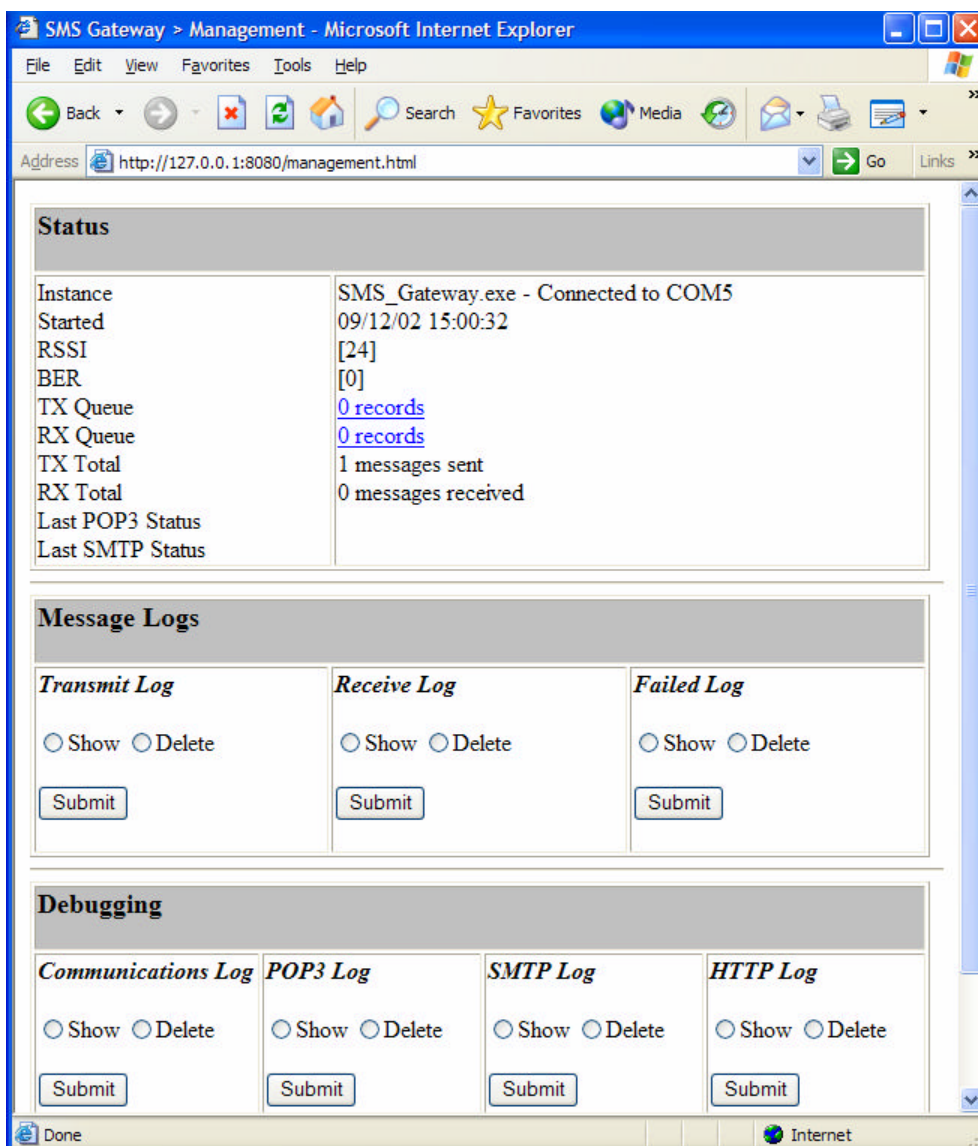
MANAGEMENT

SMS Gateway includes a web (HTML) diagnostic facility that enables viewing of the current status of SMS Gateway, inspection and clearing of log files, and viewing of the message queues.

To activate this facility, check the “Enable HTML Management” field, and enter a Username and Password which will be used to authenticate all access to the facility.

