A MITEL
PRODUCT
GUIDE

# Unify OpenScape Fault Management

## Unify OpenScape Fault Management V13, System Management

User Guide
09/2023

A31003-F30D0-U120-01-7619

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

**Contents**

# 1 Preface

This chapter discusses the following aspects:

- Goal and audience for this User Guide
- Terminology
- Structure of this User Guide
- The conventions used in this User Guide

## 1.1 Audience

This guide addresses users who want to learn how to use the System Management Plugin and System Management Agents. To work with the System Management Plugin, it is necessary to know how to use the OpenScape FM. More about this can be found in the *OpenScape FM Desktop User Guide*.

## 1.2 Terminology

- **OpenScape FM** stands for OpenScape Fault Management.
- **System Management** means the System Management for OpenScape FM.
- **Server** means the OpenScape FM Server, the Server Process of the OpenScape FM with the installed System Management Plugin.
- **Client** means the OpenScape FM Client. This may be the web based user interface or the stand-alone Client of the OpenScape FM.
- **Agent** means the System Management Agent.

## 1.3 Structure of this User Guide

This User Guide is divided into the following chapters:

- *Chapter 2, "Introduction"* introduces the concepts of the System Management and gives an overview about its functions.
- *Chapter 3, "Installation, Initialisiation and Agent Updates"* explains how the System Management is initialized and installed.
- *Chapter 4, "Overview"* provides an overview about the terms used with the System Management and how they are represented within the OpenScape FM.
- *Chapter 5, "Working with the System Management"* describes the configuration of the System Management and how to work with the gathered results.

- *Chapter 6, "Data Export"* and *Chapter 7, "Special Monitoring Functions"* describe specific functions of the System Management.

## 1.4 Conventions within this User Guide

The following font conventions are used in this document:

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

Example: **Monitoring Profile**.

It is also used for Buttons, menu names and item names

Example: the **OK** button.

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

Example: `prompt%.`

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

Example: **`Java`**.

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

Example: see *Layer 2 Manager User Guide.*

Italic type is also used for emphasis.

Example: *All* users will be affected.

# 2 Introduction

## 2.1 The System Management

The OpenScape Fault Management (OpenScape FM) in combination with the System Management enables organizations to increase the reliability of vital resources of their enterprise network. The System Management capabilities for heterogeneous system environments support the reduction and avoidance of down times, and therefore helps to reduce the cost of operation. The efficiency of the system and network management will be raised through the integration of monitoring functions for heterogeneous systems into a homogenous management platform.

The OpenScape FM is an innovative java/web based network management solution, which provides the ability to load plugins for specific technologies.

The System Management for OpenScape FM consists of the System Management Plugin (short Plugin) and of at least one System Management Agent (short *Agent*).

The agents are monitoring various parameters for selected IP nodes (e.g. used disk space, memory utilisation, process status), either locally or remotely within the network. For the monitoring of the parameters, a variety of monitoring templates that provide monitoring functions (in the following named *Templates*) are delivered with the plugin. Monitors from the list of templates can be selected and configured individually for each agent. These monitors monitor the respective parameters for a configured list of IP nodes and collect the gathered data within the OpenScape FM database.

Besides the provided monitor templates additional templates can be created to perform individual management tasks. The functionalities can be realized by using e.g. scripts, executable programs or Java functions. Once created, the individual functions can be selected from the list of templates for the agents; just like the provided functions.

The System Management Plugin communicates with the agents and displays the monitored parameters or executed actions graphically within the OpenScape FM.

