



SysManSMS Server

User and Developers Guide
Version 8.5

ENGLISH

Last modified for V8.5

CONTENT

PRODUCT PROTECTION	5
SYSMANSMS SOFTWARE LICENSE AGREEMENT	5
PRODUCT ACTIVATION	5
<i>How to check Product Activation Status</i>	5
<i>How to manually Activate the Product</i>	5
SYSMANSMS FOR WINDOWS – ARCHITECTURE PICTURE	6
QUICK OVERVIEW OF PRODUCT FUNCTIONALITY	7
SERVER AND LOCAL CLIENT INSTALLATION	8
REMOTE CLIENT INSTALLATION	8
WHERE TO FIND SYSMANSMS FILES	8
HOW SERVER SENDS SMS FROM YOUR NETWORK	9
NUMBER FILES IN THE SYSMANSMS ARCHITECTURE	9
<i>Default Number Files - Usage and naming</i>	10
<i>Selecting a Number File dependent on the SMS text</i>	12
<i>Change a Number File entry from users mobile</i>	12
<i>Number Files and Message Delivery Status</i>	12
<i>STEP send Number Files - Request mobile user to reply back</i>	12
<i>Test Number Files without sending messages</i>	12
<i>Schedule automatic change to Number Files</i>	13
SEND SMS FROM WINDOWS DESKTOP USING WINCLIENT	14
<i>Activating WINclient for the first time</i>	14
<i>Running WINclient from a remote PC</i>	14
SEND SMS FROM FILE – FREE TEXT FILES	15
<i>Use of different Number Files</i>	15
<i>Confirm message read - Request mobile to reply back</i>	15
SEND SMS FROM FILE – FORMATTED FILES	16
<i>SysManSMS Formatted files – TXT type</i>	16
<i>SysManSMS Formatted files – XML type</i>	16
<i>SysManSMS Formatted files – HTML type</i>	17
DELAY OR REPEAT SENDING FROM FILE BY USE OF FILE NAMING	18
<i>Delay sending of a message files</i>	18
<i>Repeat sending of a message files</i>	18
<i>Select a specific number file for delayed files</i>	18
SEND SMS FROM COMMAND LINE CLIENT – SYSMANSMS_CMD	19
SEND SMS FROM COMMAND LINE CLIENT – SYSMANSMS_ICLIENT	19
<i>Number File dependent on the SMS text</i>	19
SEND SMS FROM COMMAND LINE CLIENT – SYSMANSMS_ENVCLIENT	20
ADDITIONAL INFORMATION TO COMMAND LINE CLIENTS	20
FORWARD MAIL TO SMS (SMTP)	21
<i>How to send mail to SMS</i>	21
<i>SysManSMS Server to manage the SMTP Server</i>	22
<i>Mail to SMS Security</i>	22
FORWARD SMS TO MAIL	22
FORWARD TRAPS TO SMS (SNMP)	23
<i>Operation Mode selection</i>	23
<i>Define Header and Footer for SNMP SMS texts</i>	23
<i>Map a Number File to SNMP client address</i>	24
<i>Trap to SMS Text mapping</i>	24
<i>Trap to SMS Text mapping using OID Editor</i>	25
FORWARD URL TO SMS (HTTP)	27
<i>HTTP Server in SysManSMS</i>	27
Configure SysManHTTP Server	27
<i>HTTP Module for Microsoft IIS Servers</i>	29
How to install HTTP IIS functionality	29
Send from a FORM in a browser	29
Send from a URL string	29

SMS SEND TEXT FILTERS TO CONVERT OR LIMIT SENDING	30
SMS SEND TEXT FILTER ACTIVATION	30
SMS SEND FILTER FILES FORMAT.....	30
SMS SEND TEXT FILTER EXAMPLE.....	30
HOW SERVER HANDLES INCOMING SMS.....	31
EXECUTE A SERVER ADMINISTRATIVE COMMAND FROM A MOBILE	31
<i>Mobile command :ALARM xxxx</i>	31
<i>Mobile command :VCFWD xxxxxxxx</i>	31
<i>Mobile command :VCSTAT</i>	31
<i>Mobile command :CONT</i>	31
<i>Mobile command :HOLD</i>	31
<i>Mobile command :INFx</i>	32
<i>Mobile command :STAT</i>	32
<i>Mobile command :STOP xxxx</i>	32
<i>Mobile command :SERVERRESET</i>	32
<i>Mobile command :STORMRESET xxxxxxxx</i>	32
<i>Mobile command :TIME</i>	32
<i>Mobile command :ADDSLAVE COMxx</i>	32
<i>Mobile command :REMSLAVE COMxx</i>	32
<i>Mobile command :SHOWSLAVE</i>	32
<i>Server internal command execution – Security issues</i>	33
PROCESSING INCOMING SMS – AUTOMATIC EXECUTION OF A PROGRAM	34
<i>Alias Register File for Program names</i>	34
<i>Parameters to programs that starts on incoming SMS</i>	34
<i>Always start a program on incoming SMS</i>	34
STANDARD MOBILE PROGRAMS USAGE	35
NAMING OF MOBILE EXECUTABLE PROGRAM	35
DEFAULT – make any program your default program	35
ADM – SysMan TELLme application	35
ECHO – to return text from file	35
GETNUM – to lookup a number in RIGHTSDB.....	35
ID – to delete STEP queued messages.....	35
OFF – to deactivate your mobile in a Number File	36
ON – to activate your mobile in a Number File	36
SET – to set default Number Files.....	36
SF – to return file content to mobile.....	37
SMStoFILE – store message text to file	37
SMStoMAIL – to forward SMS text as email.....	37
SMStoSQL – to store SMS text to SQL database	38
SN – to forward SMS to members of a Number File	38
SR – to do SMS Storm Reset for one number.....	38
SSTART – to start a Windows Service	39
SSTOP – to stop a Windows Service	39
WC – WakeupCall request	39
WHO – to check who is active in a Number File.....	39
PROCESSING INCOMING SMS – SAVE TEXT TO FILE.....	40
PROCESSING INCOMING SMS – SECURITY SETTINGS.....	42
<i>Limit mobile user access using the SysManSMS.rig file</i>	42
<i>Internal format of the SysManSMS.rig file</i>	43
<i>Creating a default answer-back message</i>	43
SMS INCOMING TEXT FILTERS TO MODIFY RECEIVED SMS	44
SMS INCOMING TEXT FILTER ACTIVATION	44
SMS INCOMING TEXT FILTER FILES FORMAT	44
SMS INCOMING TEXT FILTER EXAMPLE.....	44
SERVER STATUS INFORMATION	45
ADVANCED SERVER OPTIONS.....	46
MODIFYING THE SERVERS INPUT MESSAGE FOLDER PATH.....	46
SHARE SERVERS INPUT FOLDER BETWEEN MULTIPLE SERVERS	46

FILE FILTERING ON INPUT FOLDER	46
SAVING PROCESSED MESSAGE FILES	46
SETUP A SYSMANSMS SLAVE SERVER CONFIGURATION	47
<i>Configure SysManSMS Slave Servers using desktop menu</i>	48
<i>Configure SysManSMS Slave Servers using Administrator SMS</i>	48
SETUP A SYSMANSMS DUAL SERVER CONFIGURATION	49
LOGGING ACCOUNT INFORMATION TO SEPARATE FILE	50
MONITOR IMPORTANT WINDOWS SERVICES ON YOUR SYSTEM	51
<i>Create a Service Monitor List</i>	51
<i>Start a Windows Service from your mobile</i>	51
<i>Stop a Windows Service from your mobile</i>	51
PROCESSING INCOMING VOICE CALL – SEND SMS	52
<i>Automatic execution at incoming Voice Call</i>	52
<i>Start a program</i>	52
<i>Send message to numbers in Number File</i>	52
<i>Voice Call – Security issues</i>	52
AUTOMATIC CONVERSION OF MESSAGE FILES	53
THE COMMAND WINDOW INTERFACE TO SYSMANSMS SERVER	53
<i>Display Server Legal commands</i>	53
<i>Check Server Software Version</i>	53
<i>Install Server in Windows Control Manager</i>	53
<i>Start the SysManSMS Server service</i>	53
<i>Stop the SysManSMS Server service</i>	53
<i>Pause the SysManSMS Server</i>	53
<i>Hold (freeze) the SysManSMS Server for 30 Seconds</i>	53
<i>Resume the SysManSMS Server from Pause</i>	53
<i>Remove Server from Windows Control Manager</i>	53
USE OF WINDOWS ENVIRONMENT VARIABLES IN SYSMANSMS	54
HOW TO MANUALLY REMOVE A SYSMANSMS INSTALLATION	54
ADDING SYSMANSMS WATCHDOG OPTION TO YOUR SMS SERVER	55
SYSMANSMS DEVELOPER INFORMATION	56
USE OF SYSMANSMS SDK FOR SENDING SMS MESSAGES	56
<i>Description of the functions in the SysManSMS_SDK.dll</i>	56
int SysManSMS_Hello (*servadr,*clientname);	56
int SysManSMS_Send (*servadr,*mobilenumber,*smstext,*clientname);	57
<i>Content of SysManSMS_SDK.h</i>	58
<i>Example code in 'C'</i>	58
<i>Test Number Files without sending messages</i>	58
<i>Example code in 'VB'</i>	59
<i>SysManSMS SDK as Microsoft COM object</i>	59
APPENDIX-1	60
7 BIT GSM CHARACTER SET	60
APPENDIX-2	61
7 BIT GSM CHARACTER SET EXTENSION TABLE	61

Product protection

SysManSMS Software License Agreement

SysMan AS is the 100% owner of the SysManSMS code, product and all rights to distribute the product.

SysMan AS grants the owner of this license a limited license to use the selected software product in binary executable form in the normal operation of the product.

Title, ownership rights, and intellectual property rights in and to the Software remain in SysMan. You agree not to distribute, decompile, disassemble, modify, reverse assemble, reverse engineer, or reduce to human readable form the Software or any part thereof or create any derivative works based on the Software.

One SysManSMS Server License KEY can only be used with one installation. Multiple installations will fail! Use *uninstall* to move a license KEY to another installation.

By purchasing the Software you acknowledge that you have read this agreement, and that you agree to the content of the agreement and its terms, and agree to use the Software in compliance with this agreement. The agreement comes into legal force at the moment when you download this Software from our site or receive it through email or on any data medium from SysMan AS.

Product Activation

SysManSMS Server product activation is a technology that protects users from pirated or counterfeit software by limiting use of the product to those users who have acquired the product legitimately. Product activation requires a unique product key for each installation of a product. The activation of the SysManSMS Server will normally be done automatically via the GSM network at installation time. If the automatic activation should fail due to limited GSM access, you will be notified. A manually activation must then be performed. You must activate the SysManSMS Server within ten (10) days after installing it. If product activation is not successful, the product will stop working.

How to check Product Activation Status

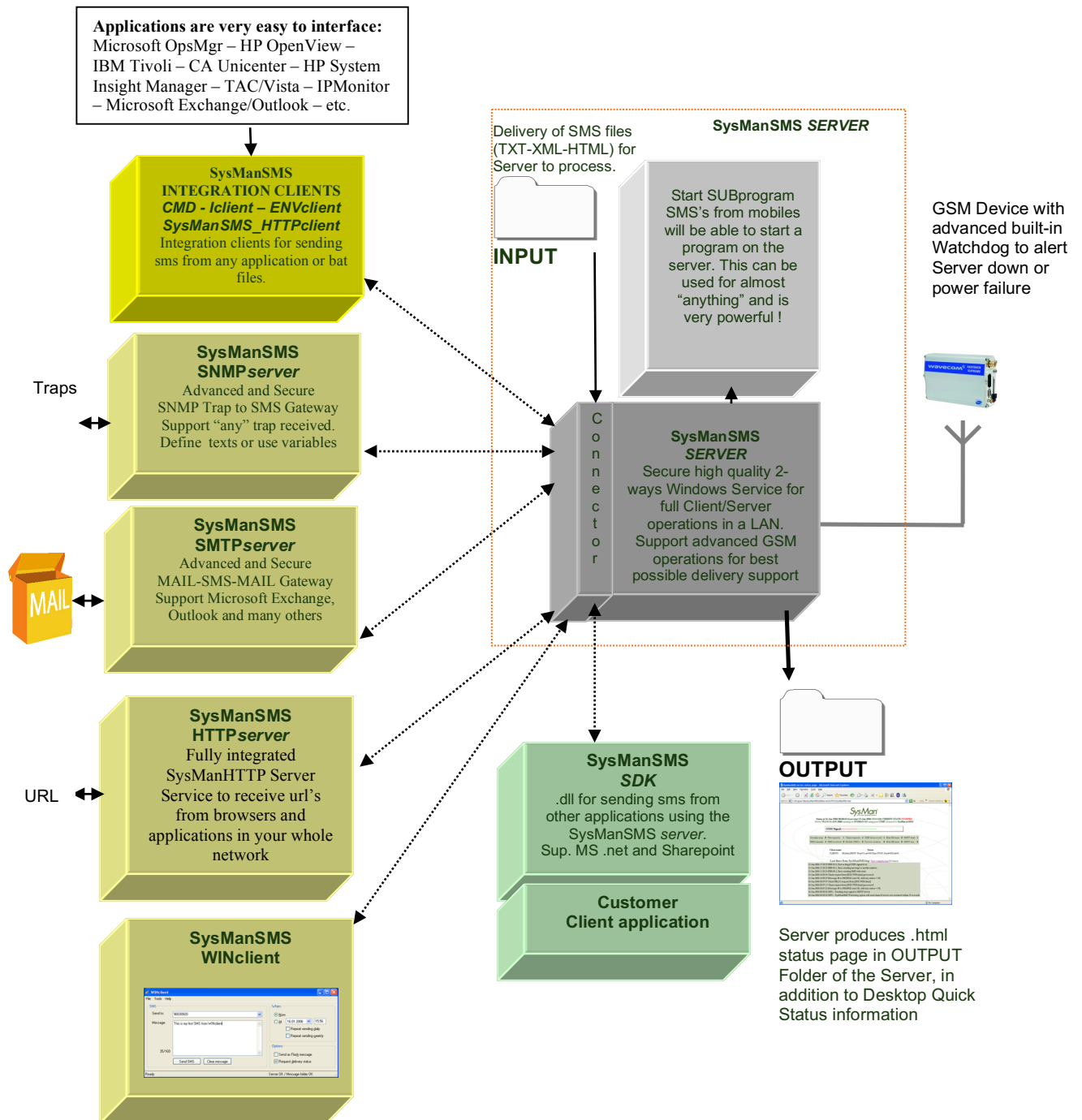
To check product activation status, right-click on the SysManSMS Server desktop menu icon and select *Quick Status*, or open the SysManSMS Server html status page located in `OUTPUT` folder. You should see the text "Product Activated" or "Product not Activated".

How to manually Activate the Product

To manually activate the product, right click on the SysManSMS Server desktop menu icon and select Server Settings and License Activation. If you are not using SysManSMS Server Desktop Menu, you may start the SysManSMS_Register.exe from the `Utilities` Folder.

Copy the Request Key and go to <http://www.sysman.no/support/activate> and use this key to request the Activation Key. As soon as you receive the Activation Key on e-mail, insert it into the Activation Key field and hit OK. SysManSMS Server will now automatically restart to activate.

SysManSMS for Windows – Architecture picture



SysManSMS Server for Microsoft® Windows 32 or 64 bits platforms with full Client/Server support and a secure connection to any global GSM network.

NO INTERNET CONNECTION REQUIRED !

Quick overview of product functionality

With the SysManSMS Server software and a GSM modem, you will be able to send and receive SMS messages or dial from/to your intranet - to/from any phone or GSM/SMS capable device.

Different licenses may limit functionality. You can at any time upgrade your license by contacting your reseller, or SysMan direct. See www.sysman.no for more product and contact information.

The Short Message Service (SMS) for mobiles and fixed phones are proven to be the most flexible and secure message system for alerts, remote control and information collection. Even if new mobile solutions like MMS and MAIL has been introduced, the SMS usage and popularity has increased significantly over years.

The SysManSMS Server is a high quality, secure and flexible 2-ways mobile gateway software, supporting the safe SMS protocol plus dialup calls. With its direct GSM link, you get an independent internet-free solution. Either the task is a general SMS solution or safe alert delivery; the built-in delivery notification and auto re-routing capability will make sure the messages are delivered, even if the standard network is broken. The SysMan supplied GSM devices come with a Watchdog to alert your critical full down situations. In this way the Server becomes your networks Emergency Exit!

The Send support of the background server component will handle client requests from command line, desktop, mail and user clients, including many flexible file send formats. The SMS Storm control and text filters will stop sending of unwanted messages. A powerful User Group manager works across all send interfaces and can be managed either by the administrator or direct from the users own mobile. A rights-database will handle user access and security settings.

The Received messages can safely start or stop programs and services, save to file or SQL database. The SMS-to-MAIL has full 2-ways INBOX support. In addition a special set of admin commands will help controlling your system, or you may create your own scripts or programs to fit your special needs.

Integration with other applications is very simple and can be done in several ways. Powerful 2-ways MAIL support included. The SDK for developers includes Microsoft® .net and Sharepoint. More on integration at our web www.sysman.no

General features and requirements

- Windows (32/64) XP®, 2K, 2003, 2008/R2, 2012, Vista, Windows 7/8/8.1
- Install and forget – Setup only takes 5-10 Minutes
- Send/Receive capacity of up to 25.000 SMS's per 24 hour
- Server/client architecture - supporting full LAN
- Live status and menu access from Windows desktop

Server software features

- Easy configurable, even from text file interface
- SMS Storm Control and text filters to limit sending
- SMS Silence Alert if Server detect no messages sent
- Delivery acknowledge and SMS read detection
- Automatic re-routing if mobile can't receive
- Secure execution of *any* program on incoming SMS
- Dual server, dual operators with automatic failover
- SMTP Server for secure 2-ways mail-sms-mail support

GSM/SMS message support

- 2-ways communication with any worldwide GSM network
- Standard SIM card from any GSM operator
- Support SMS (max 1530 char) plus WakeupCall (dialed)

Send SMS from message files

- Send from free text files to single users or users groups
- File text content may be used to select destination group
- Send from text, xml or html formatted files

Send SMS from client programs

- Desktop and Command Line utilities for sending
- Common User Group manager across all clients
- Support client or server located User Groups
- Client failover to second gateway/GSM network
- Client SDK for program developers including .NET

Send SMS from any mail client via secure SMTP

- Send to one number or User Groups
- Support mail-back SMS delivery notification
- SMS replies can be routed back to senders INBOX

Send SMS from HTTP form or single URL

- User friendly web form to send to number or User Groups
- Single URL to send from applications in one step

Secure and powerful handling of incoming SMS

- Save messages to text, xml or html files
- Save messages direct to *any* Access or SQL database
- SMS-to-MAIL forwarding; even without any address or key
- Automatic start of programs as a sub process
- Users sign on/off to User Groups from own mobile
- Start or Stop Windows Services from mobile

Security and logging

- Event log support - notifications on SMS or e-mail
- Account and Activity logs for all components
- Monitor critical Windows Services – send alert
- Advanced Watchdog option monitors hardware
- Rights-database to limit or stop mobile access

Product ORDER identifications:

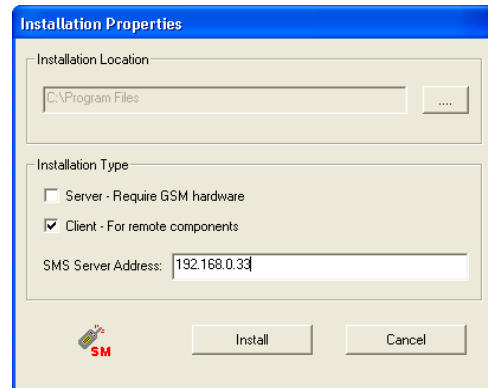
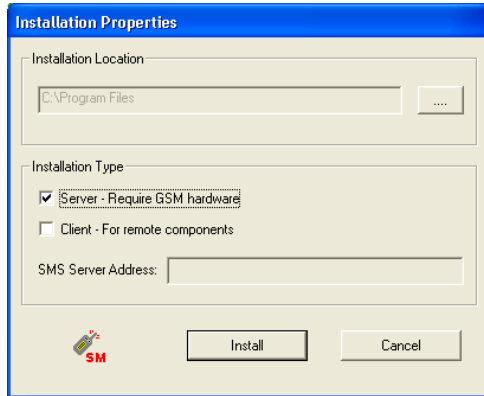
- 802000 - SysManSMS ALERT SRV license
- 802051 - SysManSMS ALERT SRV 1 year upg/support
- 804000 - SysManSMS COMMUNICATION SRV license
- 804051 - SysManSMS COMMUNICATION SRV 1 year sup.
- 100002 - GSM device w/Watchdog – SERIAL + USB, ex pwr

Installation and configurations

The SysManSMS SETUP kit can be used for 3 different tasks:

- Install and configure the SysManSMS *Server* and local *Clients* on server machine
- Install *Client* software on other machine than the server (remote components)
- To remove *Server* and *Client* installations

Please follow steps in the SysManSMS Installation Guide to properly install SysManSMS *Server*.