Also, you will need to specify the IP Addresses that are able to access the facility in the “Allowed IP Addresses” field and press the + button (or press enter). This field accepts the “*” wildcard, for example to allow access for the entire RFC 1918 Class C subnet, enter the address in the form 192.168.0.*. To allow access from all addresses, just enter a single “*” character.

Once enabled, point your browser to the following URL: <http://hostipaddress:port/management.html>, enter the defined Username and Password, and you should be presented with the web page shown below.



HTML
Management
web page

MESSAGE RECEPTION

SMS Gateway is able to forward received SMS messages in XML format via HTTP. To activate HTTP(XML) message forwarding, select the "Enable XML POST forwarding" checkbox, configure a suitable interval (this defines how often SMS Gateway will attempt to forward the contents of the Receive Queue to the HTTP server) and set a reasonable Timeout (this is how long SMS Gateway will wait for a reply from the HTTP server before clearing the connection and trying again next interval).

The URL field specifies the full path to the server process that will service the POST from SMS Gateway. If the server requires authentication, you can enter the Username and Password in the configuration form.

Messages are forwarded in the following format:

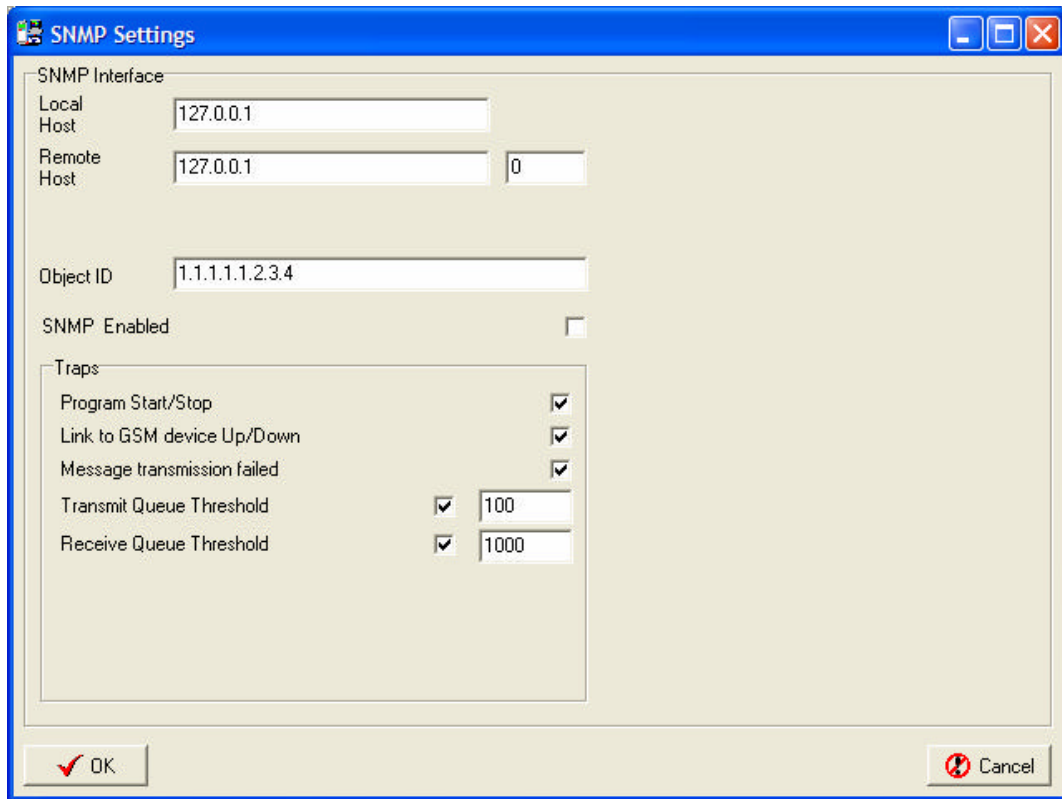
```
<!-- SMS Gateway XML Interface -->
<msggateway instance="SMS_Gateway.exe">
  <message>
    <deliver>
      <status>Loop</status>
      <opt>0</opt>
      <dc>0</dc>
      <mr>0</mr>
      <pid>0</pid>
      <oa><number>+000000000000</number></oa>
      <scts><date>09/12/02</date></scts>
      <scts><time>15:31:41</time></scts>
      <ud>This is a test</ud>
    </deliver>
  </message>
</msggateway>
```

The definitions of the individual fields are consistent with those detailed elsewhere in this manual.

