A MITEL
PRODUCT
GUIDE

Unify OpenScape Fault Management

Unify OpenScape Fault Management V12, Mobile Alarm Reaction

User Guide
10/2021

A31003-F30C0-U109-01-7619

Mitel

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Europe Limited. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos, and graphics (collectively “Trademarks”) appearing on Mitel’s Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively “Mitel), Unify Software and Solutions GmbH & Co. KG or its affiliates (collectively “Unify”) or others. Use of the Trademarks is prohibited without the express consent from Mitel and/or Unify. Please contact our legal department at iplegal@mitel.com for additional information. For a list of the worldwide Mitel and Unify registered trademarks, please refer to the website: http://www.mitel.com/trademarks.

Contents

1 Preface

This chapter discusses the following aspects:

- purpose and audience of this guide
- terminology
- organization of this guide.
- conventions used in this manual.

1.1 Purpose

This User Guide contains a description of the **Mobile Alarm Reaction (MAR)**, a software program that runs as an add-on module – a so called “plugin” – for the OpenScape FM.

1.2 Audience

This guide is addressed to users who want to learn how to use MAR for the OpenScape FM. You should be familiar with the use of OpenScape FM. For a detailed introduction please read the *OpenScape FM User Guide*.

1.3 Terminology

- **OpenScape FM** means OpenScape Fault Management.
- **Server** means the OpenScape FM Server, i.e. the server where OpenScape FM Desktop with the MAR Plugin has been installed.
- **Client** means the OpenScape FM Client, usually a web browser where OpenScape FM has been started.
- **Desktop** means the OpenScape FM Desktop.
- **HiPath 4000** stands for HiPath 4000 or OpenScape 4000.
- **HiPath 3000** stands for HiPath 3000 or OpenScape Business.

1.4 Organization of this Guide

This guide is organized as follows:

- *Chapter 2, “Introduction”* provides a description of the concepts used by the OpenScape FM.

- *Chapter 3, "Getting Started"* tells you what you have to do before you can install MAR and how to install it as Plugin for OpenScape FM.
- *Chapter 4, "Using MAR"* describes the User Interface and shows you how to configure reactions.
- *Chapter 5, "Example"* contains an example definition of reactions, recipients and time schedules and shows how they could be connected to each other and to events.
- *Appendix A, "Documentation of MAR Rights"* provides a list of all rights which can be assigned within the MAR environment.
- *Appendix B, "Hardware and Software Requirements"* provides a list of hardware and software you should have to run the MAR Plugin.
- *Appendix C, "Macros for Contents Messages"* provides information about the macros which contain the information of the trap variables of events and which can be used in contents messages.
- *Appendix D, "Character Sets"* provides the character sets, which are supported for the generated Messages.

1.5 Conventions Used in this Manual

The following font conventions are used in this document:

Bold Font: Indicates that a word is a new or important term.

Example: **Proxy Agent**.

Bold Computer Font: Indicates data to be entered by the user.

Example: **`java`**.

Computer Font: Indicates computer output, including UNIX prompts, an explicit directory or a file name.

Example: `prompt%.`

Italics: Indicates a reference to another manual or to a different section within the current manual.

Example: *HDMS documentation.*

Italic type is also used for emphasis.

Example: *All* users will be affected.

2 Introduction

The OpenScape FM, a client/server application, is a network management program for integrated monitoring and management of IP and PBX networks.

The OpenScape FM server receives several information from all its managed devices such as HiPath 4000 machines, HiPath 3000 machines or other IP nodes. The system administrator, can watch the status of all HiPath systems and other network devices via the OpenScape FM GUI (graphical user interface). When a new alarm occurs, the corresponding icon changes its color and starts blinking. Besides, every event which is registered as an SNMP trap by the OpenScape FM server is shown in the OpenScape FM Event Browser. The Event Browser GUI can be used to keep track of all new alarms.

Now, what do you do when a new alarm comes in and you are not sitting at your console, watching your network environment? Of course, you demand to know what is going on in your network and you want to be able to react as quickly as possible to keep all systems up and running. Enter the Mobile Alarm Reaction, MAR.

With this add-on program, one of several so called OpenScape FM Plugins, you will always be up-to-date about the status of your devices to-be-watched. You can define how you would like to be informed when a new alarm or event occurs – choose, for example, if you want to receive an e-mail or an SMS message.

You can also decide for which types of alarms you want to be contacted. Say for example, you do not think that it is necessary to know everything about all alarms, but you do want to know immediately when a HiPath 4000 target alarm has occurred. MAR can manage this for you. In the following chapter, we will give you a detailed introduction how to install and work with the Mobile Alarm Reaction.

3 Getting Started

3.1 Installation

The installation of the MAR will be performed automatically during the installation of the OpenScape FM.

To define reactions for objects of a specific technology (e.g HiPath 4000), the plugin for the respective technology has to be initialized.

To initialize the MAR Plugin, an OpenScape FM Client must be started and the menu item **Server->Plugins->Init MAR Plugin** from the main menu bar has to be selected. After the initialization the menu item **Init MAR Plugin** will be removed from the menu **Server->Plugins**. A new menu item **MAR** will appear in the main menu bar. Additionally an object representing the **MAR** Plugin is added to the hierarchy with the path **Root->System->Plugins**. The new object offers the same menu items like the **MAR** main menu.

In the next chapter, we will cover how to add new reactions and how to work with MAR.

The MAR item in the main menu is your starting point for the MAR configuration. It contains the menu item **Configure** that opens a window in which **Reactions**, **Time Schedules** and **Recipients** can be defined that are needed to create MAR reactions. They will be explained in detail in *Chapter 4, "Using MAR"*.

3.2 Licensing

To work with the MAR Plugin a valid license must be available. Please refer to the *OpenScape FM Desktop User Guide* to get information about how to license your OpenScape FM system.

3.3 Required Hardware and Software

The machine on which the OpenScape FM server with the MAR Plugin is running will be configured to send E-Mails and/or SMS. Thus, before all, you have to make sure that the machine is able to do that.

3.3.1 Hardware

In order to send SMS, you need a cellular phone or a GSM modem. Please see *Appendix B, "Hardware and Software Requirements"* for more information.

3.3.2 Software

OpenScape FM is a program written in Java(TM). Thus in order to be able to generate E-Mails and SMS Messages, the corresponding Java(TM) APIs have to be installed on the OpenScape FM server machine. Please read the current OpenScape FM Release Notes to get a detailed explanation of which APIs you need and how to install them. To be aware of which other software you have to install please refer to *Appendix B.2, “Software:”*.

4 Using MAR

4.1 General Concept Overview

The MAR is a tool to create automatic responses in reaction to events received by the OpenScape FM system.

The following can be accomplished by the reaction system:

When a defined event arrives at the management platform for a given HiPath system, in a given time interval and of a given priority, a number of predefined reactions will be fired. These reactions may either be the sending of E-mail notifications or SMS messages, or the invocation of a program or script. The latter insures that virtually any possible reaction can be generated. Since the messages sent or the programs/scripts parameters can even be modified in relation to the event that caused the reaction, the system is as flexible as possible.

To achieve this grade of flexibility, the different configurations of the reactions have to reflect this requirements.

For each reaction it has to be defined for which events the reaction should be triggered. Additionally, it has to be configured during which time intervals the reaction is active, which message should be sent and who the recipients are.

To avoid duplication of data about destinations (called recipients) and time intervals (called time schedules) both are defined separately and can then be connected to defined reactions.

Figure 1 gives an overview how the independently defined recipients, time schedules and reactions are connected to each other.

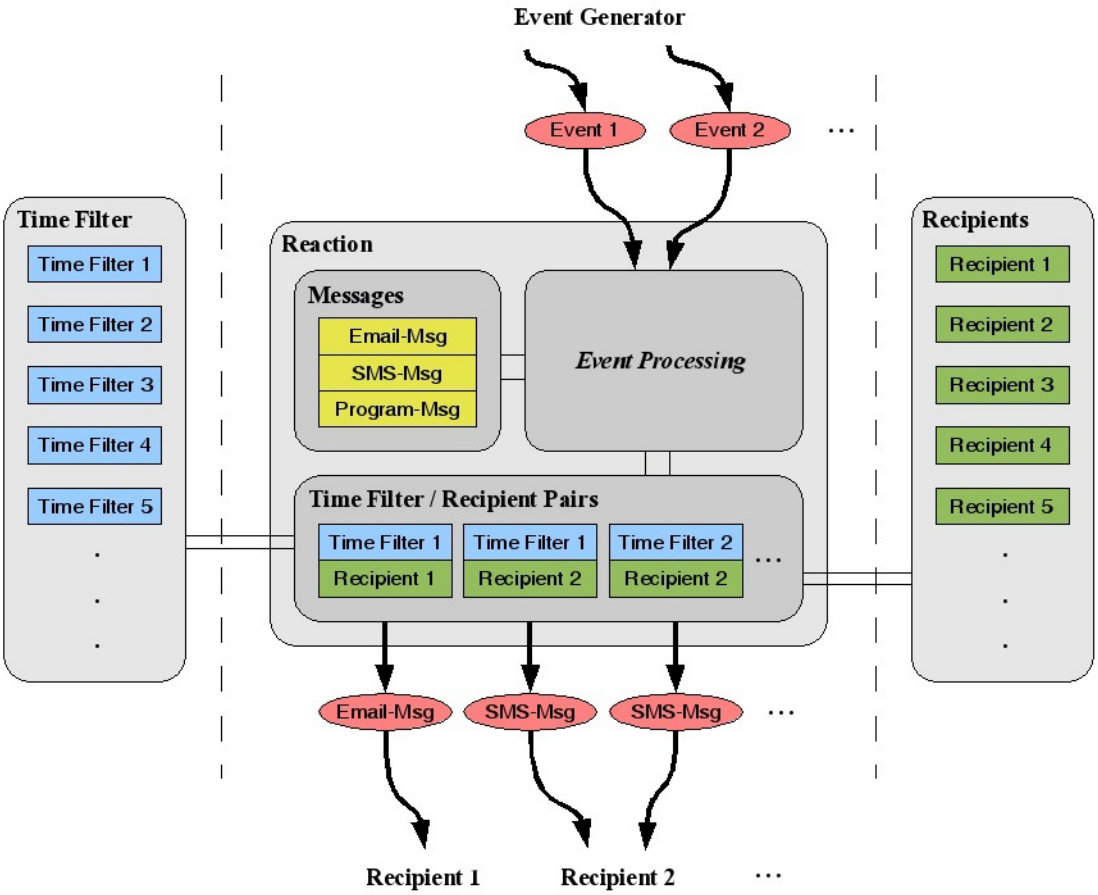

Figure 1 Concept Overview

Each e.g. HiPath 3000 or 4000 **Event** can be assigned to any number of reactions. These **Reactions** will sent **Messages** to **Recipients** but only if the event happened within the time interval defined by connected **Time Schedules**.

For each **reaction** more than one **recipient** can be defined and each recipient is combined with a **time schedule**. In a matter of fact one recipient can be connected more than once to a single reaction as long as it is combined with a different time schedule each time. For example two time schedules can be designed, one for work hours and one for off work hours. When a recipient is added twice, once with each of these time schedules, the recipient will get notified during work and off work hours.

It is also possible to combine a single time schedule to more than one recipient within a single reaction definition. This way different recipients can be notified during the same time interval.

Important Note:
Combinations between time schedules and recipients are only valid within the reaction the time schedule / recipient combinations is defined for. Different reactions may use an identical as well as a completely different set of combinations.

Important Note:
Messages will only be sent within the intervals defined by the time schedules. If no time schedule matches for a given time and given reaction, no message will be sent at all if the reaction is triggered at that time.

Since any combination of E-mail, SMS and program start recipients can be connected to a single reaction, more than one **message** can be defined for a single reaction, one each for the E-mail, SMS and program start recipient type. When a recipient gets notified, the message designed for the recipient's type will be sent.

Finally, a single reaction may be connected to any number of **events**. To distinguish which event was responsible for the sending of a message, **macros** can be included within the massage text. These macros are place holders for event specific data like a HiPath 4000 system name, an alarm class or an alarm time. The macros will be replaced by the data they represent before a message is delivered.

How MAR reactions, messages, recipients and time schedules can be defined and configured and how they are connected to events will be explained in detail in the following chapters:

Section 4.2, "Defining Message Recipients" describes how destinations or recipients can be defined.

Section 4.3, "Defining Time Schedules" shows how time intervals are defined.

Section 4.4, "Defining Reactions" explains how reactions and their messages are created, how macros can be included into messages and how time schedules and recipients are connected to reactions.

Section 4.5, "Connecting Events and Reactions" describes how defined reactions are connected to event types.

Chapter 5, "Example" finally uses an example to illustrate the whole concept.

4.2 Defining Message Recipients

This chapter will explain how **recipients** for reactions can be defined or modified.

In the context of the reaction system a recipient is the receiver of an action triggered by the system. In case of E-Mail or SMS reactions, receivers are defined by their E-Mail-address or phone number. In case of Program-Start reactions the programs that will be started are the receivers.

Definitions or modifications of message recipients can be done in the **Recipients** page. To open this page, the **MAR->Configure...** menu item from the main menu bar has to be selected. Then the respective page has to be opened.