Server and local Client Installation

Selecting “Server” will start a standard installation type. The Server installation requires a supported GSM device to be connected on a Serial or USB port on this machine. The installation procedure will take care of all local settings and create all the needed .ini files for the server and clients to run. As soon as the server is running (after a Minute or so, check your SysManSMS.html page) you are ready to start sending and receiving SMS messages to and from your network.

Remote Client Installation

If you like to send SMS from another machine as where your server is running, you run the SETUP on the remote machine and select “Client” installation type. The installation procedure will not start the server, but will copy all Clients to the standard SysManSMS folder tree.

The desktop client WINclient will need to be configured for remote operations after first start. The configuration is done via the *Tools – Options – Server settings...* menu of WINclient.

At installation time you will be asked to enter the “SMS Server Address”. This will be used as default address for the clients on your remote installation. The command line clients, CMDclient, lclient, ENVclient, all have its own private folder and a private .ini file for configuration settings. To override the default address you can edit the client's .ini file. The supplied .ini files can be changed using a standard text edit tool such as Notepad.

Access and security issues:

Make sure the accounts using clients has access to SysManSMS *Server* by using domain accounts or same username/passwords. If firewall, you may need to open port NetBios/Named Pipes (139/445).

Where to find SysManSMS files

SysManSMS components are all located in the same directory structure, SysManSMS. Normally this is below “C:\Program Files”. Here you will find documentation and other folders. Each client has its own home sub-folder, with init (.INI) files, Number Files (.LST) and log files.

How Server sends SMS from your network

The SysManSMS Server will let you send SMS messages or dial any number in many ways:

- **From Windows Desktop** using the WINclient utility (found on Desktop Menu or in *Utilities* folder)
- **From text files saved to INPUT folder**, free or formatted txt/xml (See *Fclient* folder for samples)
- **From Windows command line** via integration utilities CMDclient, Iclient or ENVclient
- **From any mail client** via the built-in secure SMTP server (enabled in SysManSMS Server SETUP/Configure)
- **From http form or URLstring** via supplied IIS modules
- **From your own application** via .dll, COM or .net libraries. See chapter for *Developers*.

Note: SysManSMS Server supports dialing a number in the same way as sending SMS messages.

IMPORTANT ABOUT SMS STORM CONTROL

If your sending algorithm or application should fail in a way that a lot of unwanted SMS's could be sent, you may turn on the Server's SMS Storm Control. (This is done during SETUP or Server Config) SMS Storm Control will limit the amount of SMS's that one number can receive within same date. The Storm Control will be reset each midnight, or if : **STORMRESET** or **SR** commands are received by the server.

Note: Storm exceptions numbers can be listed in the file `exceptions.txt` in the STORMDB folder.

Number Files in the SysManSMS architecture

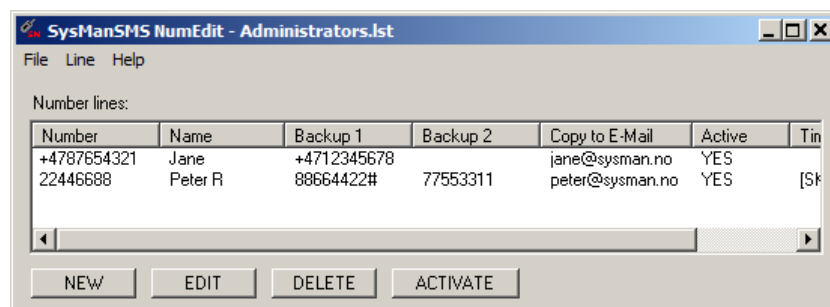
Number Files (.LST) are files that list multiple numbers that you can send to as a group. Instead of sending multiple messages to single recipients, you can address the message to a Number File, and the message will be delivered to all recipients. Most client interfaces for sending SMS can use this group messaging feature. Number Files are maintained with the supplied Number Editor, or a plain text Editor like Notepad.

Number Files are located inside the different client folders. The most common client program is the **SysManSMS_Iclient**, located in the *Iclient* folder, and you will normally store all Number Files here. You may also share SERVER located Number Files between multiple Servers. For more details on how to setup shared Number Files, please see chapter about Dual-Server Setup in *Advanced Server Options*.

The *Iclient* is automatically invoked by Server if you send a free text file. This is very powerful, as you can name a message file, so that it matches a Number File of same name. (See sending of SMS from file). You can name your Number Files as you want to, as long as you use the file type extension .LST. The default Number File for the clients, are named the same as the client programs, but with the extension .LST. Client uses default if no other specified, or if it can't find the specified Number File.

Schedule Number Files to be activated from a specified time, defined in file name string. See p12.

Start SysManSMS_NumEdit from Windows System Status bar menu or from Utilities folder:



This is the SysManSMS_NumEdit.exe main window.
Use buttons to add, change or remove as many number lines you need.
You can temporarily disable a line by hitting the Enable/Disable button.

**Each entry in a Number File can specify backup numbers and set send/skip periods for the number.
 If backup number(s) are specified, the SMS delivery status will automatically be activated
 One or two e-mail addresses can be specified to request a copy for archiving or informational usage**

Note on Time settings usage:

Specifying “SKIP send on selected times” or “SEND only on selected times” will open up the capability to limit sending of SMS to specific times for this number. Example: Skip Sunday at “00-23” will result in not sending SMS to this number on Sundays between 00:00 and 23:59.

If you specify “HOLD (delay) on selected times” the message will be put in a file queue for execution just after the end of specified HOLD time.

In the example Item window, the person will get SMS messages all days except Saturdays between 00 and 23:59 and Sundays between 00 and 23:59. The *Backup 1* number will get a Wakeup call in the same period, if *Primary* can't be reached.

A copy of the SMS text can be sent to e-mail addresses (max 2) for information or archiving.

Default Number Files - Usage and naming

The SysManSMS Server and its components, uses some default Number Files for internal alerts.

IMPORTANT:

Any SysManSMS installation should have at least one number in the default SysManSMS_Iclient.lst Number File. We also recommend that you create an Administrators.lst in Iclient folder, to forward System messages to administrative users only.

To fully make use of System message features, it is important to know the following:

Number File Name	Located	Usage
SysManSMS_Iclient.lst	Iclient	Iclient uses if no Number File specified, or specified file does not exist Also used by the following system tasks: <ul style="list-style-type: none"> - If more than 20 files in INPUT folder at Server Startup - To warn if your trial license has expired
Administrators.lst	Iclient	If exist, system messages like the following will be sent: <ul style="list-style-type: none"> - After upgrade or configuring the SysManSMS Server - If send Hard errors exceed limit monitored count of 25 - Warning about errors in SysManSMS.log at startup time - Warning about errors handling the SMTP Service - If an :ALARM time bomb did not reset within time - State changes in Monitored Services (see Service Monitoring) - If Iclient or ENVclient are being used with bad parameter cnt - If problems accessing INPUT or OUTPUT folders If file does not exist, SysManSMS_Iclient.lst will be used
SecurityStop.lst	Iclient	Only if exist, a Security Rights Stop will be warned to listed users

Voice_To_SMS.lst	Iclient	Only if exist, a message is sent to listed users if a dial-in call detected
SysManSMS_ENVclient.lst	ENVclient	ENVclient uses if no Number File specified, or specified does not exist
SysManSMS_Pclient.lst	Pclient	Pclient uses if no Number File specified, or specified does not exist
SysManSMS_Sclient.lst	Sclient	Sclient uses if no Number File specified, or specified does not exist

Selecting a Number File dependent on the SMS text

You may select a specific Number File dependent on the text inside SMS messages. The Iclient (used by free text files and for integration with other applications) will check for any defined rules, and then switch to the given Number File. Any hit in the rule search will override any specific or default Number File name. For more information on how to setup Number File rules, see the chapter about the Iclient.

Change a Number File entry from users mobile

A powerful feature is to ENABLE or DISABLE an entry in a Number File from users own mobile. Read about the ON, OFF and WHO utilities in the *How Server Handles incoming SMS* chapter.

Number Files and Message Delivery Status

One of the most advanced functions in the SysManSMS Server, is its use of the GSM networks capability to tell whether a message could be delivered or not. This will give you a very safe handling of messages, much more advanced than normal e-mails.

The Number File architecture uses this in a powerful way, so that messages not delivered will be deleted, and backup numbers may automatically be used as new destinations.

The delivery notification rules used in Number File architecture is like this:

12345678	Entry with single number. No delivery check
12345678,87654321	Entry with 1 backup number. Delivery checked on both numbers
12345678,87654321,24682468	Entry with 2 backup numbers. Delivery checked on all three.

STEP send Number Files - Request mobile user to reply back

To find if the receiver of your SMS actually did read your message, you may use STEP technique.

By doing so, you may stop sending of more messages as soon as a mobile has indicated that the message is read. This technique is possible with the STEP send function, allowing the mobile user to return the same message he just got, indicating its read.

You enable STEP send by simply start the Number File name by **STEP_ or STEPxx_**, where "xx" indicate Minutes of delay between each SMS going out. (If not specified, default is 10 Minutes)

If a STEP Number File is detected, an ID will be generated in the front of your text. (Uses 10 chars.) First message will be sent to Primary number of 1st line. The backup number(s) of the first line, plus the rest of Primary (only) numbers, will be queued with a delay of "xx" Minutes between each.

A Mobile program called **ID** in the *Programs* folder is then used to delete unsent STEP messages. A receiver of a STEP message will just have to send back the same message as he received. By doing so, he will remove any unsent message with same **ID**. A confirmation SMS will be returned. For more details on **ID**, see description of the mobile program ID.EXE

STEP files fully support WakeupCall's (dial a number) in the same way as sending SMS messages. However, only a SMS with the ID can clear the rest of the waiting messages in the queue.

Test Number Files without sending messages

As you configure and test your SysManSMS installation, you may need to just send to a dummy number. Two such numbers are defined in the server, one number where the mobile is "on-line" and the other one for "off-line" testing. Simply use the following numbers for test:

11111111	to send dummy to a mobile with state "on-line"
22222222	to send dummy to a mobile with state "off-line"

Schedule automatic change to Number Files

You may like to plan your Number Files for more than one week. This is possible by creating Number Files with a WAIT or REPEATE flag as part of the file name.

As soon as the specified date and time are reached, SysManSMS Server will automatically process your scheduled file, and make it an active Number File.

If the scheduled file is a repeating file, the SysManSMS Server will calculate a new time and create the next scheduled Number File.

By use of this technique you may create advanced notification plans for different personnel, and have them automatically change in for example 4 weeks intervals.

You may use the following naming conventions to establish files for automatic processing:

WAIT-110220-1830-numfile.lst	Execute on 20.02.2011 18:30, no repeat
REPH-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat hourly
REPD-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat daily
REPW-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat weekly
REP1-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat every 1. week
REP2-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat every 2. week
REP3-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat every 3. week
REP4-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat every 4. week
REPM-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat monthly
REPY-110220-1830-numfile.lst	Execute first on 20.02.2011 18:30, repeat yearly

Note: Since the length of a month is variable, the REPM will always add 30 days.

The scheduled Number Files are only supported by lclient. For this reason all such Number Files must be stored in the lclient's folder, either locally or on a common share for Dual Server's.

Send SMS from Windows desktop using WINclient

A powerful client is the SysManSMS_WINclient in the WINclient folder. WINclient is capable of sending SMS in many different ways, including using Number Files located in the Server's Iclient folder. From inside WINclient you got a direct access to the Number File Editor for creating or maintaining Iclient Number Files. You can also create repeating message files from inside WINclient.

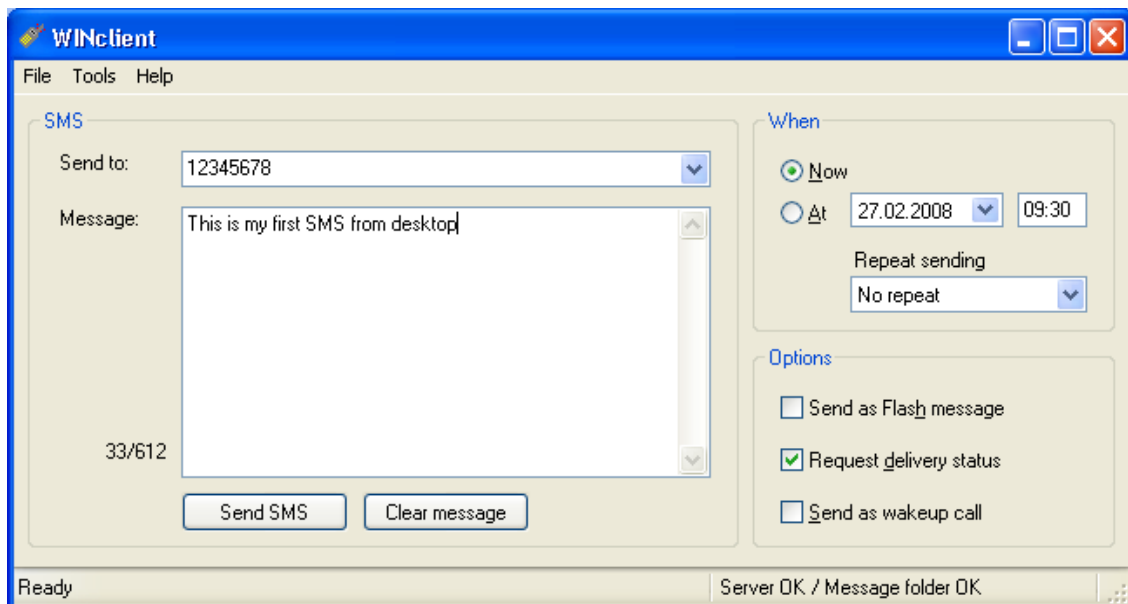
Activating WINclient for the first time

You activate WINclient by double-click on the file SysManSMS_WINclient.exe in the WINclient folder.

NOTE: SysManSMS_WINclient are dependent on Microsoft .NET Framework V2.0. If this is not installed on the computer you start WINclient you will get a message for the missing component.

In this case, install .NET 2.0 from Options folder on SysManSMS CD or USB stick, or download from Microsoft.

WINclient will now initialize for the first time and try to find a local SysManSMS Server installation. Default values will then be written to the file SysManSMS_WINclient.ini in the WINclient folder. After some seconds you will see the WINclient main window. You are now ready to send SMS's.



Selecting *Now* and *Request delivery status* will send immediately and display live delivery status. All other sending will create message files to be executed by the SysManSMS Server.

Running WINclient from a remote PC

Status display at the bottom right corner will tell you if SysManSMS Server is found. If you are running WINclient from a remote PC, you may have to enter SysManSMS Server parameters manually. This is done from the top menu line by selecting **Tools/Options/Server Settings...**

Parameter settings:

SysManSMS Server – If not default (localhost) enter SysManSMS Server Hostname or IP address

SysManSMS Server Folders – If not default, enter path to a shared SysManSMS folder of the Server

SysManSMS Message Folder – If not default INPUT folder, enter path to the active folder

Access and security issues:

Make sure the accounts using clients has access to SysManSMS Server by using domain accounts or same username/passwords. If firewall, you may need to open port NetBios/Named Pipes (139/445).

Send SMS from file – Free Text files

A powerful technique for sending SMS messages is simply to write a text file to the servers active INPUT folder. To be able to send free text files as SMSs, you will need at least 1 number to be entered in a Number File. The default Number File is called `SysManSMS_Iclient.lst` and is located in the Iclient folder. For information on how to create and use Number Files, see chapter “*Number Files in the SysManSMS architecture*”.

Text files can have one or more lines in plain (ISO-Latin-1) text. All text up to a maximum of 1530 characters, or if reached end-of-file, will be defined as the message to send.

If total text exceeds 1530 characters, the message can be split into two SMS messages.

You define maximum length and other parameters for free text files in the Iclient's parameter file `SysManSMS_Iclient.ini` in the Iclient folder.

Sample of a free text file with multiple messages:

```
This is my first sms text line in first sms message
This is my second sms text line in my first sms message
  Single line containing a space only will indicate new message
This is my first sms text line in my second sms message
```

Note that character `ı` (dec 161) is illegal to send due to protocol restrictions, and will be replaced with `!`. Sending text from a file, support SMS new lines. Insert `<cr>` (DEC=13/HEX=0D) to create a new line.

INPUT folder can have File or Send Text Filters activated. See *Advanced Server Options* for details.

If your free text files are not in a suitable format, you can use the TRANSLATE program (parser) to convert your message file to correct format. See *Advanced Server Options* for details.

You may specify to keep a backup of the processed file in the Server's menu configuration settings.

Use of different Number Files

Select Number File based on file name

A powerful technique is to create number files (.LST files) with the same name as your message files. SysManSMS Server will, by default, check if a Number File with the same name as your message file exists in the Iclient's folder, and if so, use this as your Number File. By this technique you can send different messages to different groups of people, dependent on file naming.

If no such match is found, the server will default to the `SysManSMS_Iclient.lst` Number File.

Select Number File based on text content

You may also select a specific Number File dependent on the text inside SMS message. By creating a rule file `NumberFileRules.ini` inside the Iclient folder, the Iclient will check this file for directions to use a specific Number File if a specific sub text string is found in the text about to be sent out.

IMPORTANT: Any hit in the rule search will override any specific or default Number File name. For more information on how to setup Number File rules, see chapter “*Number Files in the SysManSMS architecture*”

Confirm message read - Request mobile to reply back

You may request a mobile to return a message indicating the message has been read.

This technique is called STEP send, and is supported by the Number File architecture.

By simply naming your free text file `STEP_something` you will involve the STEP send function.

For more details on STEP send, see the chapter “*Number Files in the SysManSMS architecture*”

Send SMS from file – Formatted files

The fastest method for the sending SMS messages from your network will be to specify the content in a SysManSMS formatted or xml message file, and then store this file in the servers INPUT folder.

As the server looks into this folder as part of its standard tasks, it will pick up the file and process the content by sending one or more SMS messages.

Note: INPUT folders are monitored and can be shared. See “Advanced Server Options” for details.

