A MITEL
PRODUCT
GUIDE

Mitel OpenScape Fault Management

Mitel OpenScape Fault Management V13 Event Gateway

User Guide

10/2025

Mitel

Notices

Trademarks

Contents

1 Preface

1.1 Purpose

This User Guide provides an introduction into the Event Gateway Plugin for OpenScape FM. Basic knowledge about network management and about the OpenScape FM is required. More about the latter can be found in the *OpenScape FM Desktop User Guide*. In addition, the user should have at least basic knowledge about the configuration of SNMP trap receivers.

1.2 Audience

This guide is addressed to users who want to learn how to use the Event Gateway Plugin for OpenScape FM.

1.3 Organization of this Guide

This guide is organized as follows:

- *Chapter 1, “Preface”* explains the structure of the manual.
- *Chapter 2, “Introduction”* describes the basic concepts used by the Event Gateway Plugin for OpenScape FM.
- *Chapter 3, “Getting started”* describes how to install and start the Event Gateway Plugin for OpenScape FM.
- *Chapter 4, “General Configurations”* explains common functions within the Event Gateway Plugin Config Tool.
- *Chapter 5, “Connection Parameters”* describes the different connection types and how they can be configured.
- *Chapter 6, “Filter Parameters”* explains how to define a filter for a source connection.
- *Chapter 7, “Trap Forwarding”* shows how to add an SNMP Trap Recipient.
- *Chapter 8, “Logging/Troubleshooting”* gives hints for the troubleshooting.
- *Chapter 9, “Configuration Example”* gives examples of how to configure the Event Gateway Plugin for the different targets.
- *Appendix A, “Requirements”* shows the hardware and software requirements of the Event Gateway Plugin.
- *Appendix B, “SNMP Trap MIB Definition”* shows the SNMP Trap MIB Definition for traps which are sent when the trap forwarder is configured.
- *Appendix C, “Variables”* describes the variables which can be used within the different configuration steps.
- *Appendix D, “Glossary”* gives a glossary of different terms.

1.4 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. Bold is also used for Buttons, menu names and items

Example: **Save**.

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: *see OpenScape FM Desktop User Guide.*

Italic type is also used for emphasis.

Example: *All* users will be affected.

2 Introduction

The OpenScape Fault Management (OpenScape FM) collects and represents the events of all supported HiPath/ OpenScape technologies (e.g. HiPath 3000, HiPath 4000, OpenScape Voice, HiPath MIB). These and other events are collected in the OpenScape FM Event Browser. If the configured maximum number of events[1] has been collected by the OpenScape FM, the oldest stored events will be overwritten by newly incoming events. To preserve or edit the event data with other tools, using the Event Gateway for OpenScape FM the data can be forwarded as SNMP traps to these tools.

To ensure that only those events are forwarded that are relevant for the target application, there is a two-stage filter mechanism. This mechanism reduces the number of events provided by the Event Gateway.

- The first-level filters the events incoming to the Event Gateway. This mechanism is based on the highly configurable modular access right system of the OpenScape FM. The Event Gateway connects as a specific user to the OpenScape FM and only the events that are seen by this user will be used by the Event Gateway. E.g. the Event Gateway can be configured to connect as a user that has only OpenScape FM rights for systems within a region, then only events for the systems of this region will be visible to the Event Gateway.
- The second-level filters the outgoing events. Only events that match filter parameters defined within the Event Gateway will be forwarded. For this filtering, a combination of Severity, Source, Category, Class, OpenScape FM Tenant Domains and Event Attributes can be defined.

The combination of both mechanisms allows a wide range of filter possibilities.

In this User Guide:
HiPath 4000: stands for HiPath 4000 or OpenScape 4000,
HiPath 3000: stands for HiPath 3000 or OpenScape Business.

2.1 Overview of the Event Gateway Configuration

A Event Gateway configuration consists of three steps:

1. Configuration of the source connection
2. Creation of a filter for the source connection (optional)
3. Creation of a target configuration

[1] currently 25.000

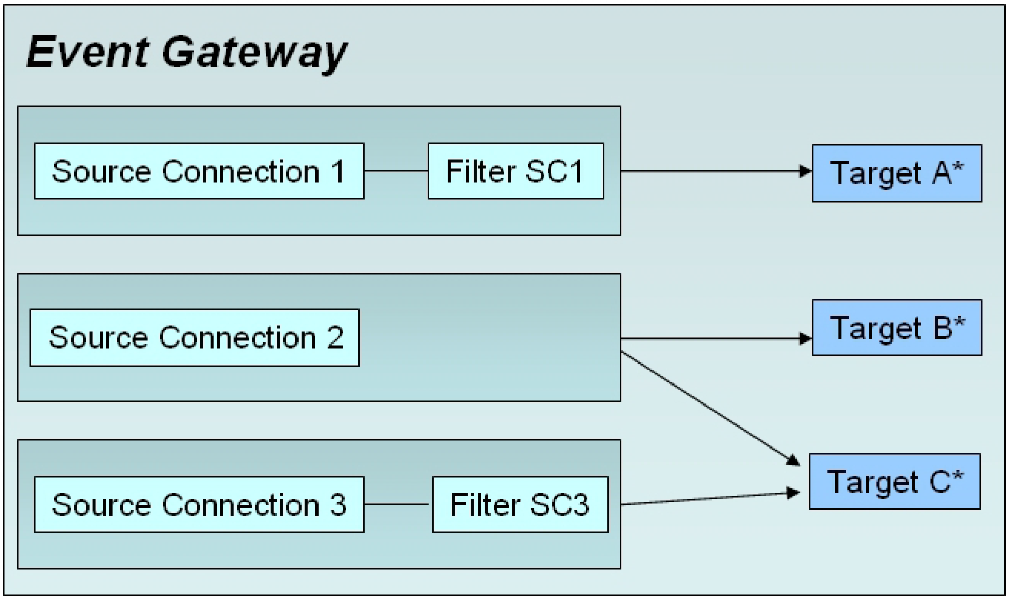


Figure 1 Overview of the Event Gateway components

Within the configuration of a source connection, it will be defined from where which events will be collected by the Event Gateway. The OpenScape FM server, the connection type and the login information the Event Gateway uses to connect to the OpenScape FM are defined. With the connection type the source is entered: It can be chosen between the OpenScape FM Event Browser or the ECE configuration. The ECE has to be installed first, if selected, and a valid ECE license must be available. Via the login information the user the Event Gateway uses to connect to the OpenScape FM server gets defined. The rights granted to the user control which events are accessible by the Event Gateway. More information about defining source connections can be found in *Chapter 5, "Connection Parameters"*.

For each source connection a filter can be defined. This filter consists of one or more different filter statements. Using a filter allows to constrain the number of the outgoing events. If no filter is defined, all events which passes the first filter level will go out. *Chapter 6, "Filter Parameters"* explains how to configure a filter for a source connection.

Via the Trap Forwarder module outgoing OpenScape FM Events are transformed into an SNMP Trap. These traps are sent (forwarded) to one or multiple SNMP Trap targets (e.g umbrella management systems). Since the Trap Forwarder by default uses a uniform trap format for all outgoing traps (see *Appendix B, "SNMP Trap MIB Definition"*), only one SNMP trap format has to be recognized by a trap recipient. If new OpenScape FM event types are added to the OpenScape FM, they will show up automatically at the recipient side, supposed they match the filtering criteria. The trap targets of forwarded OpenScape FM Events are centrally configured in the Event Gateway Configuration Tool. The original systems keep sending their traps only to the OpenScape FM server. More information about configuring a trap target is given in *Chapter 7, "Trap Forwarding"*. In addition the Trap Forwarder module provides the possibility to create/use a custom trap format, to give the user a maximum of flexibility concerning the trap structure. Additional configuration may be necessary on the trap receiver side, when custom trap formats are used.

3 Getting started

3.1 Installation

During the installation of the OpenScape FM the Event Gateway Plugin will be installed automatically. During the installation of the Event Gateway the service ’`OpenScape Event Gateway`’ will be created and started. The Event Gateway Plugin provides the user interface necessary for the configuration of the Event Gateway.

3.1.1 Event Gateway Plugin Installation

The Event Gateway Plugin is installed during the OpenScape FM installation. After the installation the plugin has to be initialized (**Server->Plugins->Initialize Event Gateway Plugin**).

3.1.2 Event Gateway Service Installation

During the installation of the OpenScape FM the Event Gateway will be installed automatically.During the installation the service „OpenScape FM Event Gateway“ gets set up and started. Within the OpenScape FM the plugin provides the necessary user interface to configure the Event Gateway.

3.2 Licensing

To work with the Trap Forwarder, an Event Gateway license has to be installed within the OpenScape FM. Without a valid license, the Trap Forwarder will not be started.

A license to forward events from the OpenScape FM Server to two trap targets is included in the base license of the fault management.

Since the Event Gateway is a plugin within the OpenScape FM, the license file is loaded by the standard OpenScape FM licensing mechanism. More about this can be found in the OpenScape FM Standard Desktop User Guide.

3.3 Starting/Stopping the OpenScape FM Event Gateway Service

Windows:

On Windows platforms, the Event Gateway is registered as system service ’OpenScape `Event Gateway`’ which can be stopped or started by using the Windows Service Manager.

If the service cannot be started succesfully, *Section 8.1, “Logging”* and *Section 8.2, “Troubleshooting”* may provide help in finding the cause.

Unix:

During the installation, a startup script is created in the operating system specific startup directory. If a valid license is available, the OpenScape FM Event Gateway server will be started automatically after a reboot.

The OpenScape FM Event Gateway server can be started and stopped via scripts. They are named `startgw` and `stopgw` and are located in the installation directory.

After starting the OpenScape FM Event Gateway a process like "`/usr/bin/java -Djava.security.policy=/opt/OpenScapeEventGateway/server/conf`" is running on the machine.

If an error message appears during the start, *Section 8.1, "Logging"* and *Section 8.2, "Troubleshooting"* can be referred.

4 General Configurations

For the configuration of the Gateway, the Gateway Config Tool has to be used. The Gateway Config Tool connects to a running OpenScape FM Event Gateway server (see *Section 3.3, “Starting/Stopping the OpenScape FM Event Gateway Service”*). If the OpenScape FM Event Gateway Service is not running, a warning will be displayed and the Gateway Config Tool will not be started. The Config Tool does not have to run on the same system where the OpenScape FM Event Gateway server is running.

The configuration tool is started by selecting the menu entry **Configure** form the main menu **Server->Event Gateway**.This opens a Login window showing the fields **Host** and **Password**. The Host field is used to enter the hostname where the Event Gateway Server is running. The Password field is used to enter the access password for the Gateway configuration. As a default the password is empty after the installation and must be set at the first login. It can be changed in the Gateway Config Tool on the panel **Server Properties**.

The following panels exist:

- **Connection Parameters:**
  Definition of the source connections (see *Chapter 5*).
- **Filter Parameters:**
  Definition of filters for the incoming events (see *Chapter 6*).
- **Server Properties:**
  Configuration of the Event Gateway (see *Section 4.1*).
- **Trap Ziel Parameter:**
  Definition of the targets and data for the trap forwarding (see *Chapter 7*).

4.1 Server Properties

Here the login password for the OpenScape FM Event Gateway server is configured.

The **Server Properties** panel of the Gateway Config Tool displays the fields to change the password. The actual password has to be typed into the field **Password**. The new password has to be typed into the field **New Password**. As a confirmation, the new password has to be typed again into the field **Retype new Password** to prevent typing errors.

Pressing the **OK** button will change the Config Tool’s password to the value entered into the **New Password** and **Retype new Password** fields.

Additionally it can be selected whether the OpenScape FM Event Gateway server should write debug output in its logfile. This helps the support team to analyze problems/errors with the Gateway. To set/reset the debug output option, select/deselect the checkbox **Activate Server Debug Output** and press the **OK** button. It has to be kept in mind that an activated debugging can cause a reduced performance.

The table **System Properties** is used to configure further properties.

4.2 Standard Buttons

Certain buttons providing common functions will appear on all panels with the exception of the Server Properties panel described above.

The button **Create** can generally be used to create an additional entry in the left most table of a panel. For example it can be used to create an additional connection within the Connection Parameter panel. In the case of the **Filter Parameters** panel the **Create** button will create a new filter statement row.

The button **Remove** removes selected entries. The removal has to be confirmed in a confirmation window dialogue.

With the button **Save**, the updated changes will be saved. This button is only active when changes have been made.
After pressing the button **Save**, the Gateway will immediately use the new configuration.

The button **Exit** can be used to cancel all changes that were made since the last time the Save button was used. The cancellation has to be confirmed in a confirmation window dialogue.

4.3 Macros for Enterprise-Specific SNMP-OIDs

The Gateway offers the possibility to define macros for Enterprise-specific SNMP OIDs (Enterprise OID). This reduces the necessity to look up frequently used OIDs, and makes configurations more readable. The macros can be selected from a menu. This menu is offered after right clicking in a field where an Enterprise OID can be entered, e.g in filter statements. When a macro is selected, the name of the macro is shown in the Config Tool, but internally the defined Enterprise OID is used.

The macros are defined in the file `enterprises.properties` in the installation directory of the Gateway server:
`<Gateway installation directory>/server/resources/macros.`

In order to define a new macro entry, a new line has to be added in the `enterprises.properties` file. First, the name of the macro has to be typed in. This name will be shown in the menu. Next an equal (=) has to be typed in followed by the Enterprise OID as which the macro should be interpreted. After saving the file, the OpenScape FM Event Gateway server has to be restarted to activate the new macro(s). It has to be kept in mind, that after an update installation of the OpenScape FM Event Gateway server, all changes in the `enterprises.properties` will be lost. Therefore a backup of this file should be made before the OpenScape FM Event Gateway server is updated.
The following lines show examples for the definition of macros.

```
HiPath4000MajorAlarmOn=1.3.6.1.4.1.231.7.2.1.21.2.2.30
HiPath4000MajorAlarmOff=1.3.6.1.4.1.231.7.2.1.21.2.2.31
```

5 Connection Parameters

On the **Connection Parameters** panel the connection between the gateway and a source (an OpenScape FM Server) can be configured. Connections are required to receive the events from the OpenScape FM.

A login parameter and a connection type to the source OpenScape FM Server are defined.

The access rights of the user selected to connect to the Gateway determine the events seen by the Gateway. This enables the highly configurable modular access rights system of the OpenScape FM to be used to effectively restrict the events received. For example, an OpenScape FM user can be created who only has specific rights for a specific system. A Gateway that uses this user for the connection only receives events for this system.

A more sophisticated method of filtering is to use the Event Correlation Engine. The ECE can be configured to filter events before they are sent to the gateway.

For each configured connection, an additional filter of the gateway itself can be used. Only events that pass this filter are forwarded. More about filters can be found in *Chapter 6, “Filter Parameters”*.

The **Connection Parameters** panel consists of two parts: the left side contains a list of configured connections. The right side displays the configured values for each entry of the list.

- **Connection Name**: A descriptive name describing the connection should be entered here. This name is used in other panels to select the connection.
- **Host Name**: The server name can either be the host name or the IP address of the OpenScape FM server
- The field **Connector** is used to define the connection type. The connection type defines how the Gateway connects to the source server. This can be one of the following:
  - **PSAccount** means the Gateway server will connect as a certain user to the OpenScape FM and the Gateway gets all OpenScape FM events of the Event Browser to which the login user has access to.
  - **MegEceConnector** means the server will also connect to OpenScape FM as a certain user, but the Gateway gets only those events that are available for the user and are routed to the corresponding MegEceConnector node in the OpenScape FM. More about the configuration of the ECE Connection can be found in *Section 5.1*. More about the ECE itself can be found in the separate *Event Correlation Engine User Guide*.
- **User:** The user name which should be used by the Gateway to login to the OpenScape FM server.
- **Password/Retype Password:** The password of the user specified above.
- **RMI Port**: The server port through which the communication will be performed. By default it is 3042.

5.1 MEG-ECE Connection

If a **MegEceConnector** has been configured for a connection (see above), only the events that are routed to the corresponding MegEceConnector node on the ECE submap are handled by the Gateway.

To use a MEG-ECE Connection, the ECE plugin has to be initialized.

If a MegEceConnector connection is configured in the Gateway, in OpenScape FM an MegEceConnector icon appears on the ECE submap. The label of this icon contains the IP address of the OpenScape FM Event Gateway server and the OpenScape FM user the Gateway uses to connect to OpenScape FM. The Event Source icon on the ECE submap represents the source of all OpenScape FM events, the Event Browser icon all events of the Event Browser. If the Event Browser icon is connected directly with the MegEceConnector icon, all events of the Event Browser will be send to the Gateway. Naturally one or more filters can be defined and arranged between the Event Browser icon and the MegEceConnector icon. In this case only events which pass the defined filters will be routed to the Gateway. More information about the definition of filters for the ECE can be found in the separate *Event Correlation User Guide*.

6 Filter Parameters

By default all incoming events of a source connection (OpenScape FM) are forwarded to the defined target (Trap Target). The number of events which are forwarded by the Gateway can be restricted in two ways:

1. By selecting a user at the **Connection Parameter** panel which has restricted event view rights. E.g. the user does not see all events but only for certain systems.
2. By defining filter rules for the Gateway at the **Filter Parameter** panel.

In most cases, it is easier to use only the second option and configure a user for OpenScape FM which sees all events. The Gateway filter rules are described in this chapter. For each source connection, one or no filter can be defined. Filter can consist of one or more filter statements which can be concatenated via AND or OR.
Each filter statement checks if the incoming event matches the defined filter criteria. A filter statement can check for the following criteria of an event:

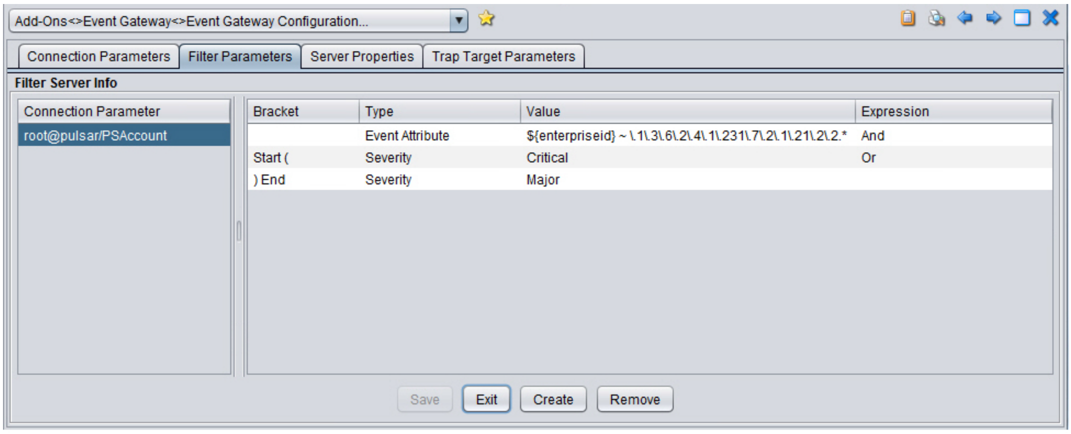
- the severity,
- the source object,
- the class of the source object,
- the tenant domain of the source object
- and different event attributes like the Enterprise OID of the original trap.

The different filter statements will be checked for equality (`a=b`). For the filter statements checking the Source, Class or Tenant Domain a regular expression can be used for comparison.

Note:
The Gateway uses the Java class `java.util.regex.Pattern` to handle Regular Expressions. Since Regular Expressions are a very powerful instrument, we cannot provide an exhaustive explanation at that point. Please refer to the specific Java documentation about the class for detailed information.

In addition for filter statements checking for event attributes a user can choose among various compare operators.The comparison value of a filter statement can be manipulated by special string formatting functions. The description of the different functions of the Formatting Language can be found in the *OpenScape FM Desktop User Guide*. There can be simple filters like „Severity=Critical“ or complex ones using the different features of the string formatting functions in combination.

The filter statements are defined on the **Filter Parameters** panel (*Figure 2*). The Filter Parameters panel consists of two parts: the left side contains a list with all configured source connections. The right side contains a table which displays all configured filter statements for the selected connection.



Add-Ons<>Event Gateway<>Event Gateway Configuration...

Filter Server Info

Connection Parameter
root@pulsar/PSAccount

Bracket	Type	Value	Expression
	Event Attribute	${enterpriseid} ~ \.1\.3\.6\.2\.4\.1\.231\.7\.2\.1\.21\.2\.2.*	And
Start (	Severity	Critical	Or
) End	Severity	Major	

Figure 2 Config Tool: Filter Parameters

The filter table consists of four columns: **Bracket**, **Type**, **Value** and **Expression**.

Expressions can be grouped with the help of the column **Bracket**. All values that are possible for a specific row can be selected from the context menu of the column. The context menu is only displayed if at least two filter expressions have been defined. Nested brackets are not possible.

The **Type** column offers the following Filter Types: „Category“, „Severity“, „Source“, „Source (Reg.-Exp.)“, „Class (Reg.-Exp.)“, „Tenant Domain (Reg.-Exp.)“ and „Event Attribute“.
In the column **Value** the comparison value for a selected type is specified. The column Value provides a context menu for the quick-selection of the comparison values. The content of the context menu depends on the selected type.

If there is more than one filter statement defined for a connection, they can be combined with either an ‘And’ relation or an ‘Or’ relation in the **Expression** Column. By default they are combined with an ’And’ relation. This can be changed by clicking in the field and selecting another value.

The following filter types are available:

- **Category:**
  A category filter checks if the category of an event matches the category specified in the field **Value**. To define a category filter select in the field **Type** the value „Category“ and enter in the field **Value** the event category, as there are for example:
  „User Events“: events which concern user operations, e.g. login
  „IP Manager“: events related to IP nodes and networks.
  „<Name of plugin>“: fault events from managed devices.

  To get a complete list of available categories please refer to the OpenScape FM event browser.

- **Severity:**
  A severity filter checks if the severity given in the field **Value** matches the severity of an event. To define a severity filter, select in the field **Type** the value „Severity“ and select in the field **Value** the respective severity. There are six different severities selectable: ’Unknown’, ’Normal’, ’Warning’, ’Minor’, ’Major’ or ’Critical’. These are the severities already known from the OpenScape FM Event Browser.

- **Source:**
  A source filter checks if the source object name of an event matches the object name defined in the field **Value**. To define a source filter, select in the field **Type** the value „Source“ and enter the OpenScape FM object name of the desired source object in the field **Value**. The source name can be accessed by the „Object->Properties“ browser which can be invoked within the OpenScape FM Client (attribute: Name).
- **Source (Reg.-Exp.):**
  Same as the option „Source“. Though in this case, the matching has not to be exact, but to a regular expression defined in the field **Value**.
- **Class (Reg.-Exp.):**
  A class filter checks if the class of the source object of an event matches the regular expression stated in the field **Value**. To define a class filter, select in the field **Type** the value „Class (Reg.-Exp.)“ and enter a constant or a regular expression representing the OpenScape FM class(es) of the object in the field Value. The class of an object can be found in the „Object->Properties“ browser, which can be invoked within the OpenScape FM Client (attribute: class capability).
- **Tenant Domain (Reg.-Exp.):**
  If only events should be used, that relate to objects that belong to a certain tenant domain, then the filter type „Tenant Domain“ can be used. The filter will only match for objects belonging to a tenant domain matching the regular expression stated in the field **Value**. To define a tenant domain filter select in the field **Type** the value „Tenant Domain (Reg.-Exp.)“ and enter a constant or a regular expression representing the OpenScape FM tenant domain name(s) in the field Value.
- **Event Attribute:**
  An event provides various information. In the Gateway, this information is accessible via the so called event attributes. An event attribute is, for example, the Enterprise OID of the original SNMP Trap which causes the event. Via the an Event Attributes filter the Gateway can check if a specific event attribute matches a specific value. This value can be another event attribute, data of type constant or variable, an operation (see Formatting Language in the *OpenScape FM Desktop User Guide*) or any combination thereof. (The menu appearing on an right click in the field Event Attribute can be extended by user defined values, see *Section 4.3, “Macros for Enterprise-Specific SNMP-OIDs”*.)
  For the Event Attribute, the **Value** column is divided into three parts. In the middle an operator can be selected. Using the selected operator, the entries in the left and the right field will then be compared. If this comparison is stated as true, the filter will match.

When a filter has been specified, an event (coming from the source connection the filter is defined for) gets only forwarded to the target system(s) (Trap Target) when the complete filter matches.

Example:
If only HiPath 4000 events of major or critical severity should be received, a category filter with the category value “HiPath 4000” and two severity filters with the corresponding values („Major“ and „Critical“) have to be specified (*Figure 2*).

Attention:
If more then one source connection of the same OpenScape FM server is selected for the same target configuration and an event matches for more then one of these source connections, multiple traps may be generated in this case. To avoid these duplication, the filters of the source connections have to be mutually exclusive.

7 Trap Forwarding

The Trap Forwarder is responsible for the conversion of OpenScape FM events into SNMP traps. This traps can be sent to one or more SNMP trap recipients.

The Trap Forwarder offers a uniform trap format by default, but the user may also define a custom trap format. If the uniform trap format is used, only this format has to be defined at the trap recipient system (e.g. umbrella management platform). Event data of the different technologies are represented by this trap.

Using the Trap Forwarder relieves the end user from adding the trap recipient systems to all the systems sending traps to the OpenScape FM. The sending systems send their traps only to the OpenScape FM. The Event Gateway collects all relevant events and the Trap Forwarder sends them to one or more trap recipients.

The standard SNMP MIB definition of the Trap Forwarder SNMP Trap can be found in *Appendix B, “SNMP Trap MIB Definition”*. By default the trap sends a number of **standard trap variable bindings** followed by the variable bindings of the original trap.

The **standard trap variable bindings** are:

- **ip address:** This field contains an IP address: In case of a HiPath 4000 event (e.g. alarm), this will be the IP address of the HiPath 4000 Manager. In case of a HiPath 3000 event (forwarded HiPath 3000 trap), this will be the IP address of the event originating HiPath 3000 system. If the event is an internal event, generated by the OpenScape FM, this field will contain the IP address of the OpenScape FM server.
- **severity**: An integer value that represents the severity which was assigned to the event by OpenScape FM (unknown(1), normal (2), warning (3), minor (4), major(5), critical(6), unmanaged (7), restricted(8), testing(9), disabled(10))
- **object name**: The OpenScape FM object name of the object the event was assigned to.
- **date**: The date when the event was received by OpenScape FM.
- **category**: The event category to which the event is assigned in the Event Browser.
- **message**: The message describing the event as assigned by the OpenScape FM; the description in the Event Browser.
- **event type**: An integer value which represents the type of the event. If the event was generated due to the reception of an SNMP trap, the event type will be snmp (2). If the event was generated internal in the OpenScape FM, the event type will be internal (1).
- **source ip**: The IP address of the source from which the event was send originally.
- **source Date**: The date which was assigned to the event by the original source.
- **event key**: A string to identify inverse events.
- **enterprise oid**: The original SNMP Enterprise OID assigned to the event.
- **event Id**: The id of the event in the Event Browser.

In addition to its default behavior, Trap Forwarder offers the possibility to change the sequence and/or the number of variables send with the trap:

- The standard variable bindings can be extended by the so called **Additional Variables**. These are freely defined additional trap variable bindings which can be added to the trap by the user. In this case, the variable bindings of the original trap are not sent with the trap. The Additional Variable bindings are configured on the **Trap Target Parameter** panel.
- The sending of the standard and original variables can be omitted. Then only the defined Additional Variables will be sent.

By using one of these options a user can define its own trap. It has to be kept in mind that in this case the standard Trap Forwarder SNMP MIB definition does not match. The new Trap Forwarder trap format is configured on the Trap Target Parameter panel of the Trap Forwarder config tool. The user has to configure the format of this custom trap at the target system (e.g. Umbrella Management tool), too.

The **Trap Target Parameters** panel is represented in the Event Gateway Config Tool. It consists of two parts: the left side contains a list with trap recipients. The right side contains the parameters to be used for sending the SNMP Traps to the selected recipient.

Trap Targets Info: Here, all defined trap recipients are listed.

Target: The recipient of the traps. The SNMP trap target name can either be a host name or an IP address which is known to the machine the OpenScape FM Event Gateway server is installed on.

Port: The remote port where the SNMP traps will be sent to.

Community: The community string used to send SNMP traps to the given trap target.

Connection: Here, one or more connection entries can be selected. These connection entries are used as source. All events which came from these sources matching their filter will be send to the target defined in the field **Target**. The connection entry will be defined on the panel **Connection Parameters**.

PDU Type: The type of the SNMP Protocol Data Unit that should be used. This can either be a v1 trap (`V1TRAP`), a v2 trap (`V2TRAP`), a v2 information request (`V2INFORM`), a v3 trap (`V3TRAP`) or a v3 information request (`V3INFORM`).

If SNMPv3 traps should be send, the respective fields can be used to select the **SNMPv3 Security Level**, the **SNMP v3 Authentication Protocol** and the **SNMPv3 Privacy Protocol**. The corresponding passwords can be defined in the fields **SNMPv3 User**, **SNMP v3 Authentication Password** and **SNMPv3 Privacy Password**.

Enterprise ID: The enterprise OID of the trap to be sent.
By default the OID will be defined by two variables to generate the enterprise OIDs as they are defined within the Trap Forwarder MIB.
`mmpgwEventOIDs.severityinteger`
The variable `mmpgwEventOIDs` stands for the fixed part of the OIDs defined in the Trap Forwarder MIB. It provides all digits of the OID besides the last one:
`.1.3.6.1.4.1.192.6.3.1.1.4.1.2.0`
The variable `severityinteger` provides the last digit of the OID. This variable will change depending on the

severity of the event which triggered the trap.
To generate an OID for a particular severity, the variables for the different severities (like `Minor` or `Critical` (see *Table 1*)) are appended to the fixes part of the OID.

Table 1 *Enterprise ID Variables*

Unset	0
Unknown	1
Normal	2
Warning	3
Minor	4
Major	5
Critical	6
Unmanaged	7
Restricted	8
Testing	9
Disabled	10

Suppress Standard Variables: If this option is set, only the variables defined via the fields **Additional Variables** will be sent in the trap.

Additional Variables: Up to twenty additional trap variable bindings, can be defined here. They will be added to the trap. The value field may contain data of type constant or variable, an operation (see Formatting Language in the *OpenScape FM Desktop User Guide*) or any combination thereof. A right click in one of the fields offers a menu which helps you to define the value. When a trap is sent, the contents of the value field will be evaluated, and then be added to the trap as an additional string variable.

8 Logging/Troubleshooting

8.1 Logging

The logfile documenting events and errors regarding the OpenScape FM Event Gateway server can be found in the directory `<installation_dir>/server/log/`. It is named `megServer.log`. Here, events and errors regarding the OpenScape FM Event Gateway server are logged.

The logfile documenting events and errors regarding the Gateway Config Tool can be found in the directory `<installation_dir>/client/log/`. It is named `megGui.log`. Here, events and errors regarding the Gateway Config Tool are logged.

Every time a component (server or config tool) will be started or it's logfile `<LogfileName>.log` reaches a size of 10 MB, the `<LogfileName>.log.old` will be saved to `<LogfileName>.log.older` and `<LogfileName>.log` will be saved to `<LogfileName>.log.old`.

In case of trouble, make sure to check these files for hints pertaining to the problem.

8.2 Troubleshooting

When problems arise during the work with the Gateway please look into the Gateway logfiles (*Section 8.1, "Logging"*) to see what kind of error has occurred:

Problem	Possible Reasons	Solution
It is displayed that a valid license is not available. The gateway does not start.	License has not be installed properly	The Gateway must be licensed correctly, see *Section 3.1, "Installation"*
Operation timed out: connect or *Connection refused to host: <hostname>*	The OpenScape FM server is not configured or running	The connection parameters must be defined in the Gateway Config Tool (see *Section 3.1.1, "Event Gateway Plugin Installation"*) and the OpenScape FM server must be started
Exception *invalid Authentication*	An invalid OpenScape FM user is configured (wrong login name or password)	A valid user and a suitable password must be configured in the connection parameters, see *Chapter 5, "Connection Parameters"*

Problem	Possible Reasons	Solution
Configurations made with the Gateway Config Tool are not saved, the next time the Gateway Config Tool is started you see the old configuration.	The Configurations have not been saved, when closing the Gateway Config Tool	It has to be ensured that the Save is confirmed before the Gateway Config Tool is closed, see *Chapter 4, “General Configurations”*
Event <id> has no license (EVENT_EXCEEDED) or *Event <id> has no license (EXPIRED)*	Not enough licenses available	The license must be extended accordingly
Event <id> has no license (IP Node Grace Period EXPIRED) or *Event <id> has no license (Port Grace Period EXPIRED)*	Not enough licenses available	The license must be extended accordingly

9 Configuration Example

This chapter contains examples of Gateway configurations according to scenarios with specific requirements.

9.1 Scenario: Forward Specific Events as Traps

9.1.1 Requirements

In this scenario only the major HiPath 4000 events of the following alarm group/class should be forwarded as SNMP trap to the Umbrella Management platform of the company: 3/1, 3/2, 3/3 and 3/4.

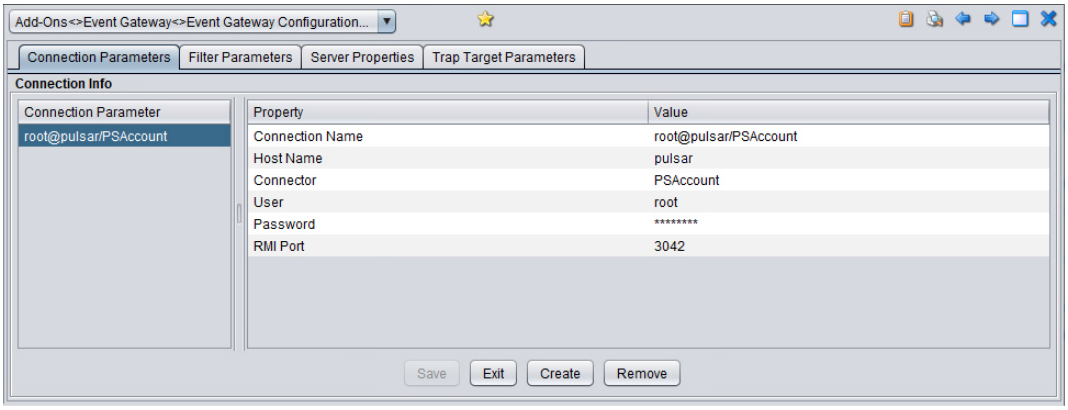
9.1.2 Configuring OpenScape FM

Within the OpenScape FM no configuration is necessary.

9.1.3 Configuring Trap Forwarder

First we define the connection configuration. We select the panel **Connection Parameter** and press the **Create** button to create a new entry. For the new entry the following fields are necessary:

- **Connection Name**: Enter a name for this connection configuration, for example in the following syntax <user>@<OpenScape FM server>/<connection type>, in our example we name it root@pulsar/PSAccount
- **Server**: Enter the name of the OpenScape FM server. In our example our server is called „pulsar“.
- **User**: Enter the OpenScape FM user here, as which the Trap Forwarder should connect to the OpenScape FM. In our example we use the user „root“.
- **Password, Retype Password**: Enter the password of the OpenScape FM user here.
- **Connector**: Select the Connection type here, in our example we need the connection type PSAccount.



Connection Parameter
root@pulsar/PSAccount

Property	Value
Connection Name	root@pulsar/PSAccount
Host Name	pulsar
Connector	PSAccount
User	root
Password	********
RMI Port	3042

Figure 3 Connection Parameters

Next let us define the filter for this connection. Change to the panel **Filter** and select from the left list the connection we have just defined. Our filter will consists of five filter statements, so press five times the button **Create** to get five rows in the right list. The first filter statement checks if the Enterprise OID belongs to an HiPath 4000 alarm. The other four statements filters for the group and class we want to be forwarded.

Let us start with the first filter statement. The enterprise OID of a trap is held in the Trap Forwarder variable ${enterpriseid}.

The enterprise id of a HiPath 4000 major on alarm looks like this:
.1.3.6.2.4.1.231.7.2.1.21.2.2.30

The enterprise id of a HiPath 4000 major off alarm looks like this:
.1.3.6.2.4.1.231.7.2.1.21.2.2.31

They differ in the last digit. So we need a statement where the last element can be any combination. This can be done by using a regular expression. Any character is symbolized by „.*“. Since the „.“ is a specific character in regular expressions, we have to escape the other „.“. In our example the regular expression will be the following:

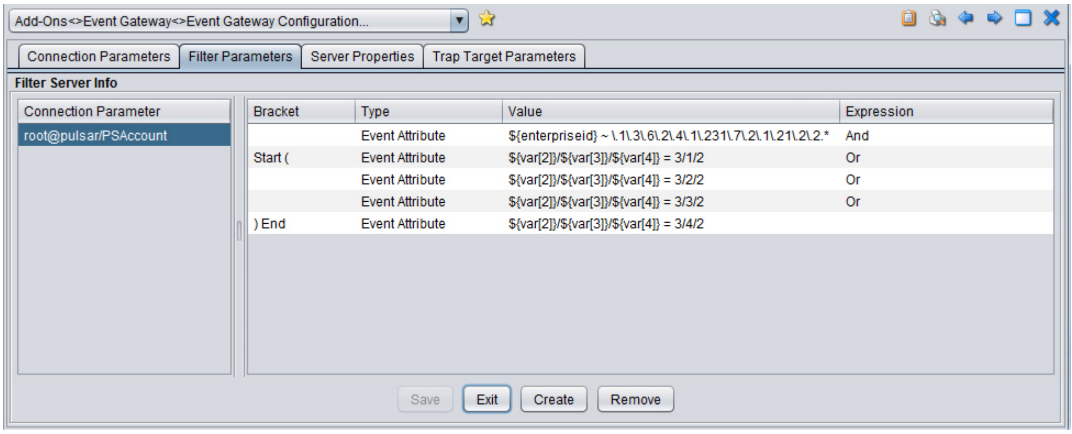
```
${enterpriseid} ~ \.1\.3\.6\.2\.4\.1\.231\.7\.2\.1\.21\.2\.2.*
```

This is our first filter statement.

The next filter statements have to check, if the alarm is an major alarm and if it belongs to one of the following alarm group/class combinations 3/1, 3/2, 3/3 or 3/4. For HiPath 4000 alarm traps the second trap variable contains the alarm group, the third trap variable contains the alarm class and the fourth trap variable the priority (2=major, 1=minor, 3=device). Knowing this allows us to define a statement which look like this for the alarm group/class combination 3/1. The „2“ at the last checks if the alarm is a major alarm. The „/“ is used as separator for the single values.:

```
${var[2]}/${var[3]}/${var[4]}=3/1/2
```

Now we have to define further statements for the other alarm group/class combinations. Finally our Filter will look like in the following screenshot.



Filter Server Info

Connection Parameter
root@pulsar/PSAccount

Bracket	Type	Value	Expression
	Event Attribute	${enterpriseid} ~ \.1\.3\.6\.2\.4\.1\.231\.7\.2\.1\.21\.2\.2.*	And
Start (	Event Attribute	${var[2]}/${var[3]}/${var[4]} = 3/1/2	Or
	Event Attribute	${var[2]}/${var[3]}/${var[4]} = 3/2/2	Or
	Event Attribute	${var[2]}/${var[3]}/${var[4]} = 3/3/2	Or
) End	Event Attribute	${var[2]}/${var[3]}/${var[4]} = 3/4/2	

Figure 4 *Filter configuration*

At last we configure the Trap Target. In our example the system which should receive the traps is called „umbrella_mgmt“. Its SNMP Agent is running on port 162. We change to the panel **Trap Target Parameter** and press the **Create** button. In the new entry we have to fill in the following fields:

Target: Here, we enter the host name or the IP address of our trap recipient, here „umbrella_mgmt“.

Port: The port on which the SNMP recipient listens. In our example it is the default value 162.

Community: Here, the community to be send with the SNMP Trap is entered.

Connection: Here, we have to select the source connection of our events. Here we choose „root@pulsar/PSAccount“ which we defined before.

The other fields can be kept empty or with their default entries.

At last we save our configuration by pressing the button **Save**.

A Requirements

The Event Gateway is a part of the OpenScape FM installation.

The necessary system requirements can be found in the *OpenScape FM Desktop User Guide*.

B SNMP Trap MIB Definition

```
MATERNA-MIB { iso org(3) dod(6) internet(1) private(4) enterprises(1) 192 }

DEFINITIONS ::= BEGIN

IMPORTS
    DisplayString
        FROM RFC1213-MIB
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Counter32, Integer32 ,IpAddress
        FROM SNMPv2-SMI
    OBJECT-GROUP
        FROM SNMPv2-CONF;
--    TruthValue
--        FROM SNMPv2-TC;
--    TRAP-TYPE
--        FROM RFC1215

manager MODULE-IDENTITY
    LAST-UPDATED    "0010061600+0100"
    -- last updated by glodde
    -- changes made:
    --  o definition of trap format for snmp forwarding

    ORGANIZATION "Materna GmbH Information and Communications BUI/NSM"
    CONTACT-INFO
        "Georg Lodde
        Vosskuhle 37
        44147 Dortmund
        Tel.:   +49 231 5599 203
        Fax.:   +49 231 5599 100
        E-mail:   Georg.Lodde@Materna.De"
    DESCRIPTION
        "This is the MIB module for the mmp gateway trap forwarder.
         It describes the SNMP trap format which is send to a nother ip system

         (trap receiver)."
    ::= { nsm 1 }

materna                 OBJECT IDENTIFIER ::= { enterprises 192 }
maternaProducts         OBJECT IDENTIFIER ::= { materna 1 }
maternaSystem           OBJECT IDENTIFIER ::= { materna 2 }
maternaMibs             OBJECT IDENTIFIER ::= { materna 3 }
maternaExperimental     OBJECT IDENTIFIER ::= { materna 4 }
maternaUnits            OBJECT IDENTIFIER ::= { materna 6 }
bui                     OBJECT IDENTIFIER ::= { maternaUnits 3 }
nsm                     OBJECT IDENTIFIER ::= { bui 1 }
ipm                     OBJECT IDENTIFIER ::= { manager 1 }
application             OBJECT IDENTIFIER ::= { manager 2 }
ers                     OBJECT IDENTIFIER ::= { manager 3 }
mmpgw                   OBJECT IDENTIFIER ::= { manager 4 }
```

```
-- this mib consists of the following subtrees/groups
mmpgwEventGroup        OBJECT IDENTIFIER ::= { mmpgw 1 }
mmpgwEventVars        OBJECT IDENTIFIER ::= { mmpgwEventGroup 1 }
mmpgwEventOIDs        OBJECT IDENTIFIER ::= { mmpgwEventGroup 2 }

-- The mmp gw event variable Group
-- There is no SNMP agent implementing the variables in a MIB.
--


ipaddress OBJECT-TYPE
    SYNTAX    IpAddress
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "This field contains an IP-Address:
             In case of a HiPath 4000 event (e.g. alarm), this will be the
         IP-Address of the HiPath 4000 Manager.
         In case of a HiPath 3000 event (forwarded HiPath 3000 trap), this
         will the the IP-Address of the event originating HiPath 3000
         system.
         If the event is an internal event, generated by the OpenScape FM,
         this field will contain the Ip-Address of the OpenScape FM server."
    ::= { mmpgwEventVars 1 }

severity OBJECT-TYPE
    SYNTAX    INTEGER {  unknown(1), normal (2), warning (3), minor (4),
              major(5), critical(6), unmanaged (7), restricted(8),
              testing(9), disabled(10)}
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "The severity assigned to the event. "
    ::= { mmpgwEventVars 2 }

objectname OBJECT-TYPE
    SYNTAX    DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "The name of the object in the OpenScape FM database to which
         the event is assigned.
         A HiPath 4000 object name is set up as follows:
         Hicom300:<IpAddressHiPath4000Manager>:<MnemonicHiPath4000>.
         MnemonicHiPath4000 represents the mnemonic name of the related
         HiPath 4000 switch.
         A HiPath 3000 object name has the structure:
         H150e:<IpAddressHipath3000>.
         IpAddressHipath3000 represents the IP-Address of the event
         originating HiPath 3000 /Hicom 150 switch.
    ::= { mmpgwEventVars 3 }

date OBJECT-TYPE
    SYNTAX    DateAndTime
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "The date when the event was received by the OpenScape FM."
    ::= { mmpgwEventVars 4 }

category OBJECT-TYPE
    SYNTAX    DisplayString
    MAX-ACCESS not-accessible
    STATUS    current
    DESCRIPTION
        "The event category to which the event is assinged.
         Exampels: HiPath 3000/5000, HiPath 4000, IP Manager,
         HiPath MIB HIA, HiPath MIB HMH, HiPath MIB HMP.
         Traps of Hicom 150 systems are assigned to the category HiPath
         3000/5000 too.
         Traps of Hicom 300 systems are assigned to the category HiPath
         4000."
    ::= { mmpgwEventVars 5 }

message OBJECT-TYPE
    SYNTAX    DisplayString
    MAX-ACCESS not-accessible
    STATUS    current
    DESCRIPTION
        "The message describing the event as assigned by the OpenScape FM
        (see OpenScape FM event browser, field Description)."
    ::= { mmpgwEventVars 6 }

eventType OBJECT-TYPE
    SYNTAX    INTEGER {  internal(2), snmp (3)) }
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "The type of event. If the event was generated due to the reception
        of an SNMP trap the eventType will be snmp. If the event was
        generated internal in the OpenScape FM, the event type will be
        internal."
    ::= { mmpgwEventVars 7 }
-- Events

sourceIp OBJECT-TYPE
    SYNTAX    IpAddress
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "The ip address of the source from which the event was send originally. "
    ::= { mmpgwEventVars 8 }

sourceDate OBJECT-TYPE
    SYNTAX    DateAndTime
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "The date which was assigned to the event by the original source. "
    ::= { mmpgwEventVars 9 }

eventKey OBJECT-TYPE
```

```
    SYNTAX    DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "A string to identify inverse events. "
    ::= { mmpgwEventVars 10 }

enterprise OBJECT-TYPE
    SYNTAX    OBJECT IDENTIFIER
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "A enterprise id assigned to the event. "
    ::= { mmpgwEventVars 11 }

eventId OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwEventVars 12 }

additionalVar1 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 1 }

additionalVar2 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 2 }

additionalVar3 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 3 }

additionalVar4 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 4 }

additionalVar5 OBJECT-TYPE
```

```
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 5 }

additionalVar6 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 6 }

additionalVar7 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 7 }

additionalVar8 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 8 }

additionalVar9 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 9 }

additionalVar10 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 10 }

additionalVar11 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 11 }

additionalVar12 OBJECT-TYPE
```

```
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 12 }

additionalVar13 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 13 }

additionalVar14 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 14 }

additionalVar15 OBJECT-TYPE
    SYNTAX    INTEGER
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 15 }

additionalVar16 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 16 }

additionalVar17 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 17 }

additionalVar18 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 18 }

additionalVar19 OBJECT-TYPE
```

```
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 19 }

additionalVar20 OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS    not-accessible
    STATUS    current
    DESCRIPTION
        "An additional optional event variable. "
    ::= { mmpgwAdditionalVars 20 }

-- Events

mmpgwEventUnknown NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with unknown severity as
             represented in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 1 }

mmpgwEventNormal NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with normal severity as
             represented in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 2 }

mmpgwEventWarning NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with warning severity as
             represented in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 3 }

mmpgwEventMinor NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with minor severity as represented
             in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 4 }

mmpgwEventMajor NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with major severity as represented
              in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 5 }

mmpgwEventCritical NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with critical severity as
              represented in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 6 }


mmpgwEventUnmanaged NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with unmanaged severity as
              represented in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 7 }

mmpgwEventRestricted NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with restricted severity as
              represented in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 8 }

mmpgwEventTesting NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with testing severity as
              represented in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 9 }

mmpgwEventDisabled NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with disabled severity as
              represented in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 10 }

mmpgwEventUnset NOTIFICATION-TYPE
    OBJECTS { ipaddress, severity, objectname, date, category, message, eventType,
    sourceIp, sourceDate, eventKey, enterprise, eventId }
    STATUS  current
    DESCRIPTION
             "This trap is generated to forward an event with unset severity as represented
              in the OpenScape FM event browser, to a trap receiver."
    ::=  { mmpgwEventOIDs 0 }

mmpgwLicenseViolation NOTIFICATION-TYPE
    OBJECTS { message }
    STATUS  current
    DESCRIPTION
             "This trap is generated when the license of the MATERNA Event Gateway is
             violated."
    ::=  { mmpgwEventOIDs 100 }
END
```

C Variables

The following variables may be used within the panel Filter and Trap Parameters. Either as part of the output value or as part of a query or condition.

C.1 Basic Variables

This variable may always be used:

- `${enterpriseid}`: The enterprise id connected to the event.
- `${date}`: The date of the event given in a string format. The local time zone is used for this.
- `${time}`: The time of the event given in a string format. The local time zone is used for this.
- `${datetimeseconds}`: The time when the event occurred. The time is given in seconds since midnight of January 1st, 1970.
- `${datetimemillis}`: The time when the event occurred. The time is given in milli seconds since midnight of January 1st, 1970.
- `${currentTime}`: The current time in milli seconds since 1.1.1970.
- `${WEEK}`: A constant for a week in milli seconds.
- `${DAY}`: A constant for a day in milli seconds.
- `${HOUR}`: A constant for an hour in milli seconds.
- `${MINUTE}`: A constant for a minute in milli seconds.
- `${SECOND}`: A constant for a second in milli seconds.
- `${category}`: The event category as displayed in the OpenScape FM Event browser.
- `${severity}`: The severity of the event (either ’minor’, ’major’ or ’device’)
- `${severityinteger}`: The severity of the event (either 0 (minor), 1 (major) or 2 (device)
- `${eventid}`: The unique id of the event.
- `${acknowledged}`: This variable returns whether the object within the OpenScape FM connected to the event is acknowledged or not. The value is either ’true’ or ’false’.
- `${flashing}`: This variable is ’true’ if the event causes that the object within the OpenScape FM connected to the event flashes. Else it is ’false’.
- `${source}`: The source name of the object to which the event is connected.
- `${description}`: A description of the event (Event Browser text).
- `${label}`: The label of the object to which the event is related to.

- `${eventkey}`: A key which is used to identify inverse events. This key differs in dependency of the technology and is not offered for all technologies.
- `${requestid}`: The AR ticket number of the last ticket used for an event with the same Request Id Key.
- `${sourceip}`: The IP address of the source which triggered the event (e.g. the IP address of a manager).
- `${serverIp}`: The IP address of the OpenScape FM that provided the event.
- `${var.length}`: The number of trap variables contained within the event trap.
- `${var[0]}` ... `${var[9]}`: The content of trap variables from within the event trap. Here the variables 0 .. 9 are offered, if the event trap has more then 10 variables, you are free to type them manually.

C.2 Enterprises

- `${HiPath4000MajorAlarmOn}`: The enterprise Id of an HiPath 4000 Major Alarm On trap.
- `${HiPath4000MajorAlarmOff}`: The enterprise Id of an HiPath 4000 Major Alarm Off trap.
- `${HiPath4000MinorAlarmOn}`: The enterprise Id of an HiPath 4000 Minor Alarm On trap.
- `${HiPath4000MinorAlarmOff}`: The enterprise Id of an HiPath 4000 Minor Alarm Off trap.
- `${HiPath4000DeviceAlarmOn}`: The enterprise Id of an HiPath 4000 Device Alarm On trap.
- `${HiPath4000DeviceAlarmOff}`: The enterprise Id of an HiPath 4000 Device Alarm Off trap.
- `${HiPath3000Trap}`: The enterprise Id of an HiPath 3000 trap.
- `${OpenScapeVoiceOIDTreePrefix}`: The prefix of the OpenScape Voice OID tree (1.3.6.1.4.1.4329.2.18.2.2.1.47).
- `${HiPathMIBHiPathHostUp}`: The enterprise Id of an HiPath MIB Host Up trap.
- `${HiPathMIBHiPathHostDown}`: The enterprise Id of an HiPath MIB Host Down trap.
- `${HiPathMIBHiPathHostStatusChange}`: The enterprise Id of an HiPath MIB Host Status Change trap.
- `${HiPathMIBHiPathApplicationInstalled}`: The enterprise Id of an HiPath MIB Application Installed trap.
- `${HiPathMIBHiPathApplicationDeinstalled}`: The enterprise Id of an HiPath MIB Application Dienstalled trap.
- `${HiPathMIBHiPathApplicationStatusChange}`: The enterprise Id of an HiPath MIB Application Status Change trap.
- `${HiPathMIBHiPathProcessUp}`: The enterprise Id of an HiPath MIB Process Up trap.
- `${HiPathMIBHiPathProcessDown}`: The enterprise Id of an HiPath MIB Process Down trap.
- `${HiPathMIBHiPathProcessStatusChange}`: The enterprise Id of an HiPath MIB Process Status Change trap.

For the exact enterprise id see the file `enterprises.properties` in the installation directory of the OpenScape Event Gateway server: `<Event Gateway installation directory>/server/resources/macros.`

D Glossary

API: Application programming interface, a set of functions that provides application programmers with access to the functionality of a product.

GUI: short for graphical user interface, allows users to navigate and interact with information on their computer screen by using a mouse to point, click and drag icons and other data around on the screen, instead of typing in words and phrases.

OpenScape FM: OpenScape Fault Management, the Unify corporate umbrella management platform for the HiPath Enterprise Convergence Architecture. It is a web-based client/server solution for the management of integrated networks with PBX, IP and VoIP devices.

JVM: Java Virtual Machine, the operating system-specific software which is required on a computer in order to run Java programs. The JVM translates (universal) Java code into machine specific code.

Event Gateway Config Tool: The Event Gateway Config Tool Configuration Tool, is used to configure the Event Gateway.

ECE: Event Correlation Engine, add-on module for OpenScape FM which is used for event correlation.

Index