
SMS Messaging Server 4.0

Administrator's Manual

June 2005

© 1999-2005 - ActiveXperts Software B.V.

<http://www.activexperts.com>

info@activexperts.com

ActiveXperts
software

1. INTRODUCING ACTIVEXPERTS SMS MESSAGING SERVER 3

- 1.1. PRODUCT OVERVIEW 3
- 1.1. PRODUCT FEATURES 3

2. SHORT MESSAGE SERVICE (SMS) 5

- 2.1. INTRODUCTION 5
- 2.2. GSM MODEMS 5
- 2.3. SMPP PROVIDERS 6

3. SMS MESSAGING SERVER DESIGN 7

- 3.1. INTRODUCTION 7
- 3.2. CHANNELS 8
- 3.3. TRIGGERS 8

4. INSTALLING ACTIVEXPERTS SMS MESSAGING SERVER 10

- 4.1. HARDWARE REQUIREMENTS 10
- 4.2. INSTALLING ACTIVEXPERTS SMS MESSAGING SERVER 10

5. CONFIGURING ACTIVEXPERTS SMS MESSAGING SERVER 12

- 5.1. INTRODUCTION 12
- 5.2. OPTIONS (GENERAL PROGRAM SETTINGS) 12
- 5.3. CHANNELS (COMMUNICATION SETTINGS) 13
- 5.4. TRIGGERS (PROCESSING INCOMING MESSAGES) 17

6. MESSAGE DATABASE 18

- 6.1. INTRODUCTION 18
- 6.2. MESSAGE DATABASE - TABLES AND FIELDS 18
- 6.3. ARCHIVING MESSAGES 21
- 6.4. MIGRATING TO MS SQL 22

7. TROUBLESHOOTING 23

- 7.1. INTRODUCTION 23
- 7.2. ACTIVEXPERTS SMS MESSAGING SERVER MONITOR 23
- 7.3. TRACING 23

8. PERFORMANCE TUNING 25

- 8.1. INTRODUCTION 25
- 8.2. MESSAGE DATABASE 25
- 8.3. QUEUE DIRECTORY 26

9. SMS MESSAGING SERVER API 27

- 9.1. INTRODUCTION 27
- 9.2. 'CONSTANTS' OBJECT 27
- 9.3. 'MESSAGES' COLLECTION OBJECT 28
- 9.4. 'MESSAGE' OBJECT 30
- 9.5. 'OPTIONS' OBJECT 30
- 9.6. 'CHANNELS' COLLECTION OBJECT 31
- 9.7. 'GSMMODEMCHANNEL' OBJECT 32
- 9.8. 'SMPPCHANNEL' OBJECT 34
- 9.9. 'POP3CHANNEL' OBJECT 35
- 9.10. 'SMTPCHANNEL' OBJECT 36
- 9.11. 'TRIGGERS' COLLECTION OBJECT 37
- 9.12. 'TRIGGER' OBJECT 38
- 9.13. SAMPLES 39

10. CASE STUDIES 40

1. Introducing ActiveXperts SMS Messaging Server

1.1. Product Overview

SMS Messaging Server is an SMS messaging framework that enables companies to send, receive and process SMS messages. The framework is designed support virtually any scenario where low-and high volume SMS messaging is required.

Use SMS Messaging Server in the following scenarios:

- ♦ Mobile Users are able to query databases for stock quantities, new orders or customer information
- ♦ Mobile users can receive critical information while they are away from the office
- ♦ Mobile users can receive notifications when important e-mails (from specific users or with specific subjects) have arrived
- ♦ Companies can route urgent calls to the service personnel
- ♦ Companies can send daily (stock) prices to their personnel or customers
- ♦ Remote workers can update their time sheets
- ♦ IT personnel can remotely restart services or reboot servers
- ♦ Multimedia companies – like radio stations and television stations – can setup their own voting system, enabling customers to bring out votes and request voting reports
- ♦ Entertainment companies - like bars and dancing's - can send out information about special events to their customers
- ♦ Emergency Response Centers can offer a service to locals to report crime by a short message

This is just selection of scenarios. There are a lot of other scenarios where SMS Messaging Server can be used.

SMS Messaging Server consists of the following components:

- ♦ A Windows Service, running in the background. This service is responsible for sending, receiving and processing SMS- and e-mail messages;
- ♦ A Configuration Database, containing all configuration parameters. Can be any OLE/DB compliant database, like MS Access, MS SQL and Oracle;
- ♦ A Message Database, containing all in- and outgoing messages. Can be any OLE/DB compliant database, including MS Access, MS SQL and Oracle;
- ♦ A Manager application to make changes to the configuration and view messages
- ♦ A Monitor application to view real-time what's happening inside the system
- ♦ An ASP-based Web interface to allow access from any where in the world
- ♦ An API to allow custom scripting
- ♦ A collection of samples showing how to process incoming messages

1.1. Product Features

SMS Messaging Server features the following:

- ♦ SMS Send and SMS Receive, on multiple channels simultaneously;

- ◆ Support for GSM Modems and GSM phones (GSM 07.05 and GSM 07.07 specifications) for low- and medium volume messaging;
- ◆ Support for SMPP-compliant SMSC centers (SMPP 3.x) for high-volume messaging;
- ◆ Central, vendor-independent database to store messages;
- ◆ Support for multi-part messages;
- ◆ Support for SMS delivery verification;
- ◆ Full TAPI (Windows Telephony) support;
- ◆ SMTP- and POP3 support;
- ◆ Advanced logging and tracing;
- ◆ Integration of VBScript for processing incoming messages;
- ◆ Multi-threading architecture; allows sending/receiving messages on multiple channels (modems, SMPP providers, SMTP, POP3) simultaneously;
- ◆ Framework which can be fully customized.

2. Short Message Service (SMS)

2.1. Introduction

The Short Message Service (SMS) is part of the GSM specification and allows messages to be sent to and from GSM mobile networks throughout the world. A single short message can contain up to 160 characters and comprise of words, numbers or an alphanumeric combination.

Short messages can be received with voice, data and fax calls. SMS also provides confirmation that a short message has been delivered to its destination. Non-textual short messages can also be sent to carry 8-bit binary data. Messages comprising of Unicode character sets which include Arabic and Chinese characters can also be carrier in SMS.

SMS is a store and forward service where a short message is sent via a Short Message Service Center (SMSC). An advantage of this is that the destination mobile device does not have to be on the network at the time when the message is sent. If a destination mobile device is not available at the time the message is sent, the SMSC service center will retry to deliver the message.

Delivery of a short message takes only a few seconds from SMSC to the mobile device.

SMS messages can be up to 140 octets or 160 characters in length and can carry information coded in different ways. The most common ‘coding scheme’ is the GSM default alphabet. This allows a simplified text alphabet to be coded into 7 bits per character.

Most advanced applications will typically use 8-bit data where the SMSC makes no assumptions on the coding scheme and allows applications to use the 140 octets as they wish.

2.2. GSM modems

SMS Messaging Server supports GSM modems to send and receive SMS Messages. It supports a broad range of GSM modems that support the ETSI GSM 07.05 guidelines. Modern GSM devices (made by Nokia, Siemens, Sony Ericsson, etc.) use the similar subset of AT+C command set and are – as a result – also supported by ActiveXperts SMS Messaging Server. You can connect a GSM modem through one of the following interfaces:

- ◆ Serial port;
- ◆ USB interface;
- ◆ Bluetooth;
- ◆ Infrared.

SMS Messaging Server supports TAPI drivers (like ‘Standard 9600 bps Modem’) to make configuration of GSM modems easier. Using TAPI, you can configure the baud rate, initialization strings, etc. through the standard Windows Control Panel.

ActiveXperts recommends MultiTech MTCBA-G-UFx modems and WaveCom 1306B GSM modems.

ActiveXperts bundles the SMS Messaging Server software with a MultiTech modem:

- ♦ License of ActiveXperts SMS Messaging Server;
- ♦ MultiTech MTCBA-G-UF1 (900/1800 MHz, Europe) or MultiTech MTCBA-G-UF2 (850/1900 MHz, North America);

For more information and pricing, please check the ActiveXperts web site at www.activexperts.com/sales/multitechsmsserver/.

2.3. SMPP providers

SMS Messaging Server supports SMPP 3.x compliant SMSC providers.