SysManSMS Formatted files – TXT type

Record format: **;*CmdType*;*Number*;*Message*;*client_name*;*host_type*;**

CmdType	= SMS	<i>This should always be "SMS"</i>
Number	= dddddd	<i>This string specify a legal mobile number.</i>
		<i>Adding star "*" to number will send SMS "flash"</i>
		<i>Adding plus "+" will enable delivery status, MAX lifetime</i>
		<i>Adding plus "-" will enable delivery status, MIN lifetime</i>
		<i>Adding hash "#" will DIAL the number, no SMS text sent</i>
		<i>Example: "+4712345678*" will send as "flash" sms.</i>
Message	= tttt	<i>This is your sms text, maximum 1530 characters.</i>
Client_Name	= cccc	<i>Name of the application creating this file, max 64 chars</i>
Client_Type	= Fclient	<i>Type of client, this is normally "Fclient"</i>

```

;SMS;11223344;Backup started;BACKUP_PROGRAM;Fclient;
;SMS;11223344*;Backup finished, number of error=0;BACKUP_PROGRAM;Fclient;

```

The character ; (dec 161) is illegal in text part due to protocol issues.

Sending text from a file, also support “new line”. Insert <cr> (DEC=13/HEX=0D) to create a new line.

SysManSMS Formatted files – XML type

The SysManSMS Server also support xml formatted files. When detected, such files will be sent to the xmlparser to be converted into SysManSMS formatted TXT files. The XML supported format is:

```

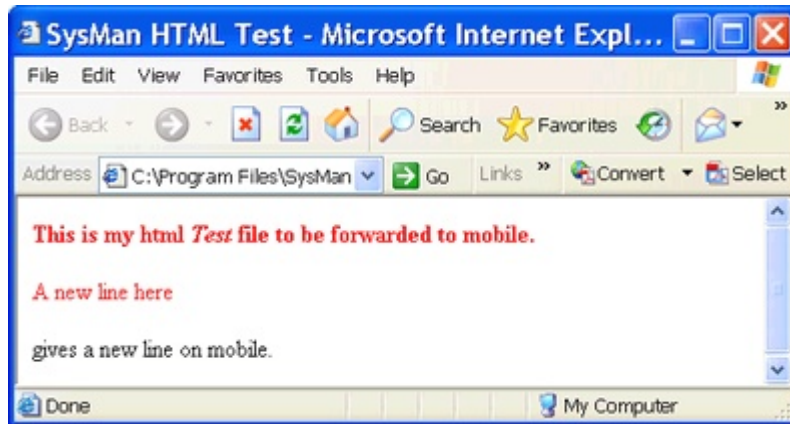
<?xml version="1.0" encoding="UTF-8"?>
<Messages>
  <SMS>
    <Number>+4712345678</Number>
    <Message>This is my SMS text for the mobile</Message>
  </SMS>

  <SMS>
    <Number>+4712345678</Number>
    <Number>+4787654321</Number>
    <Message>This is my SMS text for the mobile</Message>
    <LogDeliveryStatus>true or false</LogDeliveryStatus>
    <SendAsFlash>true or false</SendAsFlash>
    <WakeupCall>true or false</WakeupCall>
    <SendTime>2006-08-10T10:00:00</SendTime>
    <RepeatedSending>H,D,W,M or Y</RepeatedSending>
  </SMS>
</Messages>

```


SysManSMS Formatted files – HTML type

The SysManSMS Server also support html formatted files. When detected, such files will be sent to the htmlparser to be converted into SysManSMS free text files. The HTML file like the below, will be forwarded to mobiles in the Number File matching the name of your HTML file name.



Example:

HTML file `TEST.HTML`, will be sent to every active number in the Number File `TEST.LST`

Notes:

After file processing, you can delete or rename the message files. Rename will add a unique time string to the file name body (`_YYMMDDhhmmss`) in addition to your SETUP specified extension.

INPUT folder can have Text or File Filter activated. See “Advanced Server Options” for details.

STEP_ Number Files are not supported for SysManSMS formatted files (TEXT or XML types).

Delayed or repeated files? See chapter *Delay or Repeat sending from file by use of file naming*.

Delay or Repeat sending from file by use of file naming

You can create SMS message files (both *Free text* and *SysManSMS formatted*) with a special name, so that execution of the file can be delayed or repeated.

You can create delayed or repeated files with the WINclient, or manually with a Notepad.

If Server is configured with a shared COMMON input folder, make sure you create delayed or repeating files in the server's LOCAL sub-folder; for execution only by this specific server.

Delay sending of a message files

SysManSMS Server is able to delay (wait for a time) before executing a message file, based on the naming of your file.

Example of file name format for delayed files:

WAIT-111128-1830-message.txt File will execute when computer time is 18:30 28.11.2011

Note: This file will execute only once

Repeat sending of a message files

SysManSMS Server is able to repeat a message file every hour, day, week, month or year, based on the naming of your file. After execution, a new file will be created with a new calculated repeat name.

The repeating message files can be used for tasks, everything from remaining you on something, or maintaining a time bomb against another SysManSMS Server. (See server command :ALARM)

Example of file name formats for repeating files:

REPH-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat hourly
REPD-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat daily
REPW-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat weekly
REP1-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat every 1. week
REP2-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat every 2. week
REP3-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat every 3. week
REP4-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat every 4. week
REPM-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat monthly
REPY-110220-1830-filename.txt	Execute first on 20.02.2011 18:30, repeat yearly

Note: Since the length of a month is variable, the REPM will always add 30 days.

Select a specific number file for delayed files

If you create free text files with WAIT or REPx file names, you can still via file naming indicate which Number File to be used for that text file. The technique is to use the text after the time specification as a pointer to a Number File in the Iclient's folder. You may use WINclient to create such files.

Example:

File name **REPD-110930-1030-myfile.txt** will look for Number File **myfile.lst** in the Iclient folder. If this is not found, the default will be the SysManSMS_Iclient.lst file.

Send SMS from command line client – SysManSMS_CMD

SysManSMS_CMD is used to send SMS from applications or .BAT files.

SysManSMS_CMD client program will send a text string to one number. It will check for delivery status, and can use backup numbers, if it can't deliver.

SysManSMS_CMD support sending a "flash" SMS by adding a "*" (star) to the number.

```
> SysManSMS_CMD 12345678[,alt-number-1][,alt-number-2] "Your text Message"
```

Adding modifiers to a number:

Adding star "*" to number will send SMS "flash"

Adding hash "#" will DIAL the number, no SMS text sent

The delivery notification rules used by CMD client is like this:

12345678	Entry with single number. No delivery check
12345678,87654321	Entry with 1 backup number. Delivery checked on both numbers
12345678,87654321,24682468	Entry with 2 backup numbers. Delivery checked on all three.

Send SMS from command line client – SysManSMS_Iclient

SysManSMS_Iclient is the most common integration client, and SysManSMS Server also uses Iclient to send free text files from its INPUT folder, by creating a sub process.

SysManSMS_Iclient will take up to ten text parameters from the command line, add them together, and send it all as an SMS message to all numbers in the Number File to be used.

If the first parameter starts with a hash (#) it will indicate a Slave sub Address, when sending to a SysManSMS Server with active Slaves. You may address a specific Slave serving COM12 by make your first Iclient parameter to " #COM12". This request will then be handled by Slave on COM12.

If one of the parameters start with a colon (:) – Iclient will take this as a name of Number File located locally in the Iclient folder. If two colons found, it will use Number File located on Server.

```
> SysManSMS_Iclient txtpar-1 [txtpar-2] [txtpar-3] [txtpar-4] [txtpar-5] [txtpar-6]
```

Default Number File is SysManSMS\Iclient\SysManSMS_Iclient.lst found in Iclient folder

SysManSMS_Iclient ":MyNumberFile" "Text"	- uses local Number File MyNumberFile.lst
SysManSMS_Iclient "::MyNumberFile" "Text"	- uses server Number File MyNumberFile.lst

You may share SERVER located Number Files between multiple Servers. For more details on how to setup shared Number Files, please see chapter about Dual-Server Setup in *Advanced Server Options*.

Number File dependent on the SMS text

You may select a specific Iclient Number File dependent on the text in the SMS message. By creating the file `NumberFileRules.ini` inside the Iclient folder, the Iclient will check this file for directions to use a specific Number File, if a specific sub text string is found in the text about to be sent out.

The format of this file is simple, and can be created by any plain text editor, like the Notepad:

```
:NumberFileName:KeySearchText
```

If sub string *KeySearchText* found, the Iclient will use *NumberFileName* as its Number File.

The search is NOT case sensitive. The *NumberFileName* is to be specified without extension (.lst)

IMPORTANT: Any hit in the rule search will override any specific or default Number File name.

Send SMS from command line client – SysManSMS_ENVclient

SysManSMS_ENVclient is used to send SMS from applications creating text in Environment Variables. SysManSMS_ENVclient program will take up to ten Windows Environment Variables as parameters, and send the content out as one or two SMS message to all numbers in the selected Number File.

If one of the parameters start with a colon (:) – ENVclient will take this as a name of Number File located locally in ENVclient folder. If two colons found, it will use Number File located on Server.

```
> SysManSMS_ENVclient env-1 [env-2] [env-3] [env-4] [env-5] [env-6]
```

Default Number File is SysManSMS\ENVclient\SysManSMS_ENVclient.lst found in ENVclient folder

```
SysManSMS_ENVclient ":MyNumberFile" "ENV-1" - uses local Number File MyNumberFile.lst
SysManSMS_ENVclient "::MyNumberFile" "ENV-1" - uses server Number File MyNumberFile.lst
```

Additional information to command line clients

Using different Number Files

Iclient and ENVclient send SMS to numbers specified in a Number File.

If one of your parameters starts with a colon (:), this will be taken as Number File name.

If you from a remote client like to specify a SysManSMS Server located Number File, use two colons.

With this technique you can send different SMS messages to different groups, by altering one of the parameters. See chapter *Number Files in SysManSMS architecture* for more details.

You can create new or edit existing Number Files by use of the NumEdit tool in the Utilities folder.

You can also alter Number Files from your mobile. See chapter *Standard Mobile programs usage*.

Note: If ENVclient does not find your specified Environment Variable, it will be used as "freetext" and added to the final SMS text string. If string starts with one or two colon(s) - it will be used as a Number File specification.

Configure for remote usage

The command line clients finds the Server address from registry, or alternatively from its .INI file.

Make sure the accounts using clients has access to SysManSMS Server by using domain accounts or same username/passwords. If firewall, you may need to open port NetBios/Named Pipes (139/445).

By default Number Files are local (:). To use Server located Number Files, specify two colons (::)

Note: Server located Number Files are located in Servers Iclient folder only.

Change the clients EXIT status

SysManSMS_CMD, SysManSMS_Iclient and SysManSMS_ENVclient, will all exit with a status of "0" for OK execution, and "1" if it failed. These are the standard program exit codes under Windows.

If your application needs other values, you can define the *Windows Environment Variables*

SysManSMSsuccess and **SysManSMSfailure** to a value that fits your application.

Handling of long text strings

If total text exceeds 1530 characters, the message will by default be split into two SMS messages.

To inhibit this, edit the clients .ini file or define Environment Variable **SysManSMSonesms** to TRUE

Forward mail to SMS (SMTP)

All files and folders for the SMTP Service are placed in the SMTP directory below GSMserver. SysManSMS Server has a built in secure SMTP Server to receive mails from *any* mail client. In addition you can send SMS message to be forwarded to any mail account. You enable the secure SMTP functionality and set Send Mail (SM) parameters in the SETUP utility:



- Set initial values to be used by the SMStoMAIL utility **SM**
 - > Your networks standard outgoing SMTP Server
 - > The mail domain where your mobiles belongs
 - > If SM can't resolve address – send mail to this account
 - > Define your own footer text at end of SM mail text

- Enable the built-in SMTP Server to enable mail-to-sms.

- Then set the Operation and Security parameters to be used by the SMTP Server. Define SMS max length.

- If you like the SMStoMAIL (SM.EXE) to resolve mail return path, enable the save to reply database function.

After next restart of the SysManSMS Server, the SMTP Server (own Windows Service) will be started and managed by the SysManSMS Server. Do not manually start/stop the SysManSMTP service.

How to send mail to SMS

You can send mails to the SMTP server at the computer you installed the server from many of the common mail client programs, including a lot of the management software that produces a mail message if an alert occurs. The SMTP server will receive the mails and decode the content. It will then send the SMS text message via the SysManSMS Server to the destination number specified.

The SMTP Server supports plain text, as well as MIME formatted eMails. In MIME, both TEXT and HTML formats are supported. HTML will be used only if no TEXT part where found.

Multiple mail copies to Numbers and/or Number Files are supported. Some system limits may occur.

IMPORTANT: To address the SMTP Server, you may need to create a DNS subnet (sub domain) for the SMTP server's machine. Also if your network has a mail gateway (Microsoft Exchange or other) you may need to direct mails not to go out to the Internet, but instead be delivered locally.

The following mail send formats and modifications are valid:

- | | |
|------------------------------|---|
| - MobileNumber@smtpserver | Will send to the specified mobile number |
| - Mobile Number@smtpserver | Will send to the specified mobile number, with any spaces removed |
| - MobileNumber.F@smtpserver | Will send to the specified mobile number using SMS Flash message type |
| - MobileNumber.C@smtpserver | Will send to the specified mobile number, and a delivery Confirmation MAIL is returned |
| - MobileNumber.L@smtpserver | Will send to the specified mobile number, and delivery status will be requested/logged. |
| - MobileNumber.S@smtpserver | Will send SUBJECT text only. Apply to both Numbers or Number Files |
| - MobileNumber.R@smtpserver | Will write senders reply path information to SM.EXE lookup Database file |
| - MobileNumber.D@smtpserver | Will DIAL the specified number (WakeupCall) |
| - "Mobile Number"@smtpserver | Will send to the specified mobile number, with any spaces removed |
| - NumberFile@smtpserver | Will send to all numbers in the NumberFile.lst located in the SysManSMS lclient folder |
| - Number File@smtpserver | Will send to all numbers in the NumberFile.lst located in the SysManSMS lclient folder |
| - "Number File"@smtpserver | Will send to all numbers in the Number File.lst located in the SysManSMS lclient folder |
| - Step_File@smtpserver | Will send with delay between messages to all numbers in step_file.lst in lclient folder |
| - sms@smtpserver | Will take the mobile number from the mail's Subject field |

Test the SysManSMTP Server by use of TELNET command line utility

Below you find a sample on how to manually connect and send a mail in the simplest possible way:

```

220 SysMan SMTP MAIL Service, V8.0 08-NOV-2011
HELO TEST
MAIL FROM: test@test
RCPT TO: 12345678@MSGGATEWAY
DATA
<CR/LF><CR/LF>
This is my body text
<CR/LF><CR/LF>.
QUIT
250 Ok
250 2.1.0 Sender OK
250 2.1.5 Recipient OK
354 Please start mail input.
250 Mail queued for delivery.
221 Closing connection. Good bye.
```

SysManSMS Server to manage the SMTP Server

As soon as you have enabled the SMTP Server (also see file `SysManSMTP.ini` in SMTP folder) the SysManSMS Server will “look after” the SMTP server. If it for some reason should stop, it will automatically be re-started up to 10 times. If it stops more than 10 times, the SysManSMS Server will produce an SMS message file to the INPUT folder, called `SysManSMTP.txt` (default). The message in this file will by default be sent to numbers in the `SysManSMS_Iclient.lst` file in Iclient folder. To make a specific number file for this message, simply create a number file called `SysManSMTP.lst`

SMTP Server status is shown in the `SysManSMS.html`, or remote from a mobile **:INFO** cmd request.

Mail to SMS Security

If checked (configuration screen), the sender will be validated against the file `SMTPsender.sec`, and the destination against the file `SMTPdestination.sec` in SMTP folder. In addition a `SMTPfrom.sec` file can be used to limit mail clients allowed to send, either a domain (@domain.no) or a specific addr. Files are maintained by a standard text editor, and the format is simply to enter mail addresses in `SMTPsender.sec` and `SMTPfrom.sec`, and mobile numbers in the `SMTPdestination.sec` file. One address or number on each line.

Forward SMS to Mail

The incoming SMS message can be forwarded to any mail account. To make this happen, the incoming SMS will have to activate the program `SMStoMAIL.EXE` located in GSM Server's *Programs* folder. First of all make a copy of `SMStoMAIL.EXE` to a shorter name like `SM.EXE`.

`SM.EXE` will be started if the SMS starts with the letters **SM** and a space.

The next string will then have to be the destination mail address, then the text to forward.

SMS example: “**SM myaccount@mycompany.com This is my message to the mail reader**”

If the SMS text does not contain a mail address, `SM.EXE` will still try to resolve a destination mail address from one of two files in the `REPLYDB` folder. If `Defined_List.csv` file exist, it will be searched for a number match and use the defined mail address from this file. If still not found, the automatic generated history file `yymmdd.dat` will be searched.

Search functions can be turned on/off in the SMTP Settings or by editing `SysManSMTP.ini` file.

File format of records in `Defined_List.csv` is straight forward: **NUMBER;MAIL-ADDRESS;**

If still not found; the `SM.EXE` will look for a pre-defined default mail account. If a default mail account is found; this will be used as destination mail address for the received SMS text.

A default mail account can be defined in the SMTP Settings or by editing `SysManSMTP.ini` file.

Always forward SMS to MAIL?

If Send Mail is your default program (`SMStoMAIL.EXE` copied to `DEFAULT.EXE`) any incoming SMS will be forwarded to a mail account as long as the mail address can be resolved. In this case the message does not need to start with the letters **SM**. `SM.EXE` will in this case always start if no other match is found. For more info on `SMStoMAIL`, also see chapter *How Server handles incoming SMS*.

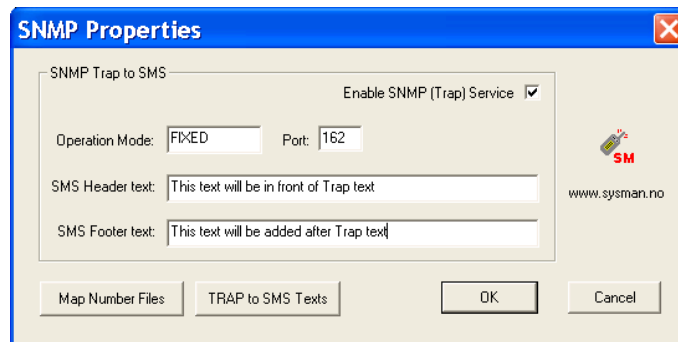
Forward Traps to SMS (SNMP)

All files and folders for the SNMP Service are placed in the SNMP directory below GSMserver.

The SysManSNMP Service is a component of the SysManSMS Server, and will be managed by this. The setup lets you control and design SNMP Trap texts to be forward to any Mobile. You may forward SNMP traps from any source to the SysManSNMP Service, and get the message directly on to your mobiles screen within seconds. The SNMP Service fully support the SysManSMS Number File (User Group) facilities, which even let you forward copies of your message to e-mail. The SNMP Trap texts can be the raw SNMP Data, a fixed text string, or even a formatted text with variable-binding values included. You only select and handle the Traps you like to monitor.

Configure SNMP Service from either running SysManSMS SETUP or use the CONFIGURE menu

Important: Windows SNMP Services need to be started for SysManSNMP Service to run



All settings from above Properties will be stored in SysManSNMP.ini in SNMP folder

Operation Mode selection

The SNMP Service can operate in one of three modes: BASIC, FIXED or CUSTOM. In the FIXED and CUSTOM modes there is a corresponding OID definition file which holds information on what to do with a received trap. A new undefined trap will always be written to a file in the RAW subfolder. This file can then be used either to manually build definition information in the OID file, or use the SysManSNMP_Editor to quickly build the definitions.

As soon as a definition is finished, the RAW file is moved to the PROCESSED raw sub folder.