The '**Recipients'** page consists of the following elements with the following functions:

- The upper part of the 'Recipients' window consists of a **table** which contains a list of already defined recipients. Every line in this table represents one recipient. The column **Recipient** shows the names of the defined recipients. The column **Target Type** shows the type of the recipient (E-Mail, SMS or Program Start) and the column **Parameter** shows the parameters defined for the recipient (E-Mail address, phone number, program executable).
- To define a new recipient, the button **New** has to be pressed. To reconfigure an already existing recipient, the **Configure...** context menu item of its list entry has to be selected, or the button **Configure...** has to be pressed while a table entry is selected. In all cases the **'Recipient Configuration'** window (see *Figure 2*) will be opened.

- The **Delete** button can be used to erase all recipients that are selected in the table.

How the different types and parameters can be configured will be shown in the following sections.

4.2.1 The ’Recipient Configuration’ Window

The **’Recipient Configuration’** window (see *Figure 2*) can be used to configure or reconfigure recipients.

Whenever the **New** button is pressed or a **Configure...** menu item is selected in the **’Recipients’** window the ’Recipient Configuration’ window will be opened. This window can be used to manipulate the data of the new or of the chosen recipient.

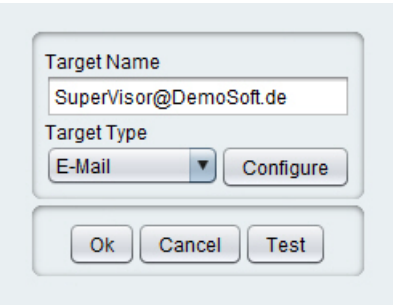

Figure 2 The ’Recipient Configuration’ window

The **’Recipient Configuration’** window consists of the following elements with the following functions:

- The **Target Name** text field contains the current name of the recipient if the ’Recipient Configuration’ window is opened. The text field can also be used to alter the name of the recipient. The name of a recipient should be unique and suggestive, since it will be used to identify the recipient while defining reactions.
- The **Target Type** selector shows the current type of reaction connected to the recipient. The selector can also be used to change the reaction type of the recipient. The three possible selections are ’E-Mail’, ’SMS’ and ’Program Start’, which will set the recipient to the type E-Mail reaction, SMS reaction or Program Start reaction respectively.
- The **Configure...** button will open a parameter configuration window if pressed. The reaction type chosen by the **Target Type** selector determines which parameter configuration window will be opened. How the parameters for the different recipient types can be defined or changed will be shown in the following three chapters (see *Section 4.2.1.1* to *Section 4.2.1.3*).
- The **Ok** button can be pressed to accept the changes made in the **Target Name** text field and **Target Type** selector. In addition the **’Recipient Configuration’** window will be closed.
- The **Cancel** button can also be pressed to close the **’Recipient Configuration’** window. In contrast to the **Ok** button, changes in the **Target Name** text field and **Target Type** selector will be ignored. Changes already confirmed in the parameter configuration windows will not be ignored.
- The **Test** button can be pressed to initiate a test reaction. Depending on the reaction type a test message will be sent to the configured address via E-Mail / SMS or the configured program will be started.

4.2.1.1 E-Mail Recipients

Recipients of the target type **E-Mail** will receive an E-Mail when a connected reaction is fired.

The parameters for E-Mail type recipients can be set or reconfigured within the **'E-Mail Parameter Configuration'** window (see *Figure 3*). This window can be opened for a specific recipient by setting the **Target Type** selector to 'E-Mail' and pressing the **Configure...** button in the **'Recipient Configuration'** window.

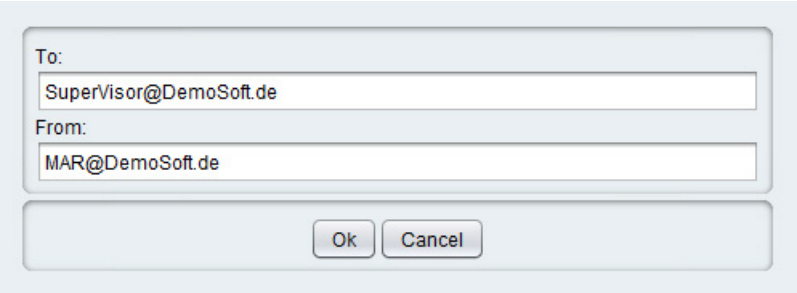

Figure 3 The 'E-Mail Parameter Configuration' window

The **'E-Mail Parameter Configuration'** window is used to define the E-mails recipient and consists of three text fields and two buttons with the following functions:

- The **To** text field is used to configure the E-mail address of the receiver. If a message should be sent to more than one E-mail-address, the addresses can be written into the text field separated by a comma (',').
- The **From** text field is used to configure the E-mail address which will be used to configure the E-mail originator address.

  The mail server which will be used to sent the E-mail messages is defined by the default mail server. This can be configured by using the main menu entry **Server->Administration->Server Properties...** and the page **Mail Configuration**.

  When the window is initially opened all three text fields will contain the actual settings of their associated parameters. In addition the **From** text field and **SMTP Server** text field will contain the last entry made to this field, when a new recipient is created.
- The **Ok** button can be pressed to accept the changes made in the text fields. Their contents will be used as the recipient's E-Mail parameters. In addition the window will be closed.
- The **Cancel** button will also close the window when pressed. Changes to the text field's content will be ignored.

4.2.1.2 SMS Recipients

Recipients of the target type **SMS** will receive an SMS when a connected reaction is fired. These messages have a maximum length of 160 characters. Longer messages will be truncated to this length.

To use this function, a **GSM-phone** or **-modem** must be connected to a **serial interface** of the OpenScape FM server system.

To protect the sending phone against message overflow, outgoing SMS messages are queued by the OpenScape FM server. The sending itself is limited to one SMS per 20 seconds.

Connecting the phone:

The mechanism used by MAR to send SMS via a connected mobile phone or GSM modem relies solely on the facilities provided by the underlying operating system. The supported operating systems are Linux and Windows. The requirements for each OS to use the SMS sending feature are described in the following sections.

Connecting using Linux:

On Linux, mobile phones or GSM modems connected via USB can be used to send SMS. Requirement is that the device and the Linux kernel support the **USB Control Device Class Abstract Control Model** - **CDC ACM**. On the Linux-side, this support is provided by the kernel module `cdc_acm`. Typically, this module is automatically initialized if the device is connected to a USB port. If the device was recognized successfully, the module creates a device file like `/dev/ttyACM0`. This device file is used to access the mobile phone or GSM modem and must be configured in the MAR user interface.

If the mobile phone was recognized, the system logfile `/var/log/messages` should contain output like *Figure 4*:

```
Dec 23 10:23:12 onyxia kernel: usb 3-4.3: new full speed USB device using
ehci_hcd and address 6
Dec 23 10:23:12 onyxia kernel: usb 3-4.3: configuration #1 chosen from 1
choice
Dec 23 10:23:12 onyxia kernel: usb 3-4.3: New USB device found,
idVendor=0421, idProduct=04f9
Dec 23 10:23:12 onyxia kernel: usb 3-4.3: New USB device strings: Mfr=1,
Product=2, SerialNumber=0
Dec 23 10:23:12 onyxia kernel: usb 3-4.3: Product: Nokia 6300
Dec 23 10:23:12 onyxia kernel: usb 3-4.3: Manufacturer: Nokia
Dec 23 10:23:13 onyxia kernel: drivers/usb/class/cdc-acm.c: Ignoring extra
header, type -3, length 4
Dec 23 10:23:13 onyxia kernel: cdc_acm 3-4.3:1.1: ttyACM0: USB ACM device
Dec 23 10:23:13 onyxia kernel: usbcore: registered new interface driver
cdc_acm
Dec 23 10:23:13 onyxia kernel: drivers/usb/class/cdc-acm.c: v0.25:USB
Abstract Control Model driver for USB modems and ISDN adapters
```

Figure 4 Connecting to Linux - Logfile

After plugging the phone into a USB port, the device file `/dev/ACM0` should exist and the output of `lsmod` should contain a line like:

```
usbcore                104672  5 cdc_acm,usbserial,uhci_hcd
```

The module `cdc_acm` is included in the standard kernel of SLES9, SLES10 and OpenSuse 11.0 and was tested successfully with a Nokia 6300.

Connecting using Windows:

On Windows systems, a COM-port is used by MAR to communicate with the mobile phone or GSM device. How the COM-port is integrated into the operating system is vendor-specific. Many mobile phone vendors offer software packages to integrate their devices into windows.

After successful installation of the mobile phone software, the phone should appear as a modem in the windows device manager (see *Figure 5*):

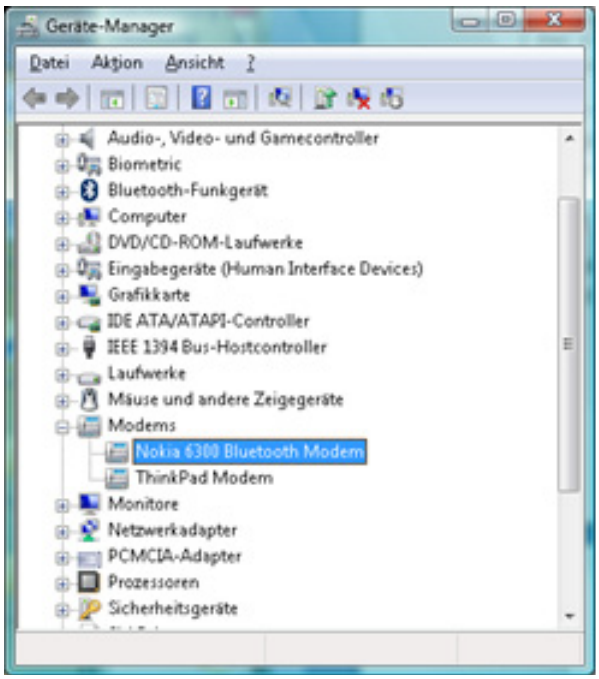

Figure 5 Connecting to Windows - Device Manager

The assigned COM port can be displayed by selecting **Properties** from the context menu in the device manager. This COM port (COM7 in this example - see *Figure 6*) has to be configured for the SMS reaction in the MAR.

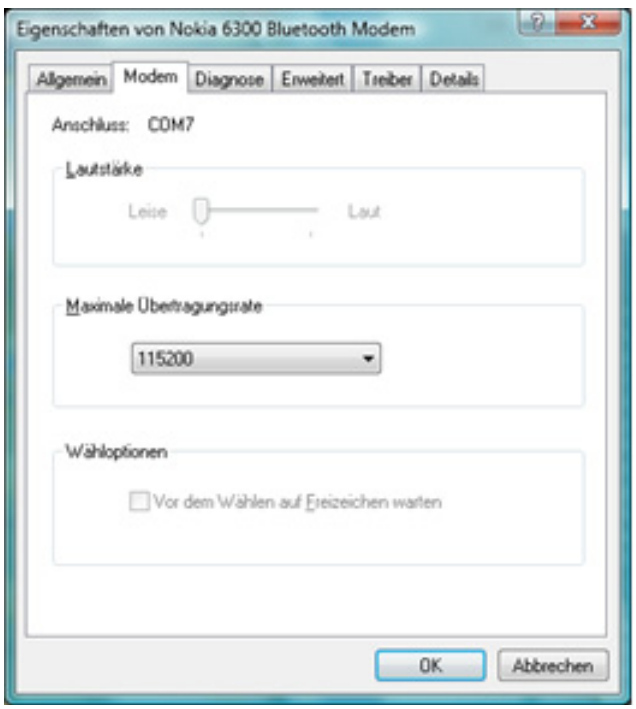

Figure 6 Connecting to Windows - Modem Properties

Configuration within the OpenScape FM

When a phone is connected, the parameters for SMS type recipients can be set or reconfigured within the **'SMS Parameter Configuration'** window (see *Figure 7* and *Figure 8*). This window can be opened for a specific recipient by setting the **Target Type** selector to 'SMS' and pressing the **Configure...** button in the **'Recipient Configuration'** window.

The **'SMS Parameter Configuration'** window is used to define the receiver of the SMS messages and consists of a number of elements divided into a **Basic** and an **Advanced** page with the following functions:

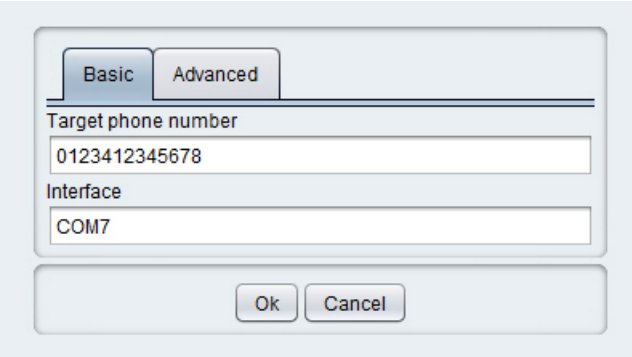

Figure 7 The 'SMS Parameter Configuration' window (Basic)

- **Basic** page:
  - The **Target phone number** text field is used to configure the phone number of the receiver to whom the SMS messages will be sent. The phone number has to be entered in the very same format as it would be when typed using the phone manually.
  - The **Interface** selector is used to specify the serial interface to which the GSM phone or modem, the messages should be send with, is connected (e.g. COM1 on a windows PC).

    Important Note:
    Data will only be send to single digit COM ports (1-9).

