



# QuadLine QPager

## User manual

### **Copyright (c) 2000 QuadCore Automation B.V.**

No part of this document may be reproduced in any form or by any means, electronically, mechanical or otherwise, without the express written permission of QuadCore Automation B.V. All rights reserved.

The data and illustrations found in this document are not binding. Information in this document is subject to change without notice and should not be construed as a commitment by QuadCore Automation B.V. QuadCore Automation B.V. assumes no responsibility for any errors that may appear in this document and is not responsible for any consequential or incidental damage incurred by the use of this document



## **Preface**

This manual describes the installation and configuration of the QuadCore QPager. It is written for persons who have to install the software and configure a system using this application.

It is assumed that the reader is familiar with relational, arithmetic and logical operators, Windows 95. Windows NT operating system. Furthermore, some knowledge of DDE and DDEML

## Table of contents

<b>1. Overview.....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 Package contents.....	1
1.3 System requirements .....	1
1.3.1 Hardware .....	1
1.3.2 Software .....	1
<b>2. Installing the QPager software.....</b>	<b>2</b>
2.1 Installation instruction.....	2
2.2 Demo limitations.....	3
2.3 QuadLine authorization instruction .....	3
2.4 Speed issues.....	4
2.5 Compatibility issues.....	4
<b>3. Getting started .....</b>	<b>5</b>
3.1 User interface.....	5
3.1.1 Menu bar .....	6
3.1.2 Tool bar .....	7
3.1.3 Client area .....	7
3.1.4 Messages .....	8
3.2 Controlling your application .....	9
3.2.1 Run.....	9
3.2.2 Stop.....	9
3.2.3 Syntax check .....	9
3.3 Interpreter .....	10
3.3.1 Arithmetic.....	10
3.3.2 Relational.....	10
3.3.3 logical .....	11
3.3.4 Examples.....	11
3.4 Data types.....	11
3.4.1 Real.....	11
3.4.2 Boolean .....	12
3.4.3 String.....	12
<b>4. Objects .....</b>	<b>13</b>
4.1 DDE .....	13
4.1.1 User interface .....	13
4.1.2 Configuration .....	13
4.1.3 Properties .....	16
4.1.4 Methods.....	16
4.1.5 Events.....	16
4.1.6 Design limitations.....	16
4.2 SMS .....	18
4.2.1 User interface .....	18
4.2.2 Configuration .....	19
4.2.3 Design limitations.....	20
4.3 DTMF.....	21
4.3.1 User interface .....	21
4.3.2 Configuration .....	22
4.3.3 Design limitations.....	22
4.4 SMTP .....	23



---

4.4.1	User interface .....	23
4.4.2	Configuration .....	24
4.4.3	Design limitations.....	25
4.5	FAX.....	26
4.5.1	User interface .....	26
4.5.2	Configuration .....	27
4.5.3	Design limitations.....	27
4.6	Printer .....	28
4.6.1	User interface .....	28
4.6.2	Configuration .....	29
4.6.3	Design limitations.....	29
4.7	Phonebook.....	30
4.7.1	User interface .....	30
4.7.2	Functions .....	32
4.8	Timer .....	32
4.8.1	User interface.....	33
4.9	Export and Import Data.....	34
4.9.1	Export Data .....	34
4.9.2	Import Data .....	34
<b>5.</b>	<b>Problem Report / Change Request .....</b>	<b>36</b>



## 1. Overview

### 1.1 Introduction

Industrial production processes don't keep working hours. They go on day and night. But at any time there is a possibility that something goes wrong. Then it's good to have the QPager application available. Because it is an automated system which will automatically alarm you if something in your production process goes wrong.

QPager is an application, which is easy to connect with your production process through your SCADA system or directly with your PLC or PC. QPager continuously checks the predefined values from those applications to trigger a call. Also is it possible to combine several values into one statement by using the build in math and logic interpreter.

According to those predefined triggers the QPager will send variable messages to predefined recipients. QPager alerts your service engineer, supervisor or whomever you want to inform of the occurring error. By this manner the QPager warns you of threatening situations. Because of this pre-warning you are able to prevent production-stops or even worse calamities.

### 1.2 Package contents

The QPager package consists of the following components

- One disk or CD containing the software
- One Manual (this manual)
- A license card specifying your configuration sequence

### 1.3 System requirements

The QPager application is a dedicated software task and expects certain hardware and software to be present to operate correctly. These hard- and software requirements will be discussed in this section.

#### 1.3.1 Hardware

An Intel based PC with a Pentium 100 MHz or faster processor is used as the hardware platform. Also a Hayes compatible FAX-modem is needed to make use of the QPager FAX and modem depended functionality.

#### 1.3.2 Software

The QPager expects Windows 95, Windows 98, or NT 4.0 sp3 as operating system. Furthermore the QPager needs a TAPI driver for your FAX-modem to work correctly. Also a winsock.dll is used to send a SMTP message.



## 2. Installing the QPager software

Install the QPager with all necessary additional components.

If a component is skipped during installation, it can be added later on by running a custom installation.

It is important that you follow the instructions during the installation.

After running the installation you have to configure all installed objects separately.

**Note: Don't forget to register and authorize your application to a full version**

### 2.1 Installation instruction

Insert the QuadLine QPager Installation diskette or CD and install the software files by running the SETUP.EXE. The QuadLine QPager Setup program guides you through the installation.

After running the Setup program a welcome dialog appears on your screen. You can choose the *Next* button to continue with your installation or the *Cancel* button to abort the installation.

If you decide to proceed the installation of the QuadLine QPager, the software license agreement appears on your screen. NOTICE: Read the contents very carefully before proceeding with your installation. Choose the *Yes* button if you accept all terms of the license agreement or choose the *No* button if you want to abort the installation.

Choose the *browse* button to select the destination directory. Choose first the *OK* button and then the *Next* button to continue.

A Setup-type dialog appears on your screen, you can influence which files are copied to your disk by choosing one of the radio buttons corresponding to the desired setup type. Choose the *Next* button to continue.

Check the contents of the Start Copying Files dialog, and start copying all files to your disk by choosing the *Next* button. Now all files are copied to your disk and icons are added to your taskbar.

After some seconds the Software protection dialog appears on your screen. Please fill in your installation sequence. You can find the installation sequence on the license card enclosed in the package. Press the *Next* button.

Press the *Finish* button to close the setup program.



## 2.2 Demo limitations

If the QPager is unauthorized it will run in a demo mode.

The demo version is the same as the runtime version except that the demo version is time-bombed and will stop running after a limited number of contiguous hours. You have to restart the QPager by hand after it has stopped.

To avoid this behavior you have to **authorize** your copy.

## 2.3 QuadLine authorization instruction

The QuadLine Driver package is provided with a software protection mechanism. To enable this protection you need to have an installation sequence. The installation sequence can be found on the license card enclosed in the package. Use the Software Protection program to edit and view your installation, registration and Authorization-code. You can run this program by choosing the Software Protection icon in the taskbar under the map specified during installation. If you don't have a valid installation sequence or your sequence is out dated, you can check our webpage, or contact us directly to obtain a new sequence.

To be able to use the QuadLine application you must enter your Installation Sequence in the Software Protection program. This procedure has already been performed during installation. A valid installation sequence enables the application to run for a limited number of contiguous hours. If this time period is passed the application stops and has to be restarted. The installation sequence has a limited period of validity. Within this period of a couple of months you have to get your authorization sequence as described beneath.

As soon as you use the QuadLine application as part of an operating system, the driver checks the installation sequence. If the sequence is valid the driver creates the registration sequence. Use it to fill out the license card and send it to QuadCore Automation B.V. by mail or fax. Be sure to use the right operating system when you register your driver package (see Note).

On receipt of your license card with the registration sequence QuadCore Automation B.V. will check this sequence and will send you an authorization sequence within five workdays. Enter your authorization sequence in the Software Protection program. A valid authorization sequence enables the driver to run without any time limits.

Note: The authorization sequence is related to your operating system. Be sure to register your application package with the target operating system. QuadCore Automation B.V. issues only one authorization sequence per package. If you want to use an authorized driver package with another operating system you have to buy a new license.