Note: The User Data element will conform to XML Escape requirements for special characters as previously detailed

SNMP Settings

From the Menu Bar select **Settings** | **SNMP Settings** to load the SMS Gateway “Simple Network Management Protocol” (SNMP) form.



The SNMP Settings Form.

SNMP Interface

SMS Gateway supports the sending of SNMP Traps for significant events.

The first step in configuring the SNMP interface is to set the local Host address. The Host address is the IP address of the local interface that you wish the SNMP service to bind to. Please consult your local Network Administrator for the correct Host address for your environment.

The Remote Host field is the address (and optionally port) of the SNMP Element manager. If you leave the Remote Port as Zero, SMS Gateway will automatically use the standard SNMP port numbers.

The Object ID specifies a unique identifier for the SMS Gateway application. The default ObjectID is a suggestion only, you may change this to any value you desire.

To activate the SNMP agent, check the “SNMP Enabled” checkbox.

Individual traps may be enabled in the “Traps” section, the resulting Trap values are detailed below:

- **Program Start/Stop**

Start	=	ObjectID-0.1
Stop	=	ObjectID-0.2

- ***Link to GSM device Up/Down***

Link Up = ObjectID-3.1
Link Down = ObjectID-2.1

- ***Message transmission failed***

Failure = ObjectID-6.1

- ***Transmit Queue Threshold***

Threshold reached = ObjectID-6.2

- ***Receive Queue Threshold***

Threshold reached = ObjectID-6.3

Testing your SMS Gateway configuration

Once SMS Gateway has been configured, you need to verify that a connection can be established between SMS Gateway



Press the “Connect to MT” button. SMS Gateway will now attempt to establish a connection to your GSM device. You will see in the Status Bar a series of “AT” commands being exchanged with the Mobile Terminal. If this exchange is successful, the button will stay down



If the exchange fails, please refer to the “Testing and Diagnosis” section of this manual.

Once connected, the next step is to send a test message. Select **Help | Send Test Message** from the SMS Gateway menu.

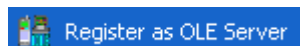
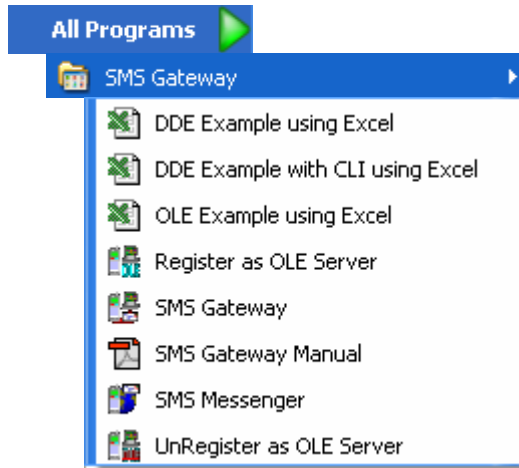
Enter a destination number (your GSM number) and some message text.

Now press “Send”, the form will close and you will see your message in the “Message Send Queue [TX]”. If the message is sent successfully, it will be removed from the queue and the message will be received at the destination shortly thereafter.

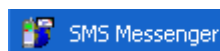
If the message sending fails, first check that you have the correct SMSC address configured as this is the most common problem. If you are sure you have the correct SMSC address, you will need to capture the problem in a Communications Log and send it to as described in the “Testing and Diagnosis” section of this manual.

First Time Configuration of SMS Messenger

SMS Messenger⁷ is a Windows OLE demonstration application for use with SMS Gateway. It provides a friendly user interface for the transmission and reception SMS messages using SMS Gateway and is supplied with full source code. The icon for SMS Messenger is located in the SMS Gateway folder on your Start Menu.



