Movicon 11

MONITORING VISION AND CONTROL

Supervision and control XML-based from Windows Vista to Windows CE

Alarms Dispatcher

Version 11.3- Ed. Feb. 2012

**Cod. DOCS 11 AD-E Build 1101**

Table Of Contents

1. Alarm Dispatcher

1.1. Overview

Alarm Dispatcher is a software program for (notification) sending messages using various means of communication (Plugin), from among those configured. The communication technologies available are:

1. **SMS messages by means of SMPP protocol**
2. **SMS messages by means of GSM**
3. **Sending a Fax**
4. **Sending Voice Messages (Voice Synthesis)**
5. **Sending E-mails by means of SMTP protocol**
6. **Sending E-mail by means of MAPI**
7. **SMS messages using Ucp-Emi protocol**
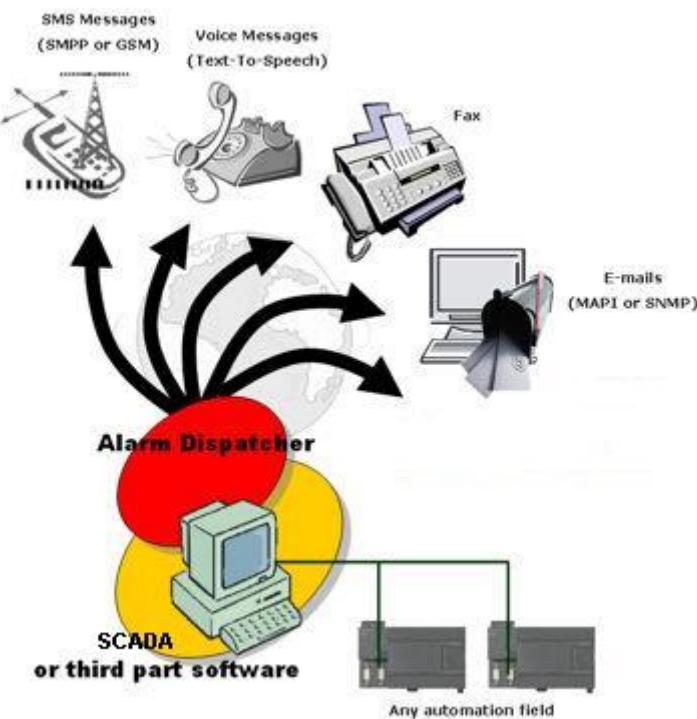
The software is designed by Progea so that it can be managed by Movicon or by any software capable of handling the functions, properties and methods available in the Alarm Dispatcher program.
The use of the Movicon is totally transparent in messages notification and general information handling . However, its use can be customized by means of VBA scripts or third party programs can command the Alarm Dispatcher for notification of messages according to the programming techniques described in this documentation.

Alarm Dispatcher maintains a list of the recipients and the list of plugins to be used, for each of its configuration projects, for sending messages. Alarm Dispatcher is designed for acting as a server for dispatching messages. The messages come from Movicon or from any Client application which must interface using the functions concerned.
However, receiving the messages and distributing them leaves a trace in the log window, apart from in the relative log file.

Use through Movicon

Movicon uses the Alarm Dispatcher functions in a completely transparent manner. The messages are transmitted by the properties of the Movicon project alarms, and the user profiles (Recipients) are those of the Movicon project users management.
However, it is possible to use the Movicon VBA Scripts to customize the management of notification according to the requirements.

Operating methods

Alarm Dispatcher has two operating methods:

1. **Configuration**
2. **Run**

Configuration

The program starts in design mode (except for startup with the option from command line) from which it is possible to create new configuration projects, open existing projects, modify and save them. Starting the program causes the last project saved to be loaded. From this mode it is possible to access the project settings windows, by means of the "Settings" menu item or by using the buttons on the toolbar. The "File" menu and the relative buttons on the toolbar, allow handling of these projects, whether new or pre-existing ones. The Run and Stop commands, present in the "File" menu, allow to change over from one operating mode to the other.

Run

In run mode the messages are actually dispatched and sent.
When possible, depending on specific criteria for each kind of message, messages for the same recipient are grouped together in order to save time and resources.
In run mode you can not modify settings or close the application. To do that, please go back to design mode. A special icon in tray area indicates the application state and gives access to log window. Right clicking on main window you can get a context menu with some commands, useful in case the application has been started by another program as an hidden application.

Windows 32/64 bit

To execute Alarm Dispatcher directly in run mode you can use the "**/R**" option from the command line. You can also specify from the command line the project file to be loaded and executed; if no project is specified here, Alarm Dispatcher will execute last loaded project.

Windows CE

On Windows CE devices Alarm Dispatcher can be executed only in run mode. If no project is specified from the command line, the Alarm Dispatcher looks for a bootable file in the compact flash root folder. The file name must be "Dispatcher.boot", it should be a valid XML file containing the "filename" attribute and "filename" should specify the path and the file name of the file to be run. If the boot file is not present, Alarm Dispatcher will execute the "AlarmDispatcher.Dspt" project, usually residing in its own folder.