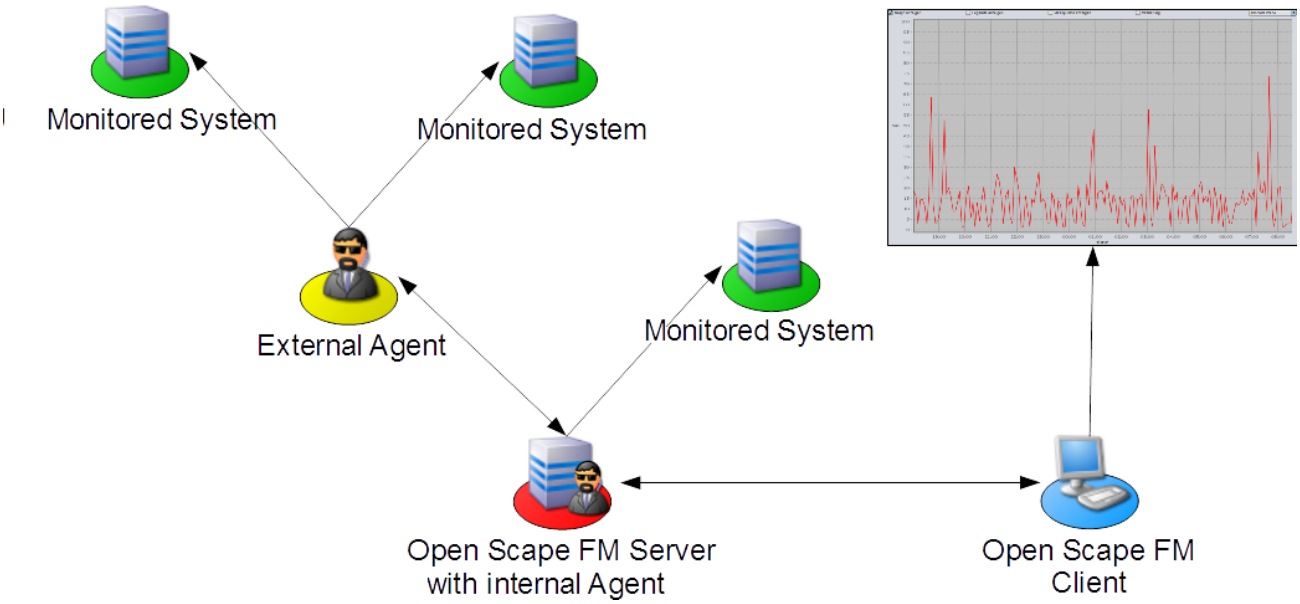



*Figure 1 Example Architecture of the System Management*

System Management is fully integrated into the OpenScape FM and can therefor use its features. E.g. the System Management Plugin uses the discovery mechanisms of the OpenScape FM to automatically discover the agents installed within the network. Each discovered agent is represented as a symbol on the submap of the respective IP node. The discovered agents can be configured by using the interface of the OpenScape FM.

For each monitor, individual multi level thresholds for the gathered results can be defined. Correspondent to the selected configuration, exceeding of thresholds will change the status of the monitored parameter within the OpenScape FM. Within the event browser of the OpenScape FM, an event will be generated that displays the status change.

The System Management Agent is a java-based generic agent. It provides core functions like the communication with the OpenScape FM Server, the time triggered execution of monitoring functions and the persistent storage of the gathered parameter data. The actual functions are implemented within monitors, which are executed by the agent at defined moments in time or in defined time intervals. These monitors deliver their results in a defined format, which will be analyzed and evaluated by the agent. The monitors can be easily modified, exchanged or expanded to generate new/modified monitoring functions.

Overview of the System Management Features:

- Java based generic agent to monitor arbitrary system parameters.
- Automatic discovery of System Management Agents in the network by the System Management Plugin for the OpenScape FM.
- Value history of monitored system parameters.
- Easily expandable (new individual monitoring functions, e.g. via new scripts).
- Event messages to the OpenScape FM in case of critical system states (e.g memory usage >90%).
- Graphical representation of the monitored parameters within a hierarchical structure within the OpenScape FM.

- Graphical status representation.
- Logfile monitoring by System Management Agents. Event messages, if a defined search pattern is detected.
-

# 3 Installation, Initialisiation and Agent Updates

The OpenScape Fault Management (OpenScape FM) is a web based network management platform. It allows fault management and administration of IP and telecommunications networks and displays the network graphically. It provides the basic functionality to add and use technology specific plugins.

## 3.1 Preparation of the OpenScape FM Server

The System Management Plugin will be installed automatically during the installation of the OpenScape FM Server. Additional preparations are not necessary.

### 3.1.1 Installation of the System Management Plugin

The System Management Plugin is a part of the OpenScape FM Installation package. It will be installed automatically when the OpenScape FM is installed.