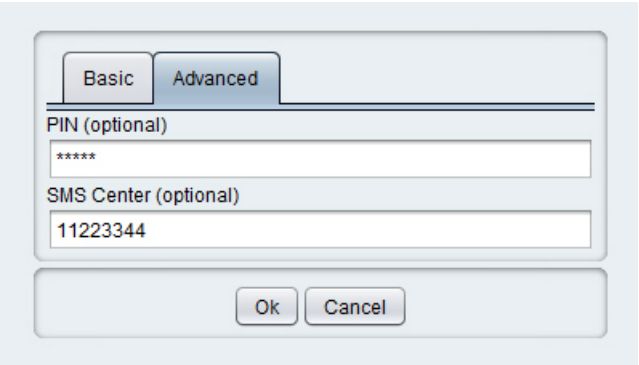

Figure 8 The 'SMS Parameter Configuration' window (Advanced)

- **Advanced** page:
  - The **PIN** text field is used to configure the PIN of the phone which will be used to send the messages. The entry to this text field is optional. The PIN will only be used, when the phone is not already checked into the network.
  - The **SMS Center** text field is used to configure the service provider, which will be used to sent the SMS messages. The entry to this text field is also optional.

  When the window is initially opened all three text fields and the selector will contain the current settings of the associated parameters. In addition the **PIN** and **SMS Center** text field will contain the last entry made to this field, when a new recipient is created.
- The **Ok** button can be pressed to accept the changes made in the text fields. Their contents will be used as the recipient's SMS parameters. In addition the window will be closed.
- The **Cancel** button will also close the window when pressed. Changes to the content of the text fields will be ignored.

4.2.1.3 Program Start Recipients

Recipients of the target type **Start Program** will start a program on the OpenScape FM server system when a connected reaction will be fired.

The parameters for **Start Program** type recipients can be set or reconfigured within the **'Start Program Parameter Configuration'** window (see *Figure 9*). This window can be opened for a specific recipient by setting the **Target Type** selector to 'Program start' and pressing the **Configure...** button in the **'Recipient Configuration'** window.

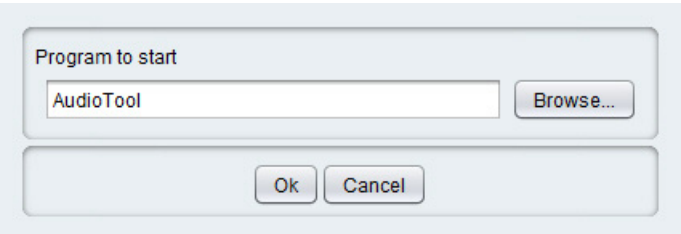

Figure 9 The 'Start Program Parameter Configuration' window

The **'Start Program Parameter Configuration'** window is used to specify the program that will be started and consists of the following elements:

- The **Choose program to start** text field is used to configure the program that will be started when a reaction connected to the recipient is triggered. When the window is initially opened the text field will contain the currently configured program. When a new recipient is created, it contains the last entry made to this field.
- The **Browse...** button will open a standard file chooser window on the server system when pressed. When a program is selected within this window the selected program name and its path are copied to the **Choose program to start** text field.

- The **Ok** button can be pressed to accept the changes made in the text field. Its content will be used as the program to be started. In addition the window will be closed.
- The **Cancel** button will also close the window when pressed. Changes to the text field's content will be ignored.

Note:
Arguments for the started program are configured in the **'Reaction Message Contents'** window (see *Section 4.4.1, "Defining Contents Messages for Reactions"*)

4.3 Defining Time Schedules

The recipients described in the last chapter are used to define who will receive a message when a reaction is triggered. **Time schedules** are used to define the time intervals when reactions shall be fired.

Definitions or modifications of time schedules can be done in the **'Time Schedule'** page. To open this page the **MAR->Configure...** menu item has to be selected from the main menu bar. Then the respective page has to be selected.

The definition of time schedules is described in the OpenScape FM Desktop User Guide.

4.4 Defining Reactions

This chapter will explain how **reactions** can be defined or modified.

In the context of the MAR system a reaction defines what will be done when an event assigned to a reaction occurs. A reaction defines what messages are sent, who the recipients of the messages are and when these messages are sent.

Definitions or modifications of reactions can be done in the **'Reactions'** page. To open this page the **MAR->Configure...** menu item in the main menu bar has to be selected. Then the respective page has to be opened.

The **'Reactions'** page consists of the following elements with the following functions:

- The upper part of the 'Reactions...' page consists of a **table** which contains a list of already defined reactions. Every row in this table represents one reaction.
  The column **Name** shows the name of the defined reaction. The columns **Executions** and **Failed Executions** shows how often messages were delivered to recipients and how often this failed. The column **Last Event** contains the time when the reaction was fired last. The columns **Time Schedule** and **Recipient** contain the information how many different time schedules and recipients are connected to the reaction.
  The column **Active** can be used to define whether the reaction should be enabled or disabled.
- To define a new reaction, the button **New** has to be pressed. To reconfigure the contents of an already existing reaction, the **Contents...** menu item of its list entry's context menu has to be selected or the **Content...** button has to be pressed when a table entry is selected. In all cases the **'Reaction Contents'** window (see *Figure 10*, *Figure 11* and *Figure 12*) will be opened.

- To configure or reconfigure the time schedules and recipients connected to a reaction, the **Configuration...** menu item of the reaction's list entry's context menu has to be selected or the **Configuration...** button has to be pressed while a table entry is selected. This will open the **'Reaction Configuration'** window (see *Figure 13*).
- The **Delete** button can be used to erase all selected reactions.

The following three chapters will describe how the contents delivered by a reaction could be defined (see *Section 4.4.1, "Defining Contents Messages for Reactions"*), which macros can be used to add specific alarm data to the delivered contents (see *Section 4.4.3, "Using Macros in Contents Messages"*) and how Time Schedules and Recipients can be connected to a reaction (see *Section 4.4.4, "Connecting Time Schedules, Recipients and Reactions"*).

4.4.1 Defining Contents Messages for Reactions

To define messages to be send, the **'Reaction Message Contents'** window is used (see *Figure 10*, *Figure 11* and *Figure 12*). This window will be opened by selecting the **Content...** menu item for a reaction in the **'Reactions'** window, or when a new reaction is created by pressing the **New** button in the same window.

The messages define the actual contents that will be transmitted to the receiver when a reaction is triggered. Since a reaction can be connected to receivers of different types, and since different types of receivers may need to deliver different data, different messages can be defined for each of the receiver types.

To include data about the event that triggered the reaction into the messages several macros may be used. These macros will be listed and explained in *Section 4.4.3, "Using Macros in Contents Messages"*.

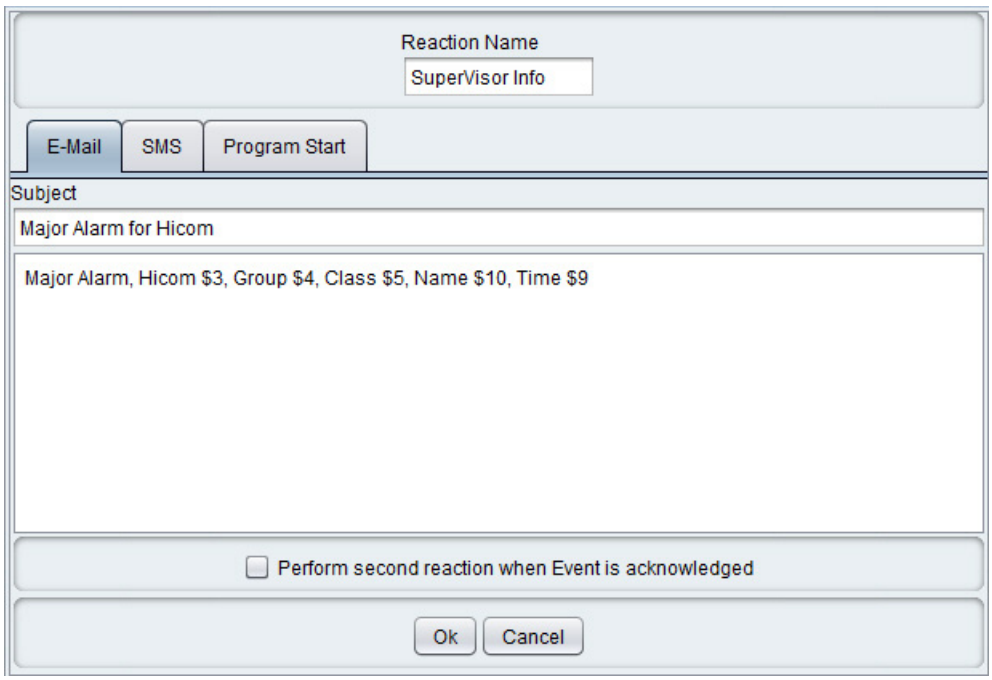

Figure 10 *The 'Reaction Message Contents' window (E-Mail)*

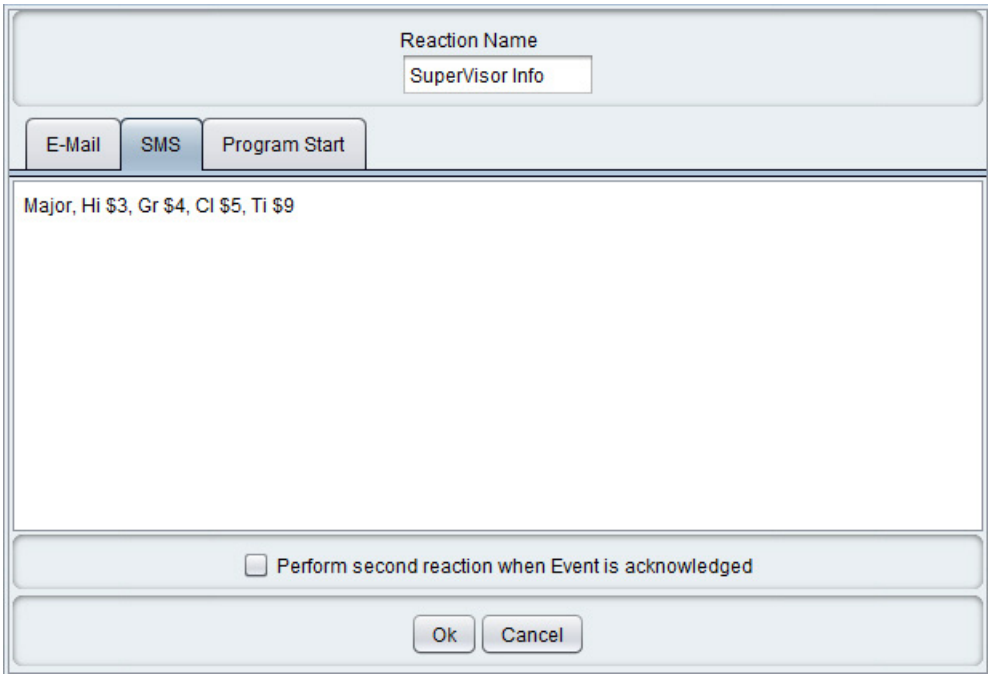

Figure 11 The 'Reaction Message Contents' window (SMS)

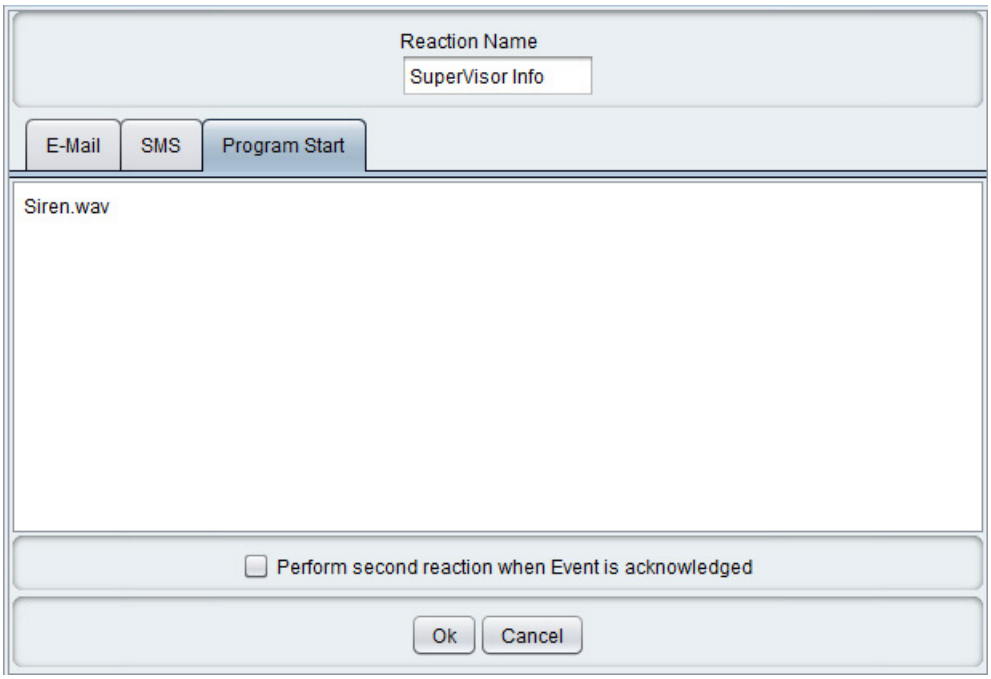

Figure 12 The 'Reaction Message Contents' window (Program Start)

The **'Reaction Message Contents'** window contains the following elements with the following functions, which will be shown on three separate pages:

- The **Reaction Name** text field contains the current name of the reaction when the window is initially opened. This text field can be used to change the name of the reaction.

- **E-Mail** page:
  - The **Subject** text field can be used to define the subject string of the mail messages that will be sent when the connected reaction is fired.
  - The **Message** text box can be used to define the body of the mail that will be sent when the reaction is fired.