Example of a "Dispatcher.boot" file:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<boot>
<filename>\SSD\ProjectFolder\ProjectName.Dspt</filename>
</boot>
```

1.2. Menu Commands

Menu File

New

Opens a new project, setting the default values for the attributes. From the standard dialog window for selecting a file, select the path and name of the new project.

Open

Opens an existing project. The default extension is <name>.dspt

Save

Saves the current project. The default extension is <name>.dspt

Save as

Saves the current project in a new project.

Run

Runs the messages dispatching process, using the current project settings.

Stop

Stops the dispatching process. The program returns to the Project design mode.

Exit

Ends program.

1.3. Licensing

The Alarm Dispatcher does not require a license if used by Movicon since it uses the Movicon license. But, if used by third party programs, the program requires an activation code which must be set by means of the COM interface.

BOOL AlDInit(LPCTSTR lpszSerialNumber)
Subject: LPCTSTR lpszSerialNumber License code for deactivating Demo mode.
Return: TRUE (1) if code entered is correct.

Allows to release demo operating mode. In demo mode, the functions are complete, except for the fact that the message sent is a constant text, and cannot be modified by the user. Contact the Progea sales offices for the license code.

- The activation code must be handled as an initialization parameter by the program user code. Contact the Progea sales offices (sales@progea.com) for more information
- The code acquired is valid for using the program on a single station only

1.4. Multilingual Message Management

When the Alarm Dispatcher is used in Movicon messages can be managed in multilingual. A default language can be associated to each user or user group in the Movicon project allowing messages to be sent to the dispatcher in the user's language (this would need a String Table to be setup in the Movicon project to manage the different languages). This means, for instance, that if a mail is sent to three different users, each one will receive it in their own language set for default. This mechanism works for sending messages by SMS, E-mail and FAX. However, messages cannot be sent vocally as the TTS engine does not support any options to modify the spoken language of the voice message with which the Dispatcher is to execute the text speech. This means that independently from the language used in the text passed to the Dispatcher, the spoken text will be sent to the user with the TTS Engine set in the Dispatcher's "Voice Message Sender" plugin (ie. by selecting the Sam item, it will be spoken with the English phonemes, even if the message text is in Italian or German, etc.).

2. Settings

2.1. General Settings

The General Settings make it possible to define certain general configurations for the entire project. There are three sections: Log, Communication and Priority delays.
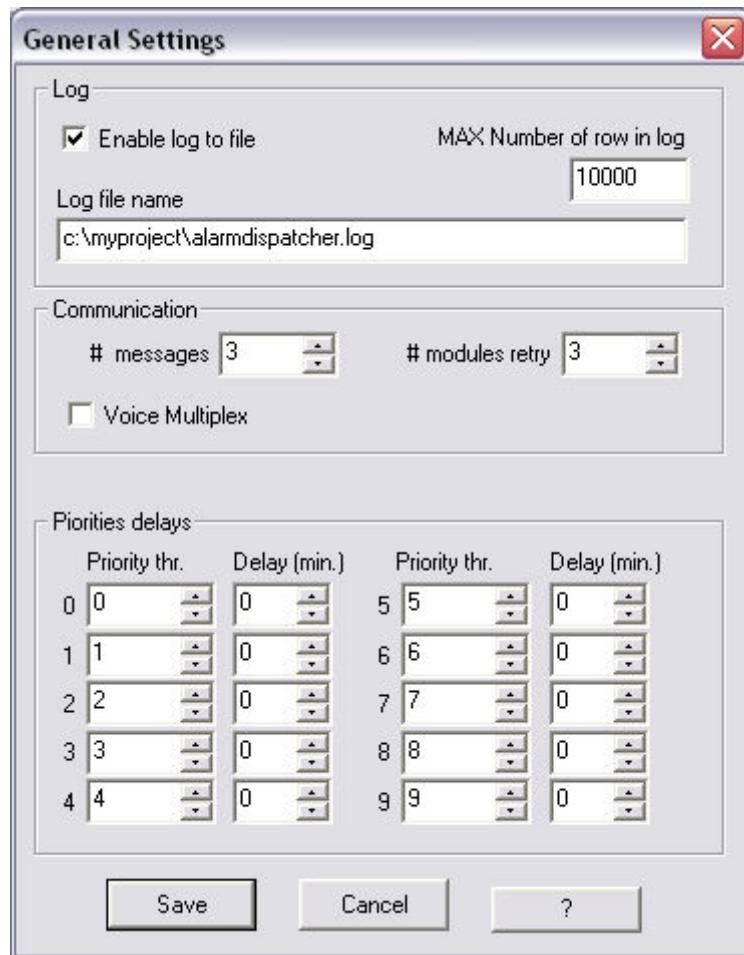
Log

Check enabling for recording the file path and the maximum number of lines on event files, before the file starts recirculating.

- **Enable log to file**: if enabled, it causes generation of a file containing a trace of events which occur during the running of the program, i.e. the start and end, sending of messages and dispatch errors.
- **Max Number of row in log**: the maximum number of rows in log. The successive recordings will be overwritten on the older rows (recirculation).
- **Log file name**: a name to be given to the log file can be specified here. If the box remains empty, the log file will have the same name as the project, with extension .dsplog.

Communications

Determines the messages notification method.

- **# Messages**: Determines the number of consecutive attempts for message notification. After the specific number of attempts, the program indicates the error in sending the message.
- **# Modules retry**: Determines the number of consecutive errors in messages notification for a single module (Plug In). After the specific number of attempts, the program indicates the error in the specified module.
- **Voice Multiples**: If this option is selected, notification of vocal messages will however be sent to all the recipients of a group, independently of individual acknowledgement by each recipient user.

Priority Delay

Determines the messages notification method.

- **Priority Thread - Dealy**: parameters relative to messages notification priority. Each priority has a corresponding delay time (min.). If the message is notified by Movicon, the priority assigned to the alarm identifies the corresponding notification delay time. In case of requests for sending messages to third party programs, it is necessary to assign a send priority (SendMessage function), the corresponding number of which will determine the relative delay time.

Example: if there are one or two alarms to be notified, the program will check the priority number and consequently the delay time assigned. It will then proceed with sending the message with the lower Delay time first, identifying the time on the basis of the priority number.

2.2. Users

The Recipients settings window is used to enter the general data regarding recipients of notifications, available for the project.
The users inserted will be listed on the left.
Each user can be added using the "**Add**" button, modified using the "**Modify**" button or removed using the "**Delete**" button.
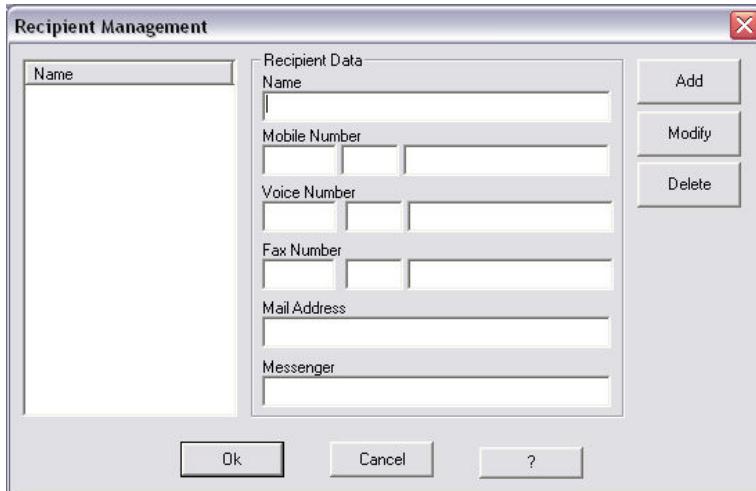
Name

Defines the Recipient's name. It may be any string of characters.

Mobile Number

Defines the recipient's mobile phone number for receiving the SMS messages notified by the program.
The number consists of three parts: the international code, area code and phone number.

Voice Number

Defines the recipient's mobile phone number for receiving the Voice messages notified by the program.
The number consists of three parts: the international code, area code and phone number.

Fax Number

> Defines the recipient's mobile phone number for receiving the fax messages notified by the program.
> The number consists of three parts: the international code, area code and phone number.

Mail Address

> Defines the E-mail address for receiving the E-mail messages notified by the program.

Messenger

> Defines the address of the recipient user connected online by means of the Microsoft Windows Messenger program.

Use by means of Movicon

> Movicon uses the Alarm Dispatcher functions in a completely transparent manner. The messages are transmitted by the properties of the Movicon project alarms, and the user profiles (Recipients) are those of the Movicon project users management.
> However, it is possible to use the Movicon VBA Scripts to customize the management of notification according to the requirements.

2.3. Plug-In Modules

The PlugIn (Notification Modules) settings window makes it possible to insert the messages notification transmission mode, using the plug-in modules available in the program.
Each plug-in module available can be freely inserted and configured in the program, depending on the requirements.
The plug-ins available are listed on the left, while those inserted will be listed on the right.
Each plug-in can be selected and then added using the "**->**" button, or removed using the "**<-**" button.
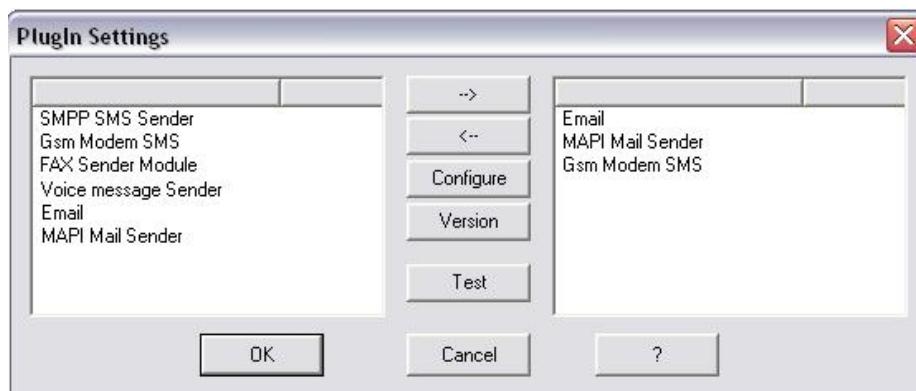
Each plug-in module inserted must be configured. To activate the module configuration window, select it from the list of modules inserted and double click or use the "**Configure**" button. The window will be displayed with the relative settings.

The "Version" button is used to check the version and release of each module, among those inserted.
The version number is useful for checking the availability of the latest versions.

The "Test" button allows you to run a test on the plugin settings by sending a test message. By pressing this button a window will open for each plugin, where you can insert recipient data, enter a test message and try to send it.

2.4. Using the Alarm Dispatcher in WinCE

The Alarm Dispatcher can also be used in Windows CE versions. However, when doing so you will need to keep in mind a few things. The Alarm Dispatcher for WinCE will always load the "AlarmDispatcher.Dspt" project. This project must be found in the Movicon CE installation folder together with the other email and SMS plugin configuration files (plugins for sending vocal messages are not supported).
All these files mentioned above are installed with the Alarm Dispatcher but may need to be modified. For instance, you might need to modify the DOM port in which the Alarm Dispatcher searches the GSM modem, or the SMTP Server to be used for sending emails. These project files, installed by the setup, can be copied to the PC desktop for modifying with the Alarm Dispatcher editor and then recopied back into the Movicon CE installation folder.

In cases where the application is an integral part of the WinCE image, you will need to execute a routine at the project or device startup to copy the Alarm Dispatcher project files from the compact flash into the "\Windows" folder.

2.5. E-mails sender (via SMTP)

This settings window allows configuration of the messages notification system by e-mail using direct access to a server using SMTP protocol.
This type of notification makes it possible to send e-mail also in the absence of a mail sender program (e.g. Ms Outlook) installed on the plant PC. The notification system however requires a modem for internet access.
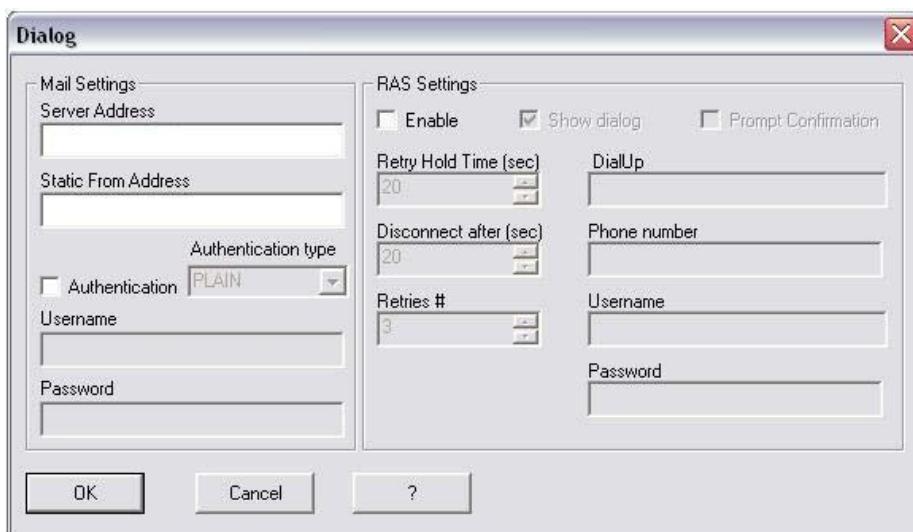
Server Address

This setting makes it possible to define the name of the mail server that handles the account.

Static From Address

This setting allows you to define an address to be used as the sender's address in the message. This setting is useful for using servers which allow post to be sent only if the sender belongs to the domain it is being sent from.

