A MITEL
PRODUCT
GUIDE

# Unify OpenScape Fault Management

## Unify OpenScape Fault Management V13, HiPath/OpenScape MIB Plugin

User Guide

09/2023

A31003-F30D0-U115-01-7619

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 of this guide
- audience of this guide
- terminology
- organization of this guide
- conventions used in this manual

## 1.1 Purpose

This manual gives an introduction into the work with the HiPath/OpenScape MIB Plugin module for the OpenScape FM.

## 1.2 Audience

This guide is addressed to users who want to learn how to use the HiPath/OpenScape MIB Plugin. Since this is not a standalone application but a Plugin for the OpenScape FM, you should know how to work with this program before using the plugin. For several functionalities of the plugin which are based on OpenScape FM features, we will base our explanations on those given in the Desktop Manual, so please read the *OpenScape FM Desktop User Guide* for a comprehensive introduction into the work with OpenScape FM.

## 1.3 Terminology

- **OpenScape FM** means OpenScape Fault Management
- **Server** means the OpenScape Server, i.e. the server where the OpenScape Desktop with the HiPath/ OpenScape MIB 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 (**OpenScape DT**).
- **CAP** means Common Application Platform, a Unify standard for the management of HiPath related applications.
- **MIB** means Management Information Base, a standard data structure format for access to SNMP data
- **HiPath MIB** is a MIB which provides information about HiPath-enabled applications on a PC, see *Section 2.1, “HiPath MIB”*.

## 1.4 Organization of This Guide

This guide is organized as follows:

- *Chapter 1, "Preface"* explains the structure of the manual.
- *Chapter 2, "Introduction"* provides a brief overview over the components of a HiPath MIB or Shared Services agent and explains what the HiPath/OpenScape MIB Plugin is.
- *Chapter 3, "Getting started"* explains how to initialize the HiPath/OpenScape MIB Plugin.
- *Chapter 4, "Work With the HiPath/OpenScape MIB Plugin"* explains how to add HiPath MIB agents to the system and describes the different HiPath/OpenScape MIB Plugin objects.
- *Chapter 5, "Fault Monitoring"* describes the handling of SNMP Traps from a HiPath MIB device in detail.

## 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** or **OK**.

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 Unity A&S documentation.*

Italic type is also used for emphasis.

Example: *All* users will be affected.

# 2 Introduction

The HiPath/OpenScape MIB Plugin is a plugin module for the OpenScape FM. With this plugin, OpenScape FM can auto discover devices with a HiPath or Shared Services MIB. It displays the applications and processes that are monitored by the HiPath MIB Agent.

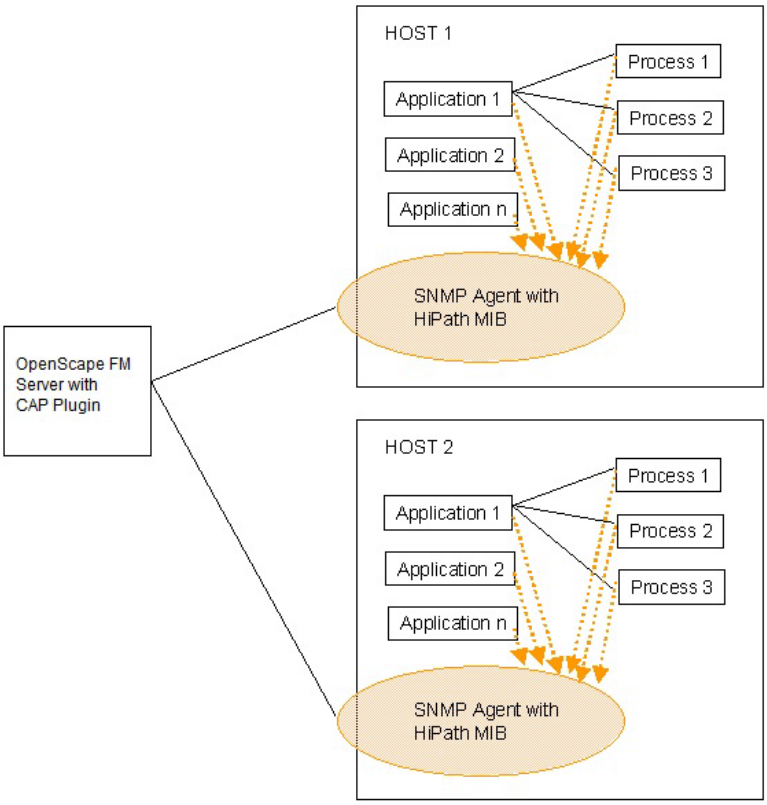



*Figure 1 HiPath MIB environment*

OpenScape FM displays MIB-enabled devices with their applications and processes on the OpenScape FM user interface and it enables the monitoring of the fault status of the managed devices, applications and processes. Shared Services MIBs are also discovered.

## 2.1 HiPath MIB

The HiPath MIB lets you monitor applications and processes on HiPath-enabled devices, e.g. PCs with CAP, in your network.

With HiPath CAP (Common Application Platform), HiPath-enabled devices, applications and processes can be integrated into an OpenScape FM environment. Thus they can be monitored concerning their state.

Applications which use the CAP-Lib, a Windows library, can provide CAP-relevant information, e. g. if a program is running or not or if a process is up or down. That information is CAP-specific but is also accessible via HiPath MIB which is the interface to management applications. To this purpose, an SNMP agent which manages the

HiPath MIB provides information about devices, applications and processes which is retrieved by the help of the HiPath MIB (see *Figure 1*).
We speak of “HiPath-managed devices”, “HiPath-managed applications” and “HiPath-managed processes”.

Thus, when you have installed a HiPath MIB SNMP agent, e.g. the CAP agent, on your PC, you can monitor the HiPath-managed applications and processes on this PC.

## 2.2 Shared Services MIB

Like the HiPath MIB the Shared Services MIB is also an SNMP MIB. It will be used for the centralized monitoring of applications that run decentralized on a number of systems. With the help of the HiPath/OpenScape MIB Plugin the monitoring can be done within the OpenScape FM.

The **Symphonia Framework** is a Unify product, which offers the functionality to provide basis infra structure functions to HiPath applications. The Symphonia Framework is a specific representation of the Shared Services MIB. All descriptions within this User Guide concerning Shares Services will apply for the Symphonia Framework.

Service orientation is one of the fundamental architectural principles within the Framework architecture. A **Service** encapsulates self-contained business functionality and offers it to its clients via one or more role-specific interfaces.

Services are the main building blocks of the Framework and Framework-based applications.

**Applications** offer and use a set of services. An application is a program or group of programs designed for end users. Application software may perform a specific task or function, such as word-processing, creation of spreadsheets or facilitating electronic mail.

Services are hosted by a **Container**. The services delegate their lifecycle/lifetime management to the container. This "inversion of control" allows a separation of concerns where the container can take care of cross cutting and technical concerns, such as resource management, logging, etc. whereas the components can focus on the domain-specific functional concerns.

Within the Framework **Nodes** represent physical machines hosting one or more containers.

# 3 Getting started

## 3.1 Initializing the HiPath/OpenScape MIB Plugin

To be able to work with the HiPath/OpenScape MIB Plugin, the OpenScape FM has to be installed on the system. More about the installation process can be found within the *Appendix Installation* in the *OpenScape FM Desktop User Guide*.

In order to initialize the HiPath/OpenScape MIB Plugin you have to start the OpenScape Client and select the menu item **Server**–>**Plugins**–>**Init HiPath/OpenScape MIB Plugin** from the main menu bar. After the initialization the menu item **Init HiPath/OpenScape MIB Plugin** will be removed from the menu **Server**–>**Plugins**. A new menu item **SNMP->HiPath MIB** will appear in the main menu bar. Additionally an object representing the HiPath/OpenScape MIB Plugin is added to the hierarchy with the path **Root**–>**System**–>**Plugins**. The new object offers the same menu items like the HiPath MIB main menu.

After the initialization HiPath MIB capable or Shared Services MIB capable devices can be added to the system. These devices will be displayed as ip nodes, to which an HiPath MIB agent or a Shared Services Server is added as an submap object.

## 3.2 Licensing

To work with the HiPath/OpenScape MIB 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.

# 4 Work With the HiPath/OpenScape MIB Plugin

## 4.1 Adding MIB-Capable Devices

You can include several HiPath MIB-enabled devices in your system – we will show you an example with only one HiPath MIB device here to make it easier to understand the context.

### 4.1.1 Adding HiPath MIB-Enabled Devices

In order to monitor the applications on a HiPath MIB-enabled device, e.g. on a PC, you have to add this PC to the OpenScape FM, i.e. you have to add the corresponding IP node by means of the IP Manager. Remember that OpenScape FM management is based on the detection of IP nodes in the LAN. Please read the *IP Manager Plugin User Guide* for a detailed introduction to the OpenScape IP Manager.

### 4.1.2 The Discovery Process For HiPath MIBs

**initial auto-discovery:**

When you have added the HiPath MIB-enabled device to the OpenScape FM, the discovery process starts. The new IP node is registered as such, i.e. an IP node object is created in OpenScape FM. In the next step the MIBs on this device are read: a MIBII and the HiPath MIB will be found. The HiPath MIB is read, and a new HiPath MIB agent object is created – as child object of the IP node. For each application which has registered itself with the HiPath MIB, an application object as child object of the HiPath MIB agent will be created. For each registered process of an application, a process object – as child object of the corresponding application object - is created. For each IP node with a HiPath MIB agent the structure of the object hierarchy will look like the one represented in *Figure 2*.

By Default, all devices with HiPath MIB agents will be marked with the OpenScape FM network Id “HiPath MIB Nodes”. Therefore, unless any network/subnetwork configurations has been done via the IP Discovery Filters, after the initial discovery, all HiPath MIB -enabled devices will be grouped together in the network container named “HiPath MIB Nodes”.

During the discovery of the HiPath MIB IP node, OpenScape FM registers itself as trap destination automatically. Thus, the OpenScape FM server is ready to receive traps from the HiPath MIB device without any further manual configuration.The HiPath MIB traps will be treated in *Chapter 5, “Fault Monitoring”*.

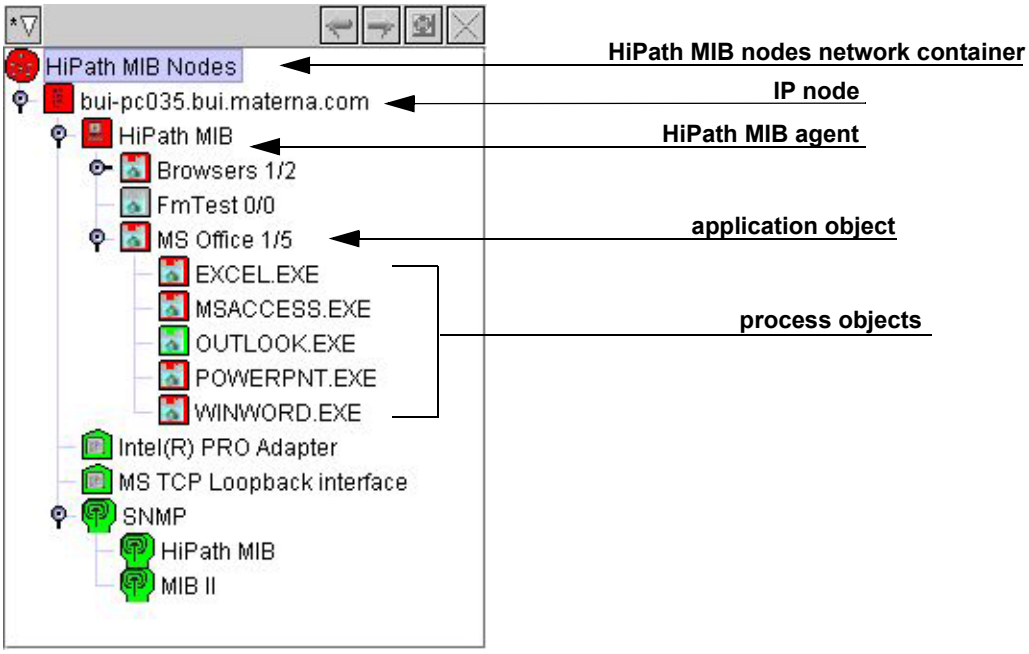



*Figure 2 HiPath MIB IP node object hierarchy*

Every object is represented on the submap and/or in the Tree View Area by its characteristic icon, see *Section 4.2, "HiPath MIB Objects: Symbols and Functionalities"*. The icon for each object is defined by the object-specific icon-URL which is also a field in the HiPath MIB.

**auto-discovery during operation:**

Whenever trap-relevant data (compare *Section 5.3, "Traps"*) has been changed in the HiPath MIB, the SNMP agent on the HiPath MIB device sends an SNMP trap to the OpenScape FM server. Depending on the type of the trap, this might cause a new information retrieval from the HiPath MIB device and the update of the corresponding submaps/trees.

## 4.1.3 The Discovery Process For Shared Services Instances

The discovery process for Shared Services Instances (Shred Services MIBs) uses the same mechanisms as the ones described above for HiPath MIBs.

If an SNMP-MIB is discovered for an added IP node, it is checked whether an OID can be found, that matches one indicating a Shared Services MIB. If this is the case, an Server Symbol will automatically be located below the respective ip node. A respective Instance Symbol will be created below the topology symbol.

The distribution and structure of the connected objects (Nodes, Services, ...) is defined by the object information found within the MIB.

## 4.2 HiPath MIB Objects: Symbols and Functionalities

### 4.2.1 HiPath MIB Agent


*Figure 3 Symbol for HiPath MIB Agents*

The HiPath MIB agent object is a child object of the IP node where the HiPath MIB agent (i.e. SNMP agent with HiPath MIB) has been found, i.e. this icon will be found on the submap of suchlike an IP node.

Its context menu provides access to several Info Browsers which will be discussed in the next paragraph.

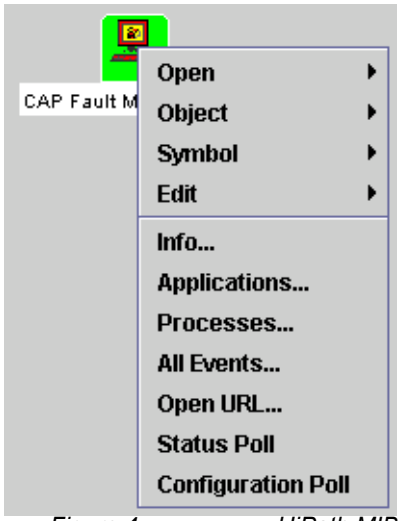



*Figure 4 HiPath MIB agent context menu*

#### 4.2.1.1 Info

**Info...:** opens an Info Browser which lists general information about this HiPath MIB agent (*Figure 5*).

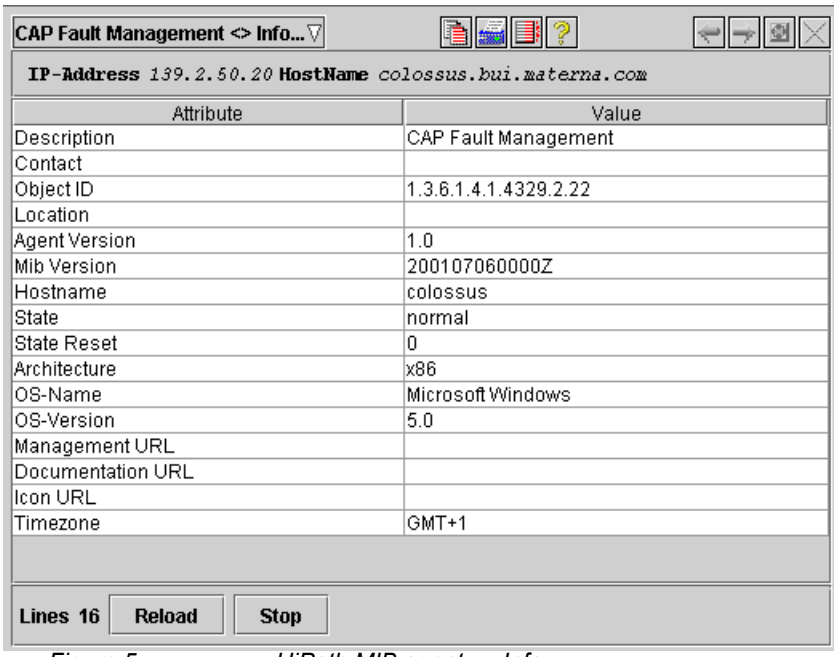



*Figure 5 HiPath MIB agent –> Info...*

**Description:** A (short) description of the component.

**Contact:** The responsible administrator.

**Object ID:** The authoritative identification of this component/device.

**Location:** The location of the component. If sysLocation starts with 'HiPath:' the remainder of the string is interpreted as /network//subnetwork. E.g. 'HiPath:/europe/germany//region south/munich'. This entry is used by OpenScape FM to place the object on the related submap. Correspondents with the sysLocation entry from SNMP MIB-2.

**Agent Version:** The version of the SNMP agent.

**Mib Version:** The version of the HiPath MIB.

**Hostname:** The hostname of the host the HiPath MIB agent is installed on. It is included in the HiPath MIB to be independent from the MIB II.

**State:** The current state of the component. The state could have the following values: unknown, normal, warning, minor, major, critical, unmanaged, restricted, testing, disabled.

**State Reset:** If this variable is written, the state is reset to 'normal'.

**Architecture:** The architecture of the host, e.g. x86, sunUltra, ...

**OS-Name:** The name of the hosts operating system, e.g. 'Windows 2000'.

**OS-Version:** The version of the hosts operating system, e.g '5.0' for Windows 2000.

**Management URL:** This URL points to a page which gives access to the element manager of the HiPath-enabled device.

**Documentation URL:** This URL points to related documentation.

**Icon URL:** This URL is the prefix of the URLs, which contain the bitmap and shape description to be used to display the component on a desktop map.

**Timezone:** The time zone where the system is located.

### 4.2.1.2 Applications

**Applications...:** opens an Info Browser which lists all registered applications (*Figure 6*).

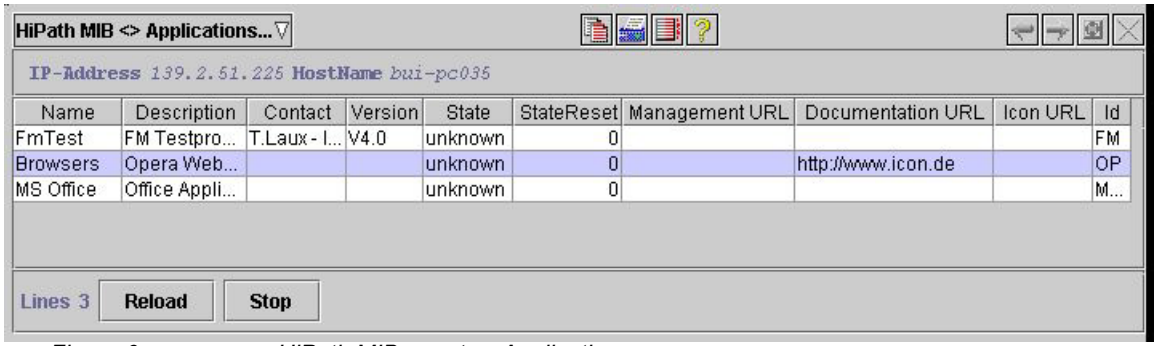



*Figure 6 HiPath MIB agent → Applications...*

**Name:** The application name.

**Description:** A (short) description of the application.

**Contact:** Information about the responsible administrator, e.g. name and phone.

**Version:** The version of the installed application.

**State:** The current state of the application. The state could have the following values: unknown, normal, warning, minor, major, critical, unmanaged, restricted, testing, disabled.

**State Reset:** If this variable is written, the state is reset to ‘normal’.

**Management URL:** This URL points to a page which gives access to the applications element manager if available.

**Documentation URL:** This URL points to the online documentation for this application.

**Icon URL:** This URL is the prefix of the URLs, which contain the bitmap and shape description to be used to display the application on a desktop map.

**Id**: [Only available with a MIB version greater than 200107060000 Z]: Technical Id of the application.

### 4.2.1.3 Processes

**Processes...:** opens an Info Browser which lists all registered processes.

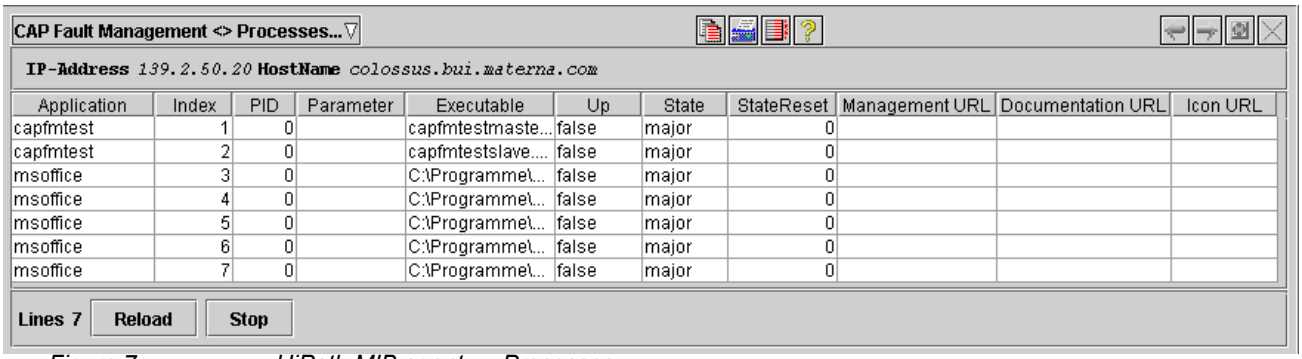



*Figure 7 HiPath MIB agent –> Processes...*

**Application:** The application the process belongs to.

**Index:** An unique index.

**PID:** The PID assigned to process by the operating system. It is 0 if the process is down.

**Parameter:** The (command line) parameters of the process.

**Executable:** Index of the executable in the appExecTable.

**Up:** True if the process is running, false if the process is configured but not running. If false there should be always a failure state on this process.

**State:** The current state of the process. The state could have the following values: unknown, normal, warning, minor, major, critical, unmanaged, restricted, testing, disabled.

**State Reset:** If this variable is written, the state is reset to ‘normal’.

**Management URL:** This URL points to a page which gives access to the processes element manager if available.

**Documentation URL:** This URL points to the online documentation for this process.

**Icon URL:** This URL is the prefix of the URLs, which contain the bitmap and shape description to be used to display the process on a desktop map.

### 4.2.1.4 All Events

**All Events...:** opens an Info Browser which lists all registered events.

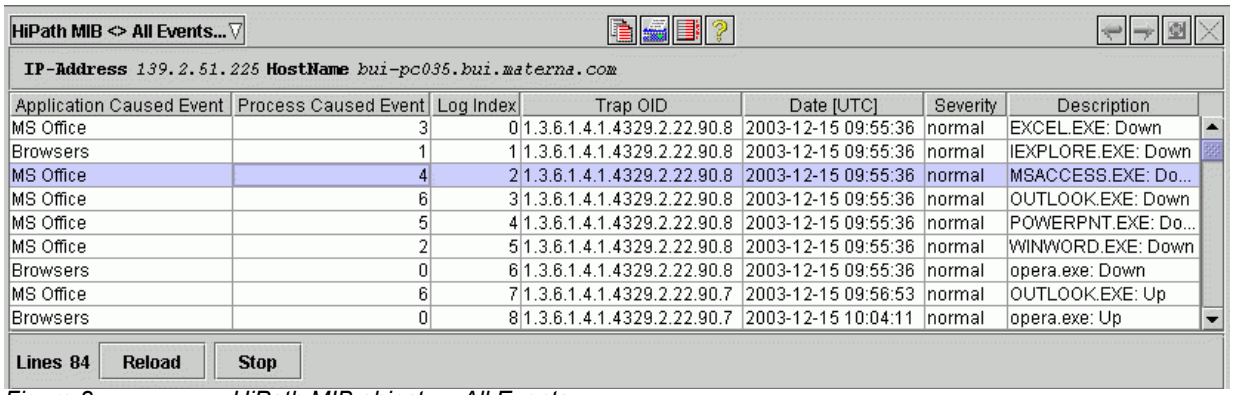



*Figure 8 HiPath MIB object –> All Events*

**Application Caused Event:** The application this event was caused by. It is empty if event occured on host/ component level.

**Process Caused Event:** The process this event was caused by. It is 0 if the event was not caused by a process. Events which are not caused by permanent processes are mapped to the application.

**Log Index:** The sequence of indexes increases in temporary order.

**Trap OID:** The OID of the trap that was sent for this event, e.g. if a hiPathHostDown trap was sent, the entry for that trap should be ‘1.3.6.1.4.1.4329.2.22.90.2’.

**Date [UTC]:** The time this event occured in UTC.

**Severity:** The severity of this event.

**Description:** A textual, human readable description of the event.

### 4.2.1.5 Open URL

**Open URL...:** opens the management URL (see *Section 4.2.1.1, “Info”*) of this HiPath MIB device.

### 4.2.1.6 Status and Configuration Poll

**Status Poll** is a standard OpenScape FM IP Manager menu item. Please refer to the *IP Manager Plugin User Guide* for detailed explanation.

**Configuration Poll:** is a standard OpenScape FM IP Manager menu item. Please refer to the *IP Manager Plugin User Guide* for detailed explanation.

## 4.2.2 Application


*Figure 9 Symbol for Applications*

The application object icon represents one application on the HiPath MIB-enabled device (i.e. usually on a PC) which has registered itself with the HiPath MIB. Its label represents the application's name and the relation of running processes to all registered processes of the application.

Its context menu (*Figure 10*) provides access to several Info Browsers which will be explained in the following paragraph.


*Figure 10 Application object context menu*

### 4.2.2.1 Info

**Info...:** opens an Info Browser which lists general information about this application.

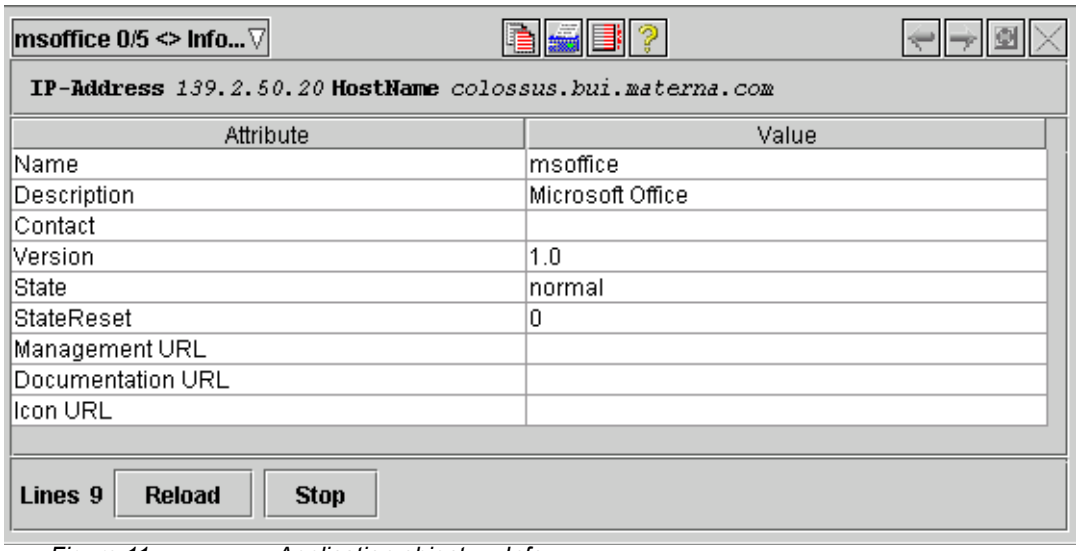



*Figure 11 Application object –> Info...*

**Name:** The application name.

**Description:** A (short) description of the application.

**Contact:** Information about the responsible administrator, e.g. name and phone.

**Version:** The version of the installed application.

**State:** The current state of the application. The state could have the following values: unknown, normal, warning, minor, major, critical, unmanaged, restricted, testing, disabled.

**State Reset:** If this variable is written, the state is reset to ‘normal’.

**Management URL:** This URL points to a page which gives access to the applications element manager if available.

**Documentation URL:** This URL points to the online documentation for this application.

**Icon URL:** This URL is the prefix of the URLs, which contain the bitmap and shape description to be used to display the application on a desktop map.

### 4.2.2.2 Processes

**Processes...:** opens an Info Browser which lists all registered processes of this application.

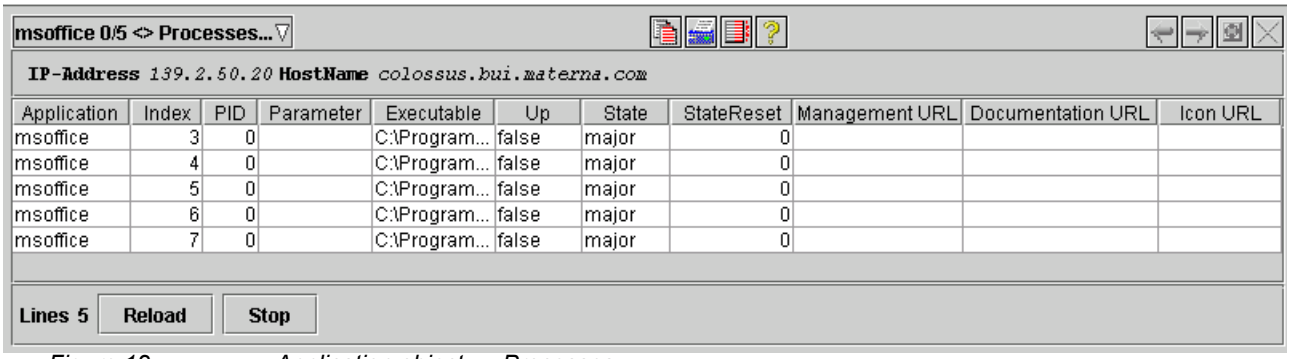



*Figure 12 Application object –> Processes*

**Application:** The application the process belongs to.

**Index:** An unique index.

**PID:** The PID assigned to process by the operating system. It is 0 if the process is down.

**Parameter:** The (command line) parameters of the process.

**Executable:** Index of the executable in the appExecTable.

**Up:** True if the process is running, false if the process is configured but not running. If false there should be always a failure state on this process.

**State:** The current state of the process. The state could have the following values: unknown, normal, warning, minor, major, critical, unmanaged, restricted, testing, disabled.

**State Reset:** If this variable is written, the state is reset to ‘normal’.

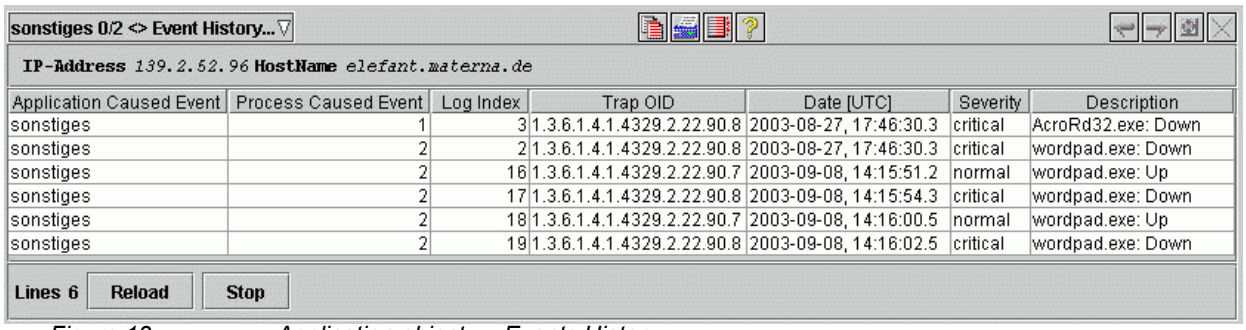
**Management URL:** This URL points to a page which gives access to the processes element manager if available.

**Documentation URL:** This URL points to the online documentation for this process.

**Icon URL:** This URL is the prefix of the URLs, which contain the bitmap and shape description to be used to display the process on a desktop map.

### 4.2.2.3 Event History

**Events History...:** opens an Info Browser which lists all registered events of this application.



sonstiges 0/2 <> Event History...

IP-Address 139.2.52.96 HostName elefant.materna.de

| Application Caused Event | Process Caused Event | Log Index | Trap OID | Date [UTC] | Severity | Description |
|---|---|---|---|---|---|---|
| sonstiges | 1 | 3 | 1.3.6.1.4.1.4329.2.22.90.8 | 2003-08-27, 17:46:30.3 | critical | AcroRd32.exe: Down |
| sonstiges | 2 | 2 | 1.3.6.1.4.1.4329.2.22.90.8 | 2003-08-27, 17:46:30.3 | critical | wordpad.exe: Down |
| sonstiges | 2 | 16 | 1.3.6.1.4.1.4329.2.22.90.7 | 2003-09-08, 14:15:51.2 | normal | wordpad.exe: Up |
| sonstiges | 2 | 17 | 1.3.6.1.4.1.4329.2.22.90.8 | 2003-09-08, 14:15:54.3 | critical | wordpad.exe: Down |
| sonstiges | 2 | 18 | 1.3.6.1.4.1.4329.2.22.90.7 | 2003-09-08, 14:16:00.5 | normal | wordpad.exe: Up |
| sonstiges | 2 | 19 | 1.3.6.1.4.1.4329.2.22.90.8 | 2003-09-08, 14:16:02.5 | critical | wordpad.exe: Down |

Lines 6 Reload Stop

*Figure 13 Application object –> Events History...*

**Application Caused Event:** The application this event was caused by. It is empty if event occured on host/ component level.

**Process Caused Event:** The process this event was caused by. It is 0 if the event was not caused by a process. Events which are not caused by permanent processes are mapped to the application.

**Log Index:** The sequence of indexes increases in temporary order.

**Tap OID:** The OID of the trap that was sent for this event, e.g. if a hiPathApplicationInstalled trap was sent, the entry for that trap should be ‘1.3.6.1.4.1.4329.2.22.90.4’.

**Date [UTC]:** The time this event occured in UTC.

**Severity:** The severity of this event.

**Description:** A textual, human readable description of the event.

### 4.2.2.4 Open URL

**Open URL...:** opens the management URL of the application (see *Section 4.2.2.1, “Info”*) in the standard web browser.

## 4.2.3 Process


*Figure 14 Symbol for Processes*

The process object icon represents one process which has registered itself with the HiPath MIB for a certain application. It is a child object of the corresponding application object and appears in its submaps and as child object in the tree views respectively.

Its context menu provides access to several Info Browsers which will be described in the following paragraph.

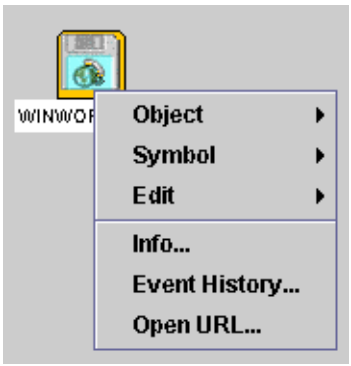



*Figure 15 Process object context menu*

### 4.2.3.1 Info

**Info...:** opens an Info Browser which lists general information about this process.

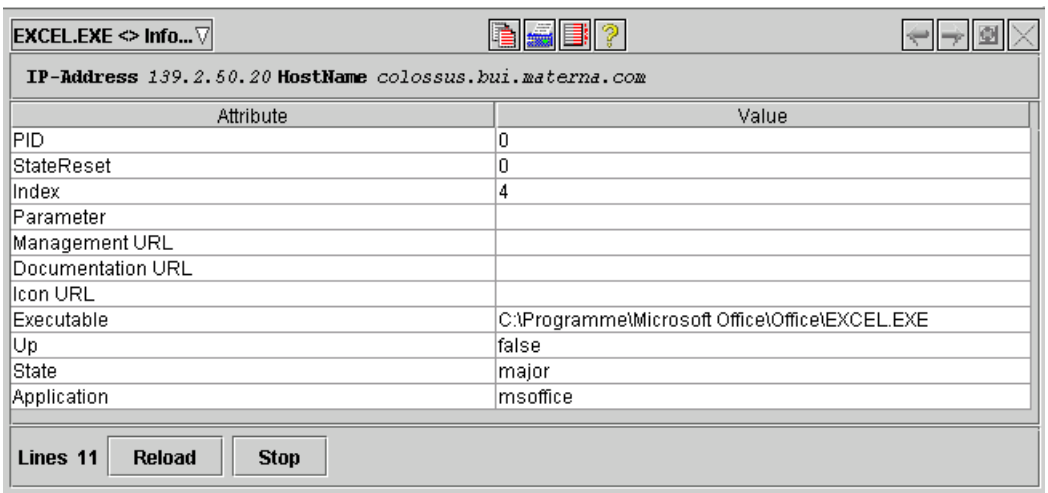



*Figure 16 Process object –> Info...*

**Application:** The application the process belongs to.

**Index:** An unique index.

**PID:** The PID assigned to process by the operating system. It is 0 if the process is down.

**Parameter:** The (command line) parameters of the process.

**Executable:** Index of the executable in the appExecTable.

**Up:** True if the process is running, false if the process is configured but not running. If false there should be always a failure state on this process.

**State:** The current state of the process. The state could have the following values: unknown, normal, warning, minor, major, critical, unmanaged, restricted, testing, disabled.

**State Reset:** If this variable is written, the state is reset to ‘normal’.

**Management URL:** This URL points to a page which gives access to the processes element manager if available.

**Documentation URL:** This URL points to the online documentation for this process.

**Icon URL:** This URL is the prefix of the URLs, which contain the bitmap and shape description to be used to display the process on a desktop map.

### 4.2.3.2 Event History

**Event History...:** opens an Info Browser which lists all registered events of this process.

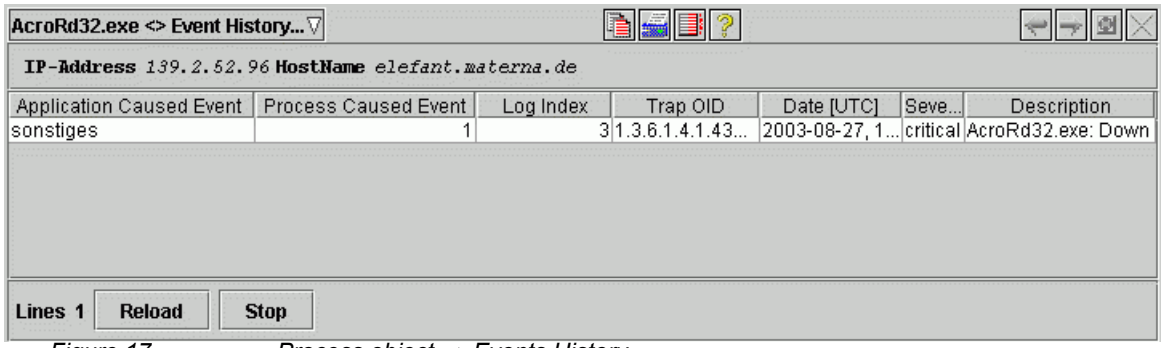



*Figure 17 Process object –> Events History...*

**Application Caused Event:** The application this event was caused by. It is empty if event occured on host/component level.

**Process Caused Event:** The process this event was caused by. Events which are not caused by permanent processes are mapped to the application.

**Log Index:** The sequence of indexes increases in temporary order.

**Trap OID:** The OID of the trap that was sent for this event, e.g. if a hiPathProcessUp trap was sent, the entry for that trap should be '1.3.6.1.4.1.4329.2.22.90.7'.

**Date [UTC]:** The time this event occured in UTC.

**Severity:** The severity of this event.

**Description:** A textual, human readable description of the event.

### 4.2.3.3 Open URL

**Open URL...:** opens the management URL of the process (see *Section 4.2.3.1, "Info"*) in the standard web browser.

## 4.3 Shared Services Objects: Symbols and Functionalities

The following chapters describe the object types, the symbols and the functionalities provided by them, which are used for Shared Services within the OpenScape FM.

There are six distinct types of objects, each being represented by a corresponding symbol and a connected functionality. The different types for Shared Services are: Instance, Node, Service, Container, Application and Server.

A **Shared Services Instance** (short Instance) is an actual customer installation of the Shared Services Framework.

An Instance contains a certain number of systems. These systems are called **Nodes**. The different components of an Instance may be divided to different Nodes.

One of the Nodes of an Instance is hosting the **Server**. Among other things the Server provides the SNMP agent for the Shared Services MIB. The Server is the central access and control component.

**Services** provide the functionalities. Using interfaces provided by the Services the functions provided by the Services can be used.

**Container** form a working environment for Services. Each Container contains a number of Services and each Container is attached to a single Node. Therefore each Service is also attached to exactly one Node during it's execution time.

**Note**:
Services may be stopped and attached to another Container. This new Container may even be connected to another Node

Besides the object classes Service, Container and Node the class **Application** exists. Applications agregate a number of Services.

All Shared Services objects are identified by **Global Unique Identifiers** (short **GUID**). These identifiers are unique within a single Instance.

The management application (OpenScape FM) can only identify the different objects by their object GUIDs. These IDs should therefore be permanently attached to the objects. If the GUID of an object changes, the management application will interpret this as an deletion and newly creation of the object. Manually configured settings (e. g. symbol positions within a submap) will then be lost.

The management application assumes that no changes for the GUIDs of Containers, Applications and Nodes will happen. GUIDs for Services stay constant as long as the Service is not transferred to a different Container.

Within an Instance only one **SNMP-Agent** (short **Agent**) may exist. Every Node belongs to exactly one Instance.

The following chapters describe the different object types for Shared Services within the OpenScape FM. In addition the functions will be described, that will be offered through the context menus of the corresponding symbol types.

## 4.3.1 Services


*Figure 18 Symbol for Shared Services 'Services'*

Services provide the different functionalities that are provided within the Shared Services Framrework.

Services will by a symbol like in *Figure 18* within the OpenScape FM. They may be present on the submap of Applications and Containers.

The status of a Services symbol will be acquired from the Shared Services MIB.

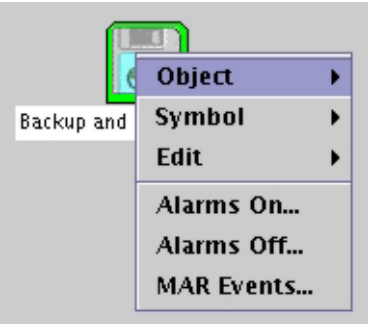



*Figure 19 Context menu for Shared Services 'Services'*

The different entries of the context menu for the object type 'Service' (*Figure 19*) have the following functionalities.

### 4.3.1.1 Alarms On

The menu entry **Alarms On...** opens an info browser (*Figure 20*). This browser contains a table which displays all alarms belonging to the respective Service, which yet have not been resetted. Every row of the table displays one such alarm.

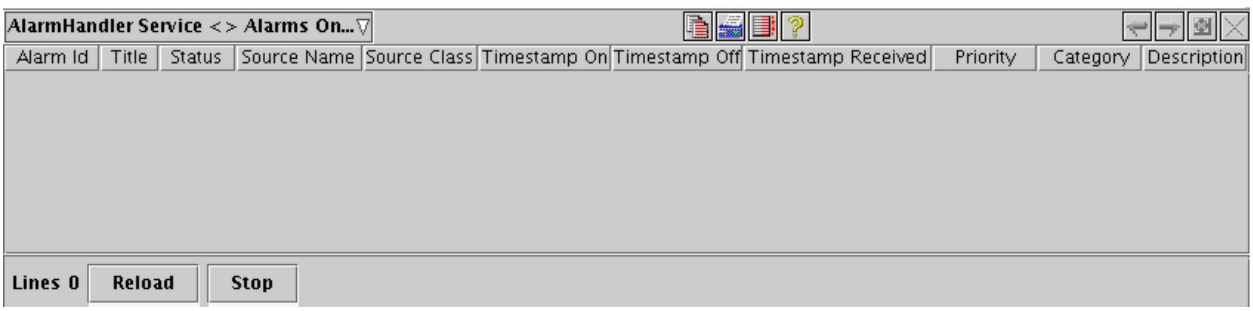



*Figure 20 Info browser for active alarms*

The table within the info browser contains the following columns:

**Alarm-Id:** The unique ID of the alarm.

**Title:** The name of the alarm.

**Status:** The status of the displayed alarm (’on’ or ’off’).

**Source Name:** The name of the source of the alarm.

**Source Class:** The class name of the source of the alarm.

**Timestamp On:** Date and time when the alarm event was generated.

**Timestamp Off:** Date and time when the alarm reset event was generated.

**Timestamp Received:** Date and time when the alarm event was received by the fault manager.

**Priority:** The priority of the alarm. Possible values are: ’normal’, ’information’, ’warning’, ’minor’, ’major’, ’critical’ and ’unknown’.

**Category:** The category of the alarm.

**Description:** A short description of the alarm type.

#### 4.3.1.2 Alarms Off

The menu entry **Alarms Off...** opens an info browser like the one in *Figure 20*. In this case the browser will only displays alarms, that have already been resetted.

#### 4.3.1.3 MAR Events

The menu entry **MAR Events...** opens up a MAR browser, which will display all event types defined for Services which may be connected to a defined reaction within the reaction system. If defined reaction exist, these reactions may be connected to the displayed events.

E. g. an reaction can be connected, that generates an EMail whenever the connected Service changes to the state ’critical’. More about the Reaction System can be found in the separate *User Guide for the Mobile Alarm Reaction System.*

#### 4.3.1.4 Documentation

The menu entry **Documentation** will be visible, when a documentation URL is defined for the respective Service within the MIB. This entry will open this URL within a browser.

#### 4.3.1.5 Configuration Management

The menu entry **Configuration Management** will be visible, when a configuration URL is defined for the respective Service within the MIB. This entry will open this URL within a browser.

## 4.3.2 Container


*Figure 21 Symbol for Shared Services ’Container’*

Within the Shared Services Framework Containers define a working environment for Services. They are always connected to a single Node.

Container will be represented by a symbol as in *Figure 21* within the OpenScape FM. They will be displayed on the submap of Nodes.

If the status of a Container within the Shared Services MIB is currently ’unknown’ or ’normal’, then the state of the Container symbol will be the aggregated state of the objects contained in the submap.

Else the status of the symbol will be set to the status of the Container within the Shared Services MIB.

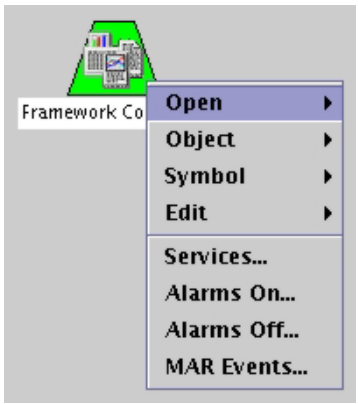



*Figure 22 Context menu for Shared Services ’Container’*

The different entries of the context menu for objects of the type ’Container’ (*Figure 22*) have the following functionalities:

### 4.3.2.1 Services

The menu entry **Services...** opens up an info browser (*Figure 23*). This browser contains a table that displays all Services that are contained within the Container. Each row within the table represents a single Service.

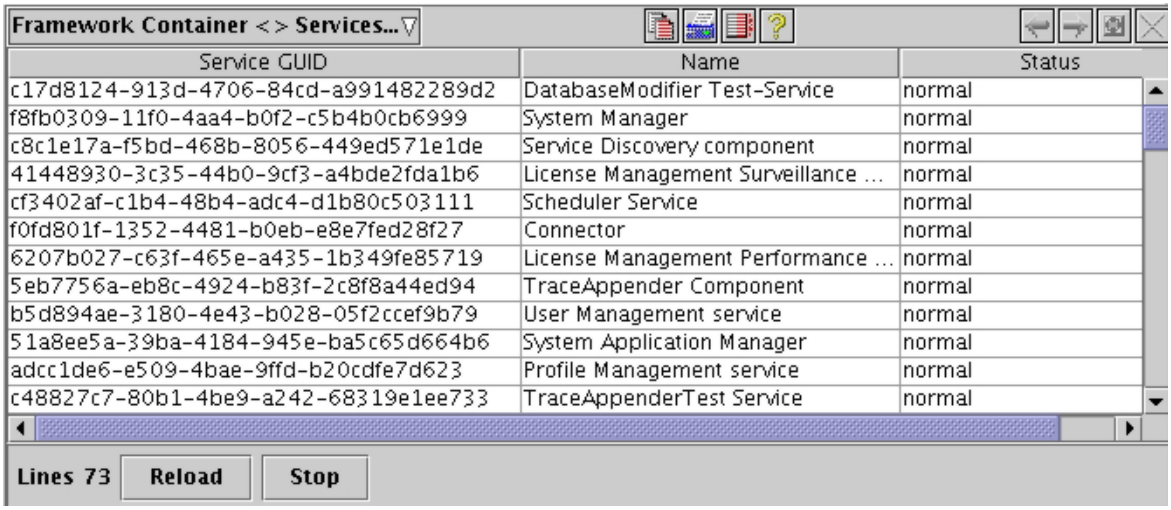



*Figure 23 Info browser for Services*

The table within the info browser contains the following columns:

**Service GUID:** The ID of the Service. The ID is unique within the Instance.

**Name:** The name of the Service.

**Status:** The current status of the Service.

### 4.3.2.2 Alarms On

The menu entry **Alarms On...** corresponds to the identical entry for Service objects described in *Section 4.3.1.1, "Alarms On"*. In this case the table lists the *non resetted* alarms for all Services that are located within the Container.

### 4.3.2.3 Alarms Off

The menu entry **Alarms Off...** corresponds to the identical entry for Service objects described in *Section 4.3.1.2, "Alarms Off"*. In this case the table lists the *resetted* alarms for all Services that are located within the Container.

### 4.3.2.4 MAR Events

The menu entry **MAR Events...** corresponds to the identical entry for Service objects described in *Section 4.3.1.3, "MAR Events"*. In this case reactions may be connected to status change events of the Container, or to status change events of the Services located within the Container.

#### 4.3.2.5 Documentation

The menu entry **Documentation** will be visible, when a documentation URL is defined for the respective Container within the MIB. This entry will open this URL within a browser.

#### 4.3.2.6 Configuration Management

The menu entry **Configuration Management** will be visible, when a configuration URL is defined for the respective Container within the MIB. This entry will open this URL within a browser.

### 4.3.3 Nodes


*Figure 24 Symbol for Shared Services 'Nodes'*

Within the Shared Services Framework, Nodes represent the different systems on which the components of a Shared Services Instance can be found.

Nodes will be displayed within the OpenScape FM by symbols as shown in *Figure 24*. They will be found on the submap of IP nodes.

If the status of a Node within the Shared Services MIB is currently 'unknown' or 'normal', then the status of the Node symbol will be the aggregated state of the objects contained in the submap.

Else the status of the symbol will be set to the status of the Node within the Shared Services MIB.

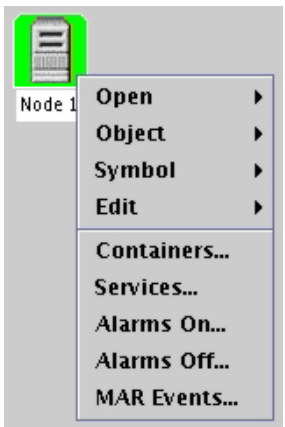



*Figure 25 Context menu for Shared Services 'Nodes'*

The different entries of the context menu for objects of the type 'Node' (*Figure 25*) have the following functionalities:

### 4.3.3.1 Containers

The menu entry **Containers...** opens up an info browser (*Figure 26*). This browser contains a table that displays all Containers that are contained within the Node. Each row within the table represents a single Container.

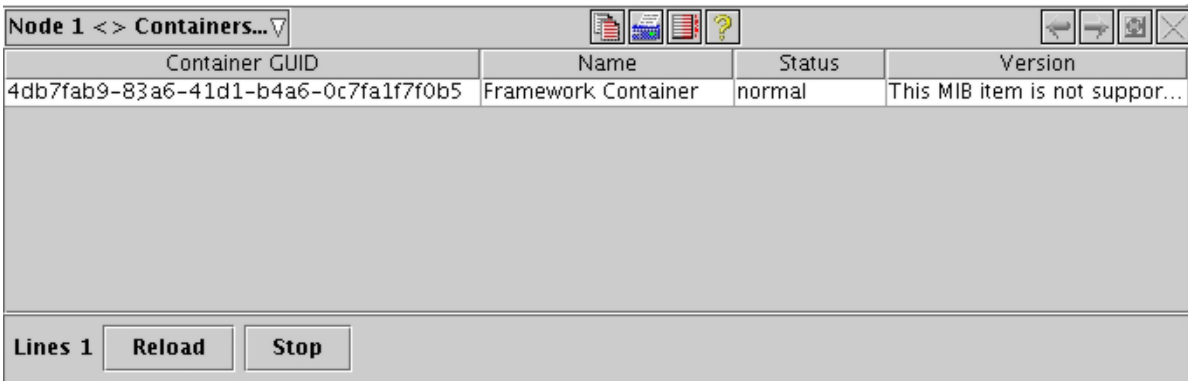



*Figure 26 Info browser for Container*

The table within the info browser contains the following columns:

**Container GUID:** The ID of the Container. Unique within the Instance

**Name:** The name of the Container.

**Status:** The current status of the Container.

**Version:** The version of the Container.

### 4.3.3.2 Services

The menu entry **Services...** corresponds to the identical entry for Container objects described in *Section 4.3.2.1, "Services"*. In this case the table lists the Services that belong to the Containers of the Node.

### 4.3.3.3 Alarms On

The menu entry **Alarms On...** corresponds to the identical entry for Service objects described in *Section 4.3.1.1, "Alarms On"*. In this case the table lists the *non resetted* alarms for all Services that are located within the Node.

### 4.3.3.4 Alarms Off

The menu entry **Alarms Off...** corresponds to the identical entry for Service objects described in *Section 4.3.1.2, "Alarms Off"*. In this case the table lists the *resetted* alarms for all Services that are located within the Node.

#### 4.3.3.5 MAR Events

The menu entry **MAR Events...** corresponds to the identical entry for Service objects described in *Section 4.3.1.3, "MAR Events"*. In this case reactions may be connected to status change events of the Node, or to status change events of the Containers and Services located within the Node.

#### 4.3.3.6 Documentation

The menu entry **Documentation** will be visible, when a documentation URL is defined for the respective Node within the MIB. This entry will open this URL within a browser.

#### 4.3.3.7 Configuration Management

The menu entry **Configuration Management** will be visible, when a configuration URL is defined for the respective Node within the MIB. This entry will open this URL within a browser.

### 4.3.4 Applications


*Figure 27 Symbol for Shared Services 'Application'*

Applications provide a connected set of Services. These Services may be distributes over different Nodes.

Applications will be displayed within the OpenScape FM by symbols as shown in *Figure 27*. They will be found on the submap of Instances.

If the status of an Application within the Shared Services MIB is currently 'unknown' or 'normal', then the state of the Application symbol will be the aggregated state of the objects contained in the submap.

Else the status of the symbol will be set to the status of the Application within the Shared Services MIB.

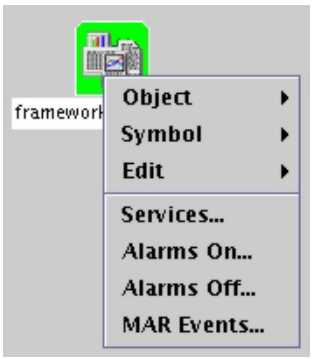



*Figure 28 Context menu for Shared Services 'Application'*

The different entries of the context menu for objects of the type 'Application' (*Figure 28*) have the following functionalities:

### 4.3.4.1 Services

The menu entry **Services...** corresponds to the identical entry for Container objects described in *Section 4.3.2.1, "Services"*. In this case the table lists the Services that are located within the submap of the Application.

### 4.3.4.2 Alarms On

The menu entry **Alarms On...** corresponds to the identical entry for Service objects described in *Section 4.3.1.1, "Alarms On"*. In this case the table lists the *non resetted* alarms for all Services that are located within the submap of the Application.

### 4.3.4.3 Alarms Off

The menu entry **Alarms Off...** corresponds to the identical entry for Service objects described in *Section 4.3.1.2, "Alarms Off"*. In this case the table lists the *resetted* alarms for all Services that are located within the submap of the Application.

### 4.3.4.4 MAR Events

The menu entry **MAR Events...** corresponds to the identical entry for Service objects described in *Section 4.3.1.3, "MAR Events"*. In this case reactions may only be connected to events for the Application.

### 4.3.4.5 Documentation

The menu entry **Documentation** will be visible, when a documentation URL is defined for the respective Application within the MIB. This entry will open this URL within a browser.

#### 4.3.4.6 Configuration Management

The menu entry **Configuration Management** will be visible, when a configuration URL is defined for the respective Application within the MIB. This entry will open this URL within a browser.

### 4.3.5 Instances

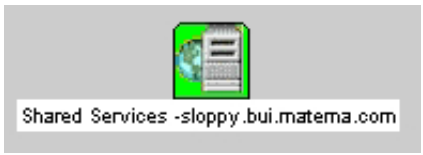



*Figure 29 Symbol for Shared Services 'Instances'*

Instances within the Shared Services Framework represent actual installations of the framework.

Instances will be represented within the OpenScape FM by a symbol like the one displayed in *Figure 29*. By default they will be placed within the submap of the Topology Manager. But using the functions of the Topology Manager, they can be moved within any network.

If the status of an Instance is the aggregated state of the objects contained within the submap.

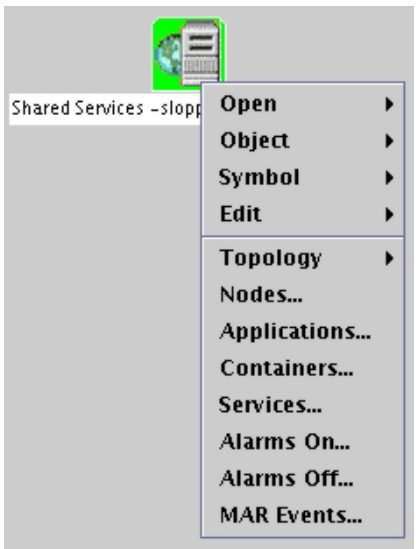



*Figure 30 Context menu for Shared Services 'Instances'*

The different entries of the context menu for objects of the type 'Instance' (*Figure 30*) have the following functionalities:

#### 4.3.5.1 Topology

Like for many other objects within the OpenScape environment, a network and/or sub network id may be assigned to an Instance. This assignment defines the location of the Instance within the topology tree.

Both menu entries within the sub menu **Topology** are used for the topology configuration of objects of the type Instance.

**Topology->Configuration...:** If this entry gets selected, a window will open in which the network, sub network and primary domain id of the Instance object can be defined. More about this subject can be found in the *OpenScape FM Desktop User Guide*.

**Topology->Node Id Configuration...:** The representation of connections between systems of different topology types can be defined by the manual assignment of Domain-Id/Node-Id pairs for certain nodes. The assignment can be made within the window, which will be opened by this menu entry. More about this subject can be found in the *OpenScape FM Desktop User Guide*.

### 4.3.5.2 Nodes

The menu entry **Nodes...** opens up an info browser (*Figure 31*). This browser contains a table that displays all Nodes that are contained within the Instance. Each row within the table represents a single Node.

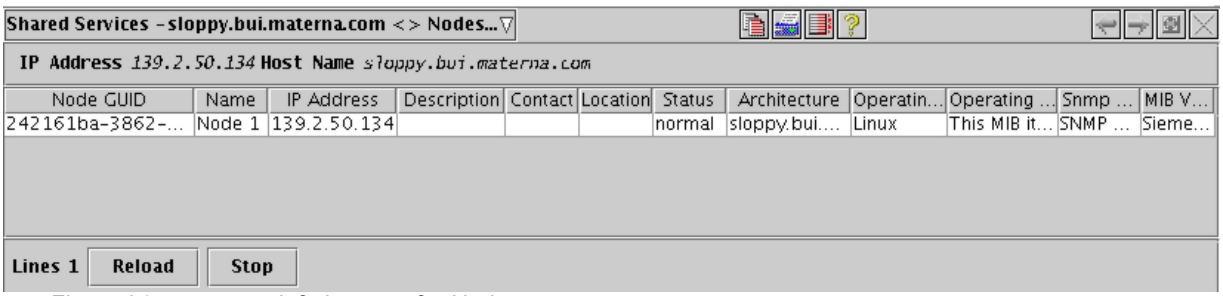



*Figure 31 Info browser for Nodes*

The table within the info browser contains the following columns:

**Node GUID:** The ID of the Node. Unique within the Instance.

**Name:** The name of the Node.

**IP Address:** The IP address of the Node.

**Description:** A (short) description of the Node.

**Contact:** The administrator responsible for the Node.

**Location:** The physical location of the Node.

**Status:** The status of the Nodes.

**Architecture:** The architecture of the Node (e. g. x86, sunUltra, ...)

**Operating System Name:** The name of the operating system (e.g. Windows, Solaris, ...)

**Operating System Version:** The version of the operating system (e. g. 5.0 for Windows2000)

**Snmp Agent Version:** The version of the SNMP agent.

**MIB Version:** The version of the management information base.

### 4.3.5.3 Applications

The menu entry **Applications...** opens up an info browser (*Figure 32*). This browser contains a table that displays all Applications that are contained within the Instance. Each row within the table represents a single Application.

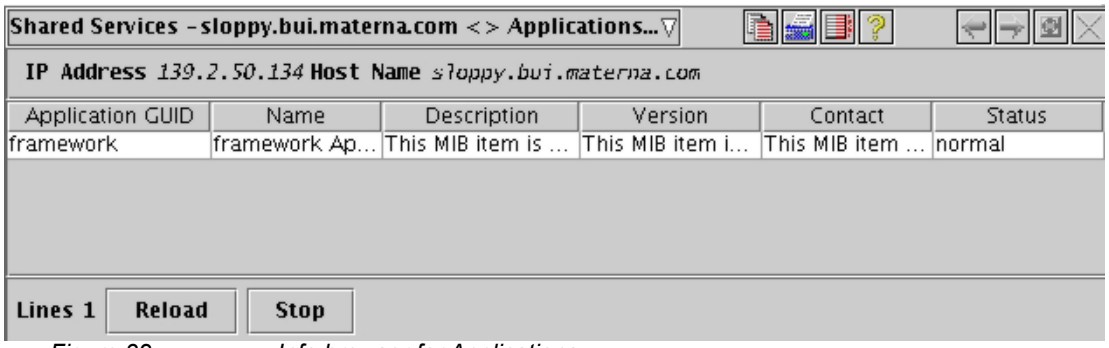



*Figure 32 Info browser for Applications*

The table within the info browser contains the following columns:

**Application GUID:** The ID of the Application. Unique within the Instance.

**Name:** The display name of the Application.

**Description:** A short description for the Application.

**Version:** The version of the Application.

**Contact:** The administrator responsible for the Application.

**Status:** The status of the Application.

### 4.3.5.4 Containers

The menu entry **Containers...** corresponds to the identical entry for Node objects described in *Section 4.3.3.1, "Containers"*. In this case not only the Containers contained within a single Node are displayed, but all Containers that belong to the Instance. The table within the info browser features the additional column **Node**. This column is used to display the Node to which the respective Container belongs.

### 4.3.5.5 Services

The menu entry **Services...** corresponds to the identical entry for Container objects described in *Section 4.3.2.1, "Services"*. In this case not only the Services contained within a single Container are displayed, but all Services that belong to the Instance. The table within the info browser features the additional columns **Application GUID**, **Container GUID** and **Node GUID**. These columns display, to which Application, Container and Node the respective Service is attached.

### 4.3.5.6 Alarms On

The menu entry **Alarms On...** corresponds to the identical entry for Service objects described in *Section 4.3.1.1, "Alarms On"*. In this case the opening info browser not only displays the non resetted alarms of a single Service, but all non resetted alarms for any Service that is connected to the Instance.

To accommodate this fact, the table of the info browser contains the following additional columns:

**Alarm Table Id:** The Id of the corresponding alarm table.

**Service GUID:** The GUID of the Service that generated the alarm.

**Container GUID:** The GUID of the Container which contains the source of the alarm.

**Node GUID:** The GUID of the Node which contains the source of the alarm.

**Source-GUID:** The GUID of the source of the alarm. Possible values are the configured GUID of the fault manager instance or the HP8k trap receiver instance.

### 4.3.5.7 Alarms Off

The menu entry **Alarms Off...** opens an info browser like the one described in *Section 4.3.5.6, "Alarms On"*. In this case the browser will only displays alarms, that have already been resettet.

### 4.3.5.8 MAR Events

The menu entry **MAR Events...** corresponds to the identical entry for Service objects described in *Section 4.3.1.3, "MAR Events"*. In this case reactions may be connected to status change events for Applications, Containers, Nodes and Services contained within the Instance. In addition events of the IP Manager and Window Log Events may be attached to reactions.

## 4.3.6 Server


*Figure 33 Symbol for Shared Services 'Server'*

If for an SNMP agent is discovered for an IP node that implements a Shared Services MIB, then a Shared Services SNMP Server object will be generated within the OpenScape FM. This object will be placed on the submap of the ip node.

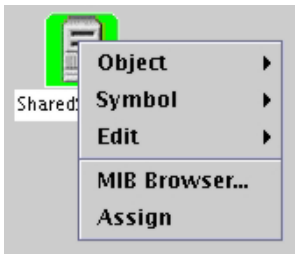



*Figure 34 Context menu for Shared Services 'Server'*

### 4.3.6.1 Assign

The menu entry **Assign** opens a window (*Figure 35*) in which Shared Services Server can be assigned to Instances.

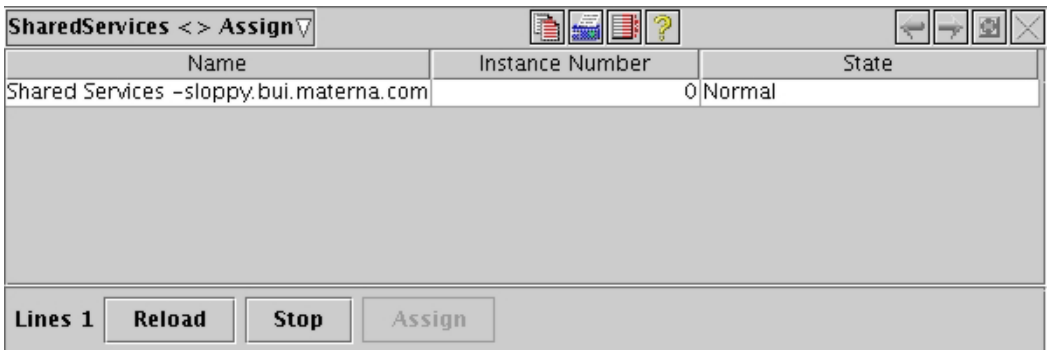



*Figure 35 Assigning Servers to Instances*

The window displays a table which contains one table row for each known Shared Service Instance. The columns display the **Name**, **Instance Number** and current **State** of the respective Instance.

If a row gets selected within the table and the button **Assign** will be pressed, then the current Server will be assigned to the selected Instance.

**Important Note:**
Only one Server may be assigned for each Instance. If there already exists an assigned Server for the selected Instance, then an assignment will overwrite the old assignment.

This function can be used, when the IP address of an Instance has been changed. In this case the OpenScape FM would initially detect a second Instance. With this function the 'old' server can be assigned to the 'new' Instance. This will keep manually made changes which would be lost otherwise that were e. g. were made to submaps, .

### 4.3.6.2 MIB Browser

The menu entry **MIB Browser...** opens a MIB browser. In this browser the structure and variable values of the MIB connected to the Server can be viewed. More about MIB browsers can be found in the separate *Enterprise MIB Plugin User Guide*.

## 4.4 List of HiPath MIB agents

To get a quick overview over all discovered Shared Services and HiPath MIB agents, the main menu item **HiPath MIB**–>**List HiPath MIB agents** can be selected. This will opens an Info Browser which displays the discovered agents (*Figure 36*).

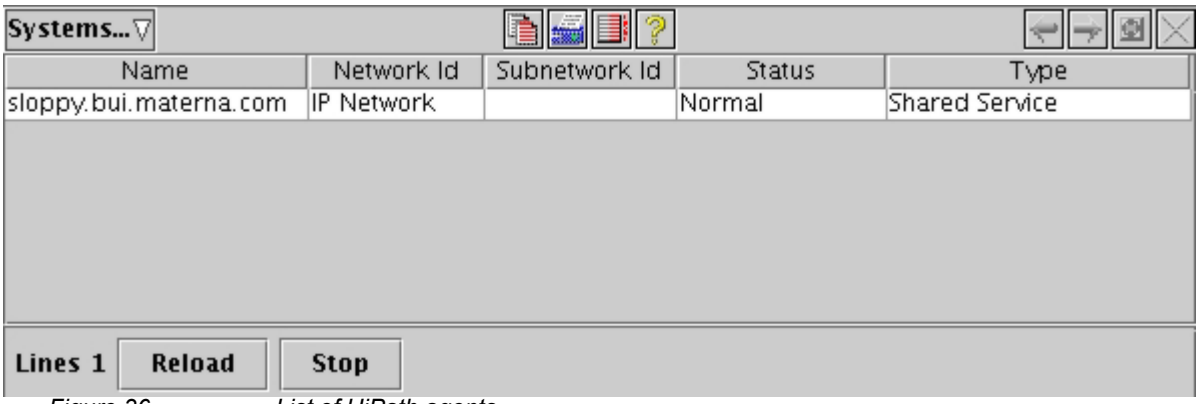



*Figure 36 List of HiPath agents*

The info browser contains a table in which each row represents an agent. The columns contain information about the name (column **Name**), the configured position within the OpenScape FM (columns **Network Id** and **Subnetwork Id**) and the current status of the agents (column **Status**). The column **Type** specifies the type of the agent. At the time of the creation of this user guide this can be a HiPath MIB agent (entry 'HiPath MIB') or a Shared Services MIB agent (entry 'Shared Service').

# 5 Fault Monitoring

## 5.1 Status of the HiPath/OpenScape MIB Plugin Components

By using a HiPath MIB or a Shared Services MIB the status of an agent/an instance and the status of all registered applications, processes and subcomponents (nodes, services, ...) can be retrieved. During the initial discovery the status of every MIB object will be read from the MIB and stored within the OpenScape FM. After this the OpenScape FM server will check the status of these objects on a regular basis. Like for every OpenScape FM IP node, the HiPath MIB IP node has set its SNMP polling parameters initially to those values which have been defined in the IP Manager (please refer to the *IP Manager Plugin User Guide* for detailed information). The SNMP parameters of every single IP node can be changed by using the context menu of the node and selecting the entry **Configure...** switching to the tab **SNMP Parameters**.

Since the status of the HiPath MIB agent itself and the status of the applications and processes correspond, there is a status propagation from the underlying processes to the parent application and from there to the parent agent.

## 5.2 Status of Shared Services Components

In the case of Shared Services Components the status of the objects will not always be defined by the status of the sub components. In some cases the status of sub components will only be propagated, when the component itself is in a ’non critical’ state (’normal’ or ’unknown’). Otherwise the status of the component itself will be displayed. Even when there are sub components with a ’more critical’ status. More about this can be found within the descriptions of the different component types (see *Section 4.3, “Shared Services Objects: Symbols and Functionalities”*).

## 5.3 Traps

During the discovery of the HiPath MIB IP node, OpenScape FM registers itself as trap destination automatically. Thus, the OpenScape FM server is ready to receive traps from the HiPath MIB device without any further manual configuration.

### 5.3.1 HiPath MIB Traps

In this paragraph we will examine the handling of SNMP traps from a HiPath MIB device by OpenScape FM.

A HiPath MIB-enabled device, like a PC, can send nine types of traps. Each trap will be shown in the OpenScape FM Event Browser (see *OpenScape FM Desktop User Guide* for detailed information about the Event Browser). Depending on the type of the trap, there can be OpenScape FM reactions. We will give a short explanation for each trap type in the proceeding paragraph.

#### 5.3.1.1 Host Traps

**hiPathHostUp:** a HiPath-enabled device (i.e. for example a PC with OpenScape-managed applications and HiPath SNMP agent) which had already been registered as a OpenScape FM object has been restarted. This will activate a status and configuration poll on the agent.

**hiPathHostDown:** a HiPath MIB-enabled device has been stopped. The status of the respective OpenScape FM objects will be set accordingly by means of a status and a configuration poll.

**hiPathHostStatusChange:** the status of a HiPath MIB-enabled device has changed. A status poll will be performed for the agent and the underlying applications and processes.

#### 5.3.1.2 Application Traps

**hipathApplicationInstalled:** a new HiPath-enabled application has registered with the HiPath MIB. A new application object will be created as a child object of the corresponding IP node, resp. HiPath MIB agent.

**hiPathApplicationDeinstalled:** a registered application has been deinstalled. The corresponding OpenScape FM object will be deleted from the OpenScape FM database.

**hiPathApplicationStatusChange:** the status of a registered application has changed. A status poll will be performed for the application and the underlying processes.

#### 5.3.1.3 Process Traps

**hiPathProcessUp:** a registered process is up and running (again). The status will be set in OpenScape FM accordingly.

**hiPathProcessDown:** a registered process is down. The staus will be set in OpenScape FM accordingly.

**hiPathProcessStatusChange:** the status of a registered process has changed and a status poll will be performed for this process.

### 5.3.2 Shared Services Traps

This chapter lists the traps that are defined for the Shared Services MIB. It describes how the respective events will be handled by the OpenScape FM. For the traps listed below an entry will be created within the event browser of the OpenScape FM.

#### 5.3.2.1 Status Traps

**svcNodeStatusChange:** The information of a Shared Services Node has been changed. This can be a change of the status or a change of the assigned Containers. If this trap is received, the status of the Node gets updated and the representation of the assigned Containers will be checked.

**svcApplicationStatusChange:** The information of a Shared Services Application has been changed. This can be a change of the status or a change of the assigned Services. If this trap is received, the status of the Application gets updated and the representation of the assigned Services will be checked.

**svcContainerStatusChange:** The information of a Shared Services Container has been changed. This can be a change of the status or a change of the assigned Services. If this trap is received, the status of the Container gets updated and the representation of the assigned Services will be checked.

**svcServiceStatusChange:** The status of a Service has been changed. If this trap is received, the status of the respective Service object will be updated.

**svcAlarmTrap:** This trap generates an entry within the event browser. A status change will not be initialized by this trap (this can be handled by an additional ’svcServiceStatusChange’ trap). If this trap type is received, the corresponding object will start blinking within the OpenScape FM. The object will keep on blinking as long as at least one unacknowledged alarm event exists within the event browser for this object.

**svcResetAlarmTrap:**If this trap is received, an entry will be generated within the event browser. A previously generated respective alarm event will get acknowledged by this trap.

# A HiPath MIB 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 *HiPath MIB* or *HiPath MIB Shared Services*.

# Index