<u>MODE</u>	<u>ACTION:</u>
BASIC	Receive traps, store all traps in .RAW subfolder, produce a basic message file in the SysManSMS Server's INPUT folder.
FIXED	Receive traps, store first trap to the .RAW subfolder. Using the trap ID, you can then define the trap's fixed text to be sent as SMS, in the mode's OID file. On the next trap, the service will write a message file to the SysManSMS Server's INPUT folder. Definitions are stored in SysManSNMPOIDs_FIXED.TXT
CUSTOM	Receive traps, store first trap to the .RAW subfolder. Using the trap ID, you can then define the trap's custom text to be sent as SMS, in the mode's OID file. On the next trap, the service will write a message file to the SysManSMS Server's INPUT folder. Definitions stored are in SysManSNMPOIDs_CUSTOM.TXT In this mode you can use variable-binding information from trap data as part of text to be sent as SMS.

Define Header and Footer for SNMP SMS texts

If you like, you may define a fixed Header and/or Footer text to be added to all the SMS's generated by the SNMP Service. For no Header/Footer – leave the fields blank.

Map a Number File to SNMP client address

A Trap from a specific IP Address can be setup to always go to numbers in a specific Number File.

The mapping will be done in the file `SysManSNMP.txt` found in the SNMP folder.

If no file found, or if no IP Address defined, the default SysManSMS_Iclient Number File will be used.

```
! This is the IP (or name) to SysManSMS Number File mappings
!
! FORMAT = IPorNAME:NumberFileName
!
! 192.168.1.3:ENVIRONMENT
! 172.16.162.1:CISCO
! *:UseThisNameForAllSourceIP
```

Trap to SMS Text mapping

The easiest way to define traps is to use the OID Editor in SNMP folder. (see next chapter)

You may also manually edit a OID definition file by clicking the *Trap to SMS Texts* button in setup.

Dependent on the mode you have selected, the corresponding definition file will be opened.

Example: SysManSNMPOIDs_FIXED.TXT

```
!
! SysManSNMP OID Definition file for FIXED Mode Operation
!
! Below you find some definition samples, add your own for the traps to be handled
!
! Sample entry
!
! [TRANSLATE]
!
!     [TRAP_OID]                [<OID to be translated. Like, 1.3.6.1.4.1 -- Check the MIB>]
!
!     [MESSAGE]                 [<Message to be printed when this trap is received>]
!
!
! [TRANSLATE_END]
!
!
! NOTE: The keywords MUST appear in the same order, the optional ones could be
!       dropped. Yes, the order must be preserved
!
!
!
! NOTE 2: If there is any error or mismatch in translation, the program shall print the
! objects the way they are returned by Microsoft API's
!
!
!
! [TRANSLATE]
!
!     [TRAP_OID]                [1.3.6.1.4.1.2854.0.1]
!
!     [MESSAGE]                 ["We have received a Trap from TrapGen utility."]
!
! [TRANSLATE_END]
```

You may now start the SNMP Service by hitting OK

Trap to SMS Text mapping using OID Editor

It has never been easier to forward SNMP Trap's to a mobile, and the easiest way to define a trap's SMS message is to use the SysManSNMP_Editor found in SNMP folder.

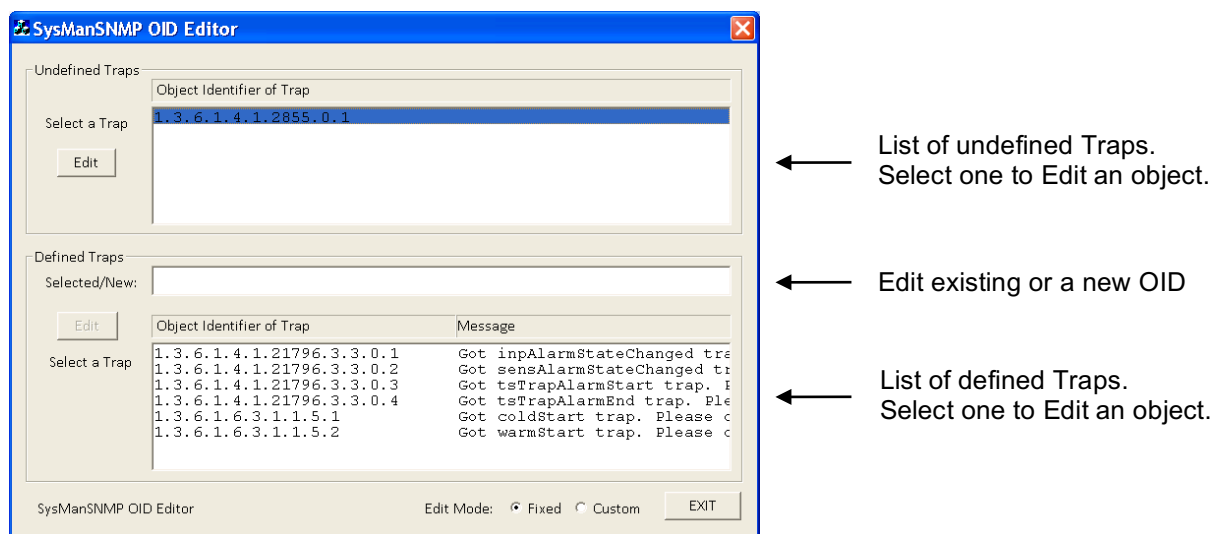
NOTE: To use the SNMP OID Editor, make sure the SNMP Service has been started. (see above)

To receive a undefined trap, you either setup a device to send a trap, or you may use the TrapGen freeware software tool provided in the TrapGen subfolder.

From a Command Line Window execute the following TrapGen example line:

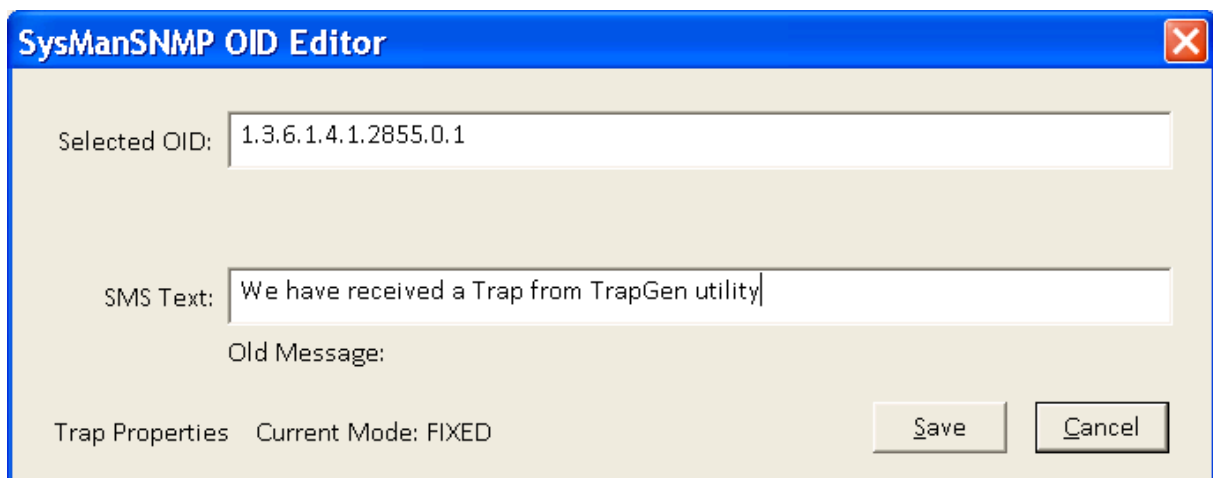
```
TrapGen.exe -d localhost -o 1.3.6.1.4.1.2855.0.1
```

This command will send a trap to the SysManSNMP Service, and if not yet defined, a file will be written to the RAW subfolder. Now start the SysManSNMP_Editor.exe found in the SNMP folder.

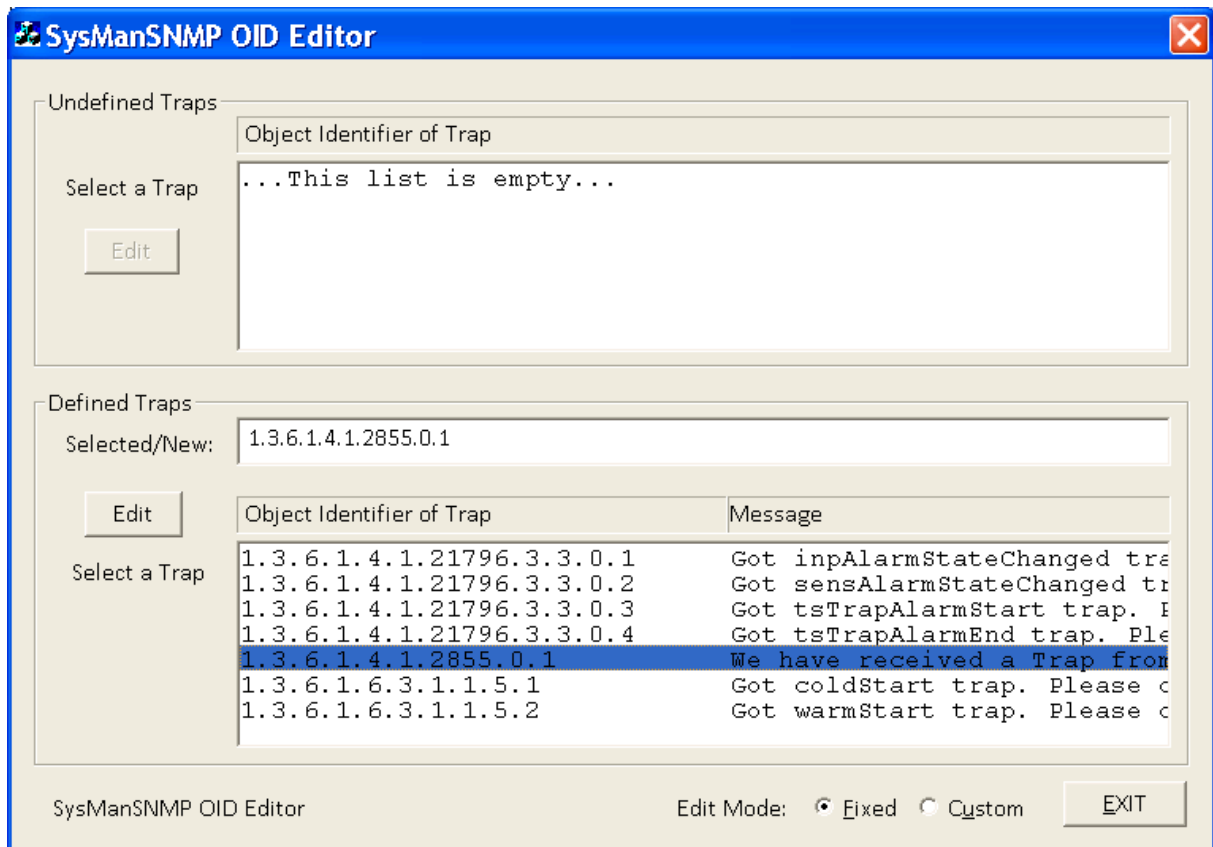


At start, the SysManSNMP Editor will list undefined and defined Traps. Select FIXED mode.

To define our trap from TrapGen, select the Trap from the undefined list and hit Edit

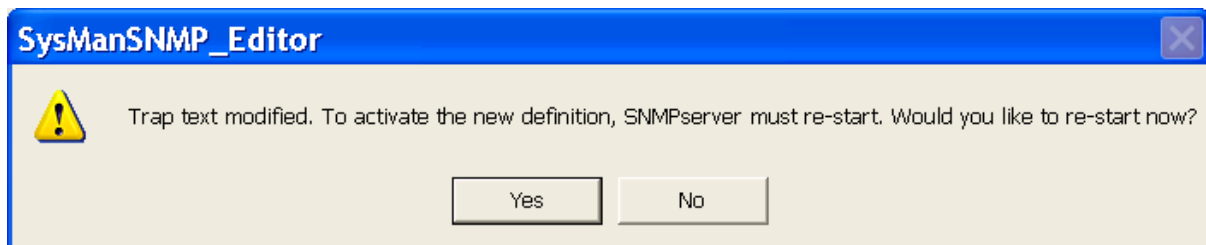


In the Trap Properties Screen you define the SMS Text to be sent to your mobiles as soon as this OID is received. Hitting the Save button will write the defined Trap to the SysManSNMPOIDs_FIXED.TXT file and move the undefined file to the PROCESSED subfolder of the RAW directory.



The update Editor screen will now show the new Trap defined with its corresponding SMS text

You may now Exit the SysManSNMP_Editor



If you allow the SNMP Service to re-start, the new Trap definition will be active after the re-start has completed. If you now execute the TrapGen utility one more time, the Trap will send a SMS message to people in the defined Number File. All SNMP Service activities will be logged to SysManSNMP.LOG

Notes:

You may at any time edit SysManSNMPOIDs_xxxx.TXT files to define/redefine Traps. After any definition has been made, the SNMP Service must re-start.

You may switch to CUSTOM Mode for more advanced definitions to be configured. In CUSTOM mode you will be able to define VarBinds (variables) from the Trap to be forwarded as part of SMS Text. Examples of such definitions are shown in the supplied OID files.

For more information on defining SMS texts from Traps, see the ReadMe file in the SNMP folder.

Forward URL to SMS (HTTP)

In SysManSMS Server you find two ways to support applications forwarding http urls as mobile SMSs. Both methods give you full flexibility to change the url parameter names to fit your application:

- 1) A fully integrated HTTP Server is by default started on port 8080 and ready to serve urls
- 2) The other option is to place the included SysManSMS_HTTPclient on a IIS web server

HTTP Server in SysManSMS

The SysManHTTP Server is by default started after installing SysManSMS Server. SysManHTTP Server is a Windows Service, totally managed by the main SysManSMS Server. The files for this Server are found in the folder ...\\SysManSMS\\GSMserver\\HTTP\\

To change the default settings for the SysManHTTP Server, you may edit the SysManHTTP.ini. New settings will be active after next restart of SysManSMS Server.

Configure SysManHTTP Server

Edit the SysManHTTP.ini

NOTE-1: If first line=YES, the service will run, to disable, set it to NO

NOTE-2: Make sure your Firewall is open for the selected port to be used

NOTE-3: For older Windows OS (2003/XP):

For the SysManHTTP Service to run, please make sure the
Microsoft Visual C++ 2008 Redistributable Package is installed

You will find a kit in HTTP Folder (vcredist_x86.exe) or at MS Downloads

```
YES
! Define the Port Number to be used (Default = 8080)
!
UrlBase          = http://*:8080/
!
! Define the module name
!
Document = sms
!
! Define url string parameter keyword names. The order is fixed
!
UrlParams = recipients, message, messageType, logDelivery, username, password, version, debug
!
! If username/password parameters specified, then define the username/password
! To disable username/password check - leave this blank
!
SysManSecurity = sysman,sysman
!
! Define http response value
!
Response_OK      = 200
Response_ERROR = 400
!
! Define a character Translation Table (Maximum 32 translations can be defined)
!
TRANSLATE_BEGIN
'+ ' = ' '
TRANSLATE_END
!
```

About the URL parameters:

Recipients	Required, holds one or more mobile number(s) or a Number File Name (User Group)
Message	Required, holds your sms message string (http/ISO-Latin-1)
MessageType	Optional, values may be TEXT/FLASH/WAKEUPCALL (Default=TEXT if not used)
LogDelivery	Optional, values may be YES/NO (Default=NO if not used)
Username	Optional, holds SysManSecurity username (Default to no username check if not used)
Password	Optional, holds SysManSecurity password (Default to no password check if not used)
Version	Optional, must be specified on own url - Return HTTPserver version info
Debug	Optional, must be specified on own url - values may be YES/NO (Default=NO)

About the URL Message string parameter:

You can use any characters in your message string allowed by the http url specification.

Example: To make a newline on the mobile, just insert a %0D in your text.

If you need to translate a character from one value to another, you may define a translation in the Translation table of the INI file. (see below)

About the URL Version parameter:

Sending the url `http://localhost:8080/sms?version` will return the HTTP Server's version information.

The version parameter cannot be combined with other parameters.

This url may be used to check if the HTTP Server is available.

About the URL Debug parameter:

Sending the url `http://localhost:8080/sms?debug=YES` will set the HTTP Server in debug mode.

This mean that it will log all received url strings (raw) to the server's log file.

This is very useful when setting up new applications to use the HTTP Server.

The debug parameter cannot be combined with other parameters.

The Server will stay in debug mode until a `debug=NO` is received, or the server is re-started.

About the URL Translation Table:

The SysMan HTTP Server can hold a translation table defined in the INI file.

This table will be used to convert one by one character in the message string.

You may list up to 32 translations in the form: `'+' = ' '` or `'a' = 'A'`

URL Samples:

```
http://localhost:8080/sms?recipients=+4712345678&message=My Text&messageType=TEXT
&LogDelivery=NO&username=sysman&password=sysman
http://localhost:8080/sms?recipients=+4712345678&message=My Text Sent from from url
http://localhost:8080/sms?recipients=ALERT&message=MyAlarmText
http://localhost:8080/sms?recipients=12345678&message=SMStext&username=sysman&password=sysman
http://localhost:8080/sms?version
http://localhost:8080/sms?debug=YES
```

Note:

If recipients are numbers, you may supply multiple numbers in one url:

```
http://localhost:8080/sms?recipients=12345678,87654321,11223344&message=MyAlarmText
```

HTTP Module for Microsoft IIS Servers

You may send SMS messages by filling in a FORM in a WEB browser, or by sending a complete URL string containing all the data. The flexible URL can be used direct by other applications like alert systems, network monitoring software, access control systems and other, for easy access to mobiles via SMS.

This URL will be compatible with services from GSM operators, but instead of going out to the Internet, you use your local secure SysManSMS Server as SMS send engine.

How to install HTTP IIS functionality

SysManSMS comes with a complete HTTP client software to receive message requests. In the `HTTPclient` folder of the SysManSMS installation, you will find the files to be copied to a Microsoft IIS root directory. The only requirements for the HTTP client software will be Microsoft ASP.net V3.5 or later. You may edit the web configuration file to adjust settings, including keywords naming.

Send from a FORM in a browser

URL-Link: The network address of your IIS6/IIS7 where SysManSMS HTTPclient is located

Send to: A valid national or international Mobile Number, including spaces and “-” characters
 Note: International prefix has “00” + country, or “+” + country (+4712345678)
 Alternatively this may be a SysManSMS Iclient Number File name (User group)

Message: This is the text to be sent as SMS. Maximum length is 1530 characters (10 SMS's)
 ISO-Latin-1 characters supported for international use.

Send from a URL string

NOTE: The HTTP client has a flexible format for a URL string to send messages. The default keywords of the URL string can be altered in the `Web.config` file.

```
http://sysmanserver/sms.aspx?recipients=<recipients>&message=<text>
&messagetype=<type>&logdelivery<deliverynotification>
&username=<username>&password=<password>
```

sysmanserver The network address of your IIS6/IIS7 where SysManSMS HTTPclient is located

recipients one or more valid national or international Mobile Number(s)
 Note: International prefix has “00” + country, or “+” + country (+4712345678)
 alternatively, this may be a SysManSMS Iclient Number File name (User group)

message This is the text to be sent as SMS. Maximum length is 1530 characters (10 SMS's)
 US-ASCII characters supported in plain text. Multinational characters in UTF-8,
 ISO-Latin-1 will need to be encoded. Example: “%C3%A5” = “å”

Optional:

messagetype Specify a `text`, `flash` or `wakeupcall` (Default is `text`)

logdelivery A `true` will tell SysManSMS Server to log SMS Delivery Status (Default is no log)

username A string matching the HTTPclient's username (Default is no username)

password A string matching the HTTPclient's password (Default is no password)

SMS Send Text Filters to convert or limit sending

Independent of your send techniques, SysManSMS Server can search your SMS text to see if it contains text patterns that should be stopped or modified before sent to the mobile.