Authentication

Makes it possible to specify if access to the server requires authentication for access.

Authentication Type

Allows selection of type of authentication required by mail server. The only type of authentication supported at the moment "Plain" and "Login".

UserName, Password

In these two edit boxes, specify the user name and password required for accessing the mail account.

RAS Settings

If access to the internet requires an RAS (modem) connection, the relative parameters must be specified for the call and the internet connection.

- **Enable**:enables use of a RAS connection for access to server via modem. If this is not the case, it means access to the server (internet) uses a permanent connection (e.g. LAN)
- **Show Dialog**:enables display of the dialog window during the connection
- **Retry Hold Time**:Timeout for retry of connection
- **Retry Hold Time**:Timeout for disconnecting
- **Retries**: Number of retry attempts
- **Dial Up**: allows use of a connection already available in the operating system (previously inserted in the network connections)
- **Phone Number**: if an existing account is not used, specify the phone number for access to the service
- **UserName, Password**: if an existing account is not used, specify the parameters for access to the service

2.6. E-mails sender (via MAPI)

This settings windows allows configuration of the messages notification system by means of e-mail using the program for sending mail configured in Windows as "Mail sender Program" (like Ms Outlook).

Remember that this type of notification uses the mail sender program configured, therefore it is indispensable to make sure this program is installed and configured correctly.
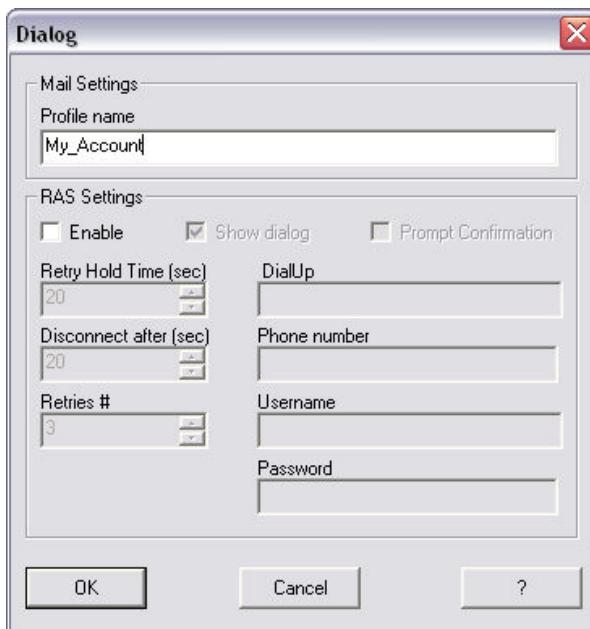
Profile Name

Allows specification of name of the profile (account) present among those recognized in the mail sender program installed. All the parameters for access to the mail server will therefore have to be configured in the account of the mail program (like, Ms Outlook).

RAS Settings

If access to the internet requires an RAS (modem) connection, the relative parameters must be specified for the call and the internet connection.

- **Enable**:enables use of a RAS connection for access to server via modem. If this is not the case, it means access to the server (internet) uses a permanent connection (e.g. LAN)
- **Show Dialog**:enables display of the dialog window during the connection
- **Retry Hold Time**:Timeout for retry of connection
- **Retry Hold Time**:Timeout for disconnecting
- **Retries**: Number of retry attempts
- **Dial Up**: allows use of a connection already available in the operating system (previously inserted in the network connections)
- **Phone Number**: if an existing account is not used, specify the phone number for access to the service
- **UserName, Password**: if an existing account is not used, specify the parameters for access to the service

2.7. SMS via Modem GSM

This settings window makes it possible to configure the messages notification system by means of the SMS technology based on GSM, using any standard GSM modem.
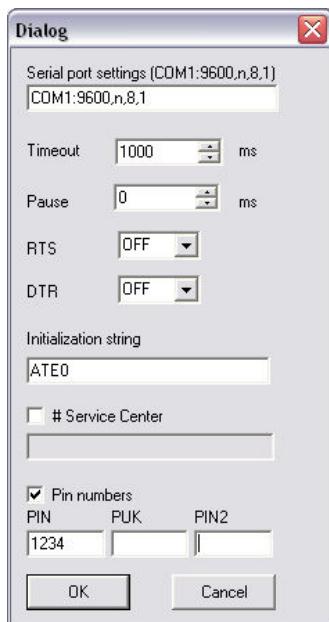
Serial Port Settings

In this settings group the parameters must be specified for using the serial port to which the GSM modem is connected.

Inizialization String

Modem initialization String ATE0 is normally used, but it is preferable to follow the documentation of the modem used.

Service Center

This number can be obtained from the SIM Card used by the modem.

Pin Numbers

Enter the PIN according to the SIM Card used by the modem.

2.8. SMS via SMPP

This settings window makes it possible to configure the messages notification system by means of the SMS technology based the SMPP (Short Message Point-To-Point) protocol.
Thanks to this notification technique, the system is capable of sending SMS messages to recipient phones by directly accessing a SMPP server, without the need for a GSM modem.
Obviously, it is necessary to check if a phone operator is available for the service, and then activate the subscription. Depending on the service provider, the relative functional and access parameters must be set.
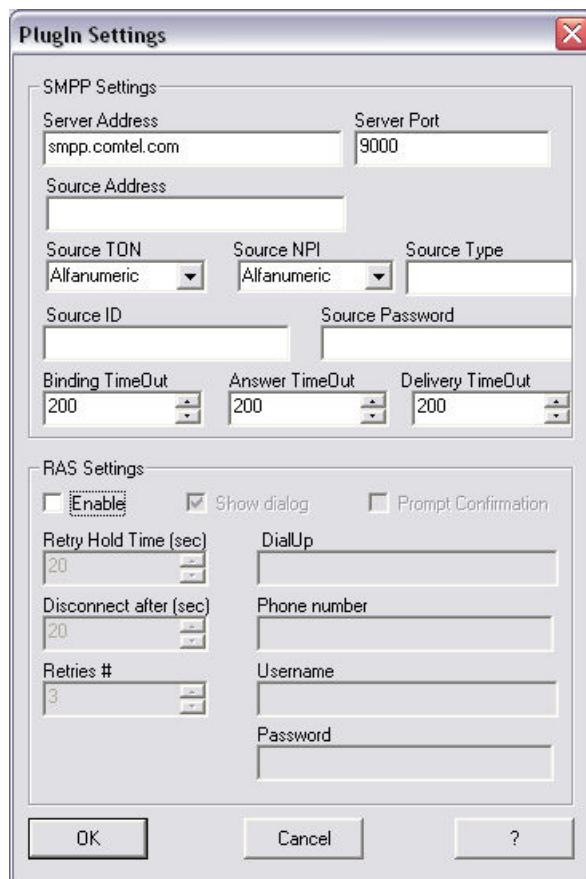
Server Address
Specify the phone operator SMPP server and service provider address.

Server Port
The 9000 port is usually required, but always check the server port required by the service provider.

Source TON, NPI, Type
Access parameters required by service provider.

Source ID, Password
Access parameters required by service provider. Received after subscription is activated.

Timeout: Binding, Answer, Delivery
Setting time (msec) relative to operating timeout.

RAS Settings
If access to the internet requires an RAS (modem) connection, the relative parameters must be specified for the call and the internet connection.

- **Enable**:enables use of a RAS connection for access to server via modem. If this is not the case, it means access to the server (internet) uses a permanent connection (e.g. LAN).
- **Show Dialog**:enables display of the dialog window during the connection

- **Retry Hold Time**:Timeout for retry of connection
- **Retry Hold Time**:Timeout for disconnecting
- **Retries**: Number of retry attempts
- **Dial Up**: allows use of a connection already available in the operating system (previously inserted in the network connections)
- **Phone Number**: if an existing account is not used, specify the phone number for access to the service
- **User Name, Password**: if an existing account is not used, specify the parameters for access to the service

2.9. SMS using Ucp-Emi

This settings window allows you to configure the message notification system with the SMS technology, using any modem standard to connect to a server supporting the Ucp protocol.
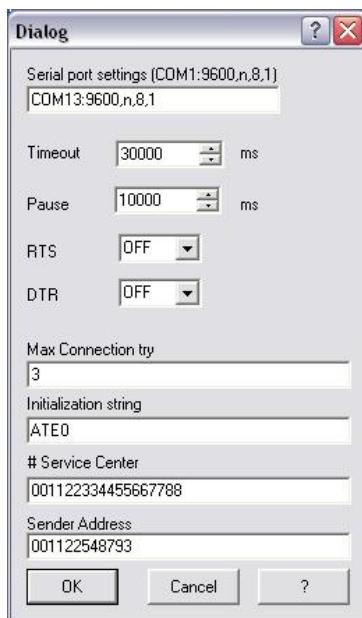
Serial Port Settings

Here you need to specify the parameters used of the serial port to which the modem is connected.

Timeout

Waiting time (in ms.) for communication operations in serial.

Pause

Pause time (in ms.) between sending two messages one after the other. Best not keep this time too low as Ucp servers usually cannot support sending messages in fast sequences. Default 5000.

RTS/DTR

Corresponding serial port signal status. Some modem models require this setting. This setting does not usually need changing for those devices currently available.

Max Connection try

Maximum number of attempts made in succession before message is deleted.

Inizialization String

The modem's initialization string. Normally ATE0 is used, but please double-check documentation of modem being used.

ServiceCenter

This is the number to call for connecting to the Ucp-Emi server.

Sender Address

This is the number used as message sender's number.

2.10. Voice Message Sender

This settings window is used to configure the messages notification system by means of the Voice-Text-To-Speech technology. Thanks to this technology, the program will vocally summarize the strings of text messages and notify these to the recipients via phone.

It is necessary to use a Voice Modem 100% compatible with the TAPI specifications.

Method for sending voice images:

1. The program notifies the voice messages to the recipient indicated. In case of group of recipients, the program sends the messages in two ways:

    *   **"Voice Multiplex"** property not selected (as default) in **General Settings**: The program notifies the messages starting from the first recipient of the group. The first recipient who acknowledges (Ack) the message determines the success of the notification procedure.
    *   **"Voice Multiplex"** property selected: The program notifies the messages starting from the first recipient of the group. The program however notifies the messages to all recipients in the group, independently of individual acknowledgement (Ack). The acknowledgement leads to a success of the notification procedure, but does not interrupt notification to all the recipients.

2. If a user answers a call, the program gives the welcome message and then starts notification of the alarm message. The alarm is repeated until the (Ack) command is given or until timeout. Acknowledgement may occur in two ways:

    *   **"Force message acknowledgement with key #"** property not selected (as default) in the Voice properties: The program notifies the message until one of the phone buttons is pressed, and will then consider the alarm as acknowledged and will pass on to the next one (if present), after giving the "New Alarm" message. The next message will also be repeated until one of the buttons is pressed. With the last message, if a button is pressed, the communication is shut off, with positive result. If no button is pressed, the system will repeat that message until timeout, and then shut off communication with negative result in the Log.
    *   **"Force message acknowledgement with key #"** property selected in the Voice properties: the program will act as above but will shut down the communication only if the "#" button is pressed on the phone. Thus, pressing a button will make the program move on to the next message, but at the end of the messages, the system will restart communicating the first message considering only the "#" button for confirmation and acknowledgement of alarms. If the "#" button is not pressed, the system will repeat the message until timeout. Pressing the "#" button will lead to acknowledgement of all the alarms, even those that are not yet notified.
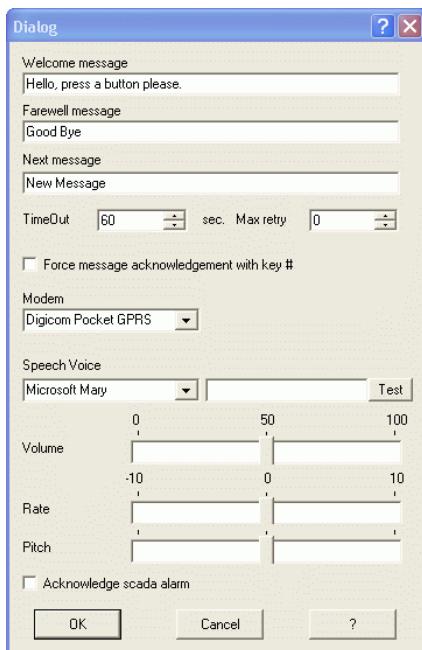
Welcome Message
Allows you to enter the string text to be transmitted as an introductory message, at the beginning of the notification call, preceding other messages.

Farewell Message
Allows you to enter a string text to be transmitted as a final message, at the end of the notification, to warn communication shutdown.

Next Message