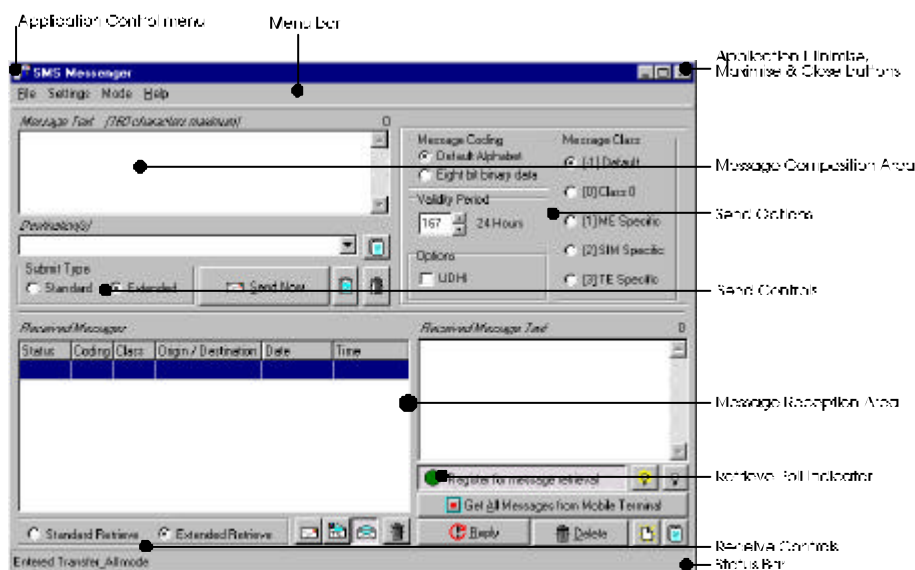
Before you can use SMS Messenger, you must first register SMS Gateway as an OLE Automation server. To do this click the left mouse button over the "Register as OLE Server" icon, you will receive no visual feedback that this action has been performed successfully, but you will receive an error message 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:



⁷ Only available under the Windows Operating System

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.

- *Options - UDHI*

The SMS standards allow for carriage of other protocols within SMS Messages, such as the upcoming Wireless Application Protocol (WAP). To tell the receiving system that a message contains another protocol inside it, you need to set this field. If you do set this field, make sure the data send does infact include correctly formatted headers for another protocol, otherwise the message sent will be invalid, and cause unpredictable results.

-
- **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 not 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.

- **Show and Hide SMS Gateway**



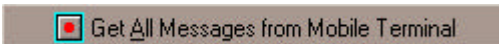
Show SMS Gateway



Hide SMS Gateway

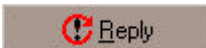
The buttons allow you to control the visibility of the SMS Gateway module from SMS Messenger. Under normal circumstances SMS Gateway is hidden in the Windows System Tray, you may want to show it to view the progress of message submissions.

- **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**



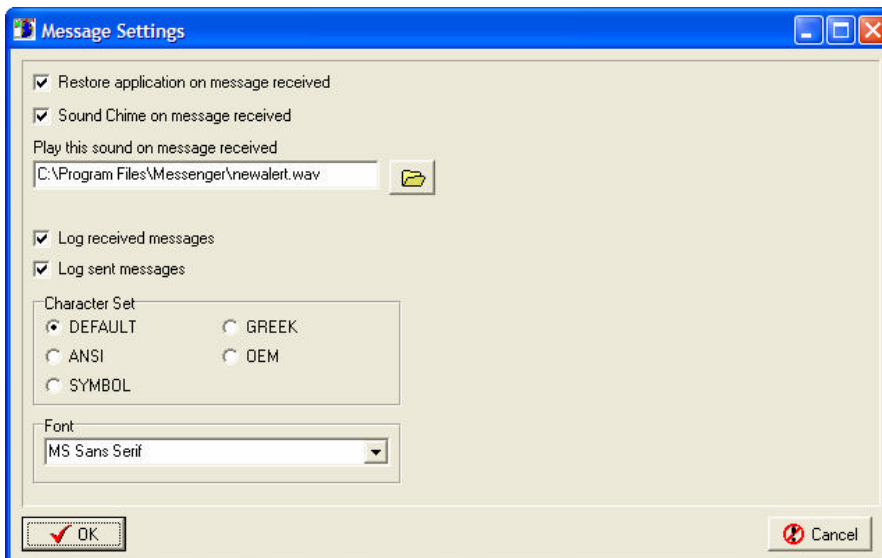
If logging is enabled (discussed later), copies of all messages appearing in the Received message grid are stored on your PC in the Receive Log file. This log file may be view and edited by pressing the small button with the notepad icon on it at the bottom right hand side of the message reception area.