4 filter rules can be activated for such tasks: *ConvertText* – *StopText* – *SendText* – *AppendText*

SMS Send Text Filter activation

To enable SMS Text Filtering, you simply create a text pattern file inside the folder `Filters` located below the `GSMserver` folder. **After SysManSMS Server is re-started, the filter(s) will be active.**

`ConvertText.txt` - contains words or text patterns to “convert text if found” actions

`StopText.txt` - contains words or text patterns to “stop send if found” actions

`SendText.txt` - contains words or text patterns for “send only if found” actions

`AppendText.txt` - contains words or text patterns for “appending (add) text” actions

The files can contain multiple text patterns, one pattern on each line.

If a line should be active for only selected mobile numbers, start the line with colon (:) then enter the number(s) and end with a colon just before your text pattern.

If more than one filter is active, filtering order is 1) *ConvertText* 2) *StopText* 3) *SendText* 4) *Append*

`SysManSMS.log` file will contain information about Text Filter activation. In addition there will be one entry for each pattern match that results in a conversion or stop sending event.

SMS Send Filter files format

The format for `Convert` and `Append` files are: **Original case sensitive text;Convert text to this**
The character “;” is used to indicate where “from text” ends and “to text” begin. For this reason you can NOT convert patterns containing this character. NOTE: The html tag’s “=0D” and “=0A” will convert to control characters <cr> and <lf>, whatever you got on the right side of pattern strings.

The format for `Stop` and `Send` files are straight forward, one pattern or words on each line.

For filter files, you can specify a filter pattern line to be active for selected mobiles only by starting the pattern line with **:mobile numbers:** This means, line should only be tested for the specified mobiles.

SMS Send Text Filter example

You got a Network Management system to produce alert messages. One of the message types that occur is an informational message type that you do not want to send out. These messages contain the word “...INFORMATION:...” inside the complete string. You decide not to send SMS with this word.

Make sure you edit this pattern 100% correct (case sensitive!) into the file `StopText.txt` located in the folder `...\SysManSMS\GSMserver\Filters`

If this was the first time you created this filter file, re-start the SysManSMS Server.

From now on, any SMS message containing the text pattern: **INFORMATION:** will be stopped from being sent to mobiles. A log entry will be written to `SysManSMS.log`

Warning:

Make sure your text filtering does not interfere with important alert messages that need to be sent out. Also be aware that WakeUpCall’s that has “text” will be processed by the filters in same way as SMS’s

Also note that text filtering works on any SMS sent, independent of source and send method being used (files, clients, mails or application SDK).

How server handles incoming SMS

Sending SMS and receiving SMS use separated techniques, so that your application does not have to wait for incoming answers. Server can handle incoming SMS in three different ways:

Priority #1 - it checks if it's an internal server command (starts with a colon) – See list of such commands below.
 Priority #2 - it checks if it's a program match in the programs folder of the server (...\\SysManSMS\\GSMserver\\Programs)
 Priority #3 - It will store the message to a file. How, is dependent on the settings in the .INI file.

Note on Multipart texts:

Multipart (SMS with more than 160 characters) will be handled as multiple SMS's of 153 chars. Each Multipart SMS will have "MP <xx>" added to the front of the message. xx = sequence number. This will make the total message length 160. Hint: If you like to add multiple messages into one long text, you may create a program MP.EXE to do this.

Execute a server administrative command from a mobile

The SysManSMS Server has some built-in **admin** commands that can be executed from your mobile, or from a another SysManSMS Server – like the :ALARM to setup server monitoring.

Mobile command :ALARM xxxx

This command creates or maintains a "time bomb" on the server. As soon as you do not keep your alarm alive, a message file called `Alarm_timeout.txt` will be written to INPUT folder for sending. The default timeout is 90 Minutes; X can optionally specify a value between 1 and 9999 Minutes. If exist, the lclient Number File `Alarm_timeout.lst` will be used for sending the alarms. For security reasons, only the number that created the alarm can reset an existing one

Mobile command :VCFWD xxxxxxxx

This command will set the Voice Call forwarding number for your SIM card's GSM Account. This command is very useful if you use your SysManSMS Server with a call centre or other. You can then have all SMSs handled by the SysManSMS Server, and any Voice Call will be forwarded to a specific number for people to people connection.

The xxxxxxxx will be your forward number. If number no given, Voice Call forwarding will be disabled. The information will be stored at the GSM Operator's database.

Mobile command :VCSTAT

This command will check your SIM card (GSM Account) current Voice Call forwarding settings. The information will be retrieved from the GSM Operator's database.

Mobile command :CONT

This command will set the SysManSMS Server in *running* mode if its *paused* by a ":STOP"
 The command will also return a confirmation message telling its new status.

Mobile command :HOLD

This command will freeze the SysManSMS Server for 30 Seconds. No messages will be lost. The :HOLD is used in special cases where you need to stop all activity for some time. The command will also return a confirmation message telling the freeze has started.

Mobile command :INFx

This command will return the latest status line from SysManSMS *service* client “x”. All service clients (Hclient, Sclient, Pclient) will update the SysManSMS *Server* with its latest status. This status line will be returned to the mobile user, upon a “:INFx” request. If “x” is 0, you will get status from client 0, if “x” is 2, you will get status from client 2, and so on.

Mobile command :STAT

This command will return a status string from the SysManSMS *Server*

Mobile command :STOP xxxx

This command will *pause* the SysManSMS *Server* for 60 Minutes (default), so that no more messages are sent out (see note). Optional specify a value of 1-9999 Minutes to change the default pause value.

NOTE: Even if the server is set to PAUSE mode, it will respond to all Admin commands, and it will also handle wait and repeat message files that has timeout within the PAUSE period. This includes (wait) files produced by the STEP function.

Mobile command :SERVERRESET

This command can be used to reset the SysManSMS *Server* from a mobile with ADM rights. This will read the .ini file, reset all counters and do a full init of the GSM device, just like a re-start.

Mobile command :STORMRESET xxxxxxxx

SMS Storm Control can be enabled at server configuration and will limit the amount of SMS messages that one mobile can receive within same date. This command can be used by administrators to reset a Storm Stop for one number, or the complete SMS Storm Database. SMS Storm Database can be shared in a Dual-Server configuration, and this command will then reset the Storm counters for both systems.

Mobile command :TIME

This command will return the current date and time from your SysManSMS *Server*.

Mobile command :ADDSLAVE COMxx

Command valid on main *Server* only, and will start a *Slave Server* on same machine, using the COMxx for the GSM Modem connection.

Pre-requisite: A valid license, a GSM Modem on COMxx, and a valid SIM card with PIN code disabled. A returned SMS will tell if command was executed successfully. HTML status shows *Slave* status.

You can read more about SysManSMS *Server* farms in chapter “Setup a SysManSMS *Slave Server* configuration” of the User Guide.

Mobile command :REMSLAVE COMxx

Command valid on main *Server* only, and will shutdown and remove a *Slave Server* on COMxx. A returned SMS will tell if your command was executed successfully or not.

Mobile command :SHOWSLAVE

Command valid on main *Server* only, and will return a list of active *Slave Servers* on the main server.

Server internal command execution – Security issues

If you for security reasons would like to disable execution of server internal commands (starting with :), you may define the Windows Environment Variable `SysManSMSnoservercmd` to TRUE, and re-boot. If requests for internal commands arrive after this, it will be logged as a security STOP.

SysManSMS Server has a first level security mechanism. The server will always check if the file `SysManSMS.rig` is in place in the server's directory. If it exists, it will look it up and see if the incoming SMS mobile number exists in the file. If not, this is a security STOP, and will be logged. "USR" and "ADM" security levels, only "ADM" permit execution of server commands (starts with colon)

For more information on the server security and the rights file, see chapter *Security settings*.

Processing incoming SMS – automatic execution of a program

A very powerful technique is the automatic start of a program as the server receives an SMS. By sending an SMS with the first word in the message equal to a program in the server's *Programs* folder, you will automatically start this program as a sub process under the account of the server. Leading spaces in the SMS text will be ignored when searching for a program match.

NOTE: Execution of mobile programs require Server with a two-way license.

Alias Register File for Program names

A program in the Server's *Programs* folder can be started even if the first word in a SMS does not match a program name. This can be setup by simply creating a file called `ProgramAlias.txt` in the *Programs* folder. This will also make it easier if you need multiple keywords to start same program.

Alias File Format:

```
WhoAliasName:who
OffAliasName:OFF
```

Parameters to programs that starts on incoming SMS

SysManSMS Server will always fill the first 3 parameters of the program that starts:

Parameter #1: The number of the mobile phone that did send the SMS
 IMPORTANT: If the number is a MASKED (Virtual) number, the Server will replace it with a dummy number (00000000) not to harm any program execution
 Parameter #2: The SMS text received
 Parameter #3: The SMS's submit time (Send time format "yy/mm/dd hh:mm:ss")

Programs can be of type cmd, exe, bat, or other. Note: Programs can't access keyboard or screen.

Example: Create the command file `test.bat` in the Server's *Programs* folder:

```
echo off
C:
cd \Program Files\SysManSMS\GSMserver\Programs
echo Received text from mobile %1 is: %2 > test.txt
```

Then send SMS text "TEST Hello" to the server. Now look in the file `test.txt` just created.

With this technique you can build very powerful solutions where all data is available to your program.

Always start a program on incoming SMS

There are some techniques to always have a default program executed as SMS arrives. Make sure your setup did not specify incoming SMS's to be stored to a file. (See server config)

If the first word of an SMS does not match any program name in the *Programs* folder, the server will try to start a program called `Default.exe` from *Programs* folder.

In the server *Programs* folder you will find many useful programs. Typical could SMStoFILE, SMStoMAIL or SMStoSQL be copied to be your `Default.exe`.

If the default program can't be started, the SysManSMS Server will store the received SMS to file `computername_InfoPage.txt` in the servers OUTPUT folder, to inhibit loss of data.

You may optionally specify your default program not to be named default.exe but something else.

In SysManSMS.ini specify "NONE:myprog.exe:", myprog.exe becomes your default program name.

You can also set up your server to always start a program if a specific mobile sends the Server an SMS. To do this, simply place a program named *mobilenumber* in the *Programs* folder. Example: If a program called `_4712345678.bat` exists in the *Programs* folder; this program will be started as a sub process of the server, if *any* messages are received from mobile +4712345678.

Standard Mobile programs usage

In the Servers *Programs* folder, you will find some very useful programs to be started from a mobile. See the following chapter for details about the supplied programs.

NAMING OF MOBILE EXECUTABLE PROGRAM

The programs in the servers *Programs* folder has a lot of small programs named after the function they can perform. To make names more useful from a mobile, simply make a copy to the name you prefer. Typical you would like to copy SMStoMAIL.EXE to SM.EXE. (Send Mail program)

NOTE: After upgrading the SysManSMS Server, you may need to make a new copy.

DEFAULT – make any program your default program

If a program called `Default.exe` exists in the *Programs* folder, this will be started if no other program match is found. The `SMStoFILE.exe`, `SMStoMAIL.exe`, `SMStoSQL.exe` are example of programs typical used as a default program.

ADM – SysMan TELLme application

The ADM.EXE found in the *Programs* folder is a complete application to handle SMS users based on topics and subtopics in a database. For details on the TELLme application – see own documentation. TELLme also require a license key open for TELLme access. Contact SysMan for details.

ECHO – to return text from file

The ECHO.EXE will look for a file called ECHO.TXT, and if found – return the text inside this file to the mobile that started this program. You may rename ECHO.EXE to any name, and the text file will then have same name, but with .TXT extension. You may have multiple lines and up to 1530 characters in the text file.

GETNUM – to lookup a number in RIGHTSDB

The GETNUM.EXE can be used to find a name of a number or a number of a name in the SysManSMS Rights Database. Example: SMS “GETNUM 12345678” will return registered name for mobile user 12345678. SMS “GETNUM PETER” will return Peters registered mobile number.

ID – to delete STEP queued messages

If you use “STEP_” or “STEPxx_” named Number Files, an ID number will be added to start of SMS. A delay will then be set up between each of the next destination numbers, so that a receiver of such an SMS are able to stop sending to the remaining numbers in a Number File (queued messages). By simply returning the same message as you receive (ID xxx), you will stop sending to the rest of the numbers in a Number File. (ID will delete queued message files with matching ID number – 6 digits) Example: SMS “ID 123456” will delete any waiting message file with ID “123456”.

STEP files fully support WakeupCall's (dial a number) in the same way as sending SMS messages. However, only a SMS with the ID can clear the rest of the waiting messages in the queue.

For more details on STEP send, also see chapter *Number Files in the SysManSMS architecture*.

OFF – to deactivate your mobile in a Number File

If you have entered your mobile number into a Number File, and are set to status ENABLED, you can use the mobile program OFF.EXE to DISABLE (deactivate) your entry in the file. By simply sending an SMS message, your phone will turn you *off* in the specified Number File, or group of files. You can rename OFF.EXE to what you want. A log will be produced telling activities and usage.

OFF	If enabled, your mobile will be disabled (deactivated) in ALL Number Files in the Iclient folder
OFF myfile	If enabled, your mobile will be disabled (deactivated) in the Server Iclient's Number File <code>myfile.lst</code> in Iclient folder
OFF file1,file2	If enabled, your mobile will be disabled (deactivated) in the Server Iclient's group of Number Files separated by comma, in Iclient folder
OFF files Pclient	If enabled, your mobile will be disabled (deactivated) in the Server Pclient's specified Number File(s) in Pclient folder

IMPORTANT: If only one number active in a number file, OFF will not allow a DISABLE

ON – to activate your mobile in a Number File

If you have entered your mobile number into a Number File, but are set to status DISABLED, you can use the mobile program ON.EXE to ENABLE (activate) your entry in the file. By simply sending an SMS message, your phone will turn you *on* in the specified Number File, or group of files. You can rename ON.EXE to what you want. A log will be produced telling activities and usage.

ON	If disabled, your mobile will be enabled (activated) in ALL Number Files in the Iclient folder
ON myfile	If disabled, your mobile will be enabled (activated) in the Server Iclient's Number File <code>myfile.lst</code> in Iclient folder
ON file1,file2	If disabled, your mobile will be enabled (activated) in the Server Iclient's group of Number Files separated by comma, in Iclient folder
ON files Pclient	If disabled, your mobile will be enabled (activated) in the Server Pclient's specified Number File(s) in Pclient folder

SET – to set default Number Files

The SETx.EXE programs will copy content of “any” Number File to default¹ Number File for a client.

SETI myfile	Copy content of <i>myfile</i> to SysManSMS_Iclient.lst ¹ (in Iclient folder)
SETENV myfile	Copy content of <i>myfile</i> to SysManSMS_ENVclient.lst ¹ (in ENVclient folder)
SETVOICE myfile	Copy content of <i>myfile</i> to Voice_To_SMS.lst ¹ (in Iclient folder)

The idea is that you create number files for your different needs (example weekend.lst, normal.lst) and then copy such a file to the default¹ number file for the specific client, by sending an SMS message.

Example of usage:

SMS message “**seti weekend**” will copy weekend.lst to SysManSMS_Iclient.lst¹ in the Iclient folder. By adding a star (*) to your SMS message (**seti weekend ***), the program will also send a message to all members of the new number file, informing about the new active membership.

Modifications:

For better adoption to your environment, you can rename or copy the “set” programs to any name. You can also alter the 3 different default SMS text messages sent back by introducing an .INI file. Your .INI file will have to be named the same as your “set” program.

1) As a fourth parameter you can define another destination number file instead of the default

The following 4 text lines can be created (note the order must be as shown):

```
The OK message that is to be sent back to the requestor's mobile
The not OK message that is to be sent back to the requestor
The message to be sent to members of the new number file if * used
MyAlternativeNumberFileName
```

SF – to return file content to mobile

The mobile started program SF.EXE will return a specified file content to your mobile.

SF myfile Will return the 1530 first characters in the file `myfile.txt`

The location of such text files, are by default `..\SysManSMS\Fclient`. To adapt SF to your file location, edit the file `..\SysManSMS\GSMserver\Programs\SF.INI`

Inside SF.INI you may also specify how the SMS should be sent (delivery status and/or flash) You can of course rename SF.EXE to any name that fits your use.

SMStoFILE – store message text to file

SMStoFILE.exe (or whatever your copy is named) will produce a log file called `computername_progname.txt` in the servers OUTPUT folder.

If you for example make a copy called `Default.exe`, you will find a log file called `computername_default.log` in the OUTPUT folder of the server.

```
20-Sep-2011 15:00:36;+4744332211;default;UPCASED SMS TEXT STRING;07/09/20 15:00:01;
```

You can also use this program to automatic return a message to the senders mobile.

By just creating a text file called `yourprogramname.txt` in the *Programs* folder, your program will read this file and send the file content back to the person who started it. (Auto reply)

Such a text file may contain <newline> to split text to new lines on the mobile.

If you end your file text with a star “*” – the SMS will be sent as a flash SMS message !

Note: The program will not return any SMS to numbers with 8 or less digits (incl. international prefix)

SMStoMAIL – to forward SMS text as email

The Send Mail program SMStoMAIL is used to forward SMS messages to any email address.

You may create copies of SMStoMAIL for different usage of different keywords.

Copied images will expect .ini files with the new name, and will produce .log files with the new name.

NOTE: A copy of SMStoMAIL.exe called SM.EXE always exist to handle internal send mail functions.

SM test@sysman.no No lunch today Will email the text to `test@sysman.no`

SM.EXE will get its operation parameters from `SM.INI`, a file created by SMTP Setup settings.

`SM.INI` can be manually edited to override default settings. Also see chapter about the SMTP Server for details on how to setup powerful 2-way mail functions of your SysManSMS Server installation.

Copies of SMStoMAIL will use the new filename to look for it's .ini file and to create the log file.

SMStoSQL – to store SMS text to SQL database

The program SMStoSQL is used to store the incoming SMS messages to any SQL database. The first time you activate SMStoSQL (by double clicking on the file) it will produce its own .ini file. The SMStoSQL.ini is to be modified for your own settings for parameters like:

SQL: Servername – Databasename – Username – Password
 TABLENAME: Tablename
 COLUMN: Mobile Number – Message - SubmitTime

You can also have the program to automatic return a message to the senders mobile. By just creating a text file called `SMStoSQL.txt` in the *Programs* folder, your program will read this file and send the file content back to the person who started it. (Auto reply)
 Such a text file may contain <newline> to split text to new lines on the mobile.
 If you end your file text with a star "*" – the SMS will be sent as a flash SMS message !

You can make multiple copies of SMStoSQL, and .ini,.txt and .log files will follow the new names.

SN – to forward SMS to members of a Number File

SN.EXE (Send NumberFile) will forward a text sent from your mobile to members of a Number File(s).

SN myfile This is text	Will send your text to all members of Server lclient Number File <code>myfile.lst</code>
SN myfile1,myfile2,myfile3 This is text	Will send your text to all members of all specified files

You may modify the operation of SN.EXE by creating an .INI file (SN.INI) in the *Programs* folder. This text file can contain one or two lines. The first line is the name of a default Number File(s). If this line exists, SN.EXE will not take the Number File(s) name from the incoming SMS text, but instead use the name(s) from line 1 in the INI file. If a second line exists, this must contain an alternative to the built-in requestor reply (answer) SMS text.