Allows you to enter a text string to be transmitted between one notification message and the next.

TimeOut
Defines the time (sec.) beyond which the system will abandon communication, if no sign of acknowledgement is received (such as, for example, in the presence of an answering machine).

Max Retry
Defines the maximum number of attempts allowed for sending notifications.

Force message acknowledgement with key
This selection makes it possible to set the type of action by the recipient user for handling the alarms acknowledgement (Ack) command.

- **Option not selected**: on message notification, the system will wait until one of the phone buttons is pressed to acknowledge (Ack) the alarm being notified. If other alarms are present, the pressing of a button will cause the system to pass on to the next message, otherwise it will shut down the notification successfully.

- **Option not selected**: on message notification, the system will wait until the # button is pressed to acknowledge (Ack) the alarm being notified. Pressing # will shut down the notification process successfully, even in the presence of other alarms.

Modem
This selection permits you to choose which modem to use. The list of modems is compiled according to the modem voice installed in the operating system.

Speech Voice
This selection permits you to choose which item, and therefore which phoneme, to use for the message speeches. The list will be compiled based on the phonemes installed in the operating system (see "Phonemes (Speech)").

Test
Pressing the "Test" button will execute the speech of the text inserted in the editbox using the defined Speech voice, volume, rate and pitch settings.

Volume

This property is used for setting the volume percentage to be used for the speech text.

Rate

This property is used for setting the how fast the text should be spoken. The 0 to 10 values speed up the speech and the 0 to -10 slows it down.

Pitch

The speech voice tone is set through this property (low pitch to high pitch).

Acknowledge scada alarm

If this property is enabled, the alarm will also get acknowledged and reset in the Movicon project when operator acknowledges message. If the alarm in question is already in the OFF state, the Alarm Window will disappear.

Using SAPI 5.1

SAPI (Speech Application Programming Interface) is a Microsoft library consenting the use of "Speech Recognition" and "Speech Synthesis within Windows applications. The Alarm Dispatcher uses the SAPI version 5.1 for sending vocal messages. The SAPI 5.1 is part of the SAPI 5 family and is installed for default in Windows 32/64 bit from 2001 (see http://www.microsoft.com/downloads/details.aspx?FamilyId=5E86EC97-40A7-453F-B0EE-6583171B4530&displaylang=en).

To use SAPI 5.1 correclty for sending vocal messages with the Alarm Dispatcher you need to:

1. Install SAPI 5.1 Speech Recognition
2. Install at least one phoneme that supports SAPI 5.1
3. Use Alarm Dispatcher 1.00.11 or later
4. Use the Message Voice Sender PlugIn (Voice.dll) version 1.0.0.8 or later
5. Use a 100% compatible Voice modem with Windows TAPI

Voice Modem

To send vocal messages correctly you will need to install a Voice Modem which is fully compactible with Windows TAPI. It is advised that you also install the accompanying drivers with the modem and not the one selected by the operating system as being the most suited for that modem. For instance, if you install the "US Robotics 56K Message Modem" voice modem, you will need to install the "US Robotics 56K Message V.92" proprietary drivers, otherwise the system drivers (European) will not recognize the phone tones in order to respond to messages.

Phonemes (Speech)

To voice texts you will need to the phonemes which can be installed in the operating system. To check out the already existing list of phonemes just open the "Speech" icon on the Windows Control Panel. Then select the "Text to Speech" tab to open the dialog window containing the list of phonemes or voices (voice selection). Usually there are come voices installed by Windows for default, Sam, Mary and Mike. These voices use all the English language phonemes and are SAPI 4 phonemes which cannot be used with the Voice Message Sender Plug-in. Therefore, you will need to install new phonemes based on the language you wish to use and must be compactible with SAPI 5.1. Keep in mind that you will find a great number of phonemes available on the market from different companies (on the internet), and their quality may vary a great deal between one company to the next. Therefore it is advised that you try various phonemes out before making the right choice. You should also take into account that some phonemes may have more restrictions than others. Normally the 'telephone' (8KHz) speech synthesis quality is better to use over the phone than higher quality speech synthesis (22KHz or other).

> When using SAPI 5.1 you can also use accented characters such as "è"and "ì" etc. and punctuations marks such as "?" and "!" without causing problems in the TTS Engine. This is due to the fact that the "vcmd.exe" process is no longer used, which used to block the TTS Engine when using SAPI 4.

2.11. Fax Sender

This settings window is used for configuring the messages notification system by means of the Windows 32/64 bit Fax Sender.
The program makes it possible to send a fax using the Windows fax printer.
The only property to be configured in the program is therefore the "Title" of the fax to be sent.
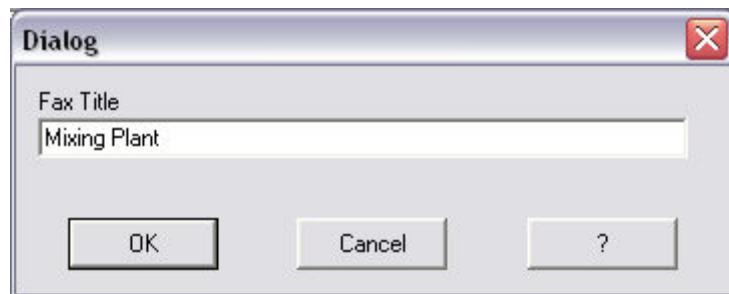
Before using the notification service via Fax, proceed with correct and complete configuration of the service in Windows.
To do so, use the Control Panel to access the "Printers and Fax" item.
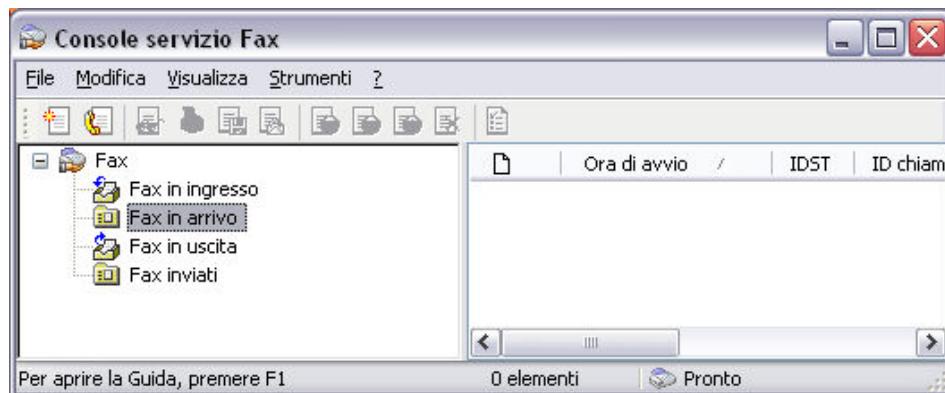
3. Errors

3.1. Plugin errors

GsmSMS

| ERROR_INIT_FAILED	-1	serial opening error.
ERROR_NO_MODEM_INIT	-2	modem initialization error. Answer received with error, or no answer. In this phase, the initialization string, ending with \r, is passed to the modem.
ERROR_NO_PIN	-3	error or no answer after sending PIN The "AT+CPIN=<pin number>\r" string is sent to the modem. If an error is received in response, or no answer is received within timeout, an error occurs.
ERROR_NO_SERVICE	-4	error or no answer after sending service centre number. The "AT+CSCA?\r" string is sent for requesting the current service centre number set. If different from that present in the settings, it is sent together with the "AT+CSCA=<num. Centro servizi>\r" string. If an error is received in response, or no answer is received within timeout, an error occurs.
ERROR_NO_ASCII_FORMAT	-5	error or no answer following ASCII mode setting. The AT+CMGF?\r" string is sent to check if the ASCII mode is already set. If it is not set, do so using the "AT+CMGF=1\r" string. If an error is received in response, or no answer is received within timeout, an error occurs.
ERROR_NO_WRITE_NUMBER	-6	phone number writing failure The "AT+CMGS=\"<telephone.num>\"\r" string is sent.
ERROR_NO_ANSWER	-7	no answer to writing of phone number.
ERROR_NO_WRITE_MESSAGE	-8	error while writing message. The "<message>\x1A " string is sent
ERROR_NO_ANSWER_MESSAGE	-9	error or no answer to message sent.

SMTP

ERROR_PARSE_MESSAGE	-1	error in format of string containing message and data to be sent.
ERROR_CONNECTING_RAS	-2	error in RAS connection.
ERROR_WINSOCK_INIT_FAILED	-23	error in stack TCP initialization.
ERROR_SMTP_SERVER_CONNECT_FAILED	-24	error during connection to SMTP server.
ERROR_SMTP_SEND_MAIL_FAILED	-25	error while sending message to SMTP server.

Voice

ERROR_PARSE_MESSAGE	-4	format error in the string containing message and data to be sent.

SmppSMS

ERROR_BIND_TIMEOUT	-1	No answer from SMPP server, on request for connection. The Binding time has run out.
ERROR_CONNECTING_RAS	-2	Error in completing RAS connection.
ERROR_SUBMIT_TIMEOUT	-4	No answer to SMS sent. Answer timeout
ERROR_ENQ_ANSWER_TIMEOUT	-5	No answer to SMS sent request. Delivery timeout.
ERROR_DELIVERY_TIMEOUT	-6	The message was not sent within the Delivery timeout.

MAPI

ERROR_PARSE_MESSAGE	-1	error in format of string containing message and data to be sent.
ERROR_CONNECTING_RAS	-2	Error in completing RAS connection.
ERROR_MAPI_NOT_INSTALLED	-4	Impossible to load MAPI library. Check installation.
ERROR_MAPI_LOGON_FAILED	-5	Connection to specified account failed. Check account data.
ERROR_MAPI_SEND	-6	Error while sending message. Check message data.

Fax

ERROR_FAX_ERROR	-2	Error in transferring to Microsoft Fax Service. Control service installation and message data.

UcpSMS

ERROR_INIT_FAILED	−1	Error when opening serial.
ERROR_NO_MODEM_INIT	−2	Error in modem initialization. Answer returned with error or no answer. The initialization string terminated with \r is passed to modem in this phase.
ERROR_NO_WRITE_NUMBER	−6	Error in written telephone number. The "AT+CMGS=\"<telephone num.>\"\r" string is sent.
ERROR_NO_ANSWER	−7	No answer to written telephone number.
ERROR_NO_WRITE_MESSAGE	−8	Error while writing message. The "<message>\x1A " string is sent.
ERROR_NO_ANSWER_MESSAGE	−9	Error or no answer when sending message.
ERROR_BAD_MESSAGE	-11	Message has invalid format.

4. DispatcherDOM

4.1. DispatcherDOM Object

The **DispatcherDOM** object is the script interface allowing to initialize the Dispatcher and to manage notification with the PlugIns installed and set in the currently running project.

The Alarm Dispatcher can be used by any third party applications, being Scada/HMI software, VB, C/C++ or other applications, simply using the Windows ROT (Running Object Table) to get a reference to the DispatcherDOM object, that the Dispatcher itself registered when it started with a valid project.
The Alarm Dispatcher can act as a notifying Server for Client applications that can access programming interfaces of applications registered in Windows ROT (Running Object Table).

VBScript, Visual Basic example