### 3.1.2 Initialization of the System Management Plugin

The initialization is performed automatically.

### 3.1.3 Installation of a System Management Agent

The installation of the OpenScape FM automatically installs and activates a so called internal agent. Usually this internal agent is sufficient. Additional external agents (stand alone agents) are needed, when individual systems that should be monitored cannot be accessed by WMI (Windows) or SSH (Linux). They are also useful to distribute the load when a large number of parameters and systems should be monitored.

To install an external agent on an IP node, the stand alone agent can be used. The installation files for this agent can be found on the distribution media of the OpenScape FM.

For Windows systems, the file `setup_agent_osfm.exe` can be used. The file `setup_agent_osfm.sh` installs the stand alone agent on UNIX operation systems. The stand alone agent supports all Windows versions and UNIX derivatives that are also supported by the OpenScape FM.

If the Performance Management Plugin is used (see *Performance Management Plugin User Guide*), this stand alone agents will also be used.

If the new stand alone System Management Agent should monitor other hosts using WMI, an account for this function can be entered during the installation process. This account must belong to the group *Administrator* on the installation host and must have the Windows security policy *‘Logon as a Service‘*. For this account the **Domain**, **Service User** and **Password** have to be entered.

The OpenScape FM Server host and the System Management Agent hosts have to support the same IP protocol(s). If the OpenScape FM Server host supports IPv4 the System Management Agent hosts have to support IPv4. If the OpenScape FM Server host supports IPv6 the System Management Agent hosts have to support IPv6.

## 3.1.4 Automatic Updates for System Management Agents

The System Management Agents are integrated into the OpenScape FM Upgrade mechanism and include a separate update service (see *Desktop User Guide*).

If an OpenScape FM load or patch is installed (automatically or manually) that contains loads or patches for the System Management Agents these agents can be updated automatically. The Internal System Management Agent is always updated as part of an OpenScape FM update, but the External Agents that should be updated have to be configured individually.

The configuration page for External System Management Agents can be opened by using the main menu entry **System Management->Software Update**.

The page lists all External System Management Agents in separate table rows which contain the **Server Name**, **Server IP** and currently installed agent **Version**.

The check boxes in the column **Auto Update** define which of the agents should be automatically updated, if a new agent load has been found during the installation of an OpenScape FM load or patch.

The button **Update** on the bottom of the table can be used to manually install the latest load or patch to the selected agents.

If **Auto Update** is enabled for an External System Management Agent and a new load or patch has been downloaded, the load or patch is transferred to the agent and it gets instructed to perform the installation of this load or patch. The agent‘s update is performed by starting the update service of the affected agent.

If **Auto Update** is not enabled for an External System Management Agent, updates for this agent have to be performed manually.

The transfer of the new load or patch to an agent is performed by using the already known/used ports of the System Management Agent. It is therefore *not* necessary to add additional ports to existing firewall rules.

## 3.1.5 Pushing KeyStores to External Agents

The Java KeyStore (JKS) is a repository of security certificates. The KeyStore located on the OpenScape FM Server machine is used to collect the authorization certificates used by the internal agent.

If external agents (see *Section 3.1.3*) are used, the certificates might also be needed by these external agents. When authorization certificates have been added or modified they can therefore be exported to external agents.

The entry **Push Keystore** from the context menu of a symbol representing an external agent pushes the content of the local Java KeyStore that is used by the local agent to the selected external agent. The context menu entry **Push Keystore** is also available on list entries representing external agents as e.g. within the list of all current agents (see *Section 5.1.2*).

Using the menu entry pushes the Java KeyStore located on the OpenScape FM Server machine to the machine on which the selected agent is running and restarts the selected external agent when the push has been performed.

# 4 Overview

This chapter introduces the basic concepts of the System Management.

## 4.1 Concepts

The System Management for OpenScape FM consists of two mayor components:

The **System Management plugin** is an extension for the OpenScape FM. It provides the user interface to define and configure the monitoring functions within the OpenScape FM. In addition it displays the evaluated results as e.g. graphs or tables. If the gathered values exceed defined thresholds, the status of the effected element will be changed accordingly and an event will be added to the event browser of the OpenScape FM to warn about current or emerging problems.