The SMPP protocol allows a client (i.e. ActiveXperts SMS Message Server) to access the SMSC's ('Short Message Server Center') systems to send and/or receive SMS messages. To allow connection between SMS Messaging Server and an SMPP compliant SMSC center, an IP connection between these two systems is required, either directly or through the Internet.

Usually, an SMSC center provides multiple sites, to ensure that the service is always available. An SMSC provider ensures high-volume messaging, usually around 10 SMS messages per second, or even more.

You need to sign-up with an SMPP-compliant center before you can actually use their service. It normally requires a small sign-up fee, and a fee for an SMS bundle per month.

SMS Messaging Server provides free SMPP messaging for a limited number of SMS messages. You can connect to the ActiveXperts SMPP server and send/receive messages right after installation, without the need to sign-up with an SMSC center first.

If you want to know about a list of SMPP providers recommended by ActiveXperts Software, [click here](#).

3. SMS Messaging Server design

3.1. Introduction

The SMS Messaging consists of the following components (see *Figure-1*):

- ♦ **SMS Messaging Server service** – multi-threading SMS Messaging engine;
- ♦ **Configuration Database** – a vendor independent configuration database;
- ♦ **Message Database** – vendor independent database where all incoming and outgoing messages are stored;
- ♦ **Applications and Scripts**;
- ♦ **GSM Modem channels and SMPP Provider channels**;
- ♦ **SMTP and POP3 channels**;
- ♦ **VBScript Triggers** – to process incoming messages;

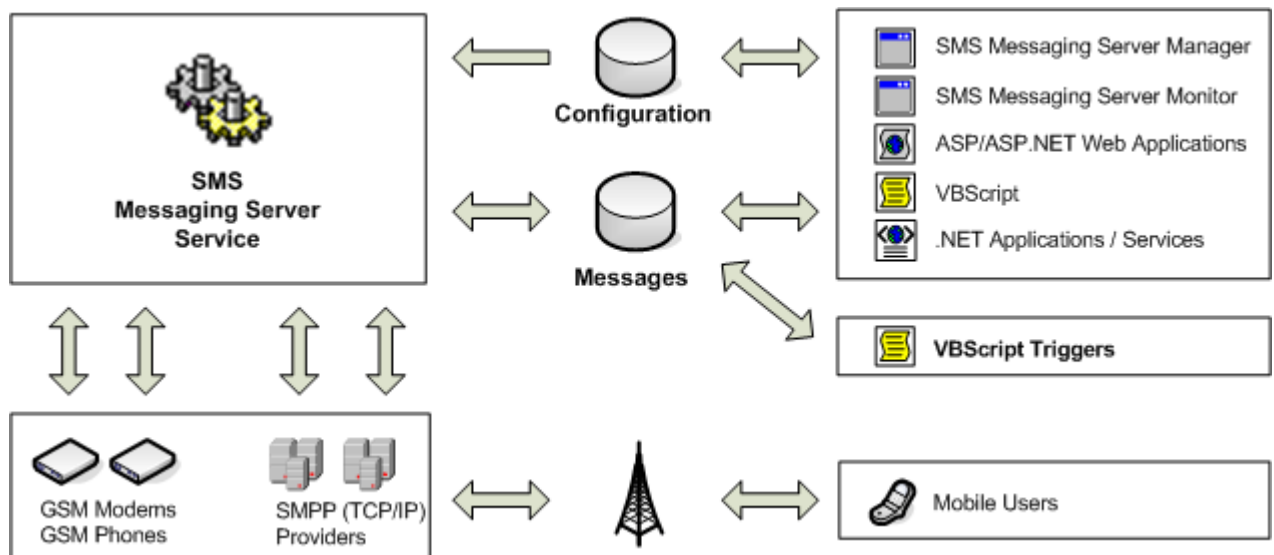


Figure 1 – SMS Messaging Server Design

The **configuration** information is stored in a **database**. This database can be any OLE/DB compliant database, including MS SQL, MS Access, MySQL and Oracle. By default, SMS Messaging Server is shipped with an MS Access configuration database:

Configuration.mdb.

Messages are also stored in an OLE/DB compliant **database**, separate from the configuration. By default, SMS Messaging Server shipped with an MS Access message database:

Messages.mdb. Migration to MS SQL can be done easily.

The **SMS Messaging Server service** is responsible for controlling the so called 'Channels'. A Channel is a communication device or communication protocol.

The SMS Messaging Server service reads messages from the Message Database and provides these to the designated channel; it also polls the channels regularly to see if there are any new incoming messages.

The SMS Messaging Server triggers VBScript programs whenever there are new messages in the Message Database. Scripts are only triggered when a certain condition is matched.

A **Channel** is a communication device or a communication protocol. You can define a 'Channel' for each GSM Modem, SMPP provider, POP3 connection and SMTP connection. These Channels are stored in the Configuration Database. The SMS Messaging Server service reads the Channel configuration from the Configuration Database and controls each Channel by a separate process (also called 'thread').

The **SMS Messaging Server Manager** enables you to change the Configuration Database. Using this Manager application, you can create new Channels, modify existing Channels, delete Channels and modify general options. This Manager application can also be used to view Messages and define Filters on the Message database.

SMS Messaging Server is shipped with a selection of **ActiveX/COM objects** to manage the Configuration Database and the Message Database. This way, Administrators can fully customize the User Interface of the product; it also enables them to generate new messages themselves from a script or a custom application.

The configuration database includes information about the channels (GSM channels, SMPP channels, POP3 channels and SMTP channels) as well as information about.

Messages are also stored in an OLE/DB compliant, central **database**. By default, SMS Messaging Server is shipped with an MS Access message database (Messages.mdb).

3.2. Channels

A 'Channel' is a communication device or a communication protocol. Messages are sent and received by such a Channel. You can define 'Channels' for each of the following devices/protocols:

- ♦ GSM Modem
- ♦ SMPP compliant SMSC provider
- ♦ POP3 server
- ♦ SMTP server

The properties of a channel depend on the type of channel.

You can define a 'Channel' for each GSM Modem, SMPP provider, POP3 connection and SMTP connection. These channels are stored in the Configuration Database. The SMS Messaging Server service reads the Channel configuration from the Configuration Database and controls each Channel by a separate process (also called 'thread').

3.3. Triggers

Triggers are defined to process incoming messages.

Each trigger is a set that includes a condition and an associated script. The condition is an SQL-like condition that selects a set of unprocessed, incoming messages.

As soon as a new message is received through one of the channels, the status of the message is set to 'Not Processed'. The SMS Messaging Server regularly (by default: once every 5 seconds)

checks if there is any unprocessed messages. As soon as there's an unprocessed message, it will trigger all scripts that match a predefined condition.

Let's demonstrate this with a small sample:

A system is running ActiveXperts SMS Messaging Server. It has several projects running simultaneously; one of the projects is a workflow system, where 3 pre-defined persons (+4400000001, +4400000002 and +4400000003) can enter their workflow through SMS.

At the same time, the system is able to accept messages from anybody else, to query the product database. To verify validity of the product database query, the keyword 'PRODUCT' is required, otherwise an error message is replied to the sender.

The Trigger-configuration is shown in the following table:

Condition	Script
Sender = '+4400000001' OR Sender = '+4400000002' OR Sender = '+4400000003'	Projects\Workflow\Triggers\WorkFlow.vbs
Body LIKE '%PRODUCT%'	Projects\ProductInfo\Triggers\ProductInfo.vbs
ANY	Projects\Triggers\UnknownMessage.vbs

In the above table, the WorkFlow.vbs will be called if the sender's mobile number matches either '+4400000001', '+4400000002' or '+4400000003'.

Let's assume that the sender's number equals '+4400000001'. WorkFlow.vbs sets the status of the message to 'Processed' and will create a reply in the message database (Saying 'Thank you for the workflow').

The other scripts will not be triggered, because WorkFlow.vb has set the status to 'Processed'.

If the sender's number does not match '+4400000001', '+4400000002' or '+4400000003', the server will check if the string 'Product' can be matched against the message body. If so, it will call ProductInfo.vbs. If not, it will call UnknownMessage.vbs.

4. Installing ActiveXperts SMS Messaging Server

4.1. Hardware Requirements

The ActiveXperts SMS Messaging Server service only runs on a Windows workstation or server platforms, and must meet either of the following requirements:

	Windows NT4 Server	Windows 2000 Server	Windows XP Professional	Windows 2003 Server
CPU	486DX/33	P133 or higher	P233 or higher	P550 or higher
Memory	96 MB	128 MB or more	128 MB or more	256 MB or more
Service Pack	SP5 or higher	SP1 or higher	Not required	-
Available Diskspace	5 MB	5 MB	5 MB	5 MB

It's recommended to install the software on a Windows Server platform, because server platforms are optimized for application services.