## 2.4 Speed issues

The response speed or the delay between a trigger and its action depends on the hardware, amount of objects, intricacy of the used expressions, and object protocols.

If a performance problem occurs, it is imported that you check the overall performance of your system and software configuration.

Some protocols are extremely slow. It is not possible to speed up the communication with the host, since the host is responsible for the speed of the communication, and not the QPager.

### Note:

**Be careful to use object, which use the same hardware device. Because they can influence each other's behavior.**

## 2.5 Compatibility issues

This QPager is due its newly introduced enhanced object approach not downwards compatible with older releases. This means that you will have to reconfigure your old QPager configuration to complete an upgrade.

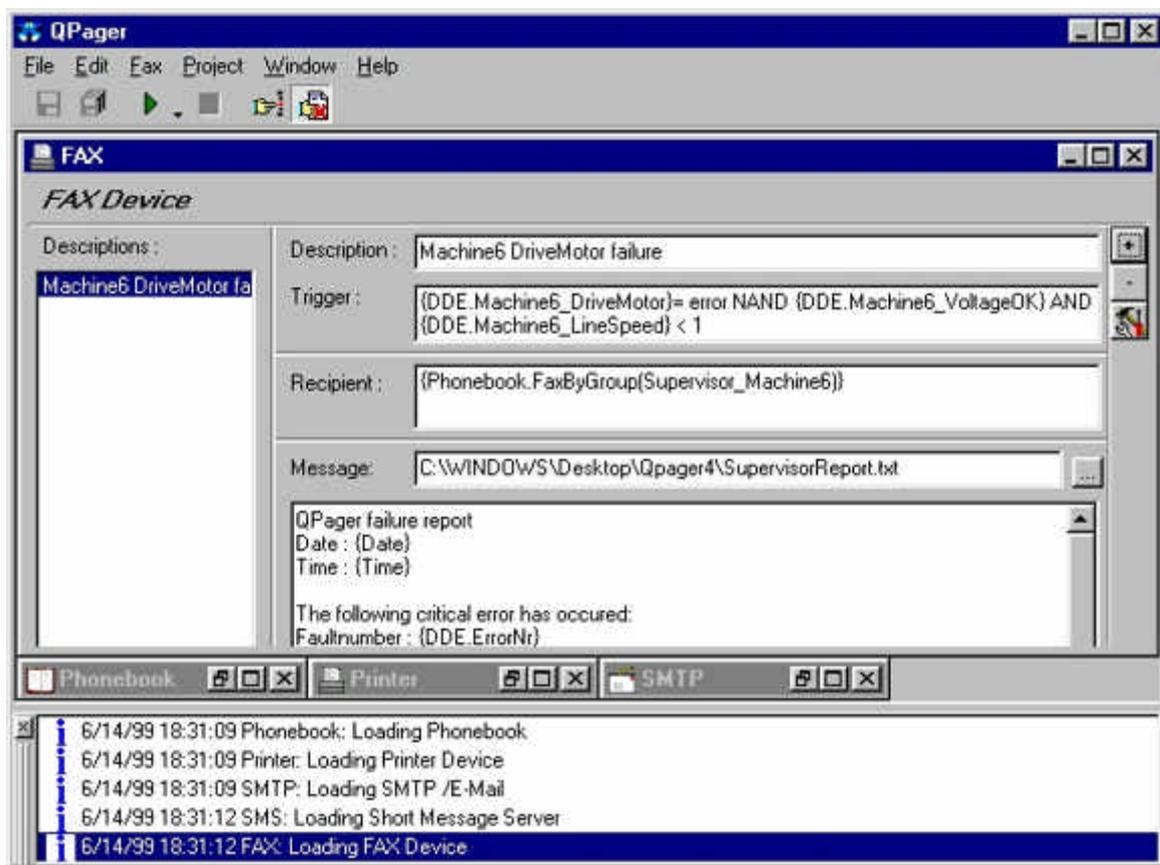
Furthermore the communication link between QPager and your application is not made by dedicated software but by standardized protocols. So maybe you have to alter the configuration in you host application.

### 3. Getting started

#### 3.1 User interface

The QPager user interface provides you with all necessary tools to edit and configure your application. The user interface is roughly divided into 4 parts:

- Menu bar
- Tool bar
- Client area
- Messages



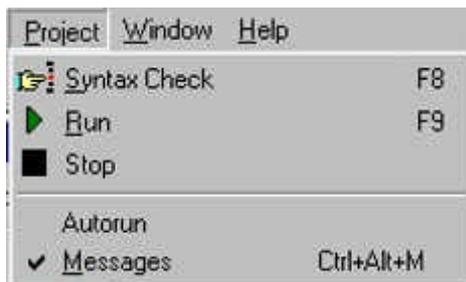
### 3.1.1 Menu bar

The menu bar of the QPager provides you access to almost all of its functionality. It contains some standard items and some object specific ones. In this chapter only the standard items are dealt with.

The menu bar has the following structure and meaning:



- |          |   |
|----------|---|
| Save     | Saves the contents of the currently active object |
| Save All | Saves the contents of all objects                 |
| Exit     | Closes down the QPager                            |



- |              |   |
|--------------|---|
| Syntax Check | Checks the contents of all objects for syntax errors in its statements<br>The results are displayed in the message window |
| Run          | Runs the QPager and activates all communication with the host application(s)  |
| Stop         | Stops the QPager but does not exit.   |
| Autorun      | Runs the application at application start   |
| Messages     | Enables or disables the messages window   |



- |               |   |
|---------------|---|
| Tile          | Tiles the open objects in the client area   |
| Cascade       | Cascade the open objects in the client area |
| Arrange icons | Arranges the icons in the client area       |
| 1..6          | Switches between the objects                |



- |                    |   |
|--------------------|---|
| Contents           | Opens the help contents                   |
| Search for Help On | Opens the search dialog                   |
| How to Use Help    | Provides you information about how to use |
| About              | Displays the about box                    |

### 3.1.2 Tool bar

The Toolbar provides a quick access to some of the functionality, which is described in the menu bar

### 3.1.3 Client area

The client area of the QPager displays all currently loaded objects. You can rearrange your objects by moving or re sizing them. It is also possible to manipulate the objects displayed in the client area by using the window item in the menu bar.

### 3.1.4 Messages

The message window displays all detail information from all objects. This window is also used for displaying Errors and warnings, which can occur during a syntax check, or at runtime.

All information displayed in the message window is also logged in a file named QPager.log

This file is located in the start-up directory of the QPager.log.

By pressing the left-mouse button on the message window, a pop-up menu appears. This pop-up menu enables you to choose which message level you want to display.





## **3.2 Controlling your application**

### **3.2.1 Run**

Choose Project|Run to execute your application.

If you have modified the source code of one of the objects you have to re-run your application. The QPager reloads those changed objects and re-links your application.

If the interpreter encounters a runtime error, it displays an Error in the message window.

### **3.2.2 Stop**

Stop program execution and aborts all running processes by choosing Project|Stop in the menu bar. If you stop your application all buffers will be emptied and all processes will be terminated.

### **3.2.3 Syntax check**

Syntax check is used to examine your application for design time errors without running the application. It is recommended that you always syntax check your application after you made some changes.

Choose Project|Syntax Check to check the objects of your project but not run them. This provides you with a means for checking your code for syntax errors. All errors are displayed in the message window.



### 3.3 Interpreter

The QPager accepts expressions in some input lines from several objects. Which lines accept expressions are described in the chapter describing the object.

An expression is a construction that returns a value.

For example:

```
{DDE.Value1}
```

The left and right brace indicates that the text between the braces is referring to an object. The text before the period points to an object name. This object name is the same as the name in the title of the object pointed to. The text after the period is object specific and described in detail in the chapter describing the object or object help file.

Left and right-parenthesis are used to indicate which part of the expression will be evaluated first. For example:

```
{(DDE.Value1} + 6) * 5
```

Expressions can contain arithmetic, relational and logical operators.

**Note: Expressions are case sensitive !!!!**

#### 3.3.1 Arithmetic

Arithmetic operators, which take real or integer operands, include +, -, \*, /.