  In additional fields a mail attachment (limited to 50MB) or an appointment invitation (ICS) can be configured.
- **SMS** page:
  - The **SMS Text** text box can be used to define the contents of the SMS that will be sent when the reaction is fired.
- **Program Start** page:
  - The **Arguments** text box can be used to define the arguments that will be added to the call of the program that will be started when the reaction is fired.
- When the **Ok** button is pressed, the changes made in this window will be used for the reaction. The window will be closed.
- When the **Cancel** button is pressed, the changes made in this window will be ignored. The window will also be closed.
- The checkbox **Perform second reaction when Event is acknowledged** can be checked to create an additional reaction when the original event gets acknowledged. More about this in the following chapter.

4.4.2 Reactions for Event Acknowledgements

It can be configured for each reaction on incoming events whether an additional reaction should be performed if the original event gets acknowledged. This can be done by checking the **Perform second reaction when Event is acknowledged** checkbox.

The acknowledgement can be done by two methods:

First by a manual acknowledgement of the event, and second by an automatic acknowledgement triggered by the Event Browser. This will be done when a matching acknowledgement trap is received (e.g. a HiPath 4000 Alarm-Off trap for a previously received matching HiPath 4000 Alarm-On trap).

The acknowledgement reaction will only be triggered by the first received acknowledgement. It will not be distinguished between manual and automatic acknowledgements.

In the case of email and SMS reactions, the original messages with the addition of the phrase ’Event Acknowledged’ will be repeated.

E.g. the email reaction

```
Subject:
Link Down
Body:
IP Node 192.168.1.1 reported link down on interface 5
```

will be acknowledged by the reaction

```
Subject:
Event Acknowledged: Link Down
Body:
Event Acknowledged:
IP Node 192.168.1.1 reported link down on interface 5
```

In the case of Program Start reactions, the same program with the same invocation parameters will be started. To distinguish between reactions and acknowledgement reactions, a parameter consisting of the variable `${acknowledged}` can be added to the program call. This variable will be replaced by the value '`false`' for reactions and by the value '`true`' for acknowledgement reactions.

4.4.3 Using Macros in Contents Messages

As stated in the previous chapters it is possible to include data from the event that triggers a reaction into the messages that will be sent. To do this, **macros** can be added to messages that are defined in the message windows. These macros will be substituted with the data they represent before the message is delivered to its recipients. Most macros can be selected from the field or function menu, which are displayed by right clicking within the input area. Text enclosed in <> brackets are placeholders which have to be substituted by the actual values.

All macros are described in Appendix C.

4.4.4 Connecting Time Schedules, Recipients and Reactions

To connect time schedules and recipients to a reaction the **'Reaction Configuration'** window can be used (see *Figure 13*). This window will be opened by selecting the **Configuration...** context menu item or by pressing the **Configuration...** button for a selected reaction in the **'Reactions'** window.

With help of the **'Reaction Configuration...'** window, pairs of time schedules and recipients can be created and connected to a reaction. When a reaction is fired it checks whether one or more time schedules are valid at that time, and will send the messages configured for the reaction to the recipients connected to the valid time schedules.

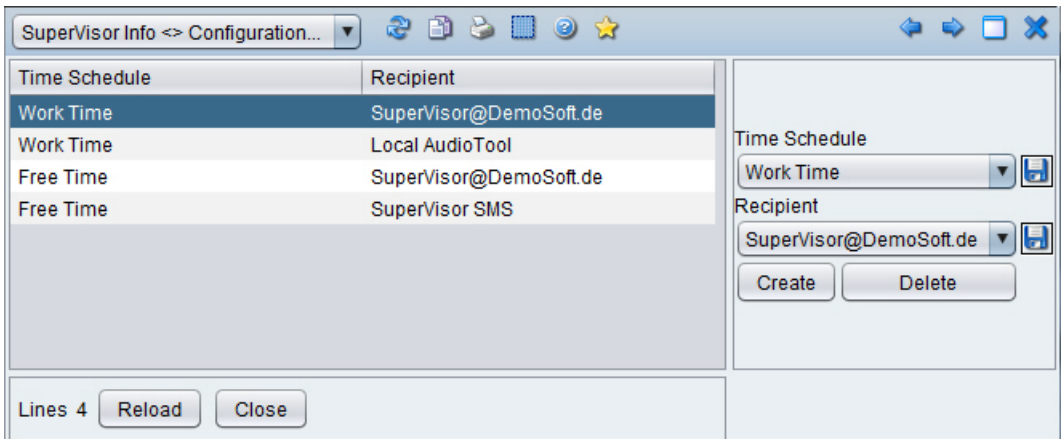

Figure 13 The 'Reaction Configuration...' window

The window consists of the following elements with the following functions:

- The **Configuration** table contains a list of all time schedule / recipient pairs connected to the reaction. The **Time Schedule** and **Recipient** columns will display the respective filter or recipient names.
- The **Time Schedule** selector is used to select one of the already created time schedules (see *Section 4.3, "Defining Time Schedules"*). When the button on the right hand side of the selector is pressed, the **Time Schedule** attribute of all lines currently selected in the **Configuration** table will be changed to the selected time schedule. When such a change would create a time schedule / recipient pair that is already part of the list, the change will be omitted.
- The **Recipient** selector is used to select one of the already created recipients (see *Section 4.2, "Defining Message Recipients"*). When the button on the right hand side of the selector is pressed, the **Recipient** attribute of all lines currently selected in the **Configuration** table will be changed to the selected recipient. When such a change would create a time schedule / recipient pair that is already part of the list, the change will be omitted.
- The **Create** button will try to add a new time schedule / recipient pair to the **Configuration** table when pressed. The current selection of the **Time Schedule** and **Recipient** selector will be used to define the respective values for the new table entry. If a pair with the same values already exists in the list, a warning message will be shown and no new entry is added to the table.
- The **'Delete'** button is only active when at least one line is selected in the '**Configuration'** table. When the button is pressed the selected time schedule / recipient pairs will be deleted from the table.
- The **'Reload'** button will refresh the contents of the window.
- The **'Close'** button will simply close the window when pressed.

4.5 Connecting Events and Reactions

While the previous chapters explained how a reaction could be defined, this chapter will explain how defined **reactions** can be assigned to actual **events**.

Important Note:
Within this User Guide:
HiPath 4000 - stands for HiPath 4000 or OpenScape 4000,
HiPath 3000 - stands for HiPath 3000 or OpenScape Business.
Some menu entries may vary accordingly, based on the actual technology or version.

Reactions can be connected to HiPath 4000, HiPath 3000 or Generic SNMP Trap events. The general procedure to connect a reaction to an event is identical for all three versions but there are minor differences.

In all cases the assignment of reactions to events is done in the **'MAR...'** window. This window can be reached from virtually any representation of a HiPath 3000 system, HiPath 4000 system or an IP node, by selecting the **MAR OpenScape Business/H3K...**, **MAR OpenScape 4000...** or **MAR Events...** menu item respectively. Also in all cases the **'MAR...'** window consists of a **table** or **tree** on the left hand side, and a **list** of the defined reactions on the right hand side. The **table** or **tree** contains the possible events to which reactions can be assigned, where each entry represents one possible event type. The layout of the table/tree and the list differs for HiPath 4000 systems, HiPath 3000 systems and IP nodes. The following two chapters will describe the different 'MARRSYS...' windows.

Important Note:
Defined macros are different for HiPath 3000, HiPath 4000 systems and for Generic SNMP Traps. Therefore it may be necessary to define different reactions for different types of traps.

Important Note:
A reaction will only be triggered when the system generating the event is in the state 'managed' and only if at least one time schedule connected to the reaction is valid at the time when an event occurs.

4.5.1 The HiPath 4000 and HiPath 3000 'MAR...' Window

For HiPath 4000 systems (see *Figure 14*) the **table** consists of the columns **Group**, **Class** and **Name**, which define the possible **event types**, which are identical to the list of configured alarms for the HiPath system, as found in the HiPath 4000 Manager/HDMS database. The column **React.** contains three numbers separated by slashes. The numbers represent how many different reactions are connected to the corresponding event type for minor, major or device alarms. E.g. '1/2/0' means that 1 reaction will be triggered for minor alarms, 2 reactions will trigger for major alarms and none for device alarms when the corresponding event type is fired for the actual HiPath machine.

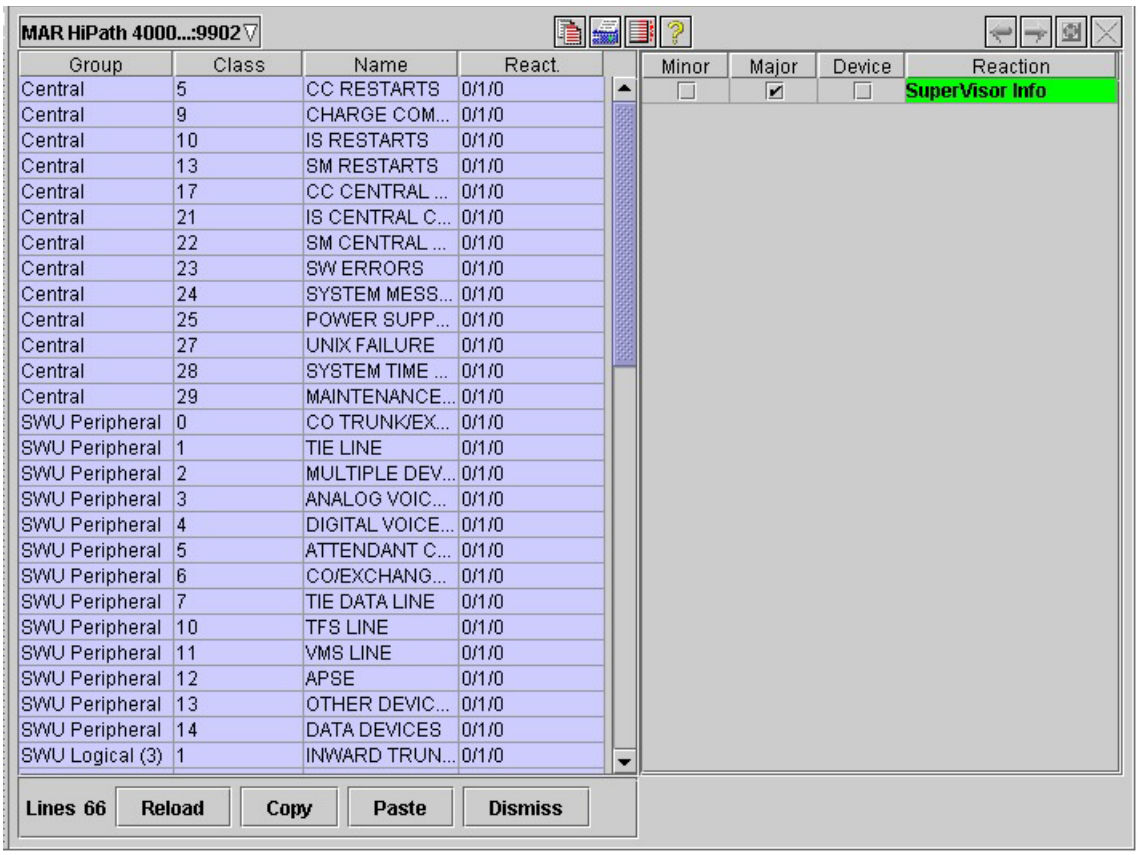

Figure 14 The 'MAR...' window (HiPath 4000)

The **list** contains one row for each defined reaction. The column **Reaction** identifies the reactions by their name. The three columns containing the check boxes (**Minor**, **Major**, **Device**) are used to assign events to reactions.

When an entry of the event table is selected, the **check boxes** display, whether their corresponding reaction will be triggered for minor, major or device priority alarms of this event type (they will when the check box is checked).

Clicking the **check boxes** will connect or disconnected the corresponding reaction to the event type currently selected in the **event table**. Using these **check boxes** makes it possible to connect different reactions, depending on the priority of an alarm event.

A click with the right mouse button on an entry in the **reaction list** will open a pull down menu with two entries. The activation of the menu item **Contents** will open the **'Reaction Message Contents'** window for the selected reaction (see *Section 4.4.1, "Defining Contents Messages for Reactions"*). The menu item **Configuration** will open the **'Reaction Configuration...'** window for the selected reaction (see *Section 4.4.4, "Connecting Time Schedules, Recipients and Reactions"*). Both can be used to view or manipulate the selected reaction.

For HiPath 3000 systems the **'MAR...'** window works in the same fashion, but has a slightly different appearance (see *Figure 15*).