The InstallShield Installation Wizard will take you through the installation.

4.2. Installing ActiveXperts SMS Messaging Server

Before you start the installation, you must assign one of your NT/2000/XP/2003 servers in your network as the central SMS messaging server. The machine will host the ActiveXperts SMS Messaging Server software.

Download the installation file `ASMSSRV.R EXE` from the internet

(<http://www.activexperts.com/download>) and run this installation program on the server that you assigned as the central SMS messaging server.

The following components will be installed on that server:

- ◆ ActiveXperts SMS Messaging Server **Service** – the actual SMS Messaging Server service;
- ◆ ActiveXperts SMS Messaging Server **Manager** – graphical application that allows you to change the configuration;
- ◆ ActiveXperts SMS Messaging Server **Monitor** – shows real-time information about all available channels and scripting engine;
- ◆ ASP Web Interface – allows you to view the message database from the Internet Browser, to create new messages and to delete messages;
- ◆ Documentation – Help files and other useful documentation files.

Step 1 – Welcome Message

This is where the installation begins.

Step 2 – Registration information

Here, you can enter your Registration code. If you want to try the software, enter 'EVALUATION' as the registration code. You will be able to use the full-functioning software for 30 days. When you decide to buy after 30 days, you don't need to re-install the product; the Manager application allows you to enter the registration code.

Step 3 – Choose Destination Folder

Choose a destination folder. Setup will copy the Service, the Manager application, the Monitor application, the ActiveX/COM objects and Documentation files to this location. Make sure to have approximately 10 MB of free hard disk space available.

Step 4 – Setup Type

Choose 'Typical' to install most common options (recommended), choose 'Compact' to install only the service component, or choose 'Custom' to choose your own installation options.

Step 5 – Select Program Folder

Specify a name for the Program folder.

Step 6 – Service Account

The 'ActiveXperts SMS Messaging service' must run with specific credentials. Make sure you assign at least Local Administrator privileges, to make sure that SMS Messaging Server has access to all local resources (including databases and devices).

Step 7 – Finish

The software is installed now. You can optionally launch the Manager application when you press 'Finish'.

5. Configuring ActiveXperts SMS Messaging Server

5.1. Introduction

ActiveXperts SMS Messaging Server's configuration information is stored in the 'Configuration Database'. This database is stored in a files called CONFIGURATION.MDB, and is located in the <INSTALL-DIR>\CFG directory.

This database includes the following tables:

- ◆ **OPTIONS** table – This table stores generic program settings, including log settings, location of the directory for temporary files, location of the message database, and so on;
- ◆ **TRIGGERS** table – This table stores the scripts that can be triggered when a new message arrives in the system, including the condition and location of the script;
- ◆ **CHANNELS_GSMMODEM** table – This table stores configuration information of all GSM modems connected to the server;
- ◆ **CHANNELS_SMPP** table – Each record in this table includes information like: IP address of the provider, login and password, and so on;
- ◆ **CHANNELS_POP3** table – Each record in this table includes the hostname (or IP address) of the POP3 server, as well as individual POP3 accounts
- ◆ **CHANNELS_SMTP** table – Contains information about the SMTP server, including authentication information (if required).

Use the ActiveXperts SMS Messaging Manager to make changes to the configuration. Alternatively, you can use the SMS Messaging Server API to access to the configuration and make changes to it.

5.2. Options (General Program Settings)

To change the General Program Options, launch the Manager application from the Start menu, open the **Configurations** folder and click on **Options**.

The following items can be configured:

- ◆ **Logging Enabled/Disabled** – Enable or disable logging;
- ◆ **Log to File / Log to Database** – Choose whether the software should write logging information to a plain ASCII file or to an OLE/DB compliant database;
- ◆ **Log File** – Specify information required to log to a plain ASCII file. Each record is written as a single line to a plain ASCII file; fields are separated by a configurable separator;
- ◆ **Log Database** – By default, SMS Messaging Server logs all activities to the following OLE/DB compliant database: <INSTALL-DIR>\LOGS\LOG.MDB;
- ◆ **Temporary Files Directory** – Location of the directory where the service can write its temporary files. This field should be left default unless there's not enough storage on the particular drive;

- ◆ **Queue Files Directory** – Location of the directory where the service writes incoming and outgoing messages. To improve performance, you can assign a different directory (for instance: a high-speed RAID configuration);
- ◆ **Message Database Connection String** – Indicates the location of the message database. By default, this connection string is set to <INSTALL-DIR>\MSG\MESSAGES.MDB. When migrating the message database to another database (for instance: MS SQL), you must modify this field;
- ◆ **Enable Archiving** – Enable Archiving if you want to move messages periodically to the Archive Database indicated by the Archive Connection String.
It's a good practice to move messages to an Archive Database periodically; it ensures a good performance of the system, especially when your database is growing large. It is recommended to have not more than 100.000 records in your Message Database. If you have more, the performance of your Message Views may decrease dramatically.
By default, archiving is enabled;
- ◆ **Archive Connection String** – Indicates the location of the archive database. By default, this connection string is set to <INSTALL-DIR>\MSG\ARCHIVE.MDB. When migrating the message database to another database (for instance: MS SQL), this field must be modified;
- ◆ **Archive after x Days** – If archiving is enabled, all messages older than x days will be moved to the Archive Database indicated by the 'Archive Connection String' field;
- ◆ **Max. Pending Time** – Maximum number of hours that a message should remain pending or waiting for acknowledgement. If it takes longer than this threshold, the message will fail.
'Max. Pending Time' only applies to outgoing messages.

The General Program Options are stored in the OPTIONS table of the configuration database. Changes to the General Program Options will be detected automatically by the SMS Messaging Server service (you do not need to restart the service).

5.3. Channels (Communication Settings)

A 'Channel' describes the properties of a communication device or protocol. ActiveXperts SMS Messaging Server has four different types of 'Channels':

- ◆ GSM Modem Channels;
- ◆ SMPP Channels;
- ◆ POP3 Channels;
- ◆ SMTP Channels.

▼ GSM Modem Channel

SMS Messaging Server uses a 'GSM Modem Channel' to send/receive SMS messages using a GSM Modem. The GSM Modem should be connected to the server where SMS Messaging Server is running, through one of the following interfaces:

- ◆ Serial COM port;
- ◆ USB port;
- ◆ Bluetooth (or Infrared port).

To enter the properties of a GSM Modem, launch the Manager application from the Start menu and open the 'Channels' folder. Here, you can add new channels, delete channels, and make modifications to a channel.

The following information can be configured for a GSM Modem Channel:

- ◆ **Enable/Disable channel** – You can enable/disable a channel at any time. When a channel is disabled, the SMS Messaging Server services will not attempt to send messages through this channel; it will neither receive messages on this channel;
- ◆ **Description** – A smart description for this channel;
- ◆ **Enable/Disable Send** – Disable the Send option to allow only receiving through this channel;
- ◆ **Enable/Disable Receive** – Disable the Receive option to allow only sending through this channel;
- ◆ **Device** – Indicates the TAPI driver or direct COM port of the GSM Modem. It is recommended to use a TAPI (i.e. Windows Telephone), like 'Standard 9600 bps Modem'. TAPI devices can be configured from the Windows Control Panel;
- ◆ **Speed** – Indicates the speed used to communicate with the device. This field can only be set when selecting a direct COM port. However, it is recommended to use a TAPI devices driver rather than a direct COM port;
- ◆ **Pin Code** – Specifies the Pin Code on the SIM card. Only needs to be set if the SIM requires a Pin Code;
- ◆ **Subscriber ID** – The Subscriber ID is the actual SMS mobile number associated with the SIM card of the GSM modem. This number cannot be read from the modem by ActiveXperts SMS Messaging Server, and therefore needs to be configured manually for your own convenience. The Subscriber ID will be used in the Message Database to indicate the sender (for outgoing SMS messages) or Recipient (for incoming SMS messages);
- ◆ **Check every x seconds for new incoming messages** – Indicates how often (in seconds) the modem should check for new incoming messages
- ◆ **SMS Limit** – The maximum number of outgoing SMS messages allowed per day (sending only) on this channel;
- ◆ **Large Messages** – Indicates how large outgoing messages (i.e. messages longer than 160 characters) should be treated. There are three options: Truncate messages larger than 160 characters, send the large message as multi-part message, or don't send the large message at all (message will fail).
- ◆ **Leave a copy of the outgoing message on the device** – By default, messages will be deleted from the SIM card after they've been received by the SMS Messaging Server. If this is not desired, enable this option.