SR – to do SMS Storm Reset for one number

SR.EXE will clear the senders Storm entry in the SMS Storm Control Database . This will typical be needed if your number has got a SMS Storm STOP, and has been disabled for more SMS's within current date. Sending SR to the server will enable you to again receive messages. Note: SMS Storm Database can be shared in a Dual-Server configuration, and this command will then reset the Storm counter for both systems.

SSTART – to start a Windows Service

SSTART.EXE program will start one of the services listed in your `ServiceMonitor.ini` file.

SSTART nl	Will start service with shortname NL on local Server
SSTART mypc:nl	Will start service with shortname NL on remote PC

For more information see chapter *Monitor important Windows Services on your system*.

SSTOP – to stop a Windows Service

SSTOP.EXE program will start one of the services listed in your `ServiceMonitor.ini` file.

SSTOP nl	Will stop service with shortname NL on local Server
SSTOP mypc:nl	Will stop service with shortname NL on remote PC

For more information see chapter *Monitor important Windows Services on your system*.

WC – WakeupCall request

You can send an SMS message to your SysManSMS Server and place a request for a WakeupCall at some upcoming time, either within the same date, or at any date and time ahead.

Format of the SMS to execute WC:

WC hhmm	Will setup a Wakeup Call at hh:mm on the current date
WC YYMMDDhhmm	Will setup a Wakeup Call at the specified date/time
WC hhmm Any text	Will setup a Wakeup SMS at hh:mm on the current date
WC YYMMDDhhmm Any text	Will setup a Wakeup SMS at the specified date/time

WHO – to check who is active in a Number File

To check who has status ACTIVE in a specific Number File, you can use the mobile program WHO.EXE. Rename WHO.EXE to what you want. A log will be produced telling activities and usage.

Format of the SMS to execute WHO:

WHO	Will send back a list of which Number Files you are a member of. The returned text will also tell which lists you are active in.
WHO myfile	Will send back a list of active numbers in the Server Iclient's Number File <code>myfile.lst</code>
WHO myfile Pclient	Will send back a list of active numbers in the Server Pclient's Number File <code>myfile.lst</code>

Processing incoming SMS – Save text to file

If an incoming SMS is not a server internal command, and not a program to execute, you will then have the possibility to store the message to a file for further processing by some other program.

In the SysManSMS Server's .INI file, you can specify if you want an incoming SMS to be stored to a file. You can specify one of 3 file types: XML – TEXT – INFOSYS (html) or MEDIA (special html) Files are stored in the servers OUTPUT folder. The location of this folder can be specified in the servers .INI file, and is normally ...\\SysManSMS\\GSMserver\\OUTPUT.

TEXT (computername_InfoPage.txt):

Single message received:

```
19-Sep-2011 11:33:44;+4790991394;12345678901234567890123456789012345678901234567890;10/09/19 11:33:39;
```

Multiple messages received:

```
19-Sep-2011 11:34:07;+4790991394;mp="01">1234567890123456789012345678901234567890123;10/09/19 11:34:03;
19-Sep-2011 11:34:09;+4790991394;mp="02">456789*1234567890;10/09/19 11:34:06;
```

XML (yymmddhhss_number.xml):

Single message received:

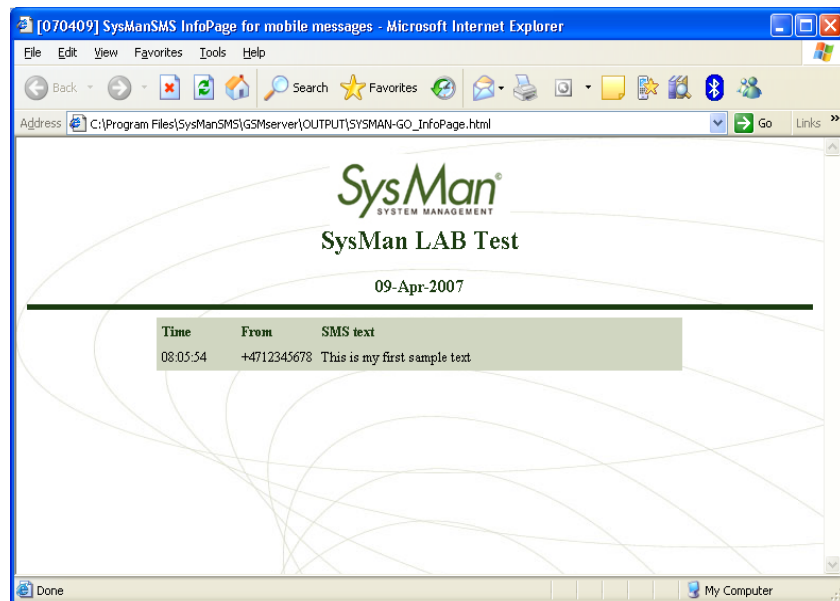
```
<?xml version="1.0" encoding="latin1"?>
<sms>
  <message>
    <time>19-Sep-2011 11:22:20</time>
    <number>+4790030920</number>
    <text>1123457890</text>
    <submitted>11/09/19 11:22:16</submitted>
  </message>
</sms>
```

Multiple messages received:

```
<?xml version="1.0" encoding="latin1"?>
<sms>
  <message>
    <time>19-Sep-2011 11:29:40</time>
    <number>+4790991394</number>
    <text mp="01">1234567890123456789012345678901234567890123</text>
    <submitted>11/09/19 11:29:36</submitted>
  </message>

  <message>
    <time>19-Sep-2011 11:29:42</time>
    <number>+4790991394</number>
    <text mp="02">456789*1234567890</text>
    <submitted>11/09/19 11:29:38</submitted>
  </message>
</sms>
```

Note: If multipart message, a flag will be inserted before start of text

HTML (computername_InfoPage.html):**Note:**

The file `computername_InfoPage.html` will automatically be renamed to an archive name after the first incoming SMS after midnight. The archive name will be: `computername_yymmddInfoPage.html`. You may also change the default logo by replacing the image file `InfoLogo.jpg` and the default background by replacing image `bakgrunn.jpg` – both found in the `.. \OUTPUT\images` folder.

Processing incoming SMS – Security Settings

SysManSMS Server has 2 first level security mechanism on incoming calls/SMS's.

Numbers listed in `SysManSMS_Stop.rig` will be stopped, and logged as a SECURITY stop.

If `SysManSMS.rig` exists, the server will check the incoming call or SMS number if it exists in the file.

If not, this is a security STOP. The call or message will not be processed, and a security log entry will be written to the servers log file. You may receive security notifications at security STOP. See below.

Limit mobile user access using the SysManSMS.rig file

If no security file exists, any mobile can send messages or dial-in to the server. **This is default !**

As soon as you create `SysManSMS_Stop.rig` or `SysManSMS.rig`, you will limit the access.

Note: An empty `SysManSMS.rig` will stop any remote attempt to send SMS messages to the server.

Rights files are plain text files, and need to be located in the GSMserver folder. Both files can be maintained by use of a Notepad. The `SysManSMS.rig` can better be modified with the utility `SysManSMS_RightsEdit.exe` located in the Utilities folder, or accessed via the Menu.

To start the Rights File Editor, use the SysManSMS Desktop Menu or double click on `SysManSMS_RightsEdit.exe` in the Utilities folder:

To register a new user, add the Number, Name, Rights code and optional a Program List.

NOTE:

If a Security Stop is detected and a Number File called `SecurityStop.lst` exist in the lclient folder, a security notification message will be sent to all numbers in this Number File, with the following text:

```
SecurityStop for number
MESSAGE=<The text from the received SMS>
```

Internal format of the SysManSMS.rig file

SysManSMS.rig is a standard flat text file located in the GSMserver folder.
Each line contain one user entry: *Number,Name,Rightscode:ProgramList*.

Mobile *Number* must include the international prefix! Separated by a comma, you add *Name*, a *Rights code* and optional, a *Program List* of programs the user is allowed to start. List start with a colon (:).
Note: If no list – user can start any program in the servers *Programs* folder.

Rights code is either “USR” (User) or “ADM” (Administrator). Both permit execution of programs, ADM also permit execution of internal server commands like “:TIME” and “:STAT” (See own chapter)

Record Format in the security file “SysManSMS.rig”:

Number, Name, Rights code : List of programs that can be started

Number	Specify the full international number for this user (max 15 characters)
Name	Specify a name to identify the user (max 20 characters)
Rights code	Specify ADM or USR rights (max 3 characters)
List of programs	Optional, specify a list of programs in the servers <i>Programs</i> folder that this user is allowed to start from his mobile. NOTE-1: If no list is specified, ANY program can be started by this user. NOTE-2: If a default program is specified to start, this is allowed to run, even if it's not specified in the list.

Example of a SysManSMS.rig file:

```
+4788662211, Ann, ADM
+4711223344, John, USR
+4712345678, John, USR : SF, setI, setENV
```

Creating a default answer-back message

If the SysManSMS.rig file is in place, you normally stop all incoming SMS messages from numbers not listed in the Rights file.

For some installations there may be a need to set up an automatic answer SMS for mobile users that are not listed in the SysManSMS.rig file.

This can be done by simply placing a file called DefaultAnswer.ini in the servers INPUT folder.
This file is a plain text file, and contain max 1530 characters of text for a default answer-back SMS.
The file can contain new lines, and this will show up as new lines on the mobile screen.

Notes:

If INPUT folder is configured with a sub-folder (named *computername*) you can either place the file in the COMMON INPUT folder, or in the LOCAL sub-folder. If found in COMMON; this file is pri #1.

By putting a star (*) as last character in the text, you will force the SMS to be a *flash* SMS

The Server will not return any SMS to masked numbers or numbers with 8 or less digits

Example of a DefaultAnswer.ini file

This is a SMS gateway with now incoming SMS allowed! Please contact our service desk.

SMS Incoming Text Filters to modify received SMS

Independent of your incoming action techniques, SysManSMS Server can search your received SMS text to see if it contains text strings that should convert, append or stop the SMS, before processing. 3 filter rules can be activated: *IncomingConvertText* – *IncomingAppendText* – *IncomingStopText*

SMS Incoming Text Filter activation

To enable SMS Incoming Text Filtering, you simply create a text pattern file inside the folder `Filters` located below the `GSMserver` folder. **After a SysManSMS Server restart, the filter will be active.**

`IncomingConvertText.txt` - contains strings to “convert” received text before processing
`IncommingAppendText.txt` - contains strings to “append” to received text before processing
`IncomingStopText.txt` - contains strings to “stop” received SMS from being processed

The files can contain multiple text patterns, one pattern on each line.

If a line should be active for only selected mobile numbers, start the line with colon (:) then enter the number(s) and end with a colon just before your text pattern.

Note: If using number directives, make sure you includes the country code. (+4712345678)

If more than one filter is active, filtering order is: 1) *Stop* 2) *Convert* 3) *Append*

`SysManSMS.log` file will contain information about Text Filter activation. In addition there will be one entry for each string match that results in a stop, convert or append action.

SMS Incoming Text Filter files format

The format for `Convert` and `Append` files are: **Original case sensitive text;Convert text to this**
 The character “;” is used to indicate where “from text” ends and “to text” begin. For this reason you can NOT specify patterns containing this character.

The format for `Stop` filter is straight forward, one string on each line. (Case sensitive)

For filter files, you can specify a filter pattern line to be active for selected mobiles only by starting the pattern line with **:mobile numbers:** This means, line should only be tested for the specified mobiles.

SMS Incoming Text Filter example

`IncomingConvertText.txt`

Convert this;to a new test like this
 @;
 :+4712345678:Peter;Potter

Text on left will be replaced with text on right
 If “@” found, it will be removed
 Peter becomes Potter for SMS from +4712345678

`IncommingAppendText.txt`

If this found;Append this to end of text
 ;Regards remote message center

If text on left found, right text will be added at end
 Right text will be added to EVERY incoming SMS

`IncomingStopText.txt`

If this text found do not handle SMS
 :+4712345678:OFFER

If string found, the SMS will not be processed at all
 Ignored if “OFFER” found in SMS from +4712345678

Warning:

If not using number specification for string definitions, the filter will execute on ALL incoming SMS's.

Server status information

From a mobile you can request a server status to be sent back. Send the text **:STAT** to the servers mobile number and wait for the SMS to be returned with a short status of your server.

Note: If security Rights list is active, make sure your mobile is enabled as ADM user.

If you got the SysManSMS Desktop Menu installed, you can get access to both a **Quick Status** and a full *computername_SysManSMS.html* status page, from right-clicking on the Menu icon.


A log file called *SysManSMS.log* in the *GSMserver* folder, will contain information about server start and stop, security issues and status of any task it performs. You can optional enable account logging.

Note: If *SysManSMS.log* contains more than 10000 entries, it will after next re-start, automatically archive the file into the name *SysManSMS_yyyymmddhhmm.log*, and create an empty new log file.

Example of SysManSMS.log file

```
02-Jun-2012 07:52:42 *****SERVER RESTART*****
02-Jun-2012 07:52:42 SysManSMS V8.1-0 31-MAY-2012 on computer DEV01-ACTIVE
02-Jun-2012 07:52:43 INFO - Server license accepted - assuming legal
02-Jun-2012 07:52:43 INFO - INPUT folder [c:\SysManSMS\INPUT\]
02-Jun-2012 07:52:43 INFO - OUTPUT folder [c:\SysManSMS\OUTPUT\]
02-Jun-2012 07:52:43 INFO - Looking for GSM device on port COM7: Speed=115200 baud
02-Jun-2012 07:53:32 INFO - Found GSM device 'MULTIBAND900E1800/ WAVECOM MODEM' - Supported=Y
02-Jun-2012 07:53:32 INFO - GSM Device INIT done, PIN OK, FW=651_09gg.Q2406B 070705 14:15
02-Jun-2012 07:53:32 INFO - Current GSM signal level is 17 (Range is between 10 and 31)
02-Jun-2012 07:53:33 INFO - GSM Device memory counters: SM=12 ME=0
02-Jun-2012 07:53:33 INFO - Current GSM network operator = [N Telenor] Operator ID = [24201]
02-Jun-2012 07:53:33 INFO - SysManSMS Watchdog found. OS=651_09gg.Q2406B 1478872 070705 14:15
02-Jun-2012 07:53:33 INFO - ACCOUNTING active on: SMStoOUT SMStoIN MAILtoSMS SMStoMAIL
02-Jun-2012 07:53:33 INFO - ACCOUNTING active with TEXT logging enabled
02-Jun-2012 07:53:33 INFO - SMTP Server enabled - Server Started OK
02-Jun-2012 07:53:33 INFO - Found Filter File [ConvertText.txt] - SMS Text Convert Filter act
02-Jun-2012 07:53:33 INFO - Sending re-start request to Slave Server [COM11]
02-Jun-2012 07:53:33 INFO - Service started successfully, entering SERVER loop
02-Jun-2012 08:13:07 CMDLINE - New Slave requested for COM12 OK
```

In addition to this, SysManSMS Server will produce a more readable .HTML file in its OUTPUT folder. In addition to a link to logfile, you will see things like GSM signal level, service client status line and a lot of server counters. This .HTML file will be updated about every 30 seconds.



[Products](#) | [Order](#) | [Downloads](#) | [Support](#) | [Contact](#)

Checking SysManSMS Server License Activation...

Status at 12-Sep-2015 04:08:19 Started on 12-Sep-2015 04:07:19 Current State: **STOPPED**
 Server V8.3-1.30-JUN-2014 - Running on ROALDB-PC - Licensed to SysMan Demo
 GSM Device on COM7 - GSM Operator - Voice Call FWD UNKNOWN - [View Server Log](#)

Sending files from: c:\SysManSMS\INPUT
 Receive data into: c:\SysManSMS\OUTPUT

GSM Signal: ---no or illegal gsm signal---

SMS SENT	0/0	SEND S-ERR	0/0	SEND H-ERR	0/0	SMS Storms	0	S-Filters	N
SMS REC	0/0	SEC STOP	0/0	SMTP REQ	0/0	Proc Create	0	Admin CMD	0
File REQ	0/0	Client REQ	0/0	SMTP start	0	SMSreport	0	MAILreport	0
GSM SIGNAL	0	DEV Reset	0	HTTP REQ	0/0	SM/ME mem	0/0	Watchdog	N

SMS Notification to: Administrators

© 2001 - 2015 SysMan AS | www.sysman.no | support@sysman.no

SysManSMS Server status

V8.1-0 31-MAY-2012 (RUNNING (SHARED))
 License: 000-000-000-000 (1 Master 4 Slave)
 Status: Product Activated
 Started: 31-May-2012 20:50:48

GSM Device Reset: 1 (COM6)
 GSM Network: NetCom
 GSM Signal: 19
 Voice Call FWD: NONE
 Data Stores: SEND=OK REC=OK
 SMS: Rec 11 Sent 125
 Send Soft Errors: 0
 Send Hard Errors: 0
 MAIL/SMS Reports: D=N/N S=Y/Y
 Watchdog Active: YES
 Next Alarm at: Not Active
 Service Monitoring: NO
 Text Filters: NO
 Admin Commands: 2
 Security Stops: 2
 SMS Storms detected: 2
 SMTP: Start Count 1 Req 58
 SNMP: Start Count 1
 HTTP: Req 43
 Last updated: 04-jun-2012 08:27:51

Example of *computername_SysManSMS.html* server status page and the Quick Status from Desktop Menu

Advanced Server options

Modifying the Servers INPUT message folder path

The Servers INPUT folder is used to hold any text file you like to send out as an SMS message. The Servers INPUT message folder are by default placed at `.\SysManSMS\GSMserver\INPUT`. This can be changed to other path or file share by modifying the servers INPUT folder in SETUP or editing the `SysManSMS.ini` file in the `GSMserver` folder. UNC specifications supported. If a remote file share are to be used, make sure it's reachable at server start-up time !

Note: At Startup, SysManSMS Server will check for number of files queued in INPUT folder, and if more than 20; an alert message file called `0000_FileAlert.xxx` will be written to INPUT folder. This message will be sent before any other, and will inform on how to delete queued messages.

Share Servers INPUT folder between multiple servers

If you got two or more SysManSMS Servers in your network, it's a good idea to let the servers share the same INPUT folder. This is fully supported by the servers, and will function as both a fail-over technique, and as a capacity increase. Multiple servers will read from same common folder. To enable a shared configuration, you will only have to manually create a sub-folder named *computername* inside common INPUT folder, for each server using this same INPUT folder. The sub-folder is called a LOCAL folder and the shared INPUT folder is called a COMMON folder. After the next restart, the server will find it's got a shared folder and then move any file it finds, to the LOCAL sub-folder before execution. This will inhibit conflicts with other servers reading same file.

You may share SERVER located Number Files between multiple Servers. For more details on how to setup shared Number Files, please see chapters about using Slave Servers with SysManSMS Server or how to do a Dual-Server Setup in *Advanced Server Options*.

File Filtering on INPUT folder

If other applications are reading from your message folder, you may need File Filtering to select which files should be picked up by SysManSMS Server.

File Filtering on message files works only on the INPUT folder (COMMON) and will in addition require a LOCAL sub-folder named *computername* inside the INPUT folder to exist.

Note: At start-up time, a sub-folder named *computername* will be created, if it not already exists.

File Filtering can be enabled by simply modifying the server's message file type specification, either by running SETUP or by directly working on the `SysManSMS.ini` and then do a restart.

The default (no File Filtering) is `*.txt`, meaning open to execute "any" file name of type `.txt`.

You can change this to any Windows Wildcard specification rule like `A?BC*.txt`.

Important:

File Filtering will work only on the top level INPUT folder, not on the server's LOCAL sub-folder.

For this reason, a local sub-folder named *computername* inside your INPUT folder must exist before File Filtering will work. (Also see chapter *Share Servers INPUT folder between multiple servers*.)