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.



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. You can select a sound file to be played when a new message is received.

- **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.

Character Set

This identifies the Windows character set of the font to be used in the Send and Receive Message Text fields. Each font supports one or more character sets. Check the information supplied by the font vendor to determine which of the listed Character Sets are valid. The following table lists the available character sets:

Default:	Font is chosen based solely on Name and Size. If the described font is not available on the system, Windows will substitute another font.
ANSI:	ANSI characters.
Symbol:	Standard symbol set.
Greek:	Greek characters. Not available on NT 3.51.
OEM:	Depends on the codepage of the operating system.

Font

Use this to specify the typeface of the font to be used in the Send and Receive Message Text fields. If the font family chosen includes multiple character sets, be sure to set the Character Set property as well. If the combination of Character Set and Font specifies a font that is not available on the system, Windows substitutes a different font.

Start your engines!

Now that you are familiar with SMS Gateway 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 - Windows

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 **Send 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, it should arrive in back in your Mobile Terminal. We will now look at receiving this message.

Sending a message - Linux

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 Gateway application
2. Press the “Connect to MT Button”, and wait for the connection sequence to end
3. Select **Help | Send Test Message**
4. Press the **Loop** button.
5. Enter some text into the “Message” area, then press **Send**
6. If everything is working properly, 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, it 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 “Transfer all incoming messages” button. If successful the button will remain down
2. Press the “Copy All Messages from M.T.” button. A few moments later you message should appear in the Received message queue.

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

Conclusion

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

If you experience trouble connecting to your Mobile Terminal, or sending messages, please refer to the next section entitled “Testing and Diagnosis.

Testing and Diagnosis

SMS Gateway includes a comprehensive set of tools for monitoring and debugging, each of which are detailed below.

Communications between the PC and GSM Mobile Terminal

If SMS Gateway is having trouble communication with the Mobile Terminal (ie. Cannot connect to it, unable to send or receive messages, etc), then the first step in resolving the problem is to enable Debugging of Communications, you can set this in the **Settings | Program Options** form.

With communications debugging now enabled, press the "Connect to MT" button. SMS Gateway will now attempt to initialise a connection to the GSM Mobile Terminal. You will see the commands sent to the terminal (and its responses if any) in the status bar at the bottom of the form.

If the connection is successful you will see a number of commands being issued by SMS Gateway, and the Mobile Terminal will respond to each command. At the end of this exchange the "Connect to MT" button should remain down, and the light globe symbol be lit, the whole process should take between 30 and 60 seconds. If this exchange fails, display the communications log from the **File** menu, and look for where commands sent by SMS Gateway are either not replied to, or an "ERROR" message is received.

If no reply is received from the Mobile Terminal, then it is likely that with the COM port or Speed settings selected in the Communication Setup form are incorrect, check what these should be and ensure they are configured correctly.

If an "ERROR" message from the Mobile Terminal is received as the last entry in the communications logfile, then you may need to send it through to us to check what the problem may be. Note: it is normal that during the early stages of the initialisation sequence some "ERROR" messages may be received, as some of the commands issued to the Mobile Terminal (such as Get Manufacturer Name) are not supported by all devices, this is no cause for concern.