The **System Management agents** (short *agents*) control the gathering of the individual measurements. They evaluate the results and store them in the internal agent database.

With the installation of the System Management plugin, an **internal agent** will be installed and started on the OpenScape FM system automatically. Additional **external agents** can be installed on other systems.

Generally external agents are only needed:
- if a large number of systems should be monitored (load balancing),
- if the network is separated (one agent per partial network),
- if missing rights prevent the remote access to the monitoring programs.

The external agents can independently monitor one or more systems and send the results to the central OpenScape FM. External agents are automatically discovered and can be administered and configured within the OpenScape FM.

Which monitoring functions should be performed by an agent is defined by the **monitoring profiles** (short *profiles*) that are assigned to the agent. These profiles are selected from a list of **profile templates** (short *templates*), configured if needed, and activated for a selection of systems that are to be monitored.

The monitoring functions that are triggered by a profile are defined by the **monitors** that are contained within the profile. Each of these monitors executes a **sensor** (program) on the systems that were selected for the profile. The monitors collect the result parameters (short *parameter*s) provided by the sensors and evaluate them. For each monitor it can be individually defined, how often and how the sensor should be executed. Additionally it can be defined how the collected values should effect the status of the monitor.

**Important Note**:
In order to run a Monitor on a target system, appropriate access rights are required (see *Section 5.7.1*).

A **Sensor** can be a program, a Visual Basic script. a Java script or a Java method. The sensor has to be executable on the monitored system and it has to provide the results in an expected form (see *Appendix E*). Besides the result parameter, a sensor can also return a status. This status will be assigned to the invoking monitor. This assignment is only performed, if no status rules are defined for the monitor itself.

With the initialization of the plugin a number of profile templates, monitors and sensors are provided. These cover a large variety of common monitoring tasks. Additional templates, monitors or sensors can be created individually.

## 4.2 Navigation Tree and Symbols

The System Management and its agents expand the object tree with the respective elements. These are explained in the following sections.

### 4.2.1 Network Topology

The System Management creates a network topology with the name *System Management*. All IP nodes on which a System Management Agent is discovered, and for which the discovery rule „*System Management Discovery*“ matches, are automatically placed into it. More about discovery rules can be found in the *IP Manager Plugin User Guide*.

### 4.2.2 Agent Objects

A Parameter Container (see *Section 4.2.4*) named *System Management Agent* will be added to the submap of IP nodes on which a managed System Management Agent is discovered. If the discovered agent is an internal agent of an OpenScape FM installation, the Parameter Container will be named *Internal System Management Agent* instead.

This container represents the respective System Management Agent. It is used as an access point for the configuration of the agent and to get an overview about the monitoring functions executed by the agent.

The Status of the container is derived from the results of the monitoring and thus from the setting of the threshold values for the individual monitored parameters (see *Section 5.5.1*).

The Reachability Status of the container results from the reachability of the host on which the monitoring agent is running. If this is not reachable, *Unknown* (blue) is displayed as its Reachability Status.

The container may include a number of additional objects and containers:

The container *Managed Monitoring Profiles* contains an object for every monitoring profile that is active on the agent. These monitoring profile objects contain the node *targets* which contain an object for each IP node that is monitored by the respective profile. In contrast, data export modules (see *Chapter 6*) may contain the object *export_target* or *sm_export_target*.

The container *Remote* will be created for all System Management Agents that monitor remote systems. It contains symbols that represent the System Management Containers of the monitored systems (see *Section 4.2.3*).

The container *Configuration Files* displays the content of the directory `<agent installation directory>/ssma/conf` within the object tree (in case of update installations, the container may be named `Agent::File System`). This directory contains the configuration files of the agent‘s active monitoring profiles. *Internal* is a special internal monitoring profile that provides some statistics about the agent itself. Generally this profile is only used for debugging purposes. The same applies for the monitor *Agent::Interfaces* which identifies the internal RMI interfaces that are supported by the agent, and for the monitor *Agent::System Parameters* that monitors general data of the agent as e.g. its runtime.

### 4.2.3 Monitored Systems

Each IP node that is monitored by a managed agent gets a Parameter Container (see *Section 4.2.4*) named *System Management* or *Internal System Management* on its submap for each agent that monitors the node (e.g. *Figure 2* below `bui-pc319.bui.materna.com`). For agents that are not running on the IP node itself, the hostname of the agent (in brackets) will be added to the name.