As soon as File Filter is detected, the servers own produced SMS message files, like *Service Monitor* alerts, will be created in the servers own sub-folder, not to be stopped by any common File Filtering.

If you create message files to be produced by one specific server, like *Repeating Messages*, you will also have to make sure this is created into this server's LOCAL sub-folder, to inhibit File Filtering stop.

Files in COMMON folder with extension `.XML` or `.TRA` will not be stopped by an active File Filter.

Saving processed message files

Normally you're SysManSMS Server will delete SMS message files after processing. To rename instead of deleting, edit `SysManSMS.ini` file to specify message file type, or use SETUP.

On the same line as you indicate which file type to be used, you then add a second parameter to tell the server to rename processed files to a new file extension: `*.txt,.bak`

Setup a SysManSMS Slave Server configuration

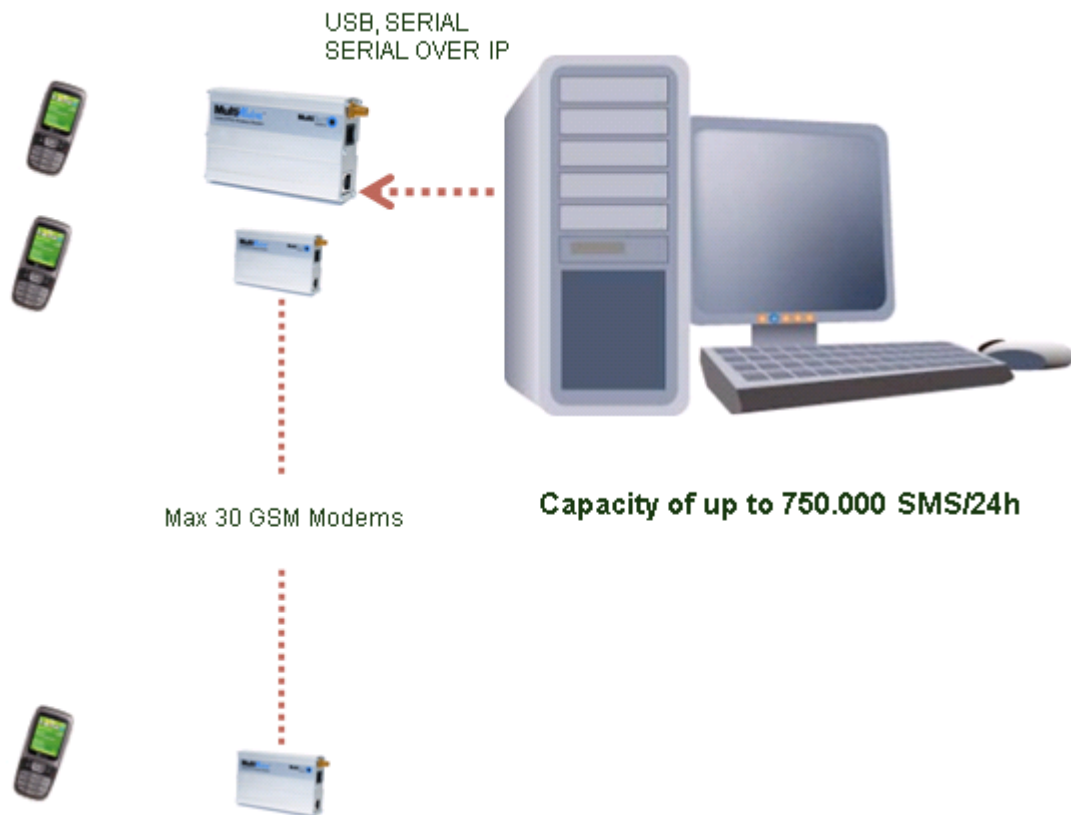
A SysManSMS Server with a supported license can create and manage up to 30 Slave Servers running on the same PC as the Master Server. After connecting a new GSM Modem to a free COM port, you simply tell the Master Server to start a new Slave service for this Modem.

All settings for the Master Server will be used also by Slaves, and from outside, the Master and its Slaves will be seen as one SMS Gateway resource, handling up to 750.000 SMS's per 24 hours.

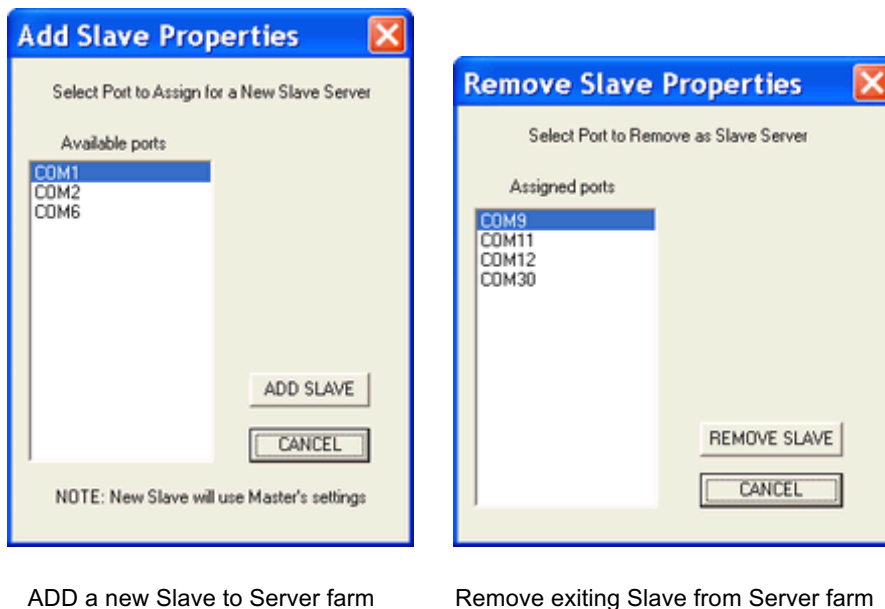
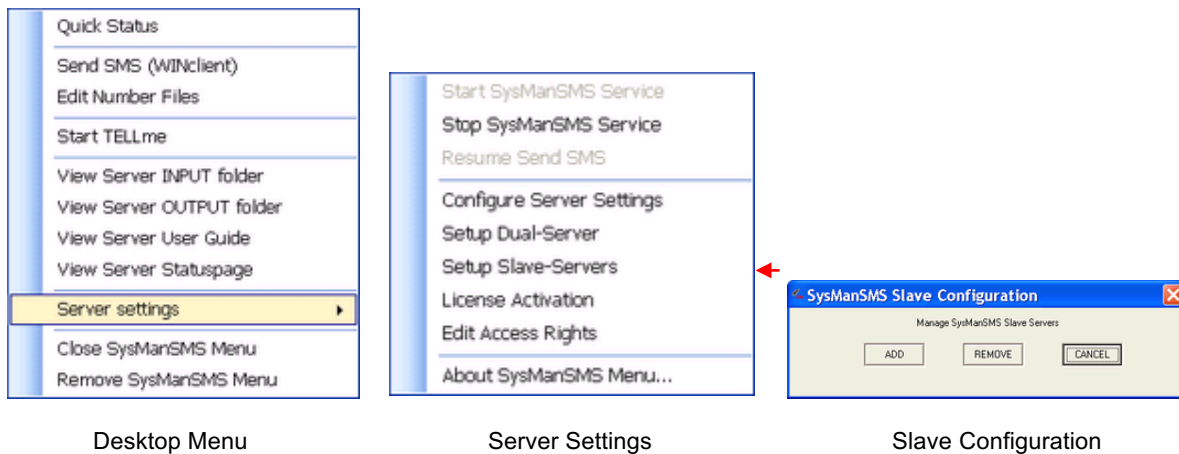
It's easy to ADD or REMOVE Slave Servers. This can be done from the SysManSMS Desktop Menu, or by sending Administrative SMS's to the Master Server. (see below)

Notes when using Slaves:

- The Master must run for the Slave to run. If Master is shutdown, Slaves will also terminate. If Master starts, Slaves will start automatically. Slaves share the Masters License key
- Both the Master's Quick Status and Server Status Page will show information about Slaves
- Make sure the Slave's Modems SIM cards has its PIN code check disabled
- Slaves will use the Masters INPUT Folder, but also create its private COMxx sub folder
- Slaves will use the Masters OUTPUT folder, but create its private named files
- Slaves will write its private log and dat files to the standard GSMserver folder, like the Master
- Remote Clients can direct its requests to a specific Slave by use parameter 1 as sub address
- Requests received via SNMP, SMTP or HTTP (urls) will as always be stored to common INPUT folder, and the first free Server (Master or Slaves) will handle the request



Configure SysManSMS Slave Servers using desktop menu



Configure SysManSMS Slave Servers using Administrator SMS

SMS adm command “:ADDSLAVE COMxx” will add a new Slave for COMxx to the Server farm
 SMS adm command “:REMSLAVE COMxx” will stop and remove existing Slave COMxx
 SMS adm command “:SHOWSLAVE” will return a SMS showing list of active Slaves

Note: The above commands are only supported on a SysManSMS Master Server.

For more information, see chapter about “Execute Server Administrator Commands”

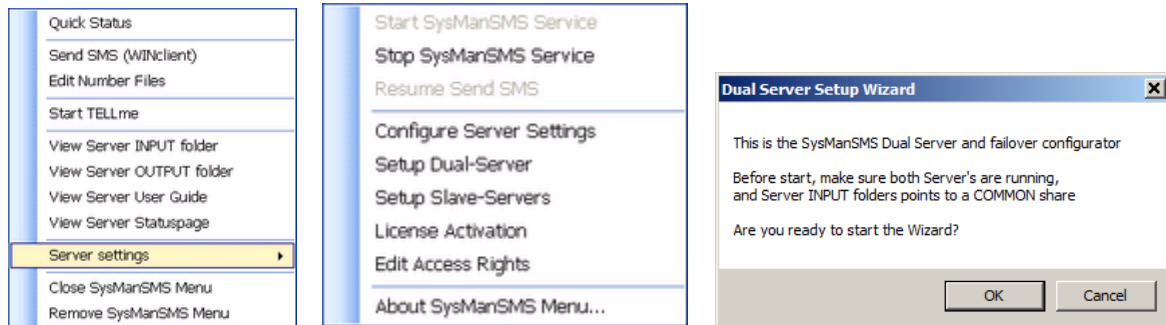
Setup a SysManSMS dual Server configuration

If you got multiple SysManSMS Servers on your network, you may setup failover between servers. The servers may also share both INPUT and OUTPUT folders in addition to Number Files. Pair of servers may also monitor each other via the SMS network by use of **:ALERT** function (keep-alive timer). Follow steps below to setup a dual-server configuration using the menu *Dual-Server Wizard*.

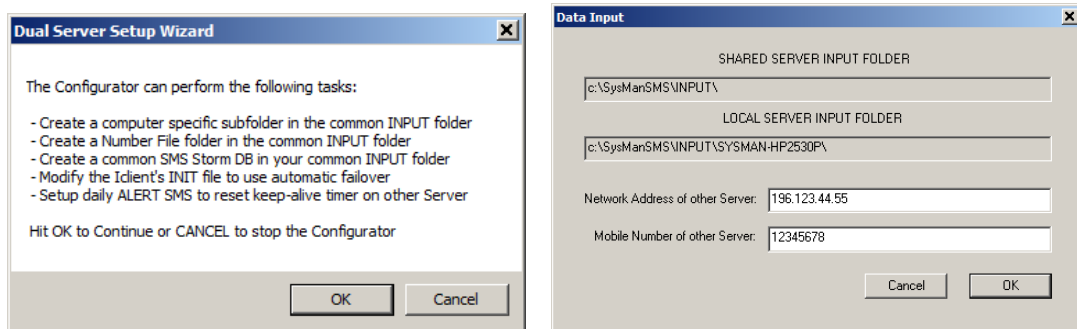
Step-1 Install your two SysManSMS Servers on two different Workstations or Servers.

Carefully follow the installation guide; make sure you test the SMS functionality.

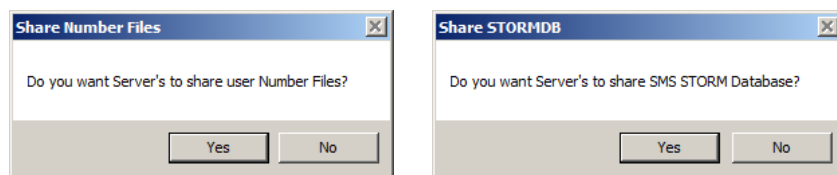
Step-2 On each server, start the “Setup Dual-Server” Wizard from SysManSMS Desktop Menu



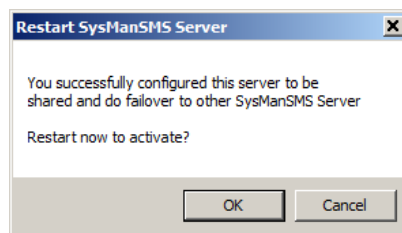
Start the Dual-Server Wizard



Hit OK and enter other servers name/IP and Mobile Number



You may now share your Number Files and SMS STORM DB as well



More info on dual servers – also see SysMan TechNotes on <http://www.sysman.no/downloads>

Logging account information to separate file

SysManSMS Server and some of its connected components are able to log activity to a special accounting file called `Accounting.csv` - located in the `OUTPUT` folder below `GSMserver` folder. For security reasons, `Accounting.csv` will not be moved if you redirect your `OUTPUT` folder. Account logging is specified at activity level in the file `Accounting.ini` found in the `GSMserver`. If not found, create one. Select only the needed activities, as accounting can produce overhead.

Example 1 of account logging specifications in the `Accounting.ini` file (accounting disabled):

ACTIVITY=NONE	No accounting active (default)
TEXT=NO	SMS text's will not be logged (default)
CLIENTNAME=NO	Client Name (sender) will not be logged (default)

Example 2 of account logging specifications in the `Accounting.ini` file (SMS out/in specified):

ACTIVITY =SMStoOUT,SMStoIN	Accounting active for outgoing and incoming SMS
TEXT=YES	SMS text's will be logged
CLIENTNAME=YES	Client Name (sender) will be logged

Note: Changes made to the `Accounting.ini` file, require SysManSMS Server to be restarted.

ACCOUNTING.CSV - Fields description								
TIME:	ACTIVITY:	MSGID:	FROM:	TO:	REALTIME:	STATUS:	TEXT:	CLIENTNAME:
x	SMStoOUT	x	action ¹	number	sms delivered	x	x	x
x	SMStoIN	x	number	action ¹	sms submitted	x	x	n/a
x	MAILtoSMS	n/a	mailaddr	number	n/a	x	x	x
x	SMStoMAIL	n/a	number	mailaddr	n/a	x	x	n/a

TIME: The date and time stamp for the log entry. Format: dd-mmm-yyyy hh:mm:ss

ACTIVITY: Indicate log entry activity: SMStoOUT SMStoIN MAILtoSMS SMStoMAIL
SMTtoOUT and SMStoIN activity logged by the GSMserver (SysManSMS.exe)
MAILtoSMS activity logged by the SMTPserver (SysManSMTP.exe)
SMTtoMAIL activity logged by the SendMail program (SMStoMAIL.exe)

MSGID: Used only by SMStoOUT or SMStoIN activity. Used to trace late delivery info.

FROM: Mobile number if SMStoIN or SMStoMAIL, mail address if MAILtoSMS

TO: Mobile number if SMStoOUT or MAILtoSMS, mail address if SMStoMAIL

REALTIME: Used only by SMStoOUT or SMStoIN activity. Log operator delivery/submit time

STATUS: Status is an integer and has two groups of codes: (X0=current X1=updated)
10 = DELIVERY-OK (11 = If late delivery update)
20 = DELIVERY-NOT-DONE (Delayed) (21 = If late delivery update)
30 = DELIVERY-NOT-POSSIBLE (31 = If late delivery update)
40 = DELIVERY-STATUS-NOT-REQUESTED
50 = DELIVERY-STATUS-UNKNOWN

TEXT: If logging of message text is active, this field contains the SMS message text.

CLIENTNAME: If CLIENTNAME active, client name logged if SMStoOUT or MAILtoSMS

¹⁾ *action* is the name of application that created the message (FROM) or was forwarded TO

Log samples from `Accounting.csv` and all activities turned on (6 lines of logging):

```
TIME;ACTIVITY;MSGID;FROM;TO;REALTIME;STATUS;TEXT;CLIENTNAME;
03-Sep-2011 10:41:44;SMStoOUT;070903104144110;File;+4512345678;07/09/03 10:42:24;20;"This is my first SMS";Myapp;
03-Sep-2011 10:43:20;SMStoIN;070903104144110;+4512345678;DeliveryNotify;07/09/03 10:43:30;11;;;
03-Sep-2011 10:44:01;SMStoIN;070903104144110;+4512345678;default.exe;07/09/03 10:44:25;10;"I return my SMS";;
03-Sep-2011 10:51:40;MAILtoSMS;;peter@sysman.no;+4712345678;;10;"This is my first mail to SMS test";SMTPserver;
03-Sep-2011 10:51:54;SMStoMAIL;;+4712345678;peter@sysman.no;;10;"This is my first SMS to mail test";;
```

Monitor important Windows Services on your system

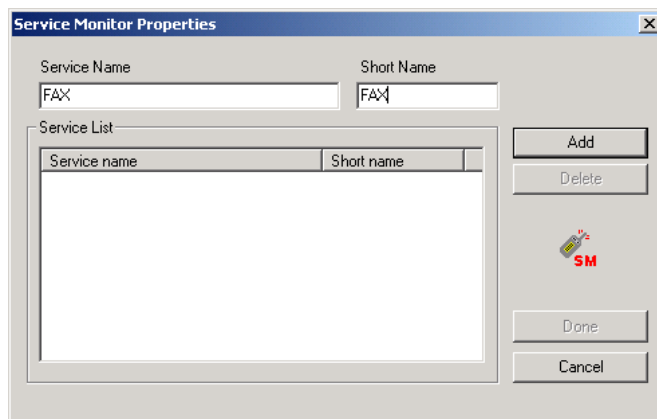
SysManSMS Server has a built-in capability of monitoring a limited amount of important services on your system. In addition you can use the supplied mobile programs SSTART and SSTOP to start and stop local or remote services - from your mobile! Log files will tell you about the activities.

Create a Service Monitor List

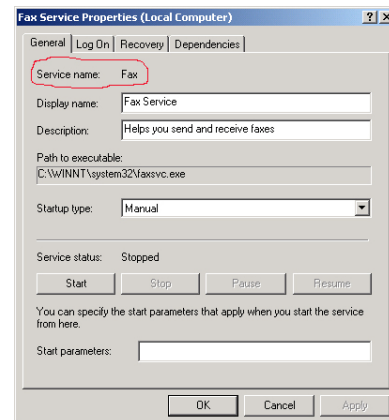
If the file `ServiceMonitor.ini` exist in the GSMserver folder, the SysManSMS Server will read this file at start-up, and get the list of up to 6 Windows Service Names to monitor.

The file will normally be created when running `SETUP.EXE`, or by a manually edit of the file.

If you got SysManSMS Desktop Menu active, you can also edit Server Settings via this Menu.



Service Monitor Registration from running SysManSMS SETUP



Example from Windows Service Control Panel
Note: Use "Service Name" when register a Service in the Service Monitor Properties.
NOTE: Case sensitive !

If the state of a service is changed, the SysManSMS Server will create an SMS message file called `Administrators.txt` in the servers INPUT folder, containing information on the event.

If exist, this message will be sent to all numbers in the `Administrators.lst` file, in Iclient folder. If this Number File does not exist, the default `SysManSMS_Iclient.lst` will be used.