```
Dim objDisp As Object
Private Sub Form_Load()
        Set objDisp = GetObject(, "Dispatcher.1")
        If objDisp Is Nothing Then
                MsgBox "Alarm Dispatcher not running!", vbinformation + vbOkOnly,
                "DispatcherDOM "
                Exit Sub
        End If
End Sub
```

4.2. DispatcherDOM.AIDInit

Allows to license the Alarm Dispatcher.

Syntax

DispatcherDOM.AIDInit(lpszSerialNumber)

Parameters

DispatcherDOM
A valid DispatcherDOM object obtained from the Windows ROT.

lpszSerialNumber
Alphanumeric string in "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX" format representing the Alarm Dispatcher license code.

Return Value

Returns a boolean value

False	Invalid licence code
True	Valid licence code: the Dispatcher is properly licensed.

Remarks

When the Alarm Dispatcher is used by third party applications, it is required to insert the license code using this method.

Please ask for Alarm Dispatcher license code to Progea sales offices

The purchased application license code is valid for one station only.

When Alarm Dispatcher is used together with Movicon, no license is required since Movicon itself automatically runs the Dispatcher and licenses it. The Dispatcher can also be run using a Movicon basic script function called **StartAlarmDispatcher.**

VBScript, Visual Basic example

See example for the SendMessage function.

4.3. dispatcherdom.emptyqueue

This allows you to eliminate messages in queue that havem't yet been sent by the Dispatcher.

Syntax

DispatcherDOM.EmptyQueue(Void)

Parameters

DispatcherDOM
A DispatcherDOM object recovered from Window's ROT.

Return Value

Observations

When eliminating messages from the queue, those being processed in the Dispatcher and still not sent would not get eliminated.

VBScript, Visual Basic Example