Operator	Operation	Operand types
+	addition	Real
-	subtraction	Real
*	multiplication	Real
/	division	Real

A runtime error occurs when y is zero in an expression of the form x/y.

#### 3.3.2 Relational

Relational operators are used to compare two operands.

Operator	Operation	Operand type
=	equality	Real
<>	inequality	Real
<	less-than	Real
>	greater-than	Real
<=	less-than-or-equal-to	Real
>=	greater-than-or-equal-to	Real
==	same	String

### 3.3.3 logical

The Boolean operators AND, OR, NAND, NOR, XOR takes Operand types of the Boolean type.

Operator	Operation	Operand types
AND	conjunction	Boolean
OR	disjunction	Boolean
NAND	negation conjunction	Boolean
NOR	negation disjunction	Boolean
XOR	exclusive disjunction	Boolean

If logical operators are not placed between parentheses, the interpreter evaluates from left to right.

### 3.3.4 Examples

#### Logical operators

	Result
(1 AND 0)	0
((1 AND 0) OR (1 NAND 0))	1
{{DDE.Tag1} OR {DDE.Tag2}}	depends on tag values
{{{DDE.Tag1} AND {DDE.Tag2}} XOR 1}	depends on tag values

#### Arithmetic operators

	Result
(5 * 10)	50
(5 * {DDE.Tag1})	depends on tag values
((5 * ({DDE.Tag1} + 10)) / 100)	depends on tag values
((5 * ({DDE.Tag1} + 10)) / DDE.Tag2)	depends on tag values

#### Relational operators

	Result
(10 > 5)	1
(10 < 5)	0
{{DDE.Tag1} > 5}	depends on tag values
{{DDE.Tag1} <= {DDE.Tag2}}	depends on tag values
{{DDE.Tag1} == ABCDEF}	depends on tag values

#### Combinations

{{{DDE.Tag1} AND {DDE.Tag2}} XOR 1} >= ({DDE.MaxLevel} \* 100) OR {DDE.Error}

## 3.4 Data types

### 3.4.1 Real

Significant digits	Size in bytes
$3.6 \times 10^{-4951} \dots 1.1 \times 10^{4932}$	19–20



Notation: 0.000E+00

### 3.4.2 Boolean

1 represents True  
0 represents False

### 3.4.3 String

A string represents a sequence of characters.

For example:

ABCDEF  
AbcDEf  
ABCD12fg

Reserved characters and words are not allowed in an input line.

## 4. Objects

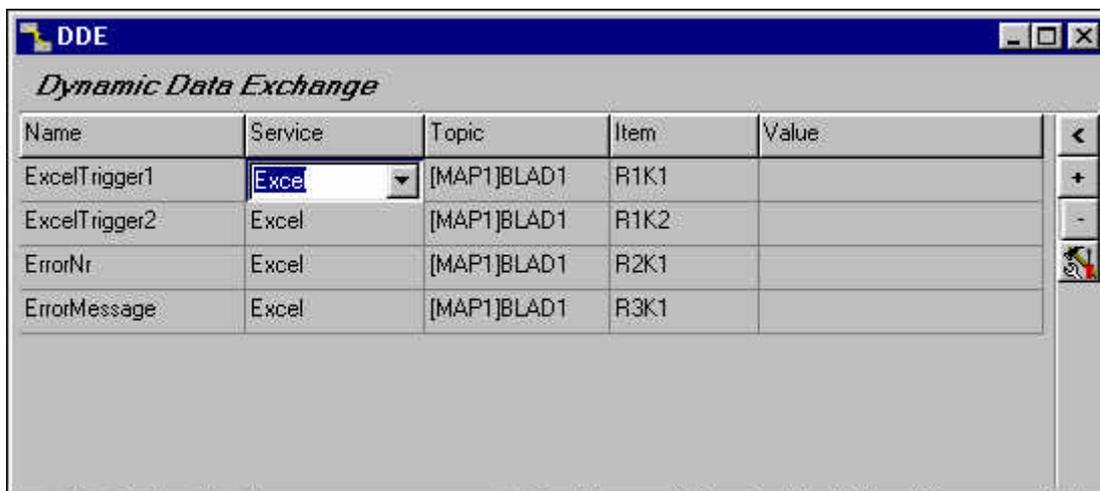
### 4.1 DDE

The DDE object is a QPager object that offers fully configurable DDE (*Dynamic Data Exchange*) functionality.

Within this single object you get all of the following features:

- A single DDE object can support multiple DDEML conversations.
- The DDE object component acts as a DDEML client and as a DDEML server or both!
- It supports both synchronous and asynchronous DDEML transactions.
- It uses a fully event based design for managing data movements and responses.
- Includes Network DDE support.

#### 4.1.1 User interface



#### **Name**

Unique name, which describes the DDE item, pointed to. This name may not contain spaces or punctuation marks

#### **Service**

Points to the service name of the host application.

#### **Topic**

Points to the topic name of the host application.

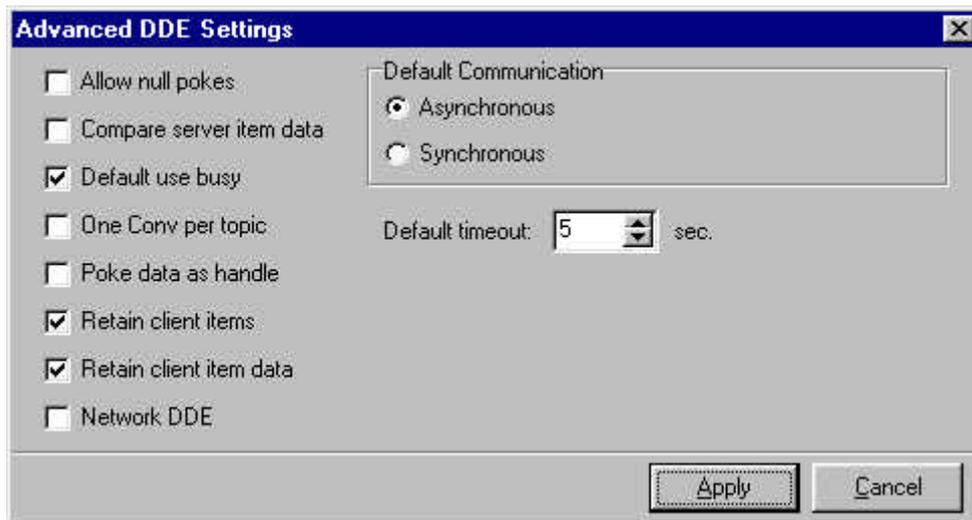
#### **Item**

Points to the item name of the host application.

#### **Value**

Shows the current value (if connected)

#### 4.1.2 Configuration



### Allow null pokes:

This property is used to control whether a 'Poke' transaction will accept NULL data (i.e. a 'NIL' pointer or a zero length string). If set to TRUE a POKE will proceed with NULL data. If set to FALSE the same POKE transaction will abort with a 'nil data' type error. By default the property should be set to FALSE.

### Compare server item data:

The DDE object uses its internal item data buffers to hold the values set into particular service items via any of the 'SetServiceItemXXXX' methods. Within these methods the DC will also compare the old stored buffer value against the new value. If the new value is different and the item is set up as a warm/hot link, then the QPager will initiate an advise message to the related client.

This property has been added to cater for situations where a server application itself outside of the QPager has already made that test. This property should be set to FALSE if the DC should not test for data changes.

For warm/hot linked server items, if this property is set to FALSE then all calls to one of the Service Items' methods will initiate advise calls.

This property should be set to TRUE

### Default use busy

This sets the default method by which the DDE object controls re-entrant processing of warm/hot link advice notices.

When a client application starts an advice link (either warm or hot), the server will send it an 'advise data' transaction each times the server item's data value changes.

With 'DefaultUseBusy' set to TRUE a client application will use an internal marker to denote the receipt of an advice data transaction. This marker remains in place until the advice data transaction processing is completed. If a subsequent advise transaction is received before the first advise transaction has completed then the client will not process the transaction and will return a 'busy' result back to the server. This process is useful in controlling large volumes of advice notices sent by a server over a short elapsed time frame.

If the 'DefaultUseBusy' set to FALSE then no such controls are made and the client application will attempt to process all advise data transactions.



This property should be set to TRUE.

### **One Conv per topic**

This Boolean property is used to test whether multiple conversations are allowed against the same server/topic pair. If set to TRUE only one conversation is permitted against a designated server/topic combination

### **Poke data as handle**

Internally DDE data can be poked by a client to a server using one of two mechanisms – by DDE data handle or by PChar type buffer. If 'PokeDataAsHandle' is set to TRUE then the DC will convert the item data into a DDE data handle and send the data handle to the server application. If 'PokeDataAsHandle' is set to FALSE then the data is sent as a PChar style buffer. The latter is normally faster. You should verify that the server is able to receive buffer style pokes before using this mechanism. When a conversation is started the new conversation will take the value of the 'PokeDataAsHandle' property as its own default state for all subsequent Poke transactions.

### **Retain client items**

The DDE object uses item level objects to retain information about an item's name and any data received from 'advise' or 'request' transaction.

When a client application uses (or receives data on) a large number of client items then this can impact memory requirements and increase search times to retrieve individual item objects.

To minimize the problem this property has been added to allow items to be disposed of once they have been used. If this property is set to FALSE then the internal item object is immediately removed at the end of the transaction process. This property is relevant for 'Request' and 'Poke' transactions and for advice transactions where an unknown item name is received.

This property should be set to TRUE.

### **Retain client item data**

The DDE object uses item level objects to retain item data received from an 'advise' or 'request' transaction.

If a client application is making use of a large number of different items then this can impose a large memory requirement on the application.

This property has been added to minimize this requirement. If the property is set to FALSE then the data associated with an item is cleared immediately after the request response has been processed.

This property should be set to TRUE

### **Network DDE**

This Boolean property is used to turn on or off the DDE object support for Network DDE functionality. If set to TRUE it will allow Network DDE conversations, if set to FALSE any such attempts at starting a conversation will be rejected.

### **Default communication**

This property holds the default data communication type (synchronous or asynchronous) that is applied to all new conversations.

### **Default timeout**



This property holds the default time out period in seconds that is applied to all new conversations.

#### 4.1.3 Properties

Each item name entered in the DDE object represents a property with the same name. If you want to use a DDE object property in an expression, the following syntax is used:

`{DDE.propertyname}`

The result of this expression is a string representation of the DDE item the property points to. This is independent of the kind of DDE communications you use.

#### 4.1.4 Methods

The DDE object can be controlled by a DDE Client. All property values can be changed by sending *poke* methods to the DDE object. Poke methods are used by client applications; this method is used to poke text and numeric data into the server party to the conversation.

Also it's possible to control the behavior of the QPager by sending commands. Commands are used by server applications. The QPager accepts the following commands:

[RUN]	runs the QPager
[STOP]	stops the QPager
[EXIT]	terminates the QPager

#### 4.1.5 Events

Every time a value pointed to by the DDE object changes an event will be triggered. This event is sent to all objects in the QPager whose expressions contain the property value, which has changed.

**Note: Be aware of this behavior, because it is possible to chain properties and their events to each other, which can cause an infinite loop.**

#### 4.1.6 Design limitations

##### Broken connections

The DDE object assumes stable connections; it will not reestablish a broken connection automatically.

We have noticed a number of bugs that arise occasionally within DDEML when shutting down the server end of a DDEML conversation. These tend to occur when there are a large number data values being sent over very short time intervals via hot links to a client. The problem only seems to manifest itself when the client initiates the hot links using the 'acknowledge' flag. In such situations if the server application is closed then on occasion this can cause a near terminal system crash. Serious as though this sounds, it is probably not good DDEML practice to have the server application terminate by itself. The client applications should be in control of when the conversations are to be terminated.

#### NT Limitations on NetDDE share usage



The NT environment adds a new overt layer of security to the definition and use of NetDDE shares. NT includes its own version of the 'DDE Share Manager' utility and it is recommended that this tool be used in all situations.

When a WIN95 client tries to connect to an NT based server using NetDDE then the process will open a login form on the WIN95 machine, even if the WIN95 user is already logged in. Microsoft has acknowledged this as a bug. We have not been able to link two NT machines from different domains using NetDDE.

### **Multi-threading**

The Q-Pager has been organized so that a DDE-Object component can in theory be created and run within a thread.

Microsoft are extremely vague about the practicalities of running a non re-entrant system like DDEML inside a thread.

Our own experiments would suggest that DDEML is NOT thread safe, and certainly any attempt to run multiple DDE sessions in separate threads just doesn't seem to work.

Therefore the QPager can contain only one DDE object because one application can contain only one DDE server and one DDE client.

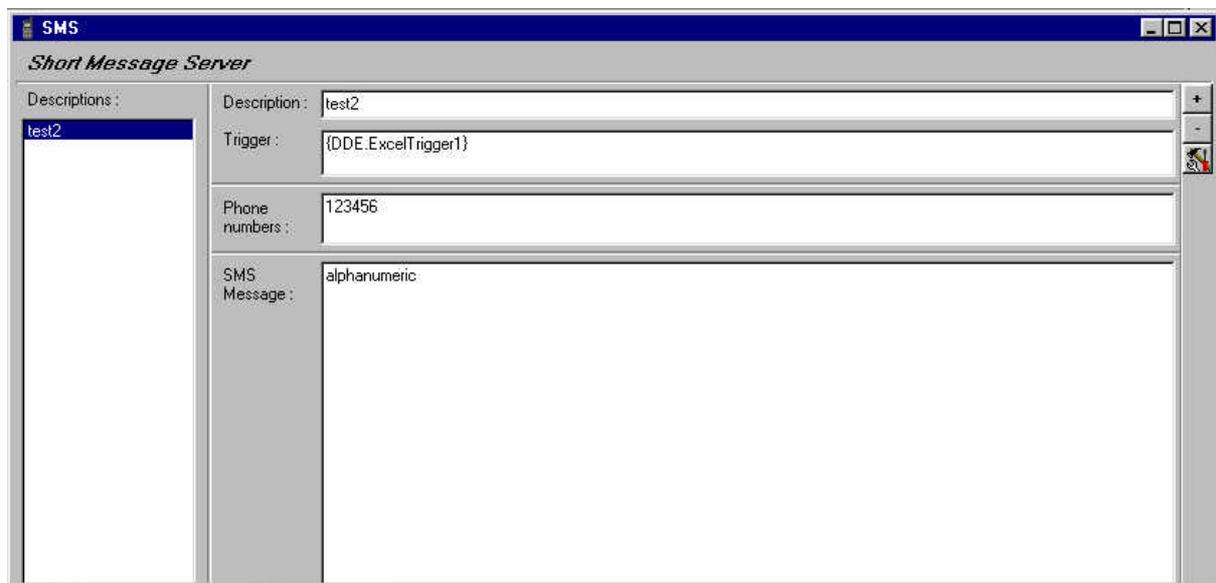
## 4.2 SMS

The SMS object is a QPager object that offers SMS functionality.

Within this single object you get all of the following features:

- UCP protocol
- Configurable message length
- Tone only , numeric, alphanumeric
- TAPI support

### 4.2.1 User interface



#### Descriptions

List of descriptions. Select a description from the list to switch to the newly selected description.

#### Description

A description describes a set of data, which controls the behavior of the SMS message.

#### Trigger

The trigger field must contain an expression, which controls the start of the SMS message. The SMS message will be send if the result of the expression contains the value '1'.

#### Phone numbers

This is the phone number(s) where the SMS message is send to. This input field can contain a straight phone number(s) (; separated) or an expression.

## SMS Message

This field contains the SMS message to send. The SMS message may contain several expressions.



adds a data set and a empty description to the list



deletes a data set



displays the configuration window

### 4.2.2 Configuration



#### Service Name

Contains the SMS service provider name.

#### Phone Number

Contains the phone number of the SMS service provider.

#### Originator

Contains the originator phone number

#### Message length

Defines the maximum message length your SMS service provider can send in one message.