Test destination numbers

Special destination SMS addresses may be used for test purposes, these are;

- +000000000000 Message will pass directly from the TX to RX queue
- +000000000001 Message will be treated as a Failed to be sent message
- +000000000002 Message will trigger an Administrative log file to be written
- +000000000005 ⇒ 9 Message will be ignored, no logging performed

Communications between SMS Gateway and email servers

If SMS Gateway is unable to retrieve messages for sending via SMS from an email server, or if it cannot forward received messages via SMTP, then the procedure is similar to that of communications, first enable debugging for wither SMTP or POP3 in the **Settings | Program Options** form, then recreate the problem scenario, and then display the appropriate log file from the **File** menu.

The log file will contain details of all exchanges between SMS Gateway and the email server, in most cases this information should be sufficient for an email administrator to diagnose any problems and recommend a solution.

If all else fails

If after trying all the steps suggested here you are still experiencing problems, please send an e-mail to support@winsms.com and we will endeavour to assist in solving the problem.

In the email, please include as much detail as possible, describing the scenario which is causing problems, and always send a copy of the “Communications Log” as discussed earlier. We endeavour to respond to most support requests within 48 hours.

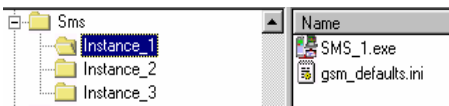
Running Multiple Instances of SMS Gateway

When using SMS Gateway under the control of OLE, it is not possible to launch more than one instance of SMS Gateway. Through the use of the DDE⁸, Email, HTTP, or Command Line Interfaces however, it is possible to support multiple instances of SMS Gateway on a single machine, this may be required where you need to send messages to terminals on more than one GSM network, or increase the overall throughput of your system.

To enable multiple instances, you need to create a directory structure that contains uniquely named instances of SMS Gateway in a number of discrete directories, eg:



Into each directory, copy the "SMS_Gateway.exe" executable file, and rename it to a unique name, but be sure to keep the .exe extension. Also copy into this directory the "gsm_defaults.ini" file, but do not rename it.



You should now have a directory structure that looks something like this.

Each instance requires its own GSM device, the first time you run each instance you will need to set the communications parameters to use a unique GSM device. This will create an ini file containing the appropriate settings for this instance in the currently active directory.

Now you may use the DDE, Email HTTP and CLI interfaces to directly target each instance of SMS Gateway.

⁸ Only available under the Windows Operating System

Integrators & Developers Reference

This section details how to integrate SMS Gateway into your environment, and also interworking between SMS Gateway and other Windows applications. It is assumed the reader has a reasonable understanding of Windows and general communications concepts

For the programmatic interfaces, it is assumed the reader is familiar with DDE and OLE Automation and passing parameters via the Command Line Interface.

The Command Line Interface

The CLI provides a simple way to integrate SMS Gateway with any environment without any special programming knowledge. The CLI format can also be used with the DDE, POP3 and HTTP interfaces (discussed previously).

A CLI command is issued in the following format:

```
sms_gateway.exe -switch1 param1 -switch2 param2
```

In the above example, "sms_gateway.exe" is the name of the SMS Gateway executable, and the subsequent commands to be processed. When using multiple instances of SMS Gateway, the executable name may be changed to target the desired running instance.

Note: Under Linux, the default executable name is "SMS_Gateway"

Any number of commands may be issued in a single call, subject to operating system limits on the maximum length of a command line. When using the CLI format with the DDE, POP3 or HTTP interfaces, it is not necessary to include the "sms_gateway.exe" component.

The CLI is broken into two sections, these are Control and Message Submission, each of which is detailed below.

Control commands

Command	Set Working Directory
Switch	-wd
Param name	Working Directory
Param type	String
Example	<code>sms_gateway.exe -wd "c:\temp"</code>
Command	Load SMS Gateway settings
Switch	-read_ini
Param name	Location of settings "ini" file
Param type	String
Example	<code>sms_gateway.exe -read_ini "c:\sms\test.ini"</code>
Command	Send gateway status via SMS
Switch	-sms_status
Param name	SMS destination to send status message to
Param type	String
Example	<code>sms_gateway.exe -sms_status "0419000000"</code>
Command	Send gateway status via Email
Switch	-email_status
Param name	Email destination to send status message to
Param type	String