You may delay any Service Monitor alert by adding a star (*) at end of the Service Name. (ex. **FAX***) The delay is fixed, and is approximately 3 Minutes, dependent on how busy the Server is. This can be useful if the monitored services are re-started from time to time, and you do not want to know about it.

Start a Windows Service from your mobile

Using the Short Name created in your `ServiceMonitor.ini` file as reference, you can send a start command to the service by use of an SMS message.

The format of the SMS should be: `SSTART shortname` or `SSTART remotePC:shortname`

The program `sstart.exe` is located in the servers *Programs* folder. You can rename it to what you like.

Stop a Windows Service from your mobile

Using the Short Name created in your `ServiceMonitor.ini` file as reference, you can send a stop command to the service by use of an SMS message.

The format of the SMS should be: `SSTOP shortname` or `SSTOP remotePC:shortname`

The program `sstop.exe` is located in the servers *Programs* folder. You can rename it to what you like.

Processing incoming Voice Call – send SMS

If Voice call's are not set to be forwarded, SysManSMS Server will recognise a dial-in call. Such a call will be logged in the server's log file, and if the number is available, this will be logged as well.

To setup Voice Call to be forwarded, please see the Server ADM commands :**VCFWD** and :**VCSTAT**

Automatic execution at incoming Voice Call

Start a program

You can set up your SysManSMS Server so that it automatically start a default program located in the *Programs* folder. If the dial-in number is "12345678" a program called `1234567.exe` in the *Programs* folder can be started. The rules for such program will be the same as if you start a program from an incoming SMS message. The dial-in number will be passed on as Parameter #1, but no text string will be passed to the program as parameter #2.

Note: Your server will need a two-way license to activate this function.

Send message to numbers in Number File

You can set up your SysManSMS Server so that it generates one or more SMS messages to be sent out if someone dials into the server from any kind of telephone.

By simply creating a Number File called `Voice_To_SMS.LST` in the *lclients* folder, you active this function for ALL dial-in numbers. Any voice call will generate SMS's to numbers in this global file.

Alternatively you can in the same folder (*lclient*) create number files named equal to the dial-in number, and the server will then use the numbers in this file, if the selected number dial-in.

This will be a number specific Number File, different from the global function.

If a number specific Number File exist, this will have priority over the `Voice_To_SMS.LST` file.

The default SMS text for such SMS messages is: "12345678 has dialed your Server"

This can be altered by editing the `SysManSMS.ini` file, setting the "File Store" parameter to `INFOSYS: Some text to be used instead of default text`

Note: Your server will need a two-way license to activate this function.

Voice Call – Security issues

You can limit the number of users that are allowed to dial-in and start programs or generate SMS messages, by using the SysManSMS Server's Rights file. By editing the `SysManSMS.RIG` file in the *GSMserver* folder, you can select the number(s) allowed to dial-in, and also which program you allow to be automatically started from the *Programs* folder.

If a Security Stop is detected and a Number File called `SecurityStop.lst` exist in the *lclient* folder, a security notification message will be sent to all numbers in this Number File:

```
SecurityStop for number
MESSAGE=<Voice Call received - No text>
```

For more information on the rights file, see chapter *Automatic start of programs after incoming SMS*.

Automatic conversion of message files

If the content of your message files do not look like you want it to, you can use a technique for automatic conversion of message files. By naming the message file with a file type of ".TRA", the server will start a program called `translate.exe` located in the GSMserver folder. This program can be developed by yourself or SysMan AS to do "anything" with your message files.

As soon as the SMS server finds a file in the INPUT folder with extension .TRA, the following happen:

- The file will be renamed to a temporary name (ex. `TRANSLATE_12345.TMP`)
- The program `translate.exe` will be started as sub process with the following two parameters:
 - o Parameter 1 = The message files temporary name
 - o Parameter 2 = The message files name after translation (ex. `TRANSLATE_12345.TXT`)

Such translation programs can be developed by customers, or on request from SysMan AS.

Note: The Translate function only support files in the common INPUT folder, not server subfolders.

The Command Window Interface to SysManSMS Server

By opening a command Window (DOS Window) you can communicate with the SysManSMS Server to do some special operations.

Click on the "Start" button in the left corner of your screen, and select the "Run..." menu, then "CMD". Use the "CD" command to move to directory ...\\SysManSMS\\GSMserver

Display Server Legal commands

```
> SysManSMS -HELP
```

Check Server Software Version

```
> SysManSMS -VERSION
```

Install Server in Windows Control Manager

```
> SysManSMS -INSTALL
```

Start the SysManSMS Server service

```
> SysManSMS -START
```

Stop the SysManSMS Server service

```
> SysManSMS -STOP
```

Pause the SysManSMS Server

```
> SysManSMS -PAUSE
```

Hold (freeze) the SysManSMS Server for 30 Seconds

```
> SysManSMS -HOLD
```

Resume the SysManSMS Server from Pause

```
> SysManSMS -CONT
```

Remove Server from Windows Control Manager

```
> SysManSMS -REMOVE
```

Use of Windows Environment Variables in SysManSMS

Windows Environment Variables can be used to change operation or add/remove functionality of the SysManSMS components. To manipulate Windows Environment Variables, see the Computer Properties settings, or use the DOS SET command.

SERVER

SysManSMSlocation

This can be used to point to the location where SysManSMS folder is placed.

Example "C:\Program Files\MyProgs\"

COMPUTERNAME

This is defined by Windows and contain the name of the computer.

SysManSMSnoservercmd

If defined to TRUE, server will stop execution of ":" internal server commands from any mobile.

(Example :STAT) If request for such commands arrive, it will then be handled as a security stop.

CMDclient – Iclient – ENVclient

SysManSMSsuccess

When the program terminates, it normally has an exit status code of "0" if its a normal termination.

You can change this be defining SysManSMSsuccess to the value tha fits your needs.

SysManSMSfailure

When the program terminates, it normally has an exit status code of "0" if its a normal termination.

You can change this be defining SysManSMSfailure to the value tha fits your needs.

SysManSMSonesms

By default, the client will split sms text that exceed 1530 characters into two SMS messages.

Be defining SysManSMSonesms to TRUE, it will always send just one SMS.

COMPUTERNAME

This is defined by Windows and contain the name of the computer.

SysManSMSaltnumfile

Normally all clients will use SysManSMS Number File format files (.LST) for sending messages.

On request, SysMan can deliver support for private (custom) formats. To open use for such files instead of standard Number Files, this Windows Environment Variable are to be used.

How to manually remove a SysManSMS installation

The setup program provides uninstall functionality by selecting *Uninstall*.

If you need to manually uninstall SysManSMS Server, please follow these steps:

1) From a DOS command window, go to the ..\SysManSMS\GSMserver location.

Execute the server command `SysManSMS -REMOVE`

Close your DOS command window.

2) Open the Windows Registry editor. (Run Regedit from Start menu)

Locate the entry `HKEY_LOCAL_MACHINE / SOFTWARE / SysMan`. Delete the entry SysMan.

Note: Use of Regedit should be done with care, and administrative knowledge is required.

3) With a Windows File Explorer, delete the folder ..\SysManSMS from your system disk.

(SysManSMS components are ALL placed in the folder ..\SysManSMS. No other files exist!)

Your SysManSMS Server installation is now completely removed from your system.

Adding SysManSMS Watchdog option to your SMS server

SysManSMS Watchdog option is a special software stored inside the GSM device. SysMan support the option on selected GSM devices from Nokia, Wavecom and MultiTech.

For updated information on device support, please see <http://www.sysman.no/support/devicesupport>

The Watchdog option will monitor the communication between the SysManSMS Server software and the GSM device. If the communication stops, (PC hang/stop or SMS service stops) the Watchdog software is capable of sending SMS alert messages on its own.

The delay from the point where communication stops and until the Watchdog sends its alarms, can be set in the `SysManSMS.ini` file to a value between 50 and 9999 seconds.

The mobile numbers to be told about the stop also have to be placed in the `SysManSMS.ini` file. A maximum of 8 mobile numbers can be given. Numbers can include international prefix like "+47".

To enable and configure Watchdog settings, use the *Configure Server Settings* from the SysManSMS Desktop Menu or manually edit the `SysManSMS.ini` file and re-start the server.

```
! Here you specify if you have the SysManSMS Watchdog hardware option installed
! To find, see label at bottom of your Nokia GSM unit.
! To enable this option, write "YES", to disable "NO"
!
YES
!
! Set your SysManSMS Watchdog timer value (minimum 50 seconds, maximum 9999 seconds)
! This is the delay after your server stop responding, until the Watchdog sends an SMS
!
180
!
! List the numbers to receive the emergency Watchdog SMS. Separate multiple numbers with a
comma (Max 8 numbers)
!
+4798878040,90030920
!
! End of file
```

To test the Watchdog, stop the SysManSMS Server from the Desktop menu and wait the amount of seconds you entered as your timeout value. The mobile should then get the alert SMS.



NOTE:

If you change settings in the `SysManSMS.ini` file – SysManSMS Server has to be re-started!

SysManSMS developer information

In addition to the send techniques (from files, from mails, form utilities) described earlier in this guide, SysMan also supply a small library for developers that will integrate sending of SMS from their own application. Such a tool kit is found on the CD/USB stick or at the SysMan web download page.

Use of SysManSMS SDK for sending SMS messages

For sending SMS messages from your own application, you can use the SysManSMS SDK. This SDK will connect you to the SysManSMS Server, and send one SMS, and get back the status of the operation. SysManSMS SDK consists of:

SysManSMS_SDK.h	for use when compiling applications in 'C'
SysManSMS_SDK.lib	for use by the linker
SysManSMS_SDK.dll	for runtime use when executing your application

NOTE: If you want to test the sdk without connecting to the server, set serveradr = "sdktest"

Description of the functions in the SysManSMS_SDK.dll

int SysManSMS_Hello (*servadr,*clientname);

This function sends a "hello" request to the SysManSMS Server. This function is being used for checking if you have a SysManSMS Server on your network, and if it can handle requests from your application. It opens a connection to the server, and this is closed at the exit.

This function you typical use once the application starts.

The SysManSMS Server will log to its log file that this "hello" request has been processed.

Parameters:

serveradr

string of maximum 16 characters, holding servers network address.

If Server and your client application are on the same PC, the address is set to "."

If not, use standard tcp/ip naming or the direct tcp/ip address.

NOTE: If you want to test the sdk without connecting to the server, set serveradr = "sdktest"

clientname

string of maximum 16 characters, holding your applications name. This will show up in the server's log file for identifying your application request.

Return status:

- 0 – SysManSMS Server found and ready to communicate
- 3 – No SysManSMS Server found with specified address. This can take some time to time out.
- 4 – Illegal logon rights to the server
- 5 – Illegal server address – maximum 16 characters
- 6 – Server indicated error
- 7 – Illegal client name – maximum 16 characters

int SysManSMS_Send (*servadr,*mobilenum,*smstext,*clientname);

This function contacts the SysManSMS Server specified in the "servadr" field via the network, and sends the sms message in "smstext" to the mobile number in "mobilenum". The "clientname" field specifies the name of the application that does the request. This function connects to the server, and the connection will be closed on exit.

The SysManSMS Server will log to its log file that this "send" request has been processed. In this way you can always identify the client program in the servers log file.

Note on text length and character set support:

SysManSMS Server supports both 7-bits GSM Character set and the international 8-bits ISO-Latin-1 character set. The Server will automatically detect 8-bits characters and send will then be performed in concatenated messages of up to 67 characters in each message, with a maximum of 10 messages. If only 7-bits characters (GSM character set) are used, text with more than 160 characters will be sent in concatenated mode of up to 153 characters in each message, with a maximum of 10 messages.

Parameters:

serveradr

string of maximum 16 characters, holding servers network address.

If Server and your client application are on the same PC, the address is set to "."

If not, use standard tcp/ip naming or direct tcp/ip address.

NOTE: If you want to test the sdk without connecting to the server, set serveradr = "sdktest"

mobilenum

string of maximum 16 characters, must be a legal mobile number

You can add parameters to a mobile number. Legal parameters are:

- # (+4712345678#) will DIAL the number instead of sending the SMS message
- * (+4712345678*) will send the sms as a "flash" sms message.
- + (+4712345678+) use delivery status, set MAX SMS lifetime
- (+4712345678-) use delivery status, set MIN SMS lifetime
- ? (+4712345678?) will not send any text sms, but just check if mobile is reachable.

smstext

string of maximum 1530 characters, characters from the GSM or ISO-Latin-1 character set.

Note1: The character "i" cannot be used, as it is used internally by the SDK.

Note2: Sending SMS text via SDK, also support "new line" function. Simply insert a Carriage Return <cr> (DEC=13/HEX=0D) character into your text to create a new line on the mobile.

clientname

string of maximum 16 characters, holding your applications name. This will show up in the server's log file, to identify your application request.

Return status:

- 0** – SysManSMS Server found, message sent without error to the operator.
- 1** – SysManSMS Server found, message sent without error to the operator, delivery delayed.
- 2** – SysManSMS Server found, message sent without error to the operator, operator does not support delivery status.
- 3** – No SysManSMS Server found at specified address. This can take some time to time out.
- 4** – Illegal logon rights to the server
- 5** – Illegal server address – maximum 16 characters
- 6** – Server indicated error
- 7** – Illegal client name – maximum 16 characters
- 8** – Illegal mobile number – error in parameter "mobilenum".
- 9** – Illegal sms text – illegal characters used or text more than 1530 characters long.
- 10** – Server in PAUSE mode after a mobile command ":STOP", can't send SMS now.

Content of SysManSMS_SDK.h

```
// SysManSMS_SDK.h  HEADER FILE
////////////////////////////////////

int SysManSMS_Hello(char * servadr,char *clientname);
int SysManSMS_Send (char * servadr,char * mobilenum,char * smstext,char * clientname);
```

Example code in 'C'

The following example will show use of SysManSMS_SDK in 'C' application:

```
#include "SysManSMS_SDK.h"
#include <stdio.h>
#include <windows.h>

int main()
{
    int err = 0;

    // Example of 'hello' function
    err = SysManSMS_Hello(".", "SDK-Client");
    if(err != 0) {
        printf("Error contacting server, err-code=%d\n",err);
        exit(1);
    }

    // Example of 'send' function
    err = SysManSMS_Send(".", "11223344", "This is a test sms", "SDK-Client");
    if(err != 0) {
        printf("Error sending sms message, err-code=%d\n",err);
        exit(1);
    }
    exit(0);
}
```

Test Number Files without sending messages

As you configure and test your SysManSMS installation, you may need to just send to a dummy number. Two such numbers are defined in the server, one number where the mobile is “on-line” and the other one for “off-line” testing. Simply use the following numbers for test:

11111111	to send dummy to a mobile with state “on-line”
22222222	to send dummy to a mobile with state “off-line”

Example code in 'VB'

```

Attribute VB_Name = "basGlobal"
Option Explicit

Public Declare Function SysManSMS_Hello Lib "D:\install\sysman_sdk\SysManSMS_SDK.dll"
    (ByVal servadr As String, ByVal clientname As String) As Long

Public Declare Function SysManSMS_Send Lib "D:\install\sysman_sdk\SysManSMS_SDK.dll"
    (ByVal servadr As String, ByVal mobilenum As String, ByVal smstext As String, ByVal
    clientname As String) As Long

Public Sub Main()
    Dim conn As New ADODB.Connection
    Dim rs As New ADODB.Recordset
    Dim strMessage As String
    Dim myMessage As String
    Dim myPhone As String
    Dim myPos As Long
    Dim retval As Long

    myMessage = ""
    myPhone = ""
    strMessage = Trim(Interaction.Command)
    If Len(strMessage) > 0 Then
        myPos = InStr(strMessage, " ")
        If myPos = 0 Then
            myPhone = strMessage
        Else
            myPhone = Left$(strMessage, myPos - 1)
            myMessage = Mid$(strMessage, myPos + 2)
            myMessage = Left$(myMessage, Len(myMessage) - 1)
        End If
    End If

    If myPhone <> "" Then
        conn.Open "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=D:\data\sysmantest\Messages.mdb"
        rs.Open "Messages", conn, adOpenStatic, adLockOptimistic
        rs.AddNew
        rs("time") = Now
        rs("Phone") = myPhone
        If myMessage <> "" Then
            rs("Message") = myMessage
            retval = SysManSMS_Send(".", myPhone, "Message received.", "mldHandling")
        End If
        rs.Update
        rs.Close
        conn.Close
    End If
End Sub

```

SysManSMS SDK as Microsoft COM object

As part of our SDK delivery, we also supply a COM object to be used with VB, VBScript and other. The object is called `SysManSMS_COM.dll` and has one Interface **ISysManSMS** and two methods called **Hello** and **Send**. The interface is exactly the same as the 'C' .dll, and return same status codes.

Register `SysManSMS_COM.dll` with `regsvr32` and open a VB project. Under *References* in the *Project* menu you should now find `SysManSMS_COM 1.0 TypeLibrary`. Make it available, click OK.

Example code for Hello method:

```

Dim res As Integer
Dim sysman As SYSMANSMS_COMLib.SysManSMS
Set sysman = CreateObject("SysManSMS_Com.SysManSMS")
res = sysman.Hello("myserver", "mytestclient")

```

Appendix-1

7 bit GSM character set

Bits per character: 7

SMS User Data Length meaning: Number of characters

CBS/USSD pad character: CR

Character table:

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
B4	b3	B2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0	@	Δ	SP	0	i	P	ı	p
0	0	0	1	1	£	—	!	1	A	Q	a	q
0	0	1	0	2	\$	Φ	"	2	B	R	b	r
0	0	1	1	3	¥	Γ	#	3	C	S	c	s
0	1	0	0	4	è	Λ	¤	4	D	T	d	t
0	1	0	1	5	é	Ω	%	5	E	U	e	u
0	1	1	0	6	ù	Π	&	6	F	V	f	v
0	1	1	1	7	ì	Ψ	'	7	G	W	g	w
1	0	0	0	8	ò	Σ	(8	H	X	h	x
1	0	0	1	9	Ç	Θ)	9	I	Y	i	y
1	0	1	0	10	LF	Ξ	*	:	J	Z	j	z
1	0	1	1	11	Ø	1)	+	;	K	Ä	k	ä
1	1	0	0	12	ø	Æ	,	<	L	Ö	l	ö
1	1	0	1	13	CR	æ	-	=	M	Ñ	m	ñ
1	1	1	0	14	Å	ß	.	>	N	Ü	n	ü
1	1	1	1	15	å	É	/	?	O	Ş	o	à

1) This code is an escape to an extension of the 7 bit default alphabet table. A receiving entity which does not understand the meaning of this escape mechanism shall display it as a space character.

Appendix-2

7 bit GSM character set extension table

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
B4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0								
0	0	0	1	1								
0	0	1	0	2								
0	0	1	1	3								
0	1	0	0	4		^						
0	1	0	1	5							2)	
0	1	1	0	6								
0	1	1	1	7								
1	0	0	0	8			{					
1	0	0	1	9			}					
1	0	1	0	10	3)							
1	0	1	1	11		1)						
1	1	0	0	12				[
1	1	0	1	13				~				
1	1	1	0	14]				
1	1	1	1	15			\					

In the event that an MS receives a code where a symbol is not represented in the above table then the MS shall display the character shown in the main default 7 bit alphabet table in section 6.2.1

- 1) This code value is reserved for the extension to another extension table. On receipt of this code, a receiving entity shall display a space until another extension table is defined.
- 2) This code represents the EURO currency symbol. The code value is that used for the character 'e'. Therefore a receiving entity which is incapable of displaying the EURO currency symbol will display the character 'e' instead.
- 3) This code is defined as a Page Break character and may be used for example in compressed CBS messages. Any mobile which does not understand the 7 bit default alphabet table extension mechanism will treat this character as Line Feed