```
Public Sub Click()
        Set objDisp = GetObject(, "Dispatcher.1")
        If objDisp Is Nothing Then
        MsgBox "Alarm Dispatcher not running!", vbInformation + vbOkOnly, "DispatcherDOM "
        Exit Sub
        Else
        objDisp.EmptyQueue
        End If
End Sub
```

4.4. DispatcherDOM.GetDispatcherState

Returns the Dispatcher state (Run or Stop).

Syntax

DispatcherDOM.GetDispatcherState

Parameters

DispatcherDOM
A valid DispatcherDOM object obtained from the Windows ROT.

Return value

Returns a 32 bits long integer, possible values are:

0	The Dispatcher is in Stop state.
1	The Dispatcher is in Run state.

Notes

When in Stop state the Dispatcher is able to collect in its buffer the notify messages, but it does not send them until its state switches to Run.

VBScript, Visual Basic example

```
Dim objDisp As DispatcherLib.DispatcherDOM
Public Sub Click()
        Dim lResult As Long

        If objDisp Is Nothing Then Set objDisp = GetObject(, "Dispatcher.1")
        If objDisp Is Nothing Then
                ' Alarm Dispatcher not running
                Exit Sub
        End If
        lResult = objDisp.GetDispatcherState
        If lResult <> 0 Then
                objDisp.SetDispatcherState(True)
        End if
End Sub
```

4.5. DispatcherDOM.GetMsgStatus

Allows to know the present state of a message that has been notified to the Dispatcher.

Syntax

DispatcherDOM.GetMsgStatus(dwTransaction)

Parameters

DispatcherDOM
A valid DispatcherDOM object obtained from the Windows ROT.

dwTransaction
A 32 bits long integer representing the numeric code of the message notified to the Dispatcher with the SendMessage function.

Return value

Returns a 32 bits long integer, possible values are:

0	Message has already been sent
1	Message aborted due to communication errors.
2	Message is waiting to be sent.
3	The *dwTransaction* parameter value is invalid, there is no

	corresponding message.
4	m4essage has been removed by the RemoveMessage function.
5	The *dwTransaction* parameter value corresponds to a grouping message (voice message).

VBScript, Visual Basic example

This example skips the Dispatcher initialization code.