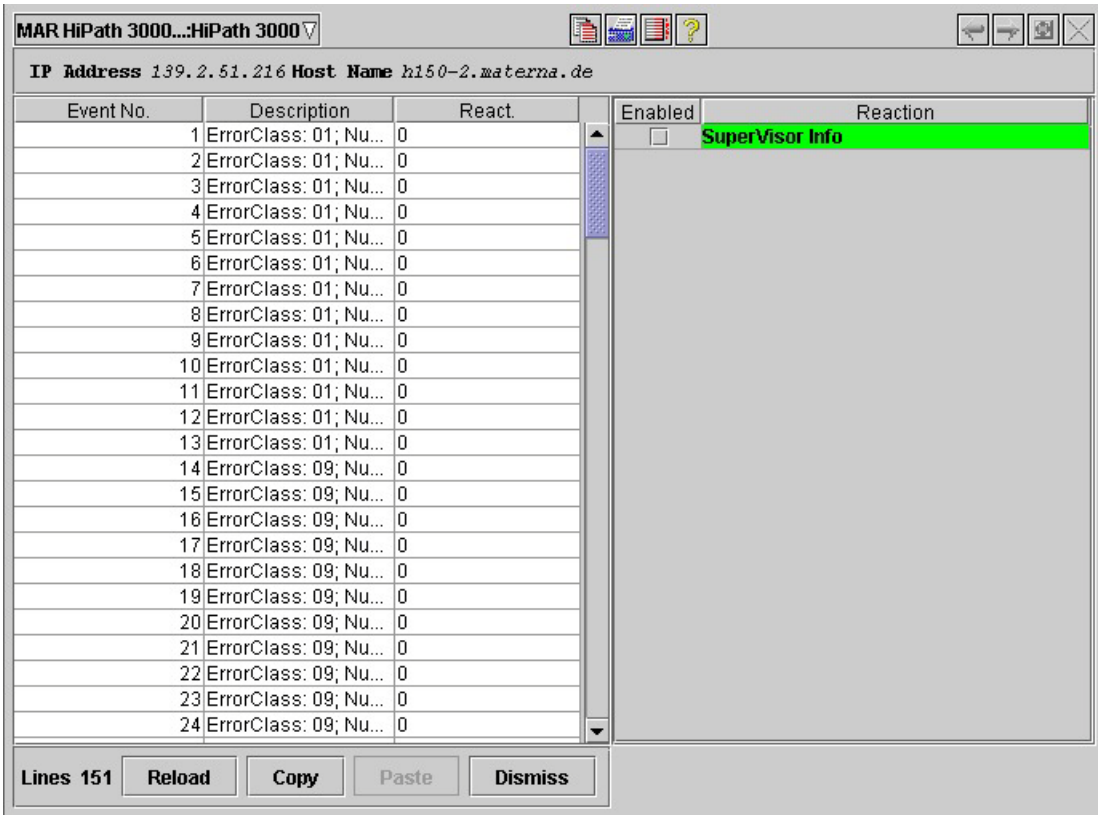

Figure 15 The 'MAR...' window (HiPath 3000)

Since there is no distinction between minor, major or device event priorities, the **reaction list** only consists of a column displaying the reaction names **(Reaction)** and one column containing the **checkboxes** (**Enabled**) to display, set and unset whether a reaction is connected to the currently selected event.

The **table** containing the possible **event types** to which reactions can be connected for the actual HiPath 3000 system consists of three columns. The first two columns (**Event No.** and **Description**) identify the event type and the third column (**React.**) displays how many reactions are currently connected to the event type.

4.5.2 The IP node 'MAR...' Window

By selecting the "MAR Events..." menu item from the IP node's context menu, reactions to all kinds of events that can occur for an IP node in general can be assigned (in contrast to HiPath 3000/4000 nodes). The kind of events depends on the installed and activated OpenScape FM Plugins and on the automatically discovered IP components under the IP node. All events fall into several event groups, whereas more than one event group can belong to one OpenScape FM Plugin. The following event groups are provided:

- HiPath MIB Host, HiPath MIB Applications, HiPath MIB Processes (HiPath MIB)
  The HiPath MIB lets you monitor applications and processes on HiPath-enabled devices in your network. HiPath-enabled devices are devices with a running HiPath MIB agent. This agent can notify the OpenScape FM server via SNMP traps e.g. if a monitored process is no longer active. For all these HiPath MIB SNMP traps reactions can be assigned.

- Host Resources
  The Host Resources Plugin shows you the programs running on IP nodes that support the host resources-MIB. During routine checks of the host resources-MIB SNMP agent (status polling) events can be generated in the OpenScape FM server. Those events can tell you, for example, that a monitored program was started or terminated or that a new program was added for monitoring.

- MS Windows Log Events (Host Resources)
  If a MS Windows Log Event was generated for an monitored application, a corresponding trap will be send to the OpenScape FM server.

- MIB-II (Enterprise MIB)
  If the selected IP node has an SNMP agent running (e.g. a MIB-II agent), it can send SNMP traps to the OpenScape FM server. To enable the processing of SNMP traps for IP nodes, the SNMP traps have to be defined via the **Enterprise MIB Plugin**. This Plugin integrates custom MIB definitions into OpenScape FM. By default, the MIB-II definition is already available and can be activated via the Enterprise MIB main menu. For details on how to integrate custom MIB definitions, see *Enterprise MIB Plugin User Guide*.

- IP Manager
  It is possible to assign reactions not only to SNMP traps defined via the Enterprise MIB Plugin, but also to various events generated by the IP Manager Plugin of the OpenScape FM. An IP Manager event can be, for example, a broken link to a router or that an ip node does not respond to a “ping”.

  Note:
  The IP Manager does *not* perform an ICMP echo request like the “ping”-program. Instead, general TCP connectivity is tested.

- Threshold Monitoring (Enterprise MIB)
  If thresholds are defined for certain MIB values and the MIB value collection of the Enterprise MIB Plugin is active, you can assign reactions to the threshold status of the corresponding threshold definitions.

For IP nodes the ’MAR...’ window looks like *Figure 16*. The events are displayed by a **tree** which fills the left side of the window**.** This tree includes all events that can occur for the current IP node. For each Enterprise MIB that was defined and discovered on an IP node, a separate sub-tree containing the corresponding traps is displayed. In dependance of the initialized Plugins and automatically discovered components the tree includes furthermore all configurable events (sorted by the corresponding event group).

Just like for HiPath 3000 systems for IP nodes there is no distinction between minor, major or device event priorities. The **reaction list** only consists of a column displaying the reaction names **(Reaction)** and one column containing the **checkboxes** (**Enabled**) to display, set and unset whether a reaction is connected to the currently selected event types for the current IP node.
For all SNMP traps, events or event groups (sub-trees), reactions can be assigned. To assign a reaction to an event or trap, select it and choose a reaction from the **reaction list** of the ’MAR...’ window. When at least one reaction is assigned to an event, it is marked by a **red check** mark. A **hollow red check** mark shows that a reaction is assigned to at least one element within a subtree.

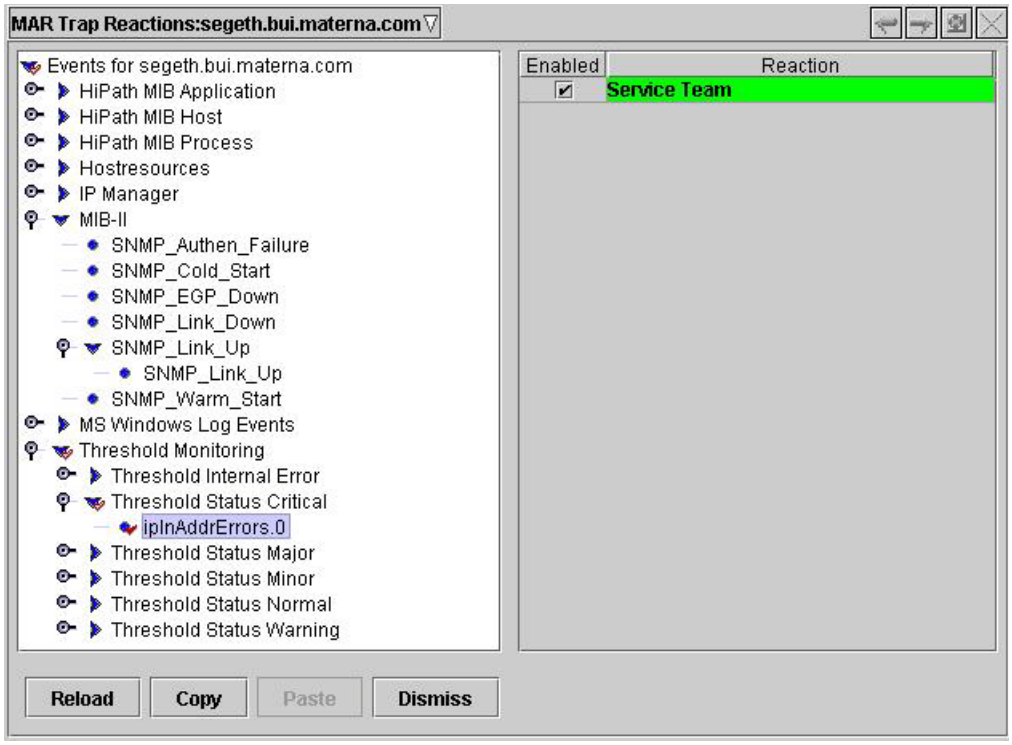

Figure 16 IP node

The current available and configurable events for IP nodes are in detail:

- HiPath MIB Application related events
  These events are generated for corresponding SNMP-traps which are send by the HiPath MIB agent to the OpenScape FM server. The events are only configurable for the monitored HiPath MIB application nodes of the OpenScape FM server (for more details please refer to the *HiPath MIB Plugin User Guide*).
  - HiPath Application Installed/Deinstalled
  - HiPath Application Changed To Status Critical, Major, Minor, Normal, Warning
- HiPath MIB Host related events
  These events are generated for corresponding SNMP-traps which are send by the HiPath MIB agent to the OpenScape FM server. The events are only configurable for the HiPath MIB agent nodes of the OpenScape FM server (for more details please refer to the *HiPath MIB Plugin User Guide*).
  - HiPath Host Up/Down
  - HiPath Host Changed To Status Critical, Major, Minor, Normal, Warning
- HiPath MIB Process related events
  These events are generated for corresponding SNMP-traps which are send by the HiPath MIB agent to the OpenScape FM server. The events are only configurable for the monitored HiPath MIB process nodes of the OpenScape FM server (for more details please refer to the *HiPath MIB Plugin User Guide*).
  - HiPath Process Up/Down

- HiPath Process Changed To Status Critical, Major, Minor, Normal, Warning

- Host Resources related events
  These events can be generated by the Host Resources Plugin during routine checks (status polling)

  - Application Added
    This event is generated in the OpenScape FM server if an application was marked for status monitoring. This event can be assigned to reactions for the application group container node.

  - Application Up/Down
    These events are generated in the OpenScape FM server if a monitored application was started or terminated. The event can be assigned to reactions for all monitored Host Resources application nodes.

- General IP Manager related events
  These events can be generated by the IP Manager during routine checks (status polling)

  - Duplicate IP Address
    This event is generated by the IP Manager when an IP interface is found which is bound to an IP Address that is already in use.

  - IP Node Down
    This event is generated by the IP Manager when the IP node responds to a “ping”. By default, the network connectivity is tested in 1-hour intervals (*status polling*).

  Note:
  The IP Manager does not perform an ICMP echo request like the “ping”-program. Instead, general TCP connectivity is tested.

  - IP Node Down
    This event is generated by the IP Manager when the IP node does not respond to a “ping”. By default, the network connectivity is tested in 1-hour intervals (*status polling*).

  - Inconsistent Netmask
    This event is generated by the IP Manager when the netmask of one of the IP node’s ip addresses does not match the netmask of the subnetwork the node belongs to.

- MIB-II related events
  The MIB-II traps which are defined in the corresponding Enterprise MIB definition file are listed in an own subtree. If a specific trap instance is defined for a MIB-II trap, the trap instance can be assigned to a reaction. The following MIB-II traps are supported for reaction assignments (for more details please refer to the *MIB-II RFC 1213*):

  - SNMP_Auth_Failure
  - SNMP_Cold_Start
  - SNMP_EGP_Down
  - SNMP_Link_Down
  - SNMP_Link_Up
  - SNMP_Warm_Start

- Events generated by threshold monitoring
  These events can be generated when MIB value collection is active for one or more MIB variables and there are thresholds defined for certain values. For detailed information on collecting MIB values and threshold monitoring, see *Enterprise MIB Plugin User Guide*.
  - Threshold Internal Error
    The threshold monitoring has run into an internal error. In this situation an unexpected error has occurred during the processing of a MIB value. E.g. the agent has returned a value in a wrong format.
  - Threshold Status Critical
    A monitored MIB variable has entered an interval or changed to a value for which a critical threshold status is defined
  - Threshold Status Major
    A monitored MIB variable has entered an interval or changed to a value for which a major threshold is defined
  - Threshold Status Minor
    A monitored MIB variable has entered an interval or changed to a value for which a minor threshold is defined
  - Threshold Status Warning
    A monitored MIB variable has entered an interval or changed to a value for which a warning threshold is defined
  - Threshold Status Normal
    A monitored MIB variable has entered an interval or changed to a value for which a normal threshold *or no threshold* is defined

It should be reminded that displayed SNMP traps and events can vary from one ip node to another, depending on the defined enterprise MIBs, the discovered SNMP agents (e.g. HiPath MIB, Host resources, MIB-II) and the defined threshold values. To define threshold values, the MIB value collection for the MIB variables to be monitored has to be configured first. For more information on how to collect MIB values and define thresholds on them see *Enterprise MIB Plugin User Guide*.

SNMP traps can carry additional information within so called trap variables. Which variables are included in a trap is defined by the MIB definition files of the *Enterprise MIB Plugin* (see *Enterprise MIB Plugin Users Guide*). When at least one trap variable is defined for a trap, the tree entry of this trap has a context menu containing the **Add Trap Instance...** menu item. If this menu item is selected, the **'Trap Instance'** window (see *Section 4.5.2.1, "The 'Trap Instance' Window"*) will be opened for this trap. There a new trap instance for this trap can be created.