#### Breakup large messages

If this option is checked a SMS message will be broken into 2 or more message if the SMS message length is longer then the message length. Some SMS service provider does this length check for you.

#### Max. Messages

Defines the maximum of messages your service provider can accept in one call.



---

**Device**

Points to the device, which is used to send SMS messages.

**Apply**

Saves the current configuration

**Cancel**

Restores the configuration to the last saved one.

**4.2.3 Design limitations**

- If a SMS message is send, the program user interface will halt for one or two seconds. This is caused by the windows TAPI driver.
- Only one SMS at once can be send.
- If a SMS message contains errors, or your SMS service central returns with an error, the message will be removed from the queue.

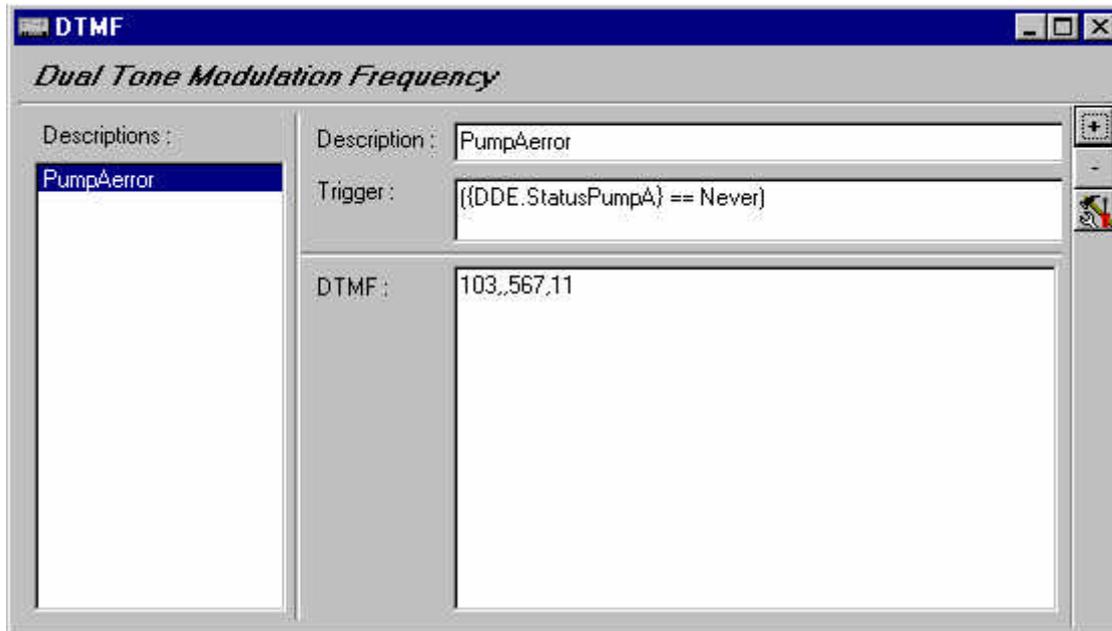
### 4.3 DTMF

The DTMF object is a QPager object that offers DTMF functionality.

Within this single object you get all of the following features:

- DTMF
- TAPI support

#### 4.3.1 User interface



#### Descriptions

List of descriptions. Select a description from the list to switch to the newly selected description.

#### Description

A description describes a set of data, which controls the behavior of the DTMF object

#### Trigger

The trigger field must contain an expression which controls the start of the DTMF object. The DTMF tones will be sent if the result of the expression contains the value '1'.

#### DTMF

This field points to the DTMF sequence to send.



adds a data set and a empty description to the list

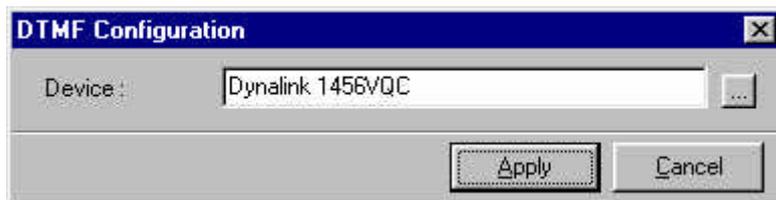


deletes a data set



displays the configuration window

### 4.3.2 Configuration

**Device**

Points to the device, which is used to send DTMF sequences.

**Apply**

Saves the current configuration

**Cancel**

Restores the configuration to the last saved one.

### 4.3.3 Design limitations

- If a DTMF sequence is send, the program user interface will halt for one or two seconds. This is caused by the windows TAPI driver.
- Only one DTMF sequence at once can be send.

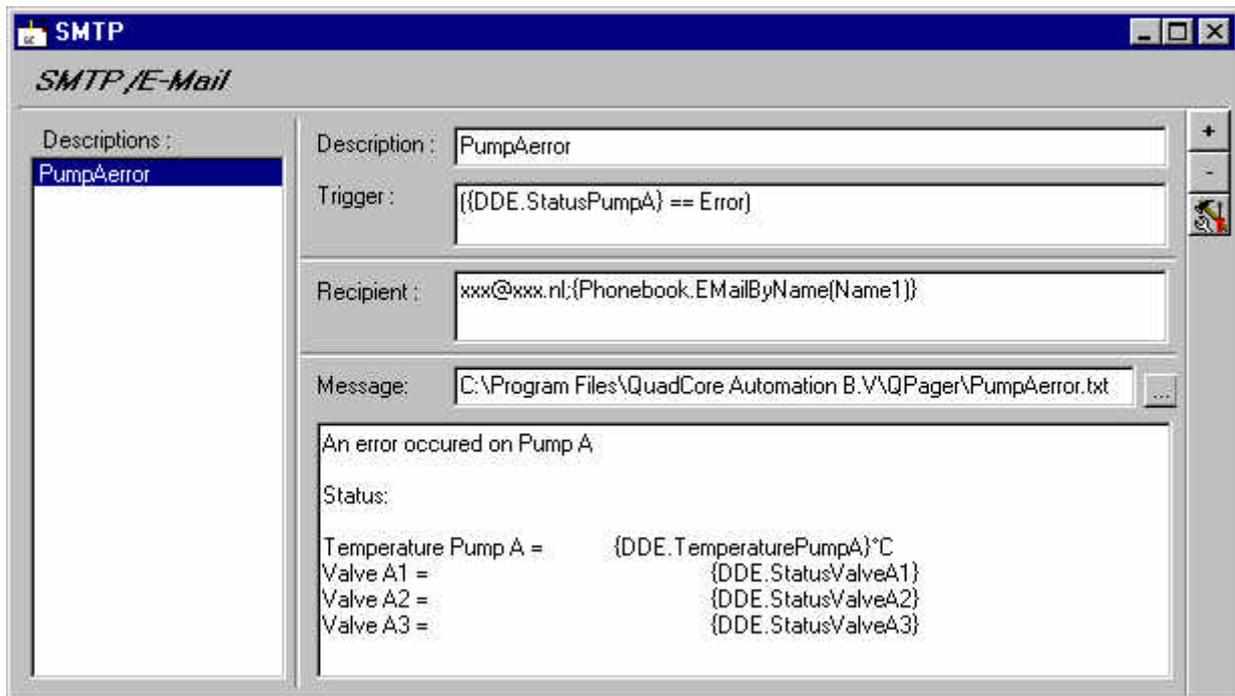
## 4.4 SMTP

The SMTP object is a QPager object that enables the sending of e-mail via an Internet mail server as specified in RFC 821.

Within this single object you get all of the following features:

- Sending text based e-mail messages

### 4.4.1 User interface



#### Descriptions

List of descriptions. Select a description from the list to switch to the newly selected description.

#### Description

A description describes a set of data, which controls the behavior of the E-mail message

#### Trigger

The trigger field must contain an expression, which controls the start of the E-mail message. The E-mail will be send if the result of the expression contains the value '1'.

**Recipient**

This is the E-mail address from the recipient where the E-mail is send to. This input field can contain a straight phone number(s) (; separated) or an expression.