Example `sms_gateway.exe -email_status "status@domain.com"`

Command	Send Communications Log via Email
Switch	-email_comms_log
Param name	Email destination to send communications log to
Param type	String
Example	<code>sms_gateway.exe -email_comms_log "log@domain.com"</code>
Command	Delete messages from the Send Queue with the nominated destination
Switch	-delete_dest
Param name	Search send queue for entries with this value in the Destination field
Param type	String
Example	<code>sms_gateway.exe -delete_dest "+61419000000"</code>
Command	Delete messages from the Send Queue with the nominated description
Switch	-delete_desc
Param name	Search send queue for entries with this value in the Description field
Param type	String
Example	<code>sms_gateway.exe -delete_desc "Some tag value"</code>
Command	Purge Receive Queue
Switch	-purge_receive
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -purge_receive</code>
Command	Purge Send Queue
Switch	-purge_send
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -purge_send</code>
Command	Minimize SMS Gateway
Switch	-sms_minimize
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -sms_minimize</code>
Command	Restore SMS Gateway
Switch	-sms_restore
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -sms_restore</code>
Command	Hide SMS Gateway
Switch	-hide
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -hide</code>

Command	Show SMS Gateway
Switch	-show
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -show</code>
Command	Connect to Mobile Terminal
Switch	-connect
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -connect</code>
Command	Disconnect from Mobile Terminal
Switch	-disconnect
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -disconnect</code>
Command	Toggle connection to/from Mobile Terminal
Switch	-toggle_connection
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -toggle_connection</code>
Command	Transfer all messages currently stored in the Mobile Terminal to the SMS Gateway Receive Message Queue
Switch	-get_all
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -get_all</code>
Command	Do not copy new incoming messages from the Mobile Terminal
Switch	-no_transfer_all
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -no_transfer_all</code>
Command	Copy all new incoming messages from the Mobile Terminal to the SMS Gateway Receive Message Queue
Switch	-transfer_all_mode
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -transfer_all_mode</code>
Command	Copy all new incoming messages marked TE from the Mobile Terminal to the SMS Gateway Receive Message Queue
Switch	-transfer_te_mode
Param name	n/a
Param type	n/a
Example	<code>sms_gateway.exe -transfer_te_mode</code>
Command	Send a command to the Mobile Terminal, and wait for an OK response from the MT.
Switch	-passthrough
Param name	Hayes Command
Param type	String
Example	<code>sms_gateway.exe -passthrough "+csq"</code>

Command	Shutdown SMS Gateway
Switch	-exit
Param name	n/a
Param type	n/a
Example	sms_gateway.exe -exit

Message Submission

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.

The command requires a minimum of two parameters to be supplied, these being the destination(s) and message content.

Command	Submit a message to SMS Gateway to be sent via SMS	
Switch	-send	
Switch	-dest	[Destination(s) to send message to]
Param type	String	
Switch	-msg	[Message content]
Param type	String	
Example	sms_gateway.exe -send -dest "0419000000" -msg "Hello there"	

Switch name	-udhi	[Set the UDHI bit]
Switch name	-no_udhi	[Clear the UDHI bit]
Param type	n/a	
Switch name	-rp	[Set the RP bit]
Switch name	-no_rp	[Clear the RP bit]
Param type	n/a	
Switch name	-rd	[Set the RD bit]
Switch name	-no_rd	[Clear the RD bit]
Param type	n/a	
Switch name	-mr	[Set the Message Reference]
Param type	Integer	[0 – 255]
Switch name	-pid	[Set the Protocol ID]
Param type	Integer	[0 – 255]
Switch name	-dcs	[Set the Data Coding Scheme]
Param type	Integer	[0 – 255]
Switch name	-vp	[Set the Validity Period]
Param type	Integer	[0 – 255]
Switch name	-desc	[Attach a Description tag to the message]
Param type	String	
Example	sms_gateway.exe -send -dest "0419000000" -msg "Hello there" -udhi -dcs 240 -desc "Msg#1"	