```
Dim objDisp As DispatcherLib.DispatcherDOM
Public Sub Click()
          ' …
          ' …
          ' …

          sMessage = "email|admin@provider.com~Dispatcher Message~Send message
          using DOM |1|UsersGroup|User|||||||||||one@hotmail.com;two@hotmail.com||"
          lMessageID = objDisp.SendMessage (sMessage)

          If lMessageID < 0 Then
                  ' Error send the message to diaspetcher
          End If

          Do
                  lResult = objDisp.GetMsgStatus(lMessageID)
                  DoEvents
          Loop Until lResult = 0
End Sub
```

4.6. DispatcherDOM.RemoveMessage

Allows to remove a message from the Dispatcher messages queue.

Syntax

DispatcherDOM.RemoveMessage (dwTransaction)

Parameters

DispatcherDOM
A valid DispatcherDOM object obtained from the Windows ROT.

dwTransaction
A 32 bits long integer representing the numeric code of the message notified to the Dispatcher with the SendMessage function.

Return value

Returns a 32 bits long integer, possible values are:

0	Message not found in the messages queue.
1	Message successfully removed from the messages queue.
2	Message could not be removed from the messages queue since it was not accessible at that time. Try again later.

Remarks

Only messages sent with a delay can be removed, since they are added to the messages queue and notified only after the delay time. For any further info about delay setting please refer to general settings.

VBScript, Visual Basic example

This example skips the Dispatcher initialization code.

```
Dim objDisp As DispatcherLib.DispatcherDOM
Public Sub Click()
            ' …
            ' …
            ' …

       sMessage = "email|admin@provider.com~Dispatcher Message~Send message
       using DOM |1|UsersGroup|User|||||||||one@hotmail.com;two@hotmail.com||"
       lMessageID = objDisp.SendMessage (sMessage)

       If lMessageID < 0 Then
               ' Error send the message to diaspetcher
       End If
       lResult = objDisp.RemoveMessage (lMessageID)
End Sub
```

4.7. DispatcherDOM.SetDispatcherState

Sets the Dispatcher state to Run or Stop mode.

Syntax

DispatcherDOM.SetDispatcherState(bGo)

Parameters

DispatcherDOM
A valid DispatcherDOM object obtained from the Windows ROT.

bGo
Boolean value to set the Dispatcher state:

- False: The Dispatcher is stopped (set to Stop mode)
- True: The Dispatcher is started (set to Run mode)

Return value

Returns a 32 bits long integer,with value "1".

Remarks

The Alarm Dispatcher application can be started directly in Run mode using the "/R" option from the command line. It is also possible to use this function to toggle the Dispatcher state between Run and Stop. If the Dispatcher state is Stop, it can collect notifications in its messages queue, but these are not sent until the dispatcher state becomes Run.

VBScript, Visual Basic example

See the example for the GetDispatcherState function.

4.8. DispatcherDOM.SendMessage

Allows to send a message using as a transmission channel any of the PlugIns installed in the currently running Alarm Dispatcher projects.

Syntax

DispatcherDOM.SendMessage (lpszMessage)

Parameters

DispatcherDOM
A valid DispatcherDOM object obtained from the Windows ROT.

lpszMessage
String including all the parameters required to send the notification. The pipe "|" character is used as a separator between parameters. The parameters should be added in the order as shown below:

"Media Type|Message|Group Name|Recipient Name|Mobile Int Code|Mobile Area Code|Mobile Number|Voice Int Code|Voice Area Code|Voice Number|Fax Int Code|Fax Area Code|Fax Number|Email Address|Messenger Address|Add Flag"

Media Type	Required. It defines the type of the plug-in the Dispatcher will use to send the notification. The plug-in needs should have been properly configured in the Dispatcher project, in order the notification can be sent properly. The following values are possible: *sms, email, voice, smpp, gsm, smtp, mapi* or *fax* SMS sending: *sms* : sends an SMS notification using the "SMPP SMS Sender" or "GSM Modem SMS" plug-in. *smpp* :sends an SMS notification using the "SMPP SMS Sender" plug-in. *gsm* : sends an SMS notification using the "GSM Modem SMS" plug-in. EMAIL sending: *email* : sends an email notification using the "Email" or "MAPI Mail Sender" plug-in. *smtp* : sends an email notification using the "Email" plug-in. *mapi* : sends an email notification using the "MAPI Mail Sender" plug-in. Voice message sending: *voice* : sends a voice notification using the "Voice message sender" plug-in. Fax sending: *fax* : sends a fax notification using the "FAX Sender Module"plug-in.
Message	Required. It represents the message text to be sent. See remarks about email sending.
Priority	Required. It defines the message priority level. In project general settings delays in messages sending can be set, related to messages priority level.
Group Name	Required. It defines the internal group name used to send the message.
Recipient Name	Required. It defines the name of the recipient to be used for message sending. If all the following parameters are left empty, the recipient information are searched in the internal list of the project recipients. If not, following parameters values are used.
Mobile Int Code	Optional. It defines the international code of the phone number to be used for SMS notification, related to the recipient as set in "Recipient Name" parameter.
Mobile Area Code	Optional. It defines the mobile phone operator or district code of the phone number to be used for SMS notification, related to the recipient as set in "Recipient Name" parameter.
Mobile Number	Optional. It defines the phone number to be used for SMS notification, related to the recipient as set in "Recipient Name" parameter.

Voice Int Code	Optional. It defines the international code of the phone number to be used for voice notification, related to the recipient as set in "Recipient Name" parameter.
Voice Area Code	Optional. It defines the mobile phone operator or district code of the phone number to be used for voice notification, related to the recipient as set in "Recipient Name" parameter.
Voice Number	Optional. It defines the phone number to be used for voice notification, related to the recipient as set in "Recipient Name" parameter.
Fax Int Code	Optional. It defines the international code of the phone number to be used for fax notification, related to the recipient as set in "Recipient Name" parameter.
Fax Area Code	Optional. It defines the mobile phone operator or district code of the phone number to be used for fax notification, related to the recipient as set in "Recipient Name" parameter.
Fax Number	Optional. It defines the phone number to be used for fax notification, related to the recipient as set in "Recipient Name" parameter.
Email Address	Optional. It defines the email address to be used for email notification , related to the recipient as set in "Recipient Name" parameter. More than one recipient can be set, using the semicolon ";" character as a separator, for example: "one@hotmail.com;two@hotmail.com"
Messenger Address	Non in use. To be left empty.
Add Flag	Parameter that allows to send command to the Dispatcher. At the moment only the "a" command is available, used to add the recipient to the internal recipients list .

Return value

If no errors occur a positive 32 bits long integer is returned, as identifier for the message notified to the Dispatcher. Otherwise one of the following error codes is returned:

- 1	ERROR_SEND_SYNTAX	Syntax error in the *IpszMessage* parameter.
- 2	ERROR_SEND_NOMESSAGE	No message text has been specified in the *IpszMessage* parameter.
- 3	ERROR_SEND_NORECIPIENT_NAME	No recipient name has been specified in the *IpszMessage* parameter.
- 4	ERROR_SEND_RECIPIENTDATA_NOTAVAILABLE	The data related to the recipient specified in the *IpszMessage* parameter can not be found, since: The specified recipient name is not present in the Alarm Dispatcher project recipient list, or The recipient data are not specified in the *IpszMessage* parameter.
- 5	ERROR_SEND_NTRANS_NOTVALID	No new message can be add to the sending messages queue. Try again later.
- 6	ERROR_SEND_MEDIA_NOTAPPLICABLE	The PlugIn specified in the *IpszMessage* parameter is not present in the project.
- 7	ERROR_SEND_SMS_MESSAGE_TOO_LONG	SMS message specified in the *IpszMessage* parameter exceeds the maximum allowed length.
- 8	ERROR_SEND_MEDIA_NOTLOADED	The PlugIn specified in the *IpszMessage* parameter has not been configured or does not exist.

Remarks

The rules for *lpszMessage* parameter setting depend on the kind of message to be sent. Sending of an email or voice message requires special attention, the specific rules are described in detail below.

Email sending

When sending an email message, the "Message" parameter should have the special format shown here :

"from~subject~mail message~file1~file2~...~file(n)"

where:

from: email message sender
subject: email message title
mail message: email message text
file(n):   optional parameter including the name of a file to be sent as attached file
~:   separator character

> On Windows CE platform it is mandatory to specify the attached files absolute (complete) path. On Windows 32/64 bit platform, if only the file name is specified, the Dispatcher will search the file in current project folder.

Voice message sending

When sending a voice message to a recipients group, the "SendMessage" function should be called many times in order to define a recipients list the message should be notified to and a last time, specifying no recipient but only a group, so that the Dispatcher knows it should start the voice call process. The voice call is sent, in sequence, to all recipients belonging to the group defined this way until a recipient acknowledges the message using the dedicated phone key. The "Group Name" parameter becomes the internal name of the group including the list of the recipients, with all their data, to be used for voice message notification.

As an example, for a voice call to recipients "User1", "User2" and "User3", the "SendMessage" function should be called 3 times to define the recipients list, specifying the recipient data for each one (here only the voice full phone number):

- DispatcherDOM.SendMessage "voice|alarm on|1|UsersGroup|User1||||+39|059|111111||||||"
- DispatcherDOM.SendMessage "voice|alarm on|1|UsersGroup|User2||||+39|059|222222||||||"
- DispatcherDOM.SendMessage "voice|alarm on|1|UsersGroup|User3||||+39|059|333333||||||"

Then the "SendMessage" function should be called once more, passing only the group name as a parameter:

- DispatcherDOM.SendMessage "voice|alarm on|1|UsersGroup|||||||||||||"

The Dispatcher starts the voice message notification process calling in sequence all the recipients defined in the "UserGroup" group, until when one of the recipients makes an acknowledge.

VBScript, Visual Basic example

This example shows how to send an email to two email addresses. The "DISPATCHER_KEY" parameter should include the 'Alarm Dispatcher activation code.