The configuration of one single GSM Modem is stored in the `Channel_Gsm` table in the Configuration Database, as one single record. The `Channel_Gsm` table can have multiple records.

▼ SMPP Provider Channel

SMS Messaging Server uses a 'SMPP Channel' to send/receive SMS messages using an SMPP-compliant SMSC provider.

You need to sign-up with an SMPP-compliant center before you can actually use their service. It normally requires a small sign-up fee, and a fee for an SMS bundle per month.

ActiveXperts SMS Messaging Server provides free SMPP messaging for a limited number of SMS messages. You can connect to an ActiveXperts SMPP server and send/receive messages right after installation, without the need to sign-up with an SMSC center first.

If you want to know about a list of SMPP providers recommended by ActiveXperts Software, [click here](#).

The following information can be configured for an SMPP Channel:

- ◆ **Enable/Disable Channel** – You can enable/disable a channel at any time. When a channel is disabled, the SMS Messaging Server service will not attempt to send messages through this channel or receive messages from this channel;
- ◆ **Enable/Disable Send** – Disable the Send option to allow only receiving through this channel;
- ◆ **Enable/Disable Receive** – Disable the Receive option to allow only sending through this channel;
- ◆ **Description** – A smart description for this channel;
- ◆ **Server** – Hostname or IP address of the SMPP provider;
- ◆ **Server Port** – TCP/IP port of the SMPP provider. Default: 2775;
- ◆ **System ID and Password** – Account and Password used to logon to the SMPP-compliant SMSC provider.
- ◆ **System Type** – Only required when the provider offers different protocols. Leave it default (“SMPP”) unless required otherwise;
- ◆ **Subscriber ID** – The Subscriber ID is the actual SMS mobile number associated with the SMPP subscription. This number cannot be detected by ActiveXperts SMS Messaging Server (it is not part of the SMPP protocol), and therefore needs to be configured manually for your own convenience. The Subscriber ID will be used in the Message Database to indicate the sender (for outgoing SMS messages) or Recipient (for incoming SMS messages);
- ◆ **Address Range** – All messages on the SMSC with the address range as recipient address will be forwarded to this channel when the client is bound to the SMSC in receiver or transceiver model;
- ◆ **Check Interval** – Specifies how often the SMS Messaging Server should check for new SMS Messages. Default value: 10 seconds;
- ◆ **Timeout** – Specifies the number of seconds that SMS Messaging Server should wait for a response of the SMPP provider during a SMPP session. If a timeout occurs, an operation will fail immediately. Default value: 3 seconds. Increase this value if the network connection is slow;
- ◆ **Keep Alive** – Specifies how long the network connection is allowed to remain idle before the network connection is disconnected. Leave it default (60 seconds) unless your provider requests a different Keep Alive value;
- ◆ **SMS Limit** – The maximum number of outgoing SMS messages allowed per day to prevent a malfunctioning custom script from sending out many messages;
- ◆ **Large Messages** – Indicates how large outgoing messages (i.e. messages longer than 160 characters) should be treated. There are three options: Truncate messages larger than 160 characters, send the large message as multi-part message, or don’t send the large message at all (message will fail).
- ◆ **Request Acknowledgement** – Specifies whether or not an acknowledgement should be requested after sending an SMS message. If this value is enabled, a message is only considered as ‘delivered’ if a positive acknowledgement is received;

- ♦ **Acknowledgement Timeout** – Specifies how long the SMS Messaging Server should wait for an acknowledgement of a sent message. If the acknowledgement is not received within the time frame, it will be set to 'failed'

▼ POP3 Channel

SMS Messaging Server is capable of receiving e-mails on different POP3 accounts and process these e-mails. A POP3 Channel describes the POP3 server settings as well as a list of POP3 accounts that need to be checked for incoming e-mail.

The following information can be configured for a POP3 Channel:

- ♦ **Enable/Disable channel** – You can enable/disable a channel at any time. A disabled Channel will not receive e-mail messages;
- ♦ **Host** – Hostname or IP address of the POP3 e-mail server;
- ♦ **Leave a copy of messages on the server** – By default, SMS Messaging Server deletes an e-mail on the POP3 server once it has received the e-mail. If you want to keep the e-mail on the server, enable this option;
- ♦ **Truncate messages** – Specify the maximum size (in KB) that a single POP3 message may occupy in the Message Database. The remainder of the message is truncated.
- ♦ **Check for new messages every X seconds** – Specify how often the POP3 server should be checked for new incoming e-mail;
- ♦ **Manage Accounts** – Specify the POP3 accounts to be checked for incoming e-mail.

The configuration of a single POP3 server is stored in the `Channel_POP3` table in the Configuration Database, as one single record. The `Channel_POP3` table can have multiple records.

▼ SMTP Channel

To send out e-mail messages, you can configure one or more SMTP channels.

The following information can be configured for an SMTP Channel:

- ♦ **Enable/Disable channel** – You can enable/disable a channel at any time. When the channel is disabled, the SMS Messaging Server service will attempt to send e-mail messages through this channel;
- ♦ **Host** – Hostname or IP address of the SMTP e-mail server;
- ♦ **Mail server requires authentication / Account / Password** – Specify login credentials if SMTP servers requires authentication. Some mail servers (incl. MS Exchange) require authentication before it can be used to send out e-mail messages;
- ♦ **Name** – Friendly display name that will be shown as the recipient on the remote POP3 e-mail client;
- ♦ **E-mail address** – E-mail address that is associated with SMS Messaging Server. When the SMS Messaging Server sends out an e-mail message, it will use this address;
- ♦ **Reply address** – Reply address of the associated e-mail address;
- ♦ **Truncate messages** – Specify the maximum size that a single POP3 message may occupy in the Message Database. The remainder of the message is truncated.

The configuration of one single SMTP server is stored in the `Channel_Smtp` table in the Configuration Database, as one single record. The `Channel_Smtp` table can have multiple records.

5.4. Triggers (Processing incoming messages)

Triggers are defined to process incoming messages.

It is a combination of a condition and an associated script. The condition is a SQL-like condition that selects a set of unprocessed, incoming messages.

As soon as a new message is received through one of the channels, the status of the message is set to 'Not Processed'. The SMS Messaging Server regularly (by default: once every 5 seconds) checks if there is any unprocessed messages. As soon as there's an unprocessed message, it will trigger all scripts that match a predefined condition.

The following information can be configured for a Trigger:

- ◆ **Enable/Disable Trigger** – If a Trigger is enabled, its condition will be checked and – if the condition is True – the associated script will be executed;
- ◆ **Condition** – A SQL formatted condition. If the condition is True, the associated script will be executed;
- ◆ **Script** – The actual VBScript script that will be executed if the condition is True.

IMPORTANT: A condition is ALWAYS FALSE if the status of the incoming message indicates that the message has already been processed. So, if you define three triggers (all enabled), and the first trigger sets the message status to PROCESSED, the other triggers will NEVER match the conditions and the associated scripts will never be called.

6. Message Database

6.1. Introduction

The Message Database contains incoming and outgoing SMS- and e-mail messages. The location of the messages is indicated by a DSN connection string, defined in the Options of the software. To see the location of the database:

- ◆ Start the **Manager application**;
- ◆ Open the **Configuration Folder** and select **Options**;
- ◆ Click on the icon next to **Message Database**.

By default, the Message Database is located in the <INSTALL-DIR>\MSG\ directory and is called Messages.mdb. It is an MS Access database (MS Access 2000 compatible). You can migrate to an MS SQL database at any time (see also 6.4: 'Migrating to MS SQL').