SPECIAL CHARACTERS

When using the CLI, you will not be able to pass Carriage Return, Line Feed or Double Quote, characters as Windows will intercept, interpret and remove these characters prior to SMS Gateway receiving the command. In order to support these characters you can use the following special codes to represent these characters:

<CR> Carriage Return <LF> Line Feed <DQ> Double Quote (")

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"	This is the executable name without the .exe extension. If using multiple instances of SMS Gateway, change this to the name you chose for the desired instance.
DDE Topic or	= "DDE_Send"	This is a simple command suitable for sending of messages with no additional parameters (such as UDHI, Class etc). An example of which is shown below.
DDE Topic	= "DDE_CLI"	This command enables you to use the powerful CLI syntax via DDE. Use the CLI command (as detailed earlier in this manual) as the only parameter you pass to SMS Gateway.

SMS Gateway receives the send instruction from a DDE client application in the form of a macro. The following example syntax is Excel 2000 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 : With Excel, SMS gateway MUST 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 are two example Microsoft Excel 2000 spreadsheets named *EXCELDDE_2002.XLS* and *EXCELDDE_CLI_2002.XLS* that demonstrate sending messages with SMS Gateway using DDE⁹.

⁹ Only available under the Windows Operating System

OLE Automation¹⁰

SMS Gateway is written in Delphi 7.0, and compiled as an OLE Automation server in the form of an out-of-process executable. For an application to interwork with SMS Gateway using OLE, 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 when using OLE, 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 improperly duplicated.

Message Transmission

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

Message Reception :- Removal of Messages from MT = Delete immediately

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

¹⁰ Only available under the Windows Operating System

Message Reception :- Removal of Messages from MT = Delete when retrieved

<u>OLE Application</u>	<u>SMS Gateway</u>	<u>Mobile Terminal</u>
Retrieve_Message Poll =>	Read Message => <i>Log Message to "receive.txt"</i> <= Message Content Delete Message => <i>Message deleted from Receive Queue.</i>	<= Incoming message notification <= Message Content <= 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);

procedure CLI_Command(const appName, cli_command: 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:

<i>Coding</i>	<i>Maximum Length</i>	<i>Data Type</i>
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:

<i>Return Code</i>	<i>Description</i>
--------------------	--------------------

-
- 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

<i>Value</i>	<i>Data Type</i>
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:

<i>Value</i>	<i>Meaning</i>
-1	Unspecified
0	Default
1	ME Specific
2	SIM Specific
3	TE Specific

validity_period is an overloaded field of type Small Integer that specifies both the SMS Validity Period, and header options within the SMS-SUBMIT PDU . The upper byte of the field contains the header options within the SMS-SUBMIT PDU, as defined in ETSI Specification 03.40. A summary of the bits within the field follows:

- Bits 0,1 TP-Message-Type-Indicator (TP-MTI)
- Bit 2 TP-Reject Duplicates (TP-RD)
- Bits 3,4 TP-Validity-Period-Format (TP-VPF)
- Bit 5 TP-Status-Report-Indication (TP-SRI)
- Bit 6 TP-User-Data-Header-Indicator (TP-UDHI)
- Bit 7 TP-Reply-Path (TP-RP)

Be very careful when setting these fields, the values set here are bitwise OR'd with the default settings within SMS Gateway with no checking for correctness. So you can set values here that result in an invalid PDU.

The only header option supported by SMS Gateway within this field is the TP-UDHI, if you set this bit, please ensure your data contains a correctly formatted header, otherwise the PDU will violate the specification and cause unpredictable results.

It is recommended you only set these fields if you know exactly what you are doing!

The lower byte of the field contains 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:

<i>Return Code</i>	<i>Description</i>
-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.

procedure CLI_Command(const appName, cli_command: WideString);

This command enables you to utilise the CLI API via OLE. Please refer to the CLI reference earlier in this manual for the syntax of the cli_command field.

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 7.0 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 spreadsheet file named *EXCELOLE_2002.XLS* that demonstrates full bidirectional interworking with SMS Gateway via OLE.

Thankyou for choosing SMS Gateway by GPA Technology