**Message**

This field points to the file to send. The file may contain several expressions.



adds a data set and a empty description to the list



deletes a data set



displays the configuration window

**4.4.2 Configuration**

The image shows the 'SMTP' dialog box with the 'Local information' tab selected. The fields are: Host: smtp.nl.net, Port: 25, Timeout: 0, and User ID: quadcore. There are 'Apply' and 'Cancel' buttons at the bottom.

**Host**

Contains the host name of your E-mail host.

**Port**

Specifies the port number.

**Timeout**

The amount of seconds the SMTP object waits for the server to reply. If this value is set to zero the SMTP object does not return with an error and waits for an infinite amount of seconds.

**User ID**

Contains your user ID

The image shows the 'SMTP' dialog box with the 'Winsock' tab selected. The fields are: From name: QPager, From address: info@quadcore.nl, and Reply address: info@quadcore.nl. There are 'Apply' and 'Cancel' buttons at the bottom.



**From name**

Contains the sender's name.

**From address**

Contains your own E-mail address.

**Reply address**

Contains your reply E-mail address.

**Apply**

Saves the current configuration

**Cancel**

Restores the configuration to the last saved one.

**4.4.3 Design limitations**

To send e-mail over the Internet an open network connection is required. The SMTP object does not open a dial-up connection by it self.

The SMTP object requires a TCP/IP stack, WSOCK32.DLL, which is available through various vendors, including Microsoft.

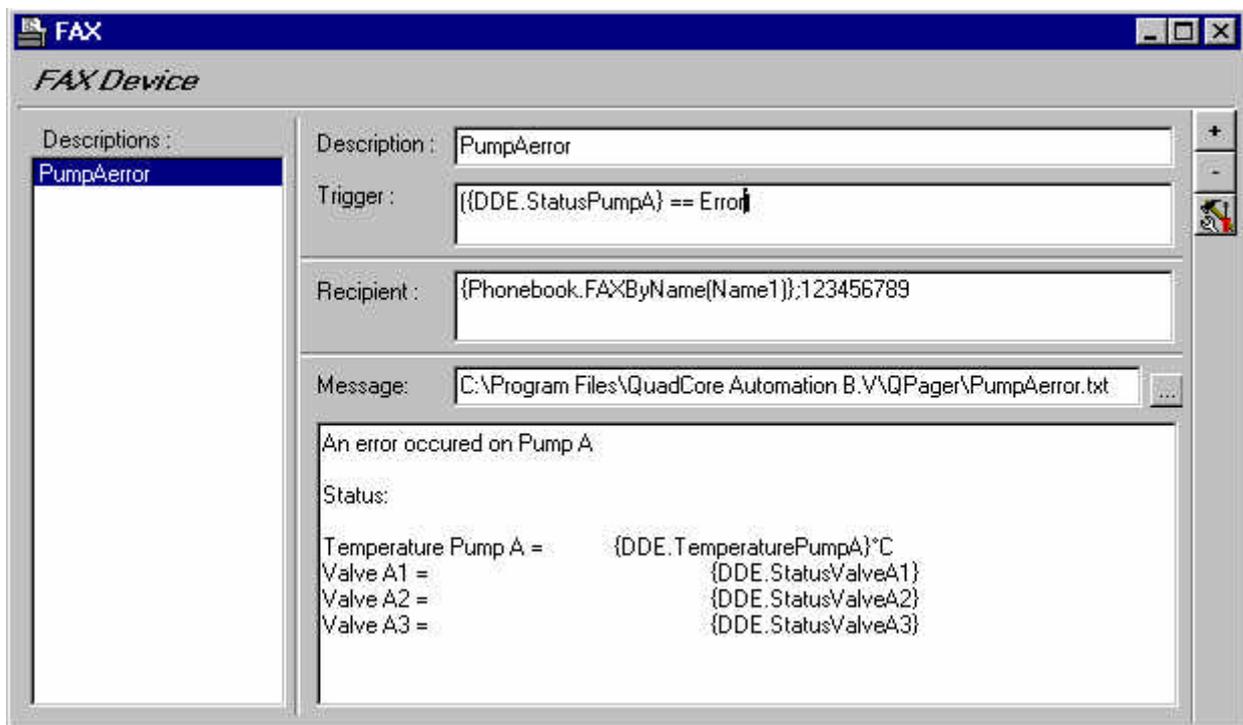
## 4.5 FAX

The FAX object is a QPager object that offers FAX functionality.

Within this single object you get all of the following features:

- Normal resolution text faxes
- Fixed station ID
- Fixed sender
- Class 1, Class 2, or Class 2.0
- TAPI support

### 4.5.1 User interface



#### Descriptions

List of descriptions. Select a description from the list to switch to the newly selected description.

#### Description

A description describes a set of data, which controls the behavior of the FAX

#### Trigger

The trigger field must contain an expression, which controls the start of the FAX. The FAX will be send if the result of the expression contains the value '1'.

#### Recipient

This is the phone number where the FAX is send to. This input field can contain a straight phone number(s) (; separated) or an expression.

## Message

This field points to the file to send. The file may contain several expressions.



adds a data set and a empty description to the list

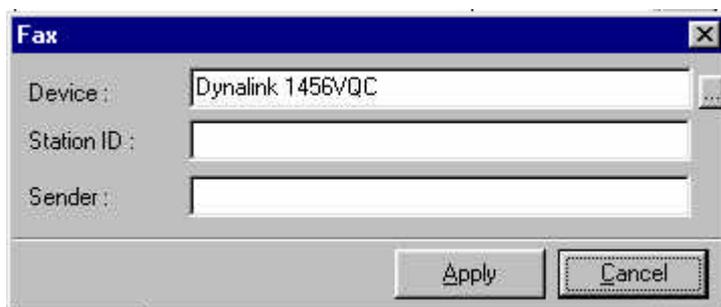


deletes a data set



displays the configuration window

## 4.5.2 Configuration



### Device

Points to the device, which is used to send faxes.

### Station ID

String indicating your station ID.

### Sender

String indicates the sender.

### Apply

Saves the current configuration

### Cancel

Restores the configuration to the last saved one.

## 4.5.3 Design limitations

- If a FAX is send, the program user interface will halt for one or two seconds. This is caused by the windows TAPI driver.
- Only one FAX at once can be send.
- If the FAX object is unable to send a FAX, it will try to send other faxes in the queue first.

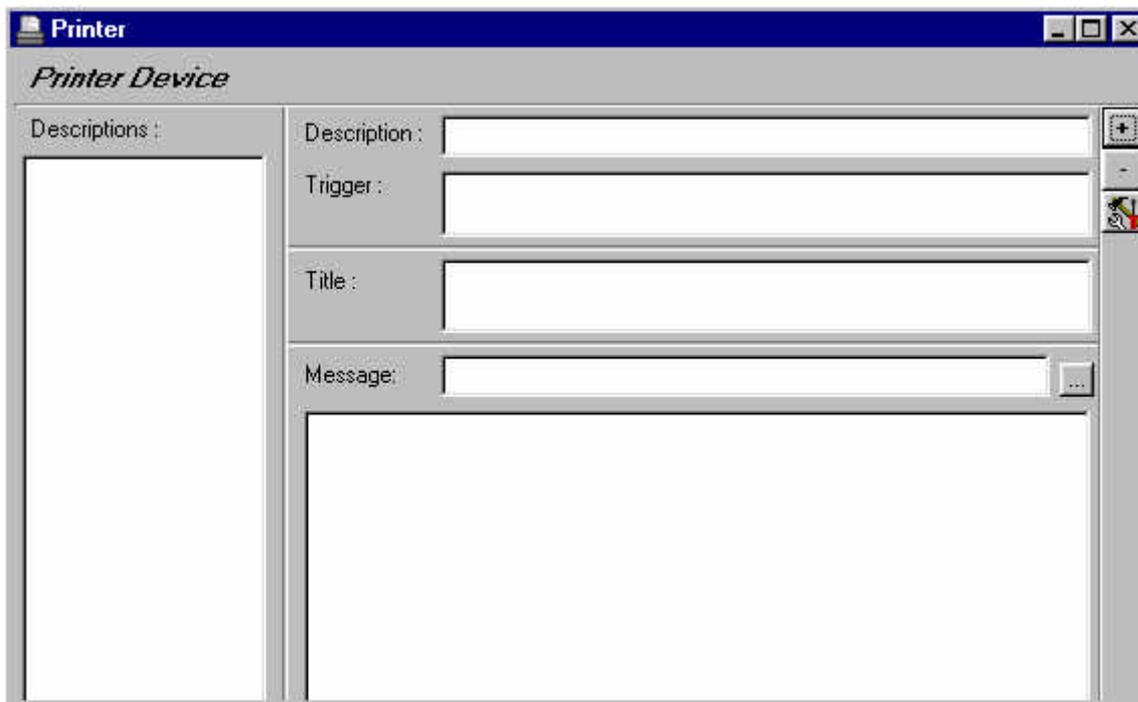
## 4.6 Printer

The printer object is a QPager object that enables the sending text to a printer.

Within this single object you get all of the following features:

- Sending text based messages to a printer.

### 4.6.1 User interface



#### Descriptions

List of descriptions. Select a description from the list to switch to the newly selected description.

#### Description

A description describes a set of data, which controls the behavior of the E-mail message

#### Trigger

The trigger field must contain an expression, which controls the start of the E-mail message. The E-mail will be send if the result of the expression contains the value '1'.

#### Title

Contains the title of the print.

## Message

This field points to the file to send. The file may contain several expressions.



adds a data set and a empty description to the list



deletes a data set



displays the configuration window

### 4.6.2 Configuration



#### Device

Contains the printer device where all prints are send to.

#### Settings

Configures the currently selected device.

#### Apply

Saves the current configuration

#### Cancel

Restores the configuration to the last saved one.

### 4.6.3 Design limitations

- Only ASCII text will be send.

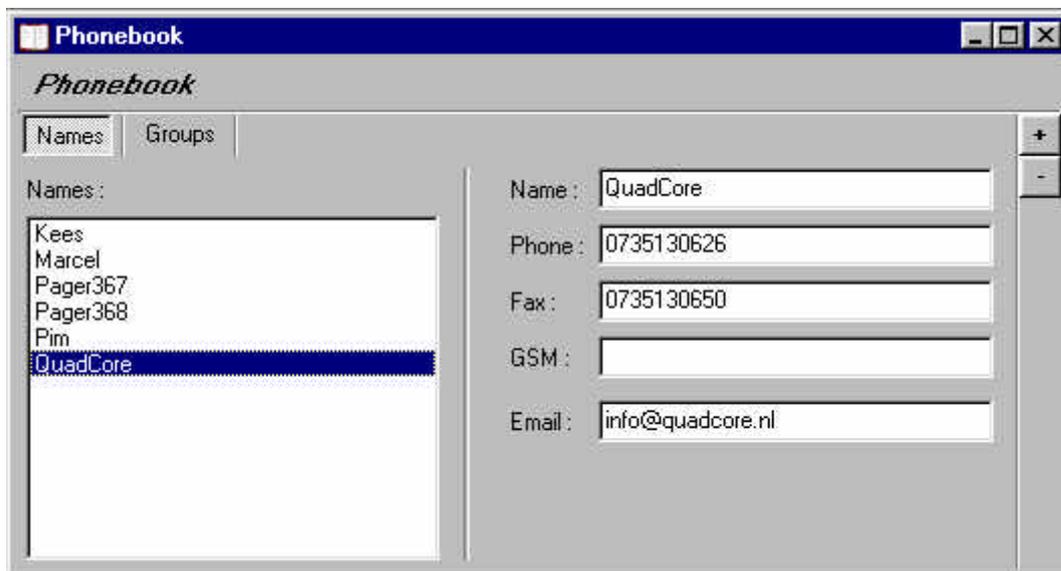
## 4.7 Phonebook

The phonebook object is a QPager object that enables the store and manages phone-numbers and e-mail addresses/

Within this single object you get all of the following features:

- Unlimited storage
- Phone-number
- Fax-number
- GSM-number
- e-mail address
- User groups

### 4.7.1 User interface



#### **Names**

List of available names. Select a name from the list to view or edit its data.

#### **Name**

A user name.

#### **Phone**

Phone number

#### **FAX**

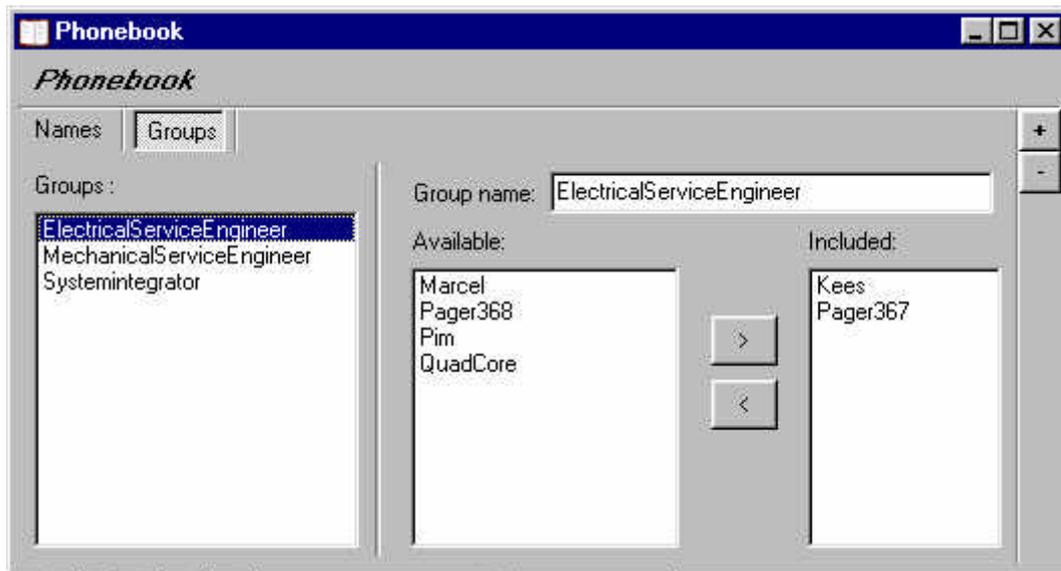
FAX number

#### **GSM**

GSM number

#### **Email**

Email address



### Groups

List of available groups. Select a group from the list to view or edit its data.

### Group

A user name.

### Available

List of available names

### Included

List of names included into the group



adds a selected name to the group



removes a selected name from the group



adds a data set and a empty description to the list



deletes a data set

## 4.7.2 Functions

The phonebook object offers a set of functions to use it in expressions

PhoneByName( <i>name</i> )	returns the phone-number from the person with the name: <i>name</i>
FAXByName( <i>name</i> )	returns the FAX-number from the person with the name: <i>name</i>
GSMByName( <i>name</i> )	returns the GSM-number from the person with the name: <i>name</i>
EMailByName( <i>name</i> )	returns the E-mail address from the person with the name: <i>name</i>
PhoneByGroup( <i>group</i> )	returns a set of phone-numbers from the group: <i>group</i>
FAXByGroup( <i>group</i> )	returns a set of FAX-numbers from the group: <i>group</i>
GSMByGroup( <i>group</i> )	returns a set of GSM-numbers from the group: <i>group</i>
EMailByGroup( <i>group</i> )	returns a set of E-mail-numbers from the group: <i>group</i>

for example the expression:

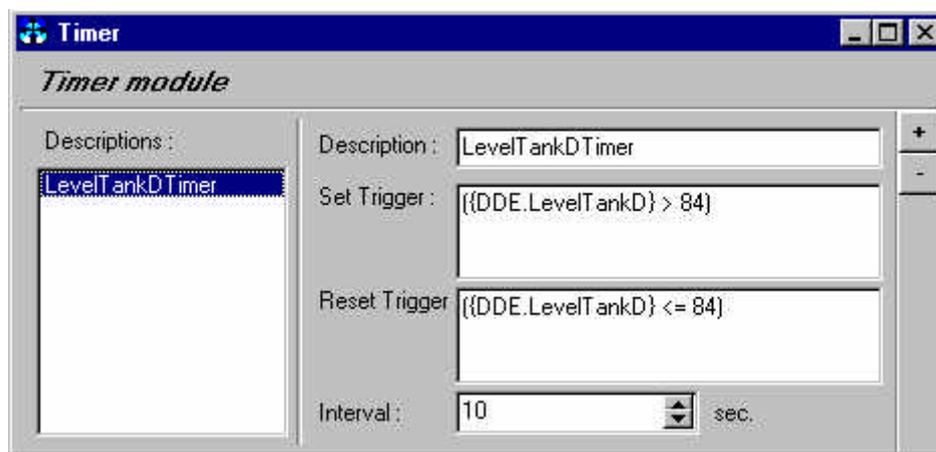
```
{PhoneBook.FAXByName(QuadCore)}
```

returns : 0735130626

## 4.8 Timer

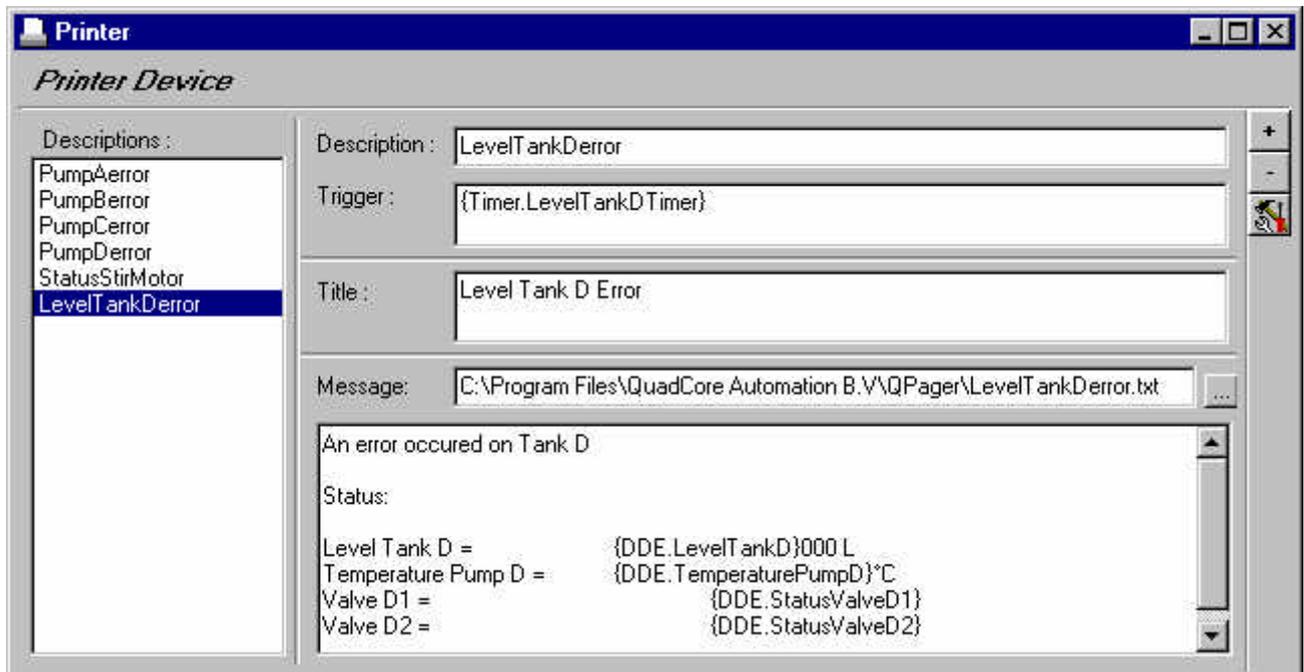
The Timer object is a QPager object where you can start a timer by a DDE trigger. This Timer then can be used in a message trigger.

For example:



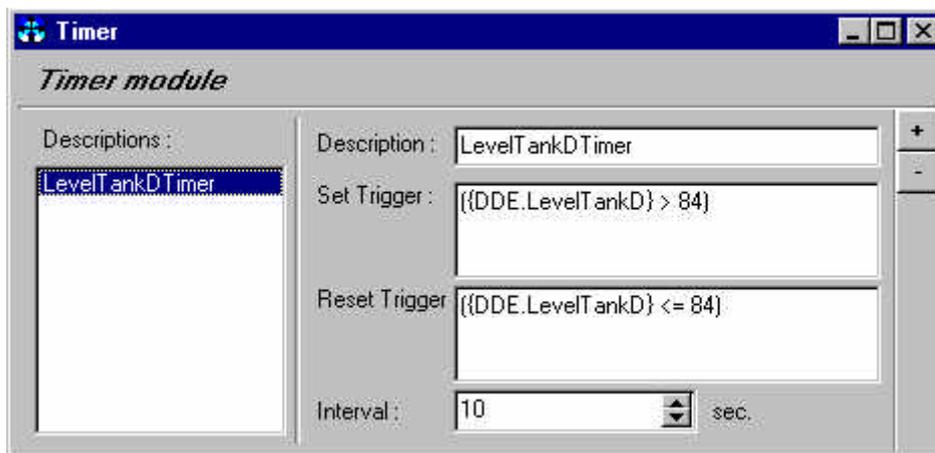
The Timer “LevelTankDTimer” will be started if the level of tank D is greater or equal to 84. The Timer will become true after 10 seconds.

Now the Timer can be used in another object, which generates a message.  
For example: The Printer object:



If the Timer has become true (after 10 seconds), the message will be sent to the printer.

#### 4.8.1 User interface



##### Descriptions

List of descriptions. Select a description from the list to switch to the newly selected description.

##### Description

A description describes a set of data, which controls the behavior of the Timer.

##### Set Trigger

The set trigger field must contain an expression, which sets the Timer.



### **Reset Trigger**

The reset trigger field must contain an expression, which resets the Timer.

### **Interval**

This field controls how long it takes for the Timer to become true.

## **4.9 Export and Import Data**

The data of the objects is stored in the registry of windows. To export or import data, the program 'Regedit' (which comes with Windows) must be used. The program can be found in "C:\Windows", for Windows95 or 98 and in "C:\Winnt" for Windows NT.

### **4.9.1 Export Data**

To Export the QPager data, start the program REGEDIT. Now open the folder "HKEY\_LOCAL\_MACHINE", then open the folder "SOFTWARE", then open the folder "QuadCore Automation B.V.", then open the folder "QPager". Now go to the menu-item "Registry" and click "Export Registry File", the file which is being generated is the backup of all the QPager Data.

If you only want to backup a part of the data, then you must open the corresponding folder, for example the "PrinterObject" folder followed by the same procedure as above.

### **4.9.2 Import Data**

To import data in QPager, you only have to Double Click the ".reg" file that is generated by the Export procedure.

**Warning:** If you import data by this way, you will lose your previous data !!!!



---

## 5. Problem Report / Change Request

Although this software product has been developed and tested with great care, QuadCore Automation B.V. does not exclude the possibility of undocumented errors.

If you happen to come across any such problem, or if you like to request a functional change, we would appreciate it very much if you would kindly fill out the form on the next pages and mail it to QuadCore Automation B.V..



---

**PROBLEM REPORT / CHANGE REQUEST**

**QuadLine QPager**

Client specifications:

Report made by : \_\_\_\_\_  
Date : \_\_\_\_\_  
Company : \_\_\_\_\_  
Address : \_\_\_\_\_  
Zip code : \_\_\_\_\_  
Postbox : \_\_\_\_\_  
Country : \_\_\_\_\_  
Phone : \_\_\_\_\_  
Telefax : \_\_\_\_\_

QuadLine product specifications

Version number : \_\_\_\_\_  
Serial number : \_\_\_\_\_  
Installation sequence : \_\_\_\_\_

Environment specifications

Hardware model : \_\_\_\_\_  
Operating system & version : \_\_\_\_\_



Description

:

---

---

---

---

---

---

---

---

Possible cause

:

---

---

---

---

---

---

---

---

Possible solutions

:

---

---

---

---

---

---

---

---

Other details

:

---

---

---

---

---

---

---

---