```
Dim objDisp As DispatcherLib.DispatcherDOM
Public Sub Click()
        Dim sMessage As String
        Dim lResult As Long

        If objDisp Is Nothing Then Set objDisp = GetObject(, "Dispatcher.1")
        If objDisp Is Nothing Then
                ' Alarm Dispatcher not running
                Exit Sub
        End If
        lResult = objDisp.AIDInit(DISPATCHER_KEY)
        If lResult <= 0 Then
                ' DISPATCHER_KEY not valid
        End If
```

```
objDisp.SetDispatcherState(1)

' "Media type"|
' "Message"|
' "Priority"|
' "Group name"|
' "Recipient name"|
' "IntCodeMoblie"|"AreaCodeMobile"|"Mobile number"|
' "IntCodeVoice"|"AreaCodeVoice"|"VoiceNumber"|
' "IntCodeFax"|"AreaCodeFax"|"FaxNumber"|
' "emailAddress"|"messenger"|"AddFlag"
sMessage = "email|admin@provider.com~Dispatcher Message~Send message
using DOM |1|UsersGroup|User|||||||||||one@hotmail.com;two@hotmail.com||"
lResult = objDisp.SendMessage (sMessage)
If lResult < 0 Then
        ' Error send the message to diaspetcher
End If
End Sub
```

4.9. DispatcherDOM.PutLog

Allows to add to the log file a line with a customized message.

Syntax

DispatcherDOM.PutLog(lpszLog)

Parameters

DispatcherDOM
A valid DispatcherDOM object obtained from the Windows ROT.
lpszLog
String to be added to the log of the currently running Dispatcher project .

Return value

Returns a 32 bits long integer with value '0'.

Remarks

The Dispatcher automatically records in a log file <ProjectName>.dsplog many information related to the sent messages. This function allows to add customized messages. All the strings added to the log file are recorded with date and time.

VBScript, Visual Basic example

This example skips the Dispatcher initialization code.

```
Dim objDisp As DispatcherLib.DispatcherDOM
Public Sub Click()
        ' …
        ' …
        ' …
        sMessage = "email|admin@provider.com~Dispatcher Message~Send message
        using DOM |1|UsersGroup|User|||||||||||one@hotmail.com;two@hotmail.com||"
        lMessageID = objDisp.SendMessage (sMessage)
        If lMessageID >= 0 Then
                objDisp.PutLog "MessageID=" & CStr(lMessageID)
        End If
End Sub
```

MOVICON11

Movicon™ is a trademark of Progea, related to the HMI/SCADA platform entirely developed and produced by Progea. © 2012 All Rights reserved.
No part of this document or of the program may be reproduced or transmitted in any form
without the express written permission of Progea.
Information in this document is subject to change without notice and is not binding in any way for the company producing it.

Via S.Anna, 88/E
41100 Modena - Italy
Tel. +39 059 451060
Fax +39 059 451061
Email:info@progea.com
Http://www.progea.com

Via XX Settembre, 30
Tecnocity Alto Milanese
20025 Legnano (MI) Italy
Tel. +39 0331 486653
Fax +39 0331 455179
Email: willems@progea.com

Progea Deutschland GmbH
Marie-Curie-Str. 12
D-78048 VS-Villingen
Tel: +49 (0) 7721 / 99 25 992
Fax: +49 (0) 7721 / 99 25 993
info@progea.de

Progea International Ltd
via Penate 16
6850 Mendrisio - Switzerland
tel +41 (91) 9676610
fax +41 (91) 9676611
international@progea.com

Progea USA LLC
2800 East Enterprise Avenue
Appleton, WI 54914
Tel. +1 (888) 305 2999
Fax. +1 (920) 257 4213
info@progea.us