For each trap instance that is displayed in the tree a context menu holding two menu items is available. The menu item **Delete Trap Instance** when selected will delete all occurrences of the trap instance from the system. The menu item **Modify Trap Instance...** when selected will open the 'Trap Instance' window (see *Section 4.5.2.1, "The 'Trap Instance' Window"*) for this trap instance to allow the modification of the trap instance definition.

Important Note:
Keep in mind that changes to the connection of reactions to events will only apply to the current IP node. The creation of a trap instance and changes to trap instance definitions will apply globally to all occurrences of this trap/trap instance.

If a reaction is connected to a **trap or event**, the reaction will be fired whenever this trap or event is received for the IP node. If a reaction is connected to a **trap instance**, it will only be fired when the respective trap is received and the trap variables have the values as they are defined for the trap instance. For more about trap instances see *Section 4.5.2.1, "The 'Trap Instance' Window"*. Trap instances are displayed in the tree as subtree entries of the trap for which they are defined.

4.5.2.1 The 'Trap Instance' Window

The **'Trap Instance'** window can be used to define or modify trap instances. Trap instances can be used to fire different reactions for a single trap depending on the variable values sent with the trap.

Generally speaking trap instances define a filter for a single trap type. Each trap instance chooses a number of trap variables and values for these trap variables. Only if all values for all trap variables chosen are received with a trap as they are defined for the trap instance, a reaction connected to the trap instance will fire.

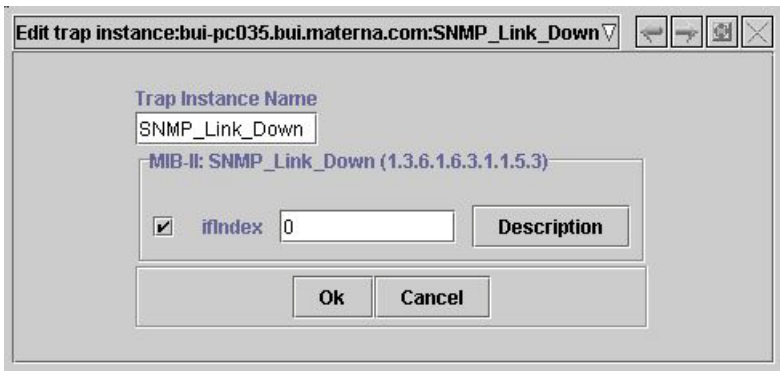

Figure 17 The 'Trap Instance' window

The **'Trap Instance'** window consists of the following elements which provide the following functionality:

- The **Trap Instance Name** textfield displays the name of the trap instance. The default name for a newly created trap instance is the name of the trap it belongs to. The textfield can be used to modify the name of the trap instance.
- The **Variable Filter Definition** area (the central area of the window) holds the elements to define which variables should be checked against what values. For each defined trap variable one set of elements exists which form a single row and has the following functions:
  - The **Variable Identifier** text displays the name of the trap variable connected to the elements of the row.
  - The **checkbox** near the **Variable Identifier** when the window is opened displays whether the trap variable should be checked against the value in the **Value** textfield or not. It can be checked or unchecked to define whether the trap variable should be checked or not checked against the value in the **Value** textfield in the future.
  - The **Value** textfield when the window is opened displays the value against which the connected variable should be checked. It can be used to modify this value. The textfield is only active when the **checkbox** is checked.

- The **Description** button can be pressed to get a short description of the connected trap variable. This description will be displayed in the **'Variable Description'** window.

  The '**Variable Description'** window consists of a **textfield** containing the trap variable description and a **Close** button which will simply close the window when pressed.

- The **OK** button will close the window when pressed. All changes made in the window will get active for all occurrences of the trap instance.
- The **Cancel** button will close the window. All changes made in the window will be ignored.

Important Note:
Keep in mind that all changes that are applied to trap instance definitions, will not only be done for the actual IP node but for all occurrences of this trap instance.

4.5.3 Multiple Selection

To speed up the process of assigning reactions to events, it is possible to change the configurations for more than one event type at the same time. To do this, the selection of more than one **table** entry at the same time in the **'**MAR**...'** window is possible. When more than one event type is selected, the **check boxes** in the **reaction list** may have one of three states:

- The check boxes will be checked when all of the selected events are connected to a reaction.
- They will be unchecked when none of the selected events is connected to a reaction.
- And they will be greyed when some of the selected events are connected to a reaction.

When a check box gets selected or deselected the corresponding reaction will be assigned or unassigned to all events that are currently selected.

Important Note:
In the 'MAR...' window for IP nodes, multiple selection is also available for the event tree. To assign a reaction to multiple events at once, either multiple leaf elements of the tree or a whole sub-tree (e.g. the MIB-II tree) can be selected. If one or more sub-tree elements are selected and assigned to a reaction, the reaction is assigned to all events within this subtree(s).

4.5.4 Copy Reaction Assignments

This feature can also be used to speed up the process of assigning reactions to events.

With this feature it is possible to copy the reaction configurations

- of one HiPath 3000 system to another HiPath 3000 system,
- of one HiPath 4000 system to another HiPath 4000 system and
- of one IP node to another IP node.

To do this, four additional buttons are available in the **'**MAR**...'** window.

- The **Copy** button will 'remember' the reactions assigned to all events currently selected in the event table when pressed. These data will be remembered until the button is pressed again for a system of the same system type (HiPath 3000 system, HiPath 4000 system, IP node) or until the client is closed.
- The **Paste** button will modify the reaction assignments of the current **'**MAR**...'** window. For all events that were 'remembered' when the **Copy** button was pressed last time for a system of the same system type and that are available for the current system, the currently assigned reactions are replaced by the copied configuration. Copied reaction assignments for events that are not available in the current system will be ignored. Events that are available in the current system but were not selected when the **Copy** button was pressed, will not be changed.

Important Note:
Even when all events were selected for the source system it is possible that not all events will be set for the target system by pressing the **Paste** button. This is the case if the target system has configured events that were not available for the source system.

- The **Dismiss** button will reset all changes that were made for event / reaction connections that were made since the window was opened for the currently active HiPath system/IP node.

Important Note:
When the 'MAR...' window is closed by clicking the **X** button of the window, all changes made to the data will stay valid. All changes in the "MAR..." window take place immediately, so there is no **Ok** button.

- The **Reload** button will simply refresh the contents of the window.

Since **'Copying'** works in conjunction with **'Multiple Selection'** (see *Section 4.5.3, "Multiple Selection"*), the reaction configuration for multiple systems can be done with a few mouse clicks.

4.5.5 Copy Complete Configurations

In many cases a configuration should be used for a node that has already been defined for another node. In this case it is possible to copy the complete MAR configuration.

By selecting the menu entry **MAR->Copy Configuration** on the source node, its complete MAR configuration will be put into the clipboard.

If the entry **MAR->Paste Configuration** is selected on the target node, the whole configuration will be used for this node.

Already existing configurations on the target node will not be deleted. But all entries will be overwritten that exist on both nodes. It is possible that reactions defined for the target node will be removed because they are not defined on the source node for the same event.

4.5.6 Assigned Reactions Overview

To get an overview about the nodes to which reactions have already been assigned, the page **Assigned Reactions** within the MAR configuration can be used.

This page contains a table which lists the nodes (column **Name**) and the reactions assigned for the node (column **Assigned Reactions**).

The button **Remove Assignment** can be used, to delete all assignments that were made for the selected node. This action has to be acknowledged.

5 Example

This chapter will recapitulate the MAR chapters by demonstrating an example how a reaction can be created and connected to a HiPath 4000 system.

The following reaction will be created:

For major alarms received from a special HiPath 4000 system the system supervisor should be notified. During work hours a signal sound should be played when an alarm arrives. For the rest of the time, but not on weekends, the supervisor should be informed by an SMS message. In addition in both cases an E-mail should be delivered.

To built this setup, three **recipients**, two **time schedules** and one **reaction** with three **messages** has to be defined. And finally the reaction has to be connected to the **event types** of the HiPath 4000 system. In the following the necessary steps to do this will be explained.

5.1 Defining Recipients

As a first step the **recipients** have to be defined. Three in all are needed for the example, one E-mail and one SMS recipient for the direct notification of the supervisor and one Program Start recipient to activate a sound tool on the managing station.

The configuration for the **recipients** may look like in the following seven screenshots.

Figure 18 and *Figure 19* show the configuration of a new **E-Mail recipient**.
Figure 20, *Figure 21* and *Figure 22* show the configuration of a new **SMS recipient**.
And finally *Figure 23* and *Figure 24* show the configuration of a new **Start Program recipient**.

In all three cases the pressing of the **New** button in the **'Recipients'** window will open the **'Recipient Configuration'** window. There the name and type of the recipient can be set (see *Figure 18*, *Figure 20* and *Figure 23*).
When the **Configure...** button is pressed, different configuration GUIs will be displayed depending on the **recipient type**, where the **recipient parameters** can be entered (see *Figure 19*, *Figure 21*, *Figure 22* and *Figure 24*).

The **'Recipients...'** page can be found within the MAR configuration (**MAR->Configure...**).

More details about the definition of recipients can be found in *Section 4.2.1, "The 'Recipient Configuration' Window"*.

Example

Defining Recipients

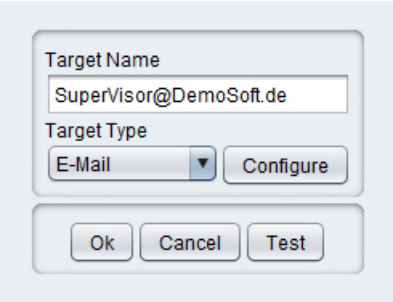

Figure 18 A new E-Mail recipient

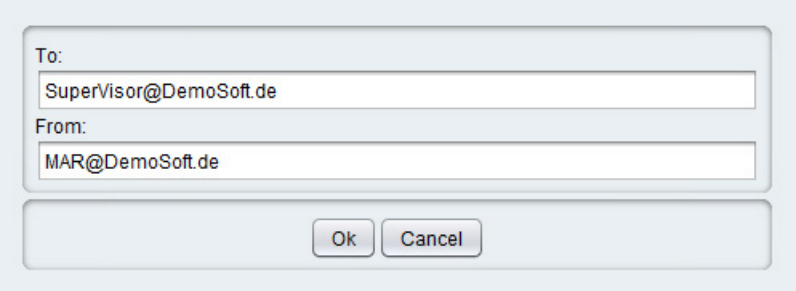

Figure 19 E-Mail recipient configuration

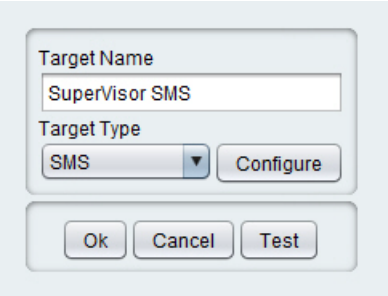

Figure 20 A new SMS recipient

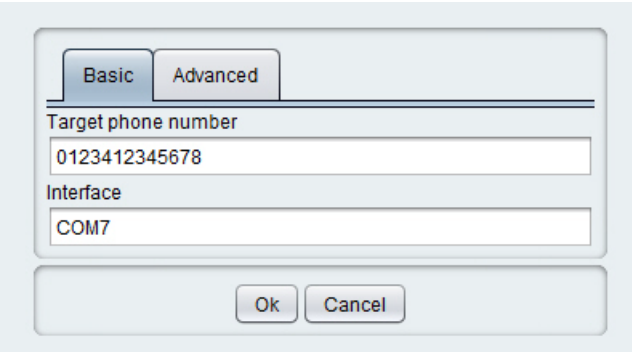

Figure 21 SMS recipient configuration (Basic)

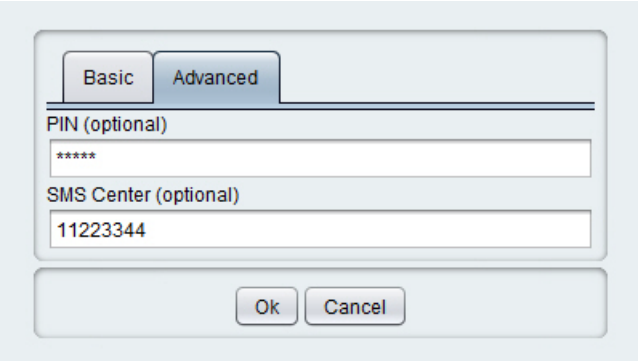

Figure 22 SMS recipient configuration (Advanced)

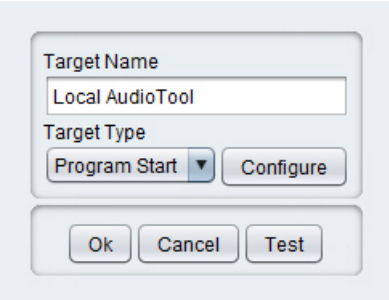

Figure 23 A new Program Start recipient

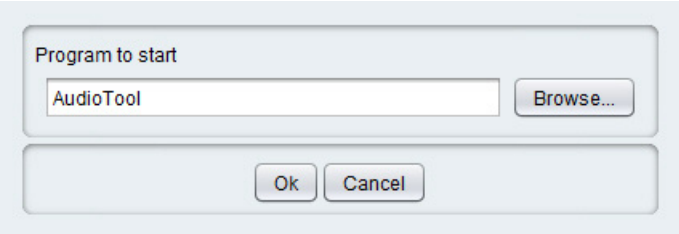

Figure 24 Program Start recipient configuration

When the recipient configurations are done, the list of recipients in the **'Recipients'** window may look like this:

Example

Defining Time Schedules

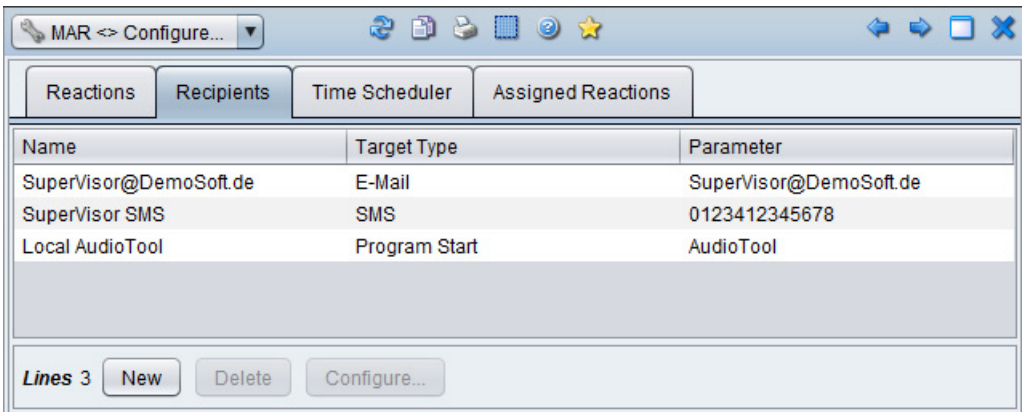

Figure 25 Recipient list

5.2 Defining Time Schedules

When this is done the **time schedules** have to be defined. In case of the example, one each for the work hours and one for the free time hours (except weekends).

More details about the definition of time schedules can be found in the OpenScape FM Desktop User Guide.

5.3 Defining Reactions

When the time schedules and recipients are defined they need to be connected to a **reaction**.

First off all the new reaction has to be created. This can be done by pressing the **New** button in the **Reactions** page which can be found within the MAR configuration (**MAR->Configure...**).

5.3.1 Defining Messages

When a new reaction is created, the **'Reaction Message'** window opens. In this window the name of the reaction can be entered. In addition, the **message** for the three possible **reaction types** can be set.

Since E-Mail, SMS and Start Program recipients will be connected to the reaction, all three types of messages have to be defined (see *Figure 26*, *Figure 27* and *Figure 28*)

A more detailed description how messages can be defined is found in *Section 4.4.1, "Defining Contents Messages for Reactions"*.

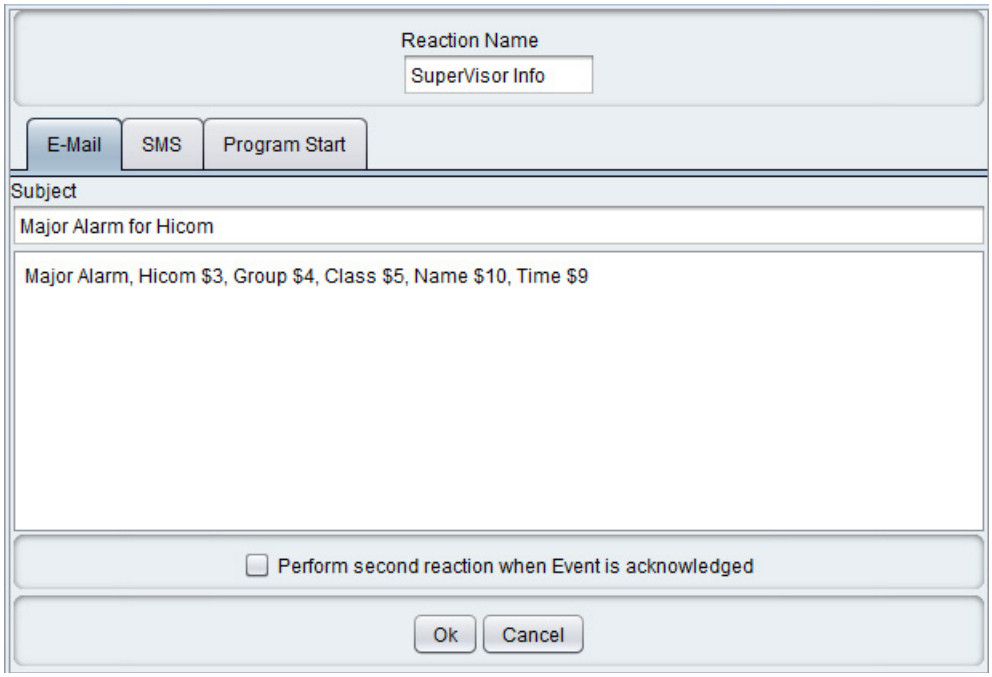

Figure 26 E-Mail reaction message

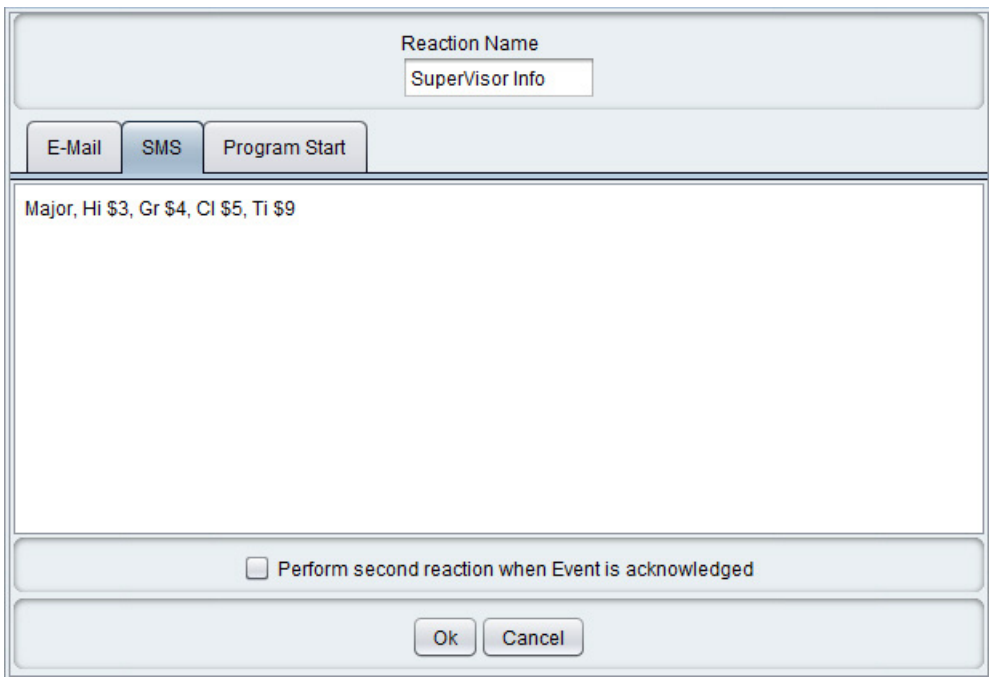

Figure 27 SMS reaction message

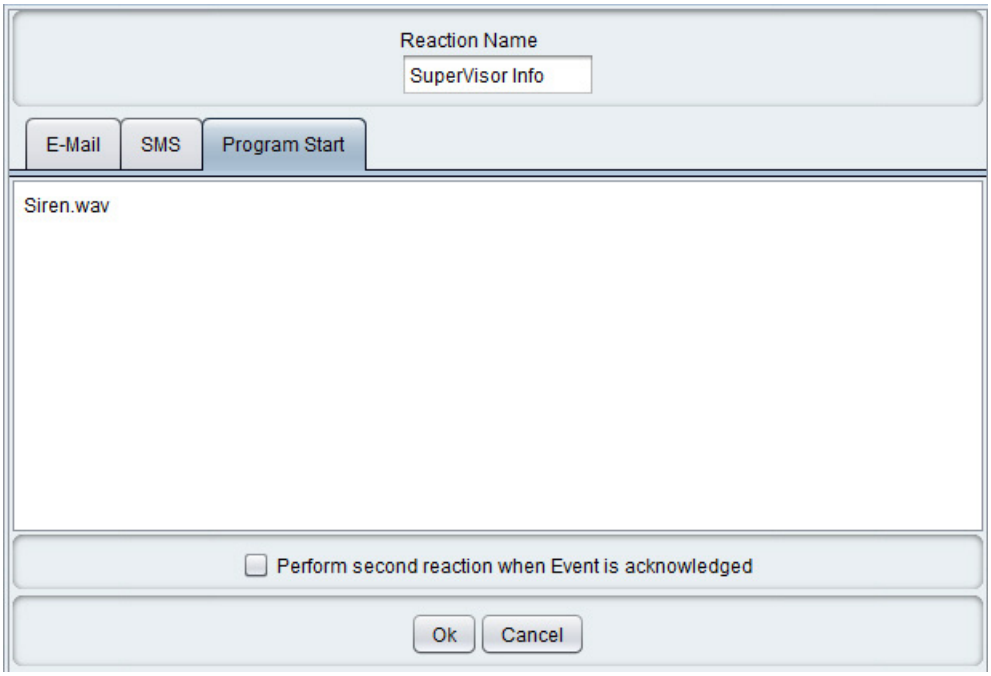

Figure 28 Start Program reaction message

For the E-mail and SMS messages **macros** are included to the text (see *Section 4.4.3, "Using Macros in Contents Messages"*). These macros will be replaced by the Group, Class and Name of the alarm that triggered the event and by the mnemonic of the system. This way the same reaction can be used for different systems and alarm classes without any modifications.

More about the possible macros that can be added to a message can be found in *Section 4.4.3, "Using Macros in Contents Messages"*

For the Program Start message, the message will be used as a parameter for the program that will be executed. In case of the audio tool this parameter is the name of the sound file that will be played.

5.3.2 Connection Time Schedules, Recipients and Reactions

When the **recipients** and **time schedules** are created, they can be attached to a **reaction**. This is done in the **'Reaction Message Recipients...'** window (see *Figure 29*). This window can be opened by pressing the **Configuration...** button for a selected reaction in the **Reactions** page which can be found within the MAR configuration (**MAR->Configure...**).

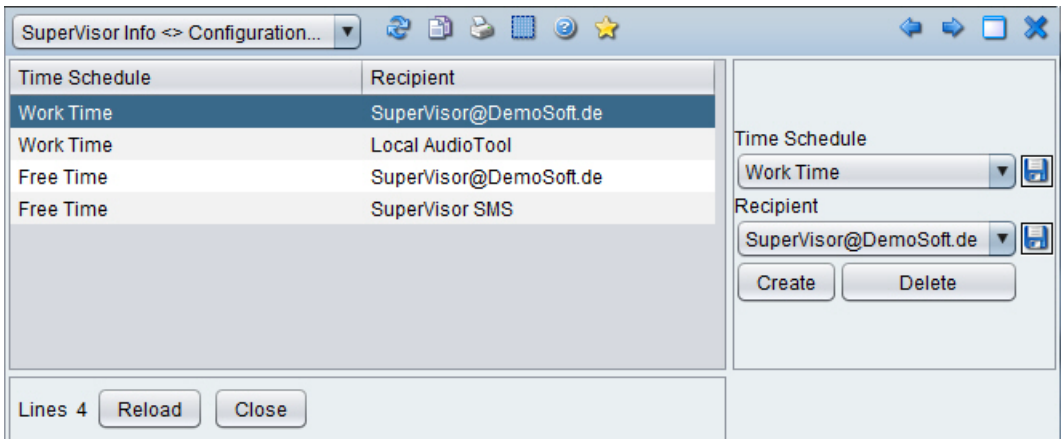

Figure 29 Connecting time schedules and recipients to the reaction

Using the **Time Schedule** and **Recipient selectors**, the different combinations of time schedules and recipients can be added to the **reaction**.

For the example, the Work Time time schedule is now connected to the E-Mail and Start Program recipient, the Free Time time schedule to the SMS and also to the E-Mail time schedule.

More about the connection of time schedules and recipients to reactions can be found in *Section 4.4.4, "Connecting Time Schedules, Recipients and Reactions"*

Important Note:
Connections of time schedules and recipients are only defined for a single reaction. Different reactions can use completely different combinations.

Important Note:
In the example no time schedule is defined for the week end. Events which are received during weekends will trigger no reaction at all.

5.4 Connecting Events and Reactions

When a reaction is completely defined, it can be connected to the alarm types and systems for which the reaction should be fired. This can be done with help of the **'MAR...'** window. This window can be opened by selecting the **MAR** menu item for a selected HiPath system.

For an example system this window may look like *Figure 30*:

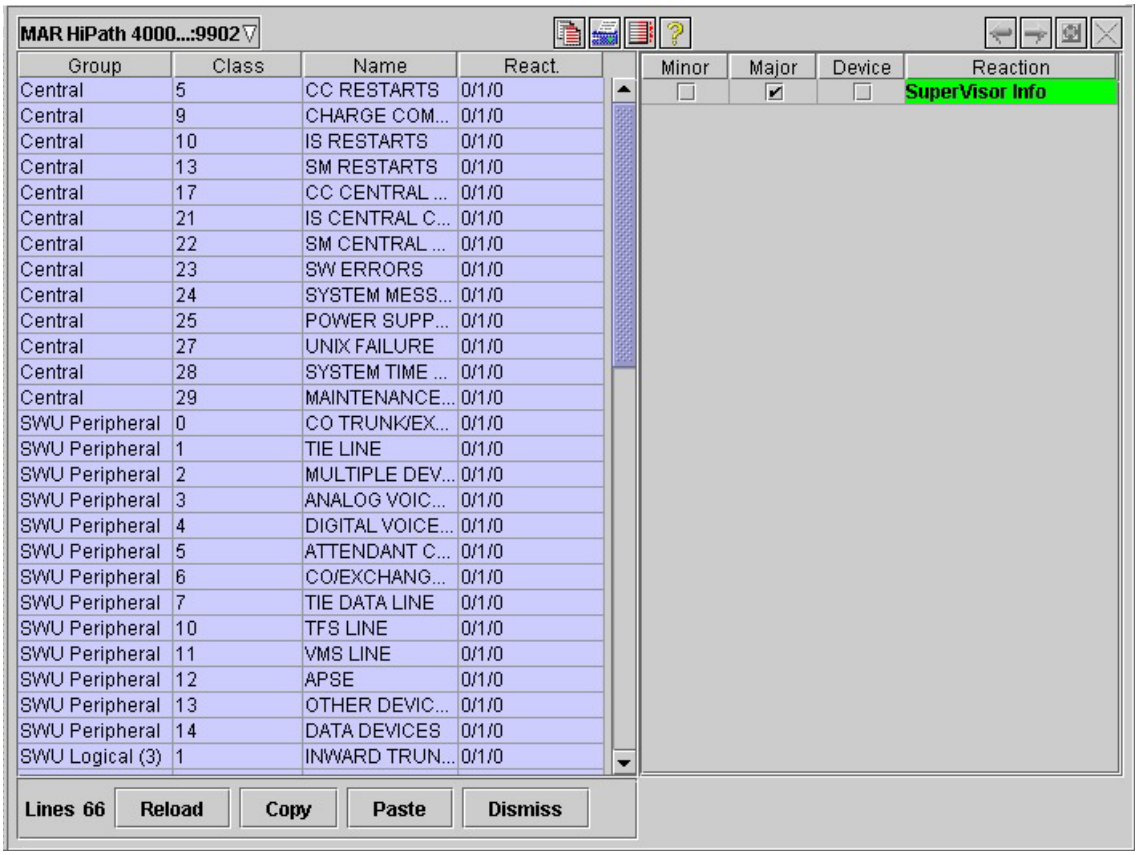

Figure 30 Connecting the reaction to a HiPath 4000 system

Since only major alarms should trigger the new reaction in the example, only the **check box** for major alarms is checked for the system and connected to all possible events.

Other reactions may be created for minor or device alarm, for a different subset of alarm types or for different systems.

A Documentation of MAR Rights

The plugin's access rights are integrated into the general access management (*see OpenScape FM Desktop User Guide*).

The description of the individual rights can be found within the tooltipps for the corresponding right symbols (tree or submap).

The names of the rights for this plugin begin with the plugin designation MAR.

B Hardware and Software Requirements

B.1 Hardware:

- OpenScape FM Server and Client, see *OpenScape FM Desktop User Guide* for detailed information concerning hardware and software requirements
- cellular phone (please read the *current OpenScape FM Release Notes* for an up-to-date list of supported cellular phones) and data cable (for connection to serial port) OR M20 terminal

B.2 Software:

- OpenScape FM Server and Client, see *OpenScape FM Desktop User Guide* for detailed information concerning hardware and software requirements
- Java(TM) APIs (please refer to the *current OpenScape FM Release Notes* for an up-to-date list of which APIs you need and how to install them)
- connection to SMTP (Simple Mail Transfer Protocol) server

Hardware and Software Requirements

Software:

C Macros for Contents Messages

Within the message of a reaction the data from the event can be added. This will be done via different macros. This appendix provides information about the macros which contain the information of the trap variables of the event which triggered the reaction. For more information about macros please refer to *Section 4.4.3, "Using Macros in Contents Messages"*.

C.1 General Macros

All macros start with the dollar symbol ('$'). The following list provides an overview over the implemented field macros. A list of the possible functions is included in *the OpenScape FM Desktop User Guide*.

- `$1` - `$99` will be substituted by the corresponding trap variable.
  Section C.2 shows details for **HiPath 3000** and **HiPath 4000**,
  Section C.3 for **HiPath MIB**
  Section C.4 for **Hostressources**
  and *Section C.5* for **IP Manager** specific traps.
- `$#` Number of variables
- `$*` All trap variables
- `$@` Time when the event was generated in seconds
- `$T` Uptime of the SNMP agent system which sent the SNMP trap
- `$x` Time when the event was generated as formatted string
- `$c` The log category of the event
- `$s` The log severity of the event
- `$e` The enterprise oid of the event
- `$H` depending on the system environment the hostname or the Full Qualified Domain Name (FQDN). If neither the hostname or the FQDN can be obtained, the IP address will be used.
- `$h` corresponds to $H. The hostname or the hostname part of an FQDN will be used. If those cannot be obtained, the IP address will be used,.
- `$D[1-99]` Interprets the given trap variable as timestamp and generates formatted output string (for HiPath 3000)
- `$G[1-99]` Interprets the given trap variable as timestamp in GMT and generates formatted output string (for HiPath 3000)
- `\r`, `\n`, `\t`: return, newline and tab (`\t` is only supported in mail configurations)
- `${date}`: The date of the event given in string format. The local time zone is used for this.

- `${datetimemillis}`: The time when the event occurred. The time is given in milli seconds since midnight of January 1st, 1970.
- `${datetimeseconds}`: The time when the event occurred. The time is given in seconds since midnight of January 1st, 1970.
- `${enterpriseid}`: The enterprise id connected to the event.
- `${hostname}`: The host name of the source that triggered the event.
- `${fullqualifiedhostname}`: The fully qualified hostname (e.g. pc123.materna.de) of the source that triggered the event.
- `${severity}`: The severity of the event (e.g. 'minor', 'major').
- `$r`, `$R`, `${source}`: The source name of the object to which the event is connected.
- `${sourceip}`: The IP address of the source which triggered the event (e.g. the IP address of a HiPath 4000 Manager).
- `${java.rmi.server.hostname}`: The name or IP address of the OpenScape FM Server.
- `${time}`: The time of the event given in string format. The local time zone is used for this.
- `${var[<X>]}`: The content of the (X+1)th event variable. E.g. `${var[4]}` will correspond to the content of the fifth variable.
- `${var.length}`: The number of variables attached to the event.
- `${acknowledged}`: Is replaced with the value 'false' when a reaction for a new event is performed. It is replaced with the value 'true' when an acknowledgement reaction for an already known event is performed.
- `${description}`: The text displayed in the Event Browser column Description.
- `${label}`: The automatically generated label of the object connected to the event.
- `${category}`: The category of the event as it is displayed in the Event Browser.
- `${sourceTimeStamp}`: The time of the event as it is reported by the source.

`${oid[<X>]}`: The oid of the (X+1)th event variable. E.g. `${oid[4]}` will correspond to the oid of the fifth variable.

C.2 HiPath 3000 and HiPath 4000 Macros

The following tables list the special macros as they are used for **HiPath 4000** and **HiPath 3000** systems. The macros which are not described contain no value.

HiPath 4000:

macro	value	type	annotation
$1	object Id		internal value for OpenScape FM

$2	Pabx Id	integer	-
$3	system Id (Mnemonic)	string	-
$4	alarm group	integer	1=Central, 2=Peripheral, 3=SWU Logical(3), 4=SWU Logical(4), 5=SW Peripheral, 6=DMS
$5	alarm class	integer	-
$6	priority	integer	1=Minor;2=Major;3=Device
$7	sender module	string	for example BPA, BPB
$8	status	integer	2=On/Set
$9	time stamp	integer	seconds since 1970
$10	alarm name	string	-
$11	event entry	string	internal value for OpenScape FM

HiPath 3000:

macro	**value**	**type**	**annotation**
$1	object Id		internal value for OpenScape FM
$2	error index	integer	-
$3	error date, time	string / dateAndTime	the output is not readable as string, for textual usage please use $D3
$4	error class	integer	-
$5	error code	integer	-
$6	access slot	integer	-
$7	access port	integer	-
$8	error description	string	-
$9	system status	integer	-
$10	error severity	integer	-
$11	event entry	string	internal value for OpenScape FM

C.3 HiPath MIB Macros

The following tables list the special macros as they are used for **HiPath MIB** systems. The macros which are not described contain no value.

Host Up/Down

macro	value	type	annotation
$1	object Id		-
$2	agent	string	-
$3	date	dateAndTime	-
$4	description	string	-
$5	event entry	string	internal value for OpenScape FM

Host Status Change:

macro	value	type	annotation
$1	object Id		-
$2	agent	string	-
$3	host state	integer	unknown (1), normal (2), warning (3), minor (4), major (5), critical (6), unmanaged (7), restricted (8), testing (9), disabled (10)
$4	date	dateAndTime	-
$5	description	string	-
$6	trap severity	string	This Severity can be different to the status of the object. unknown (1), normal (2), warning (3), minor (4), major (5), critical (6), unmanaged (7), restricted (8), testing (9), disabled (10)
$7	event entry	string	internal value for OpenScape FM

Application Installed/Deinstalled:

macro	value	type	annotation
$1	object Id		-
$2	agent	string	-
$3	application	string	-
$4	date	dateAndTime	-
$5	description	string	-
$6	event entry	string	internal value for OpenScape FM

Application Status Change:

macro	value	type	annotation
$1	object Id		-
$2	agent	string	-
$3	application	string	-
$4	state	integer	unknown (1), normal (2), warning (3), minor (4), major (5), critical (6), unmanaged (7), restricted (8), testing (9), disabled (10)
$5	date	dateAndTime	-
$6	description	string	-
$7	trap severity	integer	This Severity can be different to the status of the object. unknown (1), normal (2), warning (3), minor (4), major (5), critical (6), unmanaged (7), restricted (8), testing (9), disabled (10)
$8	event entry	string	internal value for OpenScape FM

Process Up/Down:

macro	value	type	annotation
$1	object Id		-
$2	agent	string	-
$3	application	string	-
$4	application state	integer	current state of the process: unknown (1), normal (2), warning (3), minor (4), major (5), critical (6), unmanaged (7), restricted (8), testing (9), disabled (10)
$5	date	dateAndTime	-
$6	description	string	-
$7	event entry	string	internal value for OpenScape FM

Process Status Change:

macro	value	type	annotation

$1	object Id		-
$2	agent	string	-
$3	process	string	-
$4	index	integer	Index of the process
$5	process state	integer	Current state of the process. unknown (1), normal (2), warning (3), minor (4), major (5), critical (6), unmanaged (7), restricted (8), testing (9), disabled (10)
$6	date	dateAndTime	-
$7	description	string	-
$8	trap severity	integer	This Severity can be different to the status of the object. unknown (1), normal (2), warning (3), minor (4), major (5), critical (6), unmanaged (7), restricted (8), testing (9), disabled (10)
$9	event entry	string	internal value for OpenScape FM

C.4 Hostressources Macros

The following tables list the special macros as they are used for **Hostressources** systems. The macros which are not described contain no value.

Application Added (OpenScape FM internal event):

macro	**value**	**type**	**annotation**
$1	object Id		-
$2	event category	string	-
$3	event type	string	internal value for OpenScape FM
$4	severity	string	-
$5	application name	string	-
$6	event entry	string	internal value for OpenScape FM

Application Up/Down (OpenScape FM internal event):

macro	**value**	**type**	**annotation**
$1	object Id		-

$2	event category	string	-
$3	event type	string	internal value for OpenScape FM
$4	severity	string	-
$5	event entry	string	internal value for OpenScape FM

C.5 IP Manager Macros

The following tables list the special macros as they are used for **IP nodes**. The macros which are not described contain no value.

IP node Up/Down (OpenScape FM internal event):

macro	value	type	annotation
$1	object Id		-
$2	event category	string	-
$3	event type	string	internal value for OpenScape FM
$4	severity	string	-
$5	ip node	string	-
$6	event entry	string	internal value for OpenScape FM

Duplicate ip address (OpenScape FM internal event)**:**

macro	value	type	annotation
$1	object Id		-
$2	event category	string	-
$3	event type	string	internal value for OpenScape FM
$4	severity	string	-
$5	duplicate ip address	string	-
$6	source object	string	-
$7	event entry	string	internal value for OpenScape FM

Inconsistent netmask (OpenScape FM internal trap):

macro	value	type	annotation

$1	object Id		-
$2	event category	string	-
$3	event type	string	internal value for OpenScape FM
$4	severity	string	-
$5	begin ip address	string	-
$6	end ip address	string	-
$7	event entry	string	internal value for OpenScape FM

D Character Sets

Supported Character Set for SMS Messages.

D.1 Supported Character Set for SMS Messages

The following character set is supported for SMS messages. The characters in the shaded areas are NOT supported.

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00	@	Δ		0	¡	P	¿	p
0x01	£	_	!	1	A	Q	a	q
0x02	$	Φ	"	2	B	R	b	r
0x03	¥	Γ	#	3	C	S	c	s
0x04	è	Λ	¤	4	D	T	d	t
0x05	é	Ω	%	5	E	U	e	u
0x06	ù	Π	&	6	F	V	f	v
0x07	ì	Ψ	'	7	G	W	g	w
0x08	ò	Σ	(	8	H	X	h	x
0x09	Ç	Θ	)	9	I	Y	i	y
0x0A	LF	Ξ	*	:	J	Z	j	z
0x0B	Ø	^1	+	;	K	Ä	k	ä
0x0C	ø	Æ	,	<	L	Ö	l	ö
0x0D	CR	æ	-	=	M	Ñ	m	ñ
0x0E	Å	ß	.	>	N	Ü	n	ü
0x0F	å	É	/	?	O	§	o	à

Figure 31 Supported characters for SMS

Index