The container provides the following additional hierarchy:

1. The top level contains the Monitoring Profiles from the respective agent that are monitoring the system (e.g. in *Figure 2*: *DemoMonitoring*).
2. Below these, the monitors of the monitoring profile are located. They represent the monitoring functions (**Sensors**) which are used on the IP node (e.g. in *Figure 2*: *DemoMonitoring -> Memory Usage*).
3. These finally contain the Parameters, or in other words the results of the sensors for the respective system (e.g. in *Figure 2*: *DemoMonitoring -> Memory Usage -> Physical memory*). The value, status and in some cases the name of a parameter depends on the values measured on the monitored system. How the status gets determined is described in *Section 5.2*.

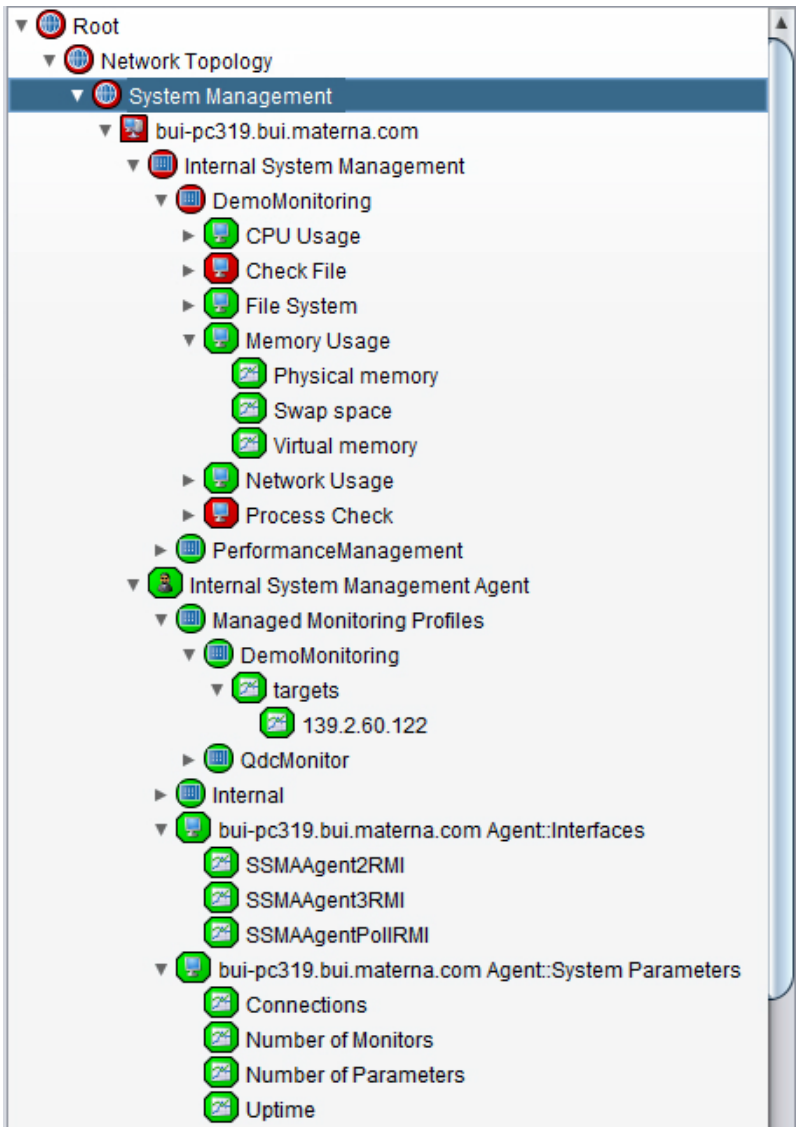



*Figure 2 System Management Objects in the Navigation Tree*

## 4.2.4 Parameter Container

Parameter Containers are used to structure the System Management objects that are assigned to either a System Management Agent (see *Section 4.2.2*) or a Host monitored by the System Management (see *Section 4.2.3*).

For both, the agent hosts and the monitored hosts one container is added automatically.