6.2. Message Database - Tables and Fields

The Message Database contains 4 tables:

- ◆ Messages – all incoming and outgoing messages;
- ◆ Directions – definitions of incoming and outgoing values for an individual message;
- ◆ Status – definitions of status values for an individual message;
- ◆ Type - definitions of type values for an individual message;

Table: Messages

The 'Messages' table contains the following fields:

Property	Type	Description
ID	Auto Number	Record primary key. This value is automatically generated when a new record is inserted.
Direction	Number	Message direction: Incoming or outgoing. Values are defined in the 'Directions' table.
Type	Number	Message type: SMS or e-mail. Values are defined in the 'Type' table.
Status	Number	Message type: SMS or e-mail. Values are defined in the 'Status' table.
ChannelID	Number	<i>For incoming messages:</i> Channel which received the message; <i>For outgoing, scheduled messages:</i> Preferred channel. If you set this value to 0, the first available channel will pickup the message. <i>For outgoing, sent/failed messages:</i> The channel which picked up the message to deliver it to the recipient.
MessageReference	Number	Message reference - if any - assigned by the provider. Only assigned by SMS channels. SMS providers provide a message reference; it is used by the provider later on to acknowledge the message.

ScheduleTimeSecs	Number	Scheduled time (in seconds after 01/01/1970). Used for outgoing messages only. 0 indicates: immediate send.
LastUpdateSecs	Number	Last update time (in seconds after 01/01/1970). Each time a record is updated by the software, this property is updated automatically.
Sender	String	Sender address of the message.
Recipient	String	Recipient address of the message.
Subject	String	Subject of the message (only applicable for e-mail messages)
Body	String	Body of the message. It can only contain ASCII data (including MIME e-mail messages)
Trace	String	Trace of the message. This multi-line field contains the progress of a message, including date and time.
sysLock	Boolean	Internal System field (used by the SMS Messaging Service)
sysInfo	String	Internal System field (used by the SMS Messaging Service)
sysHash	String	Internal System field (used by the SMS Messaging Service)
sysForwarded	Boolean	Internal System field (used by the SMS Messaging Service)

NOTE: All of the above fields can be used in a Trigger condition. For example, you can build a Trigger condition like this one:
 Direction=1 AND Type=2 AND (ChannelID=1001 OR ChannelID=1002)

Table: [Directions](#)

The 'Directions' table contains the following fields:

Property	Type	Description
Direction	Number	Numeric value of a direction.
Description	String	Description of a direction.

The 'Directions' table is a static table; it contains definitions for each direction. Each direction is described by a separate record:

Direction	Description
0	Undefined
1	In
2	Out

Table: [Status](#)

The 'Status' table contains the following fields:

Property	Type	Description
Status	Number	Numeric value of a status.
Description	String	Description of a status.

The 'Status' table is a static table; it contains definitions for each status. Each status is described by a separate record:

Direction	Description
0	Undefined
1	Received / Not processed. Message has just received and will be processed soon.
2	Received / Processed Message has been processed.
3	Received / Processing failure Message has been processed but there was an error during processing. Check the 'Trace' field of the Message for additional information (for instance: line number in the script where the error occurred).
4	Outgoing / Scheduled Message is scheduled for a later date/time. As soon as the time has elapsed, the service will set the status to Pending (5) to deliver the message to the channel responsible for sending the message.
5	Outgoing / Pending Message status has changed from Scheduled into Pending; this means that it is time to send the message. The message has been delivered to the channel that is responsible for sending the message.
6	Submitted / Waiting for ACK Message has been submitted successfully, but the channel properties require an acknowledge. The channel successfully sent out the message to the recipient but has not received an acknowledgement yet.
7	Send Failed / Generic Error The service tried to send out the message, but didn't succeed. Check the 'Trace' field of the Message for additional information.
8	Send Failed / Message undeliverable The service tried to send out the message, but could not deliver the message. Check the 'Trace' field of the Message for additional information.
9	Send Failed / NACK received The service successfully submitted the message to the provider, but a negative acknowledgement was received. The negative acknowledgement was provided by the provider. This happens for instance when an SMS number has a valid format but the recipient does not exist on the mobile network. Check the 'Trace' field of the Message for additional information.
10	Send Failed (ACK expired) The service successfully submitted the message to the provider, but there was no positive/negative acknowledgement within a certain period (configurable).
11	Sent Message was sent successfully.
12	Delivered (ACK received)
255	For internal use only

NOTE: There are two status values that look similar but are slightly different: **Sent** and **Delivered (ACK received)**. **Sent** means that the message was successfully submitted, but – according to the properties of the Channel – no acknowledgement was required.

Delivered (ACK received) means that the message was submitted successfully and – according to the properties of the Channel – an acknowledgement was requested and – indeed - received.

To require an acknowledgement, go to the properties of a particular Channel, and change the timeout value in the ACK time-out field.

Table: Type

The 'Type' table contains the following fields:

Property	Type	Description
Type	Number	Numeric value of a type.
Description	String	Description of a type.

The 'Type' table is a static table; it contains definitions for each message type. Each Type is described by a separate record:

Direction	Description
0	Undefined
1	E-mail
2	SMS

6.3. Archiving Messages

Database Archiving can dramatically improve performance when dealing with a large Message Database.

Consider an SMS Messaging Server scenario where more than 1000 SMS messages are broadcasted daily. It would end up with a database of 50.000 messages after one month, and over a 500.000 messages after a year. Most probably, you do not need instance access to these messages regularly. However, you do not want to delete these messages.

ActiveXperts SMS Messaging Servers' Archiving feature allows you to automatically archive messages. Archiving is a continuous process in the system. By default, messages are archived after 7 days: they are moved from the Message Database to the Archive Database.

To change the Archiving Settings, launch the Manager application from the Start menu, open the **Configurations** folder and click on **Options**. Then, change the following settings:

- ◆ **Enable Archiving** – Enable/disable archiving;
- ◆ **Archive Connection String** – Indicates the location of the archive database. By default, this connection string is set to <INSTALL-DIR>\MSG\ARCHIVE.MDB;
- ◆ **Archive after x Days** – If archiving is enabled, all messages older than x days will be moved to the Archive Database indicated by the 'Archive Connection String' field;

6.4. Migrating to MS SQL

You can migrate the Message Database to MS SQL using the 'MS SQL Migration Wizard'.

To start the MS SQL Migration Wizard, select the **Migrate to MS SQL Migration Wizard** from the **Tools and Utilities Folder** in the Manager application. The Migration Wizard will perform the following actions:

- ♦ **Create new Database** – The wizard will ask for a database name to store tables and properties. If the database does not exist yet, the database will be created;
- ♦ **Create Tables** – The wizard will create all necessary tables;
- ♦ **Move Messages** – All messages will be moved to the MS SQL database;
- ♦ **Move Archive** – All archived messages will be moved to the MS SQL database;

7. Troubleshooting

7.1. Introduction

ActiveXperts SMS Messaging Server is designed with a special thing in mind: let the user have complete control over the product. This means, that the system must show exactly what it is doing, why things work, and why things do not work. The following things will help you with that:

- ♦ **ActiveXperts SMS Messaging Server Monitor** – The Monitor application shows you real-time information about Channels, Message Database and the VBScript engine;
- ♦ **Trace Files** – Trace files show you detailed information about Channels, Message Database and VBScript engine, over a long period of time.

7.2. ActiveXperts SMS Messaging Server Monitor

The ActiveXperts SMS Messaging Server Monitor shows **real-time** information:

- ♦ Every single channel;
- ♦ Message database;
- ♦ VBScript engine,

To launch the Monitor application from the Manager application, select **Switch To...** from the **File Menu**, and select **Monitor**.

You can also launch the Monitor applications from the **ActiveXperts SMS Messaging Server program group** in the **Start Menu**.

By default, the ActiveXperts SMS Messaging Server Monitor shows the following items:

- ♦ The first GSM Modem Channel;
- ♦ The first SMPP Channel;
- ♦ The first POP3 Channel;
- ♦ The first SMTP Channel;
- ♦ The Message Database;
- ♦ The VBScript Engine.

You can show additional Channel by selecting them from the **Show...** item in the **View Menu**.

7.3. Tracing

The ActiveXperts SMS Messaging Server Monitor provides tracing for the following items:

- ♦ Every single channel;
- ♦ Message database;
- ♦ VBScript engine.

To enable tracing for a particular channel or thread, you must enable it through the registry, by assigning a Log File to it:

- ◆ Open a registry editor (REGEDIT.EXE or REGEDT32.EXE);
- ◆ Open the HKLM\Software\ActiveXperts\SMS Messaging Server\Trace key;
- ◆ Open the key of the item that you want to trace, and specify a valid trace filename;
- ◆ Restart the Service.

It's important to restart the service after changing one of the Trace items.

NOTE: Tracing will decrease the performance of the ActiveXperts SMS Messaging Server service. So, just use tracing for troubleshooting purposes, and make sure you turn-off the trace file(s) by assigning an empty string to the <code>LogFile</code> value in the Trace keys.

8. Performance Tuning

8.1. Introduction

ActiveXperts SMS Messaging Server can be tuned to improve performance.

When we talk about performance, we mean: the performance of the ActiveXperts SMS Messaging Server service. This service is doing the actual job of managing the Message Database, controlling the Channels and executing the Triggers.

ActiveXperts SMS Messaging Service has different sub-processes (so called ‘threads’). Each thread is running as a separate process.

8.2. Message Database

It’s important to keep the message database small. We recommend to keep the message database at a maximum of 50.000-100.000 messages. If you exceed this number of records, the performance of the thread that is controlling the Message Database will decrease.

You can control the number of messages in your message Database by configuring Archiving, by 1) enabling it in the Options, and 2) specifying after how many days messages should be archived. See also 6.3: Archiving Messages.

There are 3 registry entries that influence to the performance of the message database. It shouldn’t be necessary to change these values, but in some circumstances it may improve the performance:

- ◆ HKLM\Software\ActiveXperts\SMS Messaging Server\CheckAcksInterval – Indicates how often the system must check for new incoming ACK messages. Default: 30 seconds;
- ◆ HKLM\Software\ActiveXperts\SMS Messaging Server\CheckArchiverInterval – Indicates how frequent the archiver should check for messages to be archived. Default: 3600 seconds;
- ◆ HKLM\Software\ActiveXperts\SMS Messaging Server\CheckNewMessagesInterval – Indicates how frequent the message database should be checked for new outgoing message, to pass them to the channels. Default: 5 seconds.

You MUST restart the service after changing one of the above registry entries before changes take effect.

8.3. Queue directory

ActiveXperts SMS Messaging Server uses a file oriented mechanism to queue incoming and outgoing messages. By default, messages are stored as plain ASCII files in the <INSTALL-DIR>\Queue directory.

If you have a much faster drive (or a fast RAID configuration) other than the drive that you are currently using, consider moving the Queue directory to the other drive. To do so, go to **Options** and edit the **Directory** options. Here you can specify and different Queue directory.

9. SMS Messaging Server API

9.1. Introduction

ActiveXperts SMS Messaging Server defines interfaces to its Configuration Database and its Messages Database, to make it a so called **Open System**. The interfaces are defined and implemented in the **SMS Messaging Server API**.

The API is primary used for Database Triggers (see also Chapter 5): as soon as a new message arrives in the system, the SMS Messaging Server will trigger a VBScript program (or multiple VBScript programs). These VBScript programs make use of the API, for instance to create a reply message, or to change properties of the incoming message.

Besides for triggering, the API can be used for many other purposes.

A few scenario's where the API can be used :

- ◆ To define custom triggers (as described in Chapter 5);
- ◆ To define a custom ASP/ASP.NET based web interface, where web users can create new SMS messages and read/filter incoming messages
- ◆ To send out a batch of SMS messages every morning, for instance to notify IT Administrators about the completion status of a backup, or to tell a department about upcoming meetings on that day;
- ◆ To build a customized user interface with various message views;

The SMS Messaging Server API is a collection of objects. Each object has several properties and functions defined.

The following objects are defined:

- ◆ Constants – object representing all constants that can be used in the API
- ◆ Options – object representing all general program settings;
- ◆ Messages – object representing a collection of message, i.e. the Message Database;
- ◆ Message – object representing a single message record from the Message Database;
- ◆ Channels – object representing a collection of Channels;
- ◆ GsmModemChannel – object representing a single GSM Modem Channel;
- ◆ SmppProviderChannel – object representing a single SMPP Provider Channel;
- ◆ Pop3Channel – object representing a single POP3 Channel;
- ◆ SmtplibChannel – object representing a single SMTP Channel;
- ◆ Scripts – object representing a collection of scripts;
- ◆ Script – object representing a single script;

9.2. 'Constants' object

Properties:

Constant	Value	Description
MESSAGETYPE_EMAIL	1	Type of message: e-mail
MESSAGETYPE_SMS	2	Type of message: SMS
MESSAGETYPE_UNDEFINED	0	Type of message: undefined (invalid)
MESSAGEDIRECTION_IN	1	Direction of message: incoming
MESSAGEDIRECTION_OUT	2	Direction of message: outgoing
MESSAGEDIRECTION_UNDEFINED	0	Undefined message direction (invalid)
MESSAGESTATUS_IN_NOTPROCESSED	1	Status of message: successfully received, not processed yet
MESSAGESTATUS_IN_PROCESSED	2	Status of message: successfully received, processed successfully
MESSAGESTATUS_IN_PROCESSINGFAILURE	3	Status of message: successfully received, but failed to process
MESSAGESTATUS_OUT_SCHEDULED	4	Status of message: message scheduled (not ready for sending yet)
MESSAGESTATUS_OUT_PENDING	5	Status of message: message pickup up by SMS Messaging Server, and will be sent out soon
MESSAGESTATUS_OUT_DELIVEREDWAITING	6	Status of message: sent out but still waiting for acknowledgement
MESSAGESTATUS_OUT_GENERICFAILURE	7	Status of message: send failed, reason not specified
MESSAGESTATUS_OUT_UNDELIVERABLE	8	Status of message: undeliverable
MESSAGESTATUS_OUT_NACK	9	Status of message: sent out but not delivered; a negative acknowledgement was received
MESSAGESTATUS_OUT_ACKEXP	10	Status of message: sent out, but not acknowledged yet
MESSAGESTATUS_OUT_NOACKREQ	11	Status of message: sent out and probably delivered (no acknowledgement requested)
MESSAGESTATUS_OUT_ACKRECEIVED	12	Status of message: sent out and delivered (acknowledgement received)
MESSAGESTATUS_OUT_UNDEFINED	0	Status of message: undefined

Example:

```
' Sample: print some constants
Set objConstants = CreateObject( "AxSmsServer.Constants" )
WScript.Echo objConstants.MESSAGEDIRECTION_IN
WScript.Echo objConstants.MESSAGEDIRECTION_OUT
WScript.Echo objConstants.MESSAGESTATUS_IN_NOTPROCESSED
```

9.3. 'Messages' collection objectProperties:

Property	Type	R/W	Description
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameters	Description
Count	Filter (String)	Count the number of messages in the Message Database. You can apply a

		filter. When an empty string is passed as filter, no filter is applied
Create	None	Create a new message in the Message Database
Delete	Condition (String)	Delete all messages that match the condition
Load	Message-ID (Number)	Return the message indicated by the Message-ID
FindFirstMessage	Condition (String)	Return the first message that matches the condition
FindNextMessage	None	Return the next message
GetDirectionDescription	Direction-code (Number)	Look-up the friendly direction description of the given direction-code
GetStatusDescription	Status-code (Number)	Look-up the friendly status description of the given direction-code
GetTypeDescription	Type (Number)	Look-up the friendly type description of the given direction-code

Sample:

```
' Sample: count all incoming messages in the message database

Set colMessages = CreateObject( "AxSmsServer.Messages" )
Set objConstants = CreateObject( "AxSmsServer.Constants" )

n = colMessages.Count( "Direction = " & objConstants.objConstants.MESSAGEDIRECTION_OUT )
If( colMessages.LastError = 0 ) Then
    WScript.Echo "#Records: " & r
End If
```

Sample:

```
' Sample: create new outgoing SMS message

Set objConstants = CreateObject( "AxSmsServer.Constants" )
Set colMessages = CreateObject( "AxSmsServer.Messages" )
Set objMessage = colMessages.Create

If( colMessages.LastError <> 0 ) Then
    WScript.Echo "Create failed, error: " & colMessages.LastError
    Exit Function
End If

WScript.Echo "Message successfully created, recordID: " & objMessage.ID

objMessage.Direction = objConstants.MESSAGEDIRECTION_OUT
objMessage.Type = objConstants.MESSAGE_TYPE_EMAIL
objMessage.Status = objConstants.MESSAGESTATUS_OUT_SCHEDULED
objMessage.ChannelID = 0 ' Any available SMS channel

objMessage.Recipient = "+31624896641"
objMessage.Body = "Hello, world!"

colMessages.Save objMessage

WScript.Echo "Save, result: " & colMessages.LastError
```

9.4. 'Message' object

Properties:

Property	Type	R/W	Description
ID	Number	RW	Record primary key. This value is automatically generated when a new record is inserted.
Direction	Number	RW	Message direction: Incoming or outgoing
Type	Number	RW	Message type: SMS or e-mail
Status	Number	RW	Status of the message
ChannelID	Number	RW	<i>For incoming messages:</i> Channel that received the message; <i>For outgoing, scheduled messages:</i> Preferred channel. If you set this value to 0, the first available channel will pickup the message. <i>For outgoing, sent/failed messages:</i> The channel that picked up the message to deliver it to the recipient.
MessageReference	Number	RW	Message reference - if any - assigned by the provider. Only assigned by SMS channels. SMS providers provide a message reference; it will be used by the provider later on to acknowledge the message.
ScheduleTimeSecs	Number	RW	Scheduled time (in seconds after 01/01/1970). Used for outgoing messages only. 0 indicates: immediate send.
ScheduleTime	String	RW	Friendly string representation of the 'ScheduleTimeSecs' property. This value is NOT stored in the database; it is a translation of the 'ScheduleTimeSecs' property into a friendly date/time notation.
LastUpdateSecs	Number	RW	Last update time (in seconds after 01/01/1970). Each time a record is updated by the software, this property is updated.
Sender	String	RW	Sender address of the message.
Recipient	String	RW	Recipient address of the message.
Subject	String	RW	Subject of the message (if applicable).
Body	String	RW	Body of the message.
LastError	Number	R	Completion code of the last called function.

Functions:

None.

9.5. 'Options' object

Properties:

Property	Type	R/W	Description
ID	Number	R	Database record key
LicenseKey	String	RW	License key (registration code)

LicenseActivation	String	RW	License Activation code
LogEnabled	Boolean	RW	Logging enabled (y/n)
LogUseDatabase	Boolean	RW	If true, use database for logging, otherwise use plain ASCII file
LogFileName	String	RW	Name of the plain ASCII log file
LogFileSeparator	String	RW	Separator used in plain ASCII log file
LogFileMaxKB	Number	RW	Maximum size of plain ASCII log file (KB)
LogDbConnectionString	String	RW	Connection string of log database
LogDbConnectionStringPassword	String	RW	Password used in log database connection string
DirTemp	String	RW	Directory used internally by service to store Temporary files
DirMessageQueue	String	RW	Directory used internally by service to store incoming messages
MsgDbConnectionString	String	RW	Database connection string of the message database
DailySMSLimit	Number	RW	Maximum number of outgoing SMS messages per day
MaxPendingTime	Number	RW	Maximum number of hours that a message can remain pending before it will fail
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameter(s)	Description
Load	Record-ID (Number)	Load the options.
Save	None	Save options

Example:

```

Set objOptions = CreateObject( "AxSmsServer.Options" )
objOptions.Load( numID )

If( objOptions.LastError <> 0 ) Then
    WScript.Echo "Failed to load Options record, error: " & objOptions.LastError
Else
    WScript.Echo "ID: " & objOptions.ID
    WScript.Echo "MaxPendingTime: " & objOptions.MaxPendingTime

    objOptions.MaxPendingTime = 24
    objOptions.Save
    WScript.Echo "Save, result: " & objOptions.LastError
End If

```

9.6. 'Channels' collection objectProperties:

Property	Type	R/W	Description
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameters	Description
FindFirstGsmModem	Filter (String)	Find the first GSM Modem Channel in the database. Optionally specify a filter (SQL-like notation)
FindNextGsmModem	None	Find the next GSM Modem Channel
FindFirstSmpp	Filter (String)	Find the first SMPP Channel in the database. Optionally specify a filter (SQL-like notation)
FindNextSmpp	None	Find the next SMPP Channel
FindFirstPop3	Filter (String)	Find the first POP2 Channel in the database. Optionally specify a filter (SQL-like notation)
FindNextPop3	None	Find the next POP3 Channel
FindFirstSmtip	Filter (String)	Find the first SMTP Channel in the database. Optionally specify a filter (SQL-like notation)
FindNextSmtip	None	Find the next Smtip Channel

Sample:

```

' Sample: showing all POP3 channels in the configuration database
Option Explicit

Dim colChannels, objChannel

Set colChannels = CreateObject( "AxSmsServer.Channels" )
Set objChannel = colChannels.FindFirstPop3( "" )
While colChannels.LastError = 0
    PrintPop3Channel( objChannel )
    Set objChannel = colChannels.FindNextPop3
Wend

Sub PrintPop3Channel( objChannel )
    WScript.Echo "POP3 CHANNEL, ID: " & objChannel.ID
    WScript.Echo "  Description: " & objChannel.Description
    WScript.Echo "  Enabled: " & objChannel.Enabled
    ' Print any other POP3 Channel property
End Sub

```

Sample:

```

' Sample: iterate over all enabled GSM Modem Channels
Option Explicit

Dim colChannels, objChannel

Set colChannels = CreateObject( "AxSmsServer.Channels" )
Set objChannel = colChannels.FindFirstGsmModem( "Enabled <> 0" )
While colChannels.LastError = 0
    Set objChannel = colChannels.FindNextGsmModem
Wend

```

9.7. 'GsmModemChannel' objectProperties:

Property	Type	R/W	Description
ID	Number	R	Database record key
Description	String	RW	GSM Modem's friendly name
Enabled	Boolean	RW	Specifies if device is enabled or not. By disabling a channel, SMS Messaging Server will not be able to send/receive SMS messages on this channel. This property overrides the EnableSend and EnableReceive properties
EnableSend	Boolean	RW	By setting EnableSend to False, SMS Messaging Server will only be able to receive SMS messages
EnableReceive			By setting EnableReceive to False, SMS Messaging Server will only be able to send SMS messages on this channel
Device	String	RW	Device name. Can be a TAPI device name (like 'Standard 9600 bps Modem') or a direct COM port (like 'COM1:')
DeviceSpeed	Number	RW	Device speed. Only required when using a direct COM port in the 'Device property'. Use '0' to use the default speed
PinCode	String	RW	Specifies the PIN code on the SIM card (if required)
MultiPart	Boolean	RW	Enable MultiPart to allow messages longer than 160 characters. If disabled, a message will be truncated if it exceeds 160 characters
DailyLimit	Number	RW	Maximum number of outgoing SMS messages per day
SubscriberID	String	RW	SMS mobile number that is associated with the SIM card. It cannot be read from the SIM card so it must be specified manually
LeaveMessageOnDevice	Boolean	RW	Specifies whether or not to leave a copy of the message on the device after receive
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameter(s)	Description
Clear	None	Clear all properties
Create	None	Create a new POP3 channel
Load	ChannelID (Number)	Load GSM Channel from the configuration database into the object
Save	None	Save GSM Channel properties into the configuration database
Delete	ChannelID (Number)	Remove a GSM Modem Channel from the configuration database

Example:

```
' Load a GSM Modem Channel, change some properties and save it
```

```
Set objChannel = CreateObject( "AxSmsServer.GsmModemChannel" )
objChannel.Load( 1001 )
WScript.Echo "Load, result: " & objChannel.LastError
If( objChannel.LastError = 0 ) Then
    objChannel.Host      = "newhost"
    objChannel.Account   = "newaccount"
    objChannel.Password  = "newpassword"
    objChannel.Save
```

```

WScript.Echo "Save, result: " & objChannel.LastError
WScript.Echo "Save, result: " & objChannel.LastError
End If

```

9.8. 'SmppChannel' object

Properties:

Property	Type	R/W	Description
ID	Number	R	Database record key
Description	String	RW	SMPP provider's friendly name
Enabled	Boolean	RW	Specifies if the SMPP provers is enabled or not. By disabling a channel, SMS Messaging Server will not be able to send/receive SMS messages on this channel. This property overrides the EnableSend and EnableReceive properties
EnableSend	Boolean	RW	By setting EnableSend to False, SMS Messaging Server will only be able to receive SMS messages
EnableReceive			By setting EnableReceive to False, SMS Messaging Server will only be able to send SMS messages on this channel
Server	String	RW	Host name or IP address of the SMPP provder
ServerPort	Number	RW	TCP/IP port of the SMPP provider. Default: 2775
ServerTimeout	Number	RW	Maximum number of seconds that the SMS Messaging Server should wait for a response. Default: 3 seconds
SystemID	String	RW	Account used to logon to the SMPP provider
SystemPassword	String	RW	Password used to logon to the SMPP provider
SystemType	String	RW	Required if SMPP cannot distinguish between different protocols. Leave it default ("SMPP")
SubscriberID	String	RW	SMS mobile number that is associated with the SIM card. It cannot be read from the SIM card so it must be specified manually
CheckInterval	Number	RW	Specifies how often the SMS Messaging Server should check at the SMPP provder for new incoming messages. Default: 10 seconds
ServerKeepAlive	Number	RW	Maximum idle time before a network connection is dropped
MultiPart	Boolean	RW	Enable MultiPart to allow messages longer than 160 characters. If disabled, a message will be truncated if it exceeds 160 characters
RequestAckAfterSend	Boolean	RW	Request acknowledgement for every SMS message sent to the SMPP provider
AckTimeout	Number	RW	Specifies time out for an acknowledgement
DailyLimit	Number	RW	Maximum number of outgoing SMS messages per day
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameter(s)	Description
Clear	None	Clear all properties
Create	None	Create a new POP3 channel
Load	ChannelID (Number)	Load SMPP Channel from the configuration database into the object
Save	None	Save SMPP Channel properties into the configuration database
Delete	ChannelID (Number)	Remove a SMPP Modem Channel from the configuration database

Example:

```

' Load an SMPP Channel, change some properties and save it

Set objChannel = CreateObject( "AxSmsServer.SmppChannel" )
objChannel.Load( 1001 )
WScript.Echo "Load, result: " & objChannel.LastError
If( objChannel.LastError = 0 ) Then
    objChannel.System      = "smpp.activexperts-labs.com"
    objChannel.SystemPort  = 2775
    objChannel.SystemID    = "testuser"
    objChannel.SystemPassword = "123abc"
    objChannel.Save
    WScript.Echo "Save, result: " & objChannel.LastError
End If

```

9.9. 'Pop3Channel' object**Properties:**

Property	Type	R/W	Description
ID	Number	R	Database record key
Description	String	RW	POP3 server's friendly name
Enabled	Boolean	RW	Specifies if device is enabled or not. By disabling a channel, SMS Messaging Server will not be able to receive e-mail messages on this channel
Host	String	RW	Host name or IP address of the POP3 server
Accounts	String	RW	List of POP3 accounts that should be checked for new e-mail messages
CheckInterval	Number	RW	Specifies how often the SMS Messaging Server should check at the POP3 server for new incoming messages. Default: 60 seconds
LeaveMessageOnServer	Boolean	RW	Specifies whether or not to leave a copy of the message on the server after receive
MaxMessageSizeKB	Number	RW	Specify the maximum size (in KB) that a single POP3 message may occupy in the Message Database. The remainder of the message is truncated.
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameter(s)	Description
----------	--------------	-------------

Clear	None	Clear all properties
Create	None	Create a new POP3 channel
Load	ChannelID (Number)	Load POP3 Channel from the configuration database into the object
Save	None	Save POP3 Channel properties into the configuration database
Delete	ChannelID (Number)	Remove a POP3 Channel from the configuration database

Example:

```

' Load a POP3 Channel, change some properties and save it

Set objChannel = CreateObject( "AxSmsServer.Pop3Channel" )
objChannel.Load( 3001 )
WScript.Echo "Load, result: " & objChannel.LastError
If( objChannel.LastError = 0 ) Then
    objChannel.Host      = "pop3.myserver.com"
    objChannel.Save
    WScript.Echo "Save, result: " & objChannel.LastError
End If

```

9.10. 'SmtChannel' object**Properties:**

Property	Type	R/W	Description
ID	Number	R	Database record key
Description	String	RW	POP3 server's friendly name
Enabled	Boolean	RW	Specifies if device is enabled or not. By disabling a channel, SMS Messaging Server will not be able to send e-mail messages on this channel
Host	String	RW	Host name or IP address of the POP3 server
UseLogin	Boolean	RW	To indicate that authentication is required on the SMTP server
Account	String	RW	Account used to logon to the SMTP server
Password	String	RW	Password used to logon to the SMTP server
FromName	String	RW	Specifies the sender's friendly name
FromAddress	String	RW	Specifies the sender e-mail address
MaxMessageSizeKB	Number	RW	Messages larger than MaxMessageSizeKB will be sent out completely, but will be truncated in the Message Database
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameter(s)	Description
Clear	None	Clear all properties
Create	None	Create a new SMTPchannel
Load	ChannelID (Number)	Load SMTP Channel from the configuration database into the object

Save	None	Save SMTP Channel properties into the configuration database
Delete	ChannelID (Number)	Remove a SMTP Channel from the configuration database

Example:

```

' Load an SMTP Channel, change some properties and save it

Set objChannel = CreateObject( "AxSmsServer.SmtpChannel" )
objChannel.Load( 4001 )
WScript.Echo "Load, result: " & objChannel.LastError
If( objChannel.LastError = 0 ) Then
    objChannel.Host      = "pop3.myserver.com"
    objChannel.Save
    WScript.Echo "Save, result: " & objChannel.LastError
End If

```

9.11. 'Triggers' collection object**Properties:**

Property	Type	R/W	Description
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameters	Description
FindFirstTrigger	Filter (String)	Find the first Trigger in the database. Optionally specify a filter (SQL-like notation)
FindNextTrigger	None	Find the next Trigger
Create	None	Create a new Trigger. A new Trigger object is returned.

Sample:

```

' Sample: list all triggers; then, create a new trigger
Option Explicit

Option Explicit

Dim colScripts, objScript

Set colScripts = CreateObject( "AxSmsServer.Triggers" )
Set objScript = colScripts.FindFirstTrigger( "" )
While colScripts.LastError = 0
    PrintTrigger( objScript )
    Set objScript = colScripts.FindNextTrigger
Wend

Set objScript = colScripts.Create()
objScript.Description = "My new trigger"

```

```

...

WScript.Echo "Ready."

' //////////////////////////////////////

Sub PrintTrigger( objTrigger )
    WScript.Echo "TRIGGER ID: " & objScript.ID
    WScript.Echo "  Enabled: " & objScript.Enabled
    WScript.Echo " ProcessingOrder: " & objScript.ProcessingOrder
    WScript.Echo "  Description: " & objScript.Description
    WScript.Echo "  Condition: " & objScript.Condition
    WScript.Echo "  Script: " & objScript.Script
End Sub

```

9.12. 'Trigger' object

Properties:

Property	Type	R/W	Description
ID	Number	R	Database record key
Description	String	RW	Trigger's friendly name
Enabled	Boolean	RW	Specifies if triggered is enabled or not.
ProcessingOrder	Number	RW	Trigger priority. Value must be in range [1..99]. The Trigger with the highest priority will be evaluated first.
Condition	String	RW	Indicates and SQL-like condition. If the condition is matched, the script will be run.
LastError	Number	R	Completion code of the last called function

Functions:

Function	Parameter(s)	Description
Clear	None	Clear all properties
Load	TriggerID (Number)	Load Trigger from the configuration database into the object
Save	None	Save Trigger properties into the configuration database
Delete	ChannelID (Number)	Remove a Trigger from the configuration database

Example:

```
' Load a Trigger, change some properties and save it

Set objTrigger = CreateObject( "AxSmsServer.Trigger" )
objTrigger.Load( 1 ) ' Load Trigger 1
WScript.Echo "Load, result: " & objTrigger.LastError
If(objTrigger.LastError = 0 ) Then
    objTrigger.Description = "new description"
    objTrigger.Save
    WScript.Echo "Save, result: " & objTrigger.LastError
End If
```

9.13. Samples

ActiveXperts SMS Messaging Server includes various samples that show how to use the API.

To browse through the samples, click on the **SMS Messaging Server API** icon in the **Tools and Utilities** map of the Manager application. Here, you will find API samples to access the Configuration Database and the Message Database.

In the <INSTALL-DIR>ASP Folder, you will find an ASP web application with full source code. The ASP code is based in the SMS Messaging Server API.

10. Case Studies

SMS Messaging Server uses a 'SMPP Channel' to send/receive SMS messages using an SMPP-compliant SMSC provider.

In the online Case Studies section, you find implementations of ActiveXperts SMS Messaging Server at customers of ActiveXperts. Each Case Study is provided with background information of the company, together with a problem statement, design goals and finally the implementation.

You can find a list of case studies at: www.activexperts.com/ASmsSrvr/casestudies