Additional Parameter Containers can be generated within these containers by using the item **New->Parameter Container** from the context menu of a container. These new containers can then be used to further structure the System Management components by moving Monitors or Containers into these.

All Monitors located within a Parameter Container can be started manually with a single click by using the entry **Execute all Monitors** from the container‘s context menu.

**Important Note**:
If the last element is removed from a manually created Parameter Container, the container itself is deleted.

# 5 Working with the System Management

This chapter explains the work with the System Management and its components. The following topics will be addressed:

- Overview about the current inventory.
- Gathering and analysis of parameter values.
- Introduction to the system monitoring and its configuration.
- Management of System Management Agents.

## 5.1 Overviews

The following sections introduce summaries that provide an overview about the inventory of the System Management.

### 5.1.1 Monitoring Functions

Within the main menu **System Management** the entry **Monitoring Functions...** can be selected to gain an overview about the existing monitoring functions. This opens a tree view containing all monitoring functions managed by the OpenScape FM. This view also contains the IP nodes managed by the profiles.

Located in the top level of the tree view are the System Management Agents managed by the OpenScape FM. Listed below are active monitoring profiles that belong to the respective agent. Finally, located on the lowest level is a list of IP nodes that are monitored by the respective profile. This list is labeled `targets` - or for a data export profile `export_target` (Performance Management) or `sm_export_target` (Java DB-Export).

The lists of IP nodes can be edited within the dialog. By using drag & drop, IP nodes can be added to a list. This can be done by selecting one or more IP node objects within the object tree and dragging them to the node of the list within the overview to which the IP nodes should be added. Alternatively, after selecting the IP nodes within the object tree, they can be added by clicking the button **Add** located below the overview.

An IP node that is not yet monitored by System Management can only be added if a respective System Management license is still available.

By selecting one or more IP nodes within a list, the button **Delete** located below the overview gets activated. Clicking the button removes the selected IP nodes from the respective list.

**Note**:
It has to be kept in mind, that adding or deleting IP nodes from a list, requests a restart of the respective System Management Agent. The adding or deletion can therefore be delayed by a few seconds.

## 5.1.2 List All Agents

The entry **Show Agents...** from the main menu **System Management** opens a list of all System Management Agents (managed and unmanaged) that are known by the OpenScape FM.

For every agent, the name and IP address of its system, the status of the agent and the data relevant for the licensing - the number of monitored systems and the sum of the parameters from all monitors of the agent - will be displayed.

The status *Unmanaged* of an agent shows that the respective agent is not managed by the OpenScape FM. For unmanaged agents no monitored systems or parameters will be counted against the System Management license.

Located below the list are a number of buttons that provide easy access to frequently used configuration pages:

- **Create Monitoring Profile...**
  Opens the Wizard to create a new monitoring profile for the selected agent. More about the Wizard can be found in *Section 5.6.2*.
- **Monitoring Profiles...**
  Opens the object tree and displays the monitoring profiles managed by the agent.
- **Monitored System Accounts...**
  Opens the user identification dialog for the selected agent. The identification is needed to connect to monitored systems. More about this can be found in*Section 5.7.1***.**
- **Agent Accounts...**
  Opens the dialog to enter the account data to access the selected agent. More about this can be found in *Section 5.7.3*.

## 5.1.3 List all Monitors

To list all active System Management Monitors of the System Management Agents managed by the OpenScape FM, the entry **Show Monitors...** from the main menu **System Management** can be used.

For each monitor the following will be displayed: Its name, the IP address of the monitored system, the current status of the monitor and the hostname of the System Management Agent to which the monitor belongs.

## 5.1.4 List all Parameters

The entry **Show Parameters...** from the main menu **System Management** opens a listing of all System Management Parameters from the active monitors of all System Management Agents managed by the OpenScape FM.

For all parameters the Name, the IP Address of the monitored systems, the current Status, the name of the Monitor that created the parameter and the Hostname of the System Management Agent to which the parameter belongs to, will be displayed.

If the entry **Monitored Parameters** is selected from the context menu of an agent object, a corresponding overview is displayed that is restricted to the parameters of the agent.

## 5.1.5 List all Hosts

The entry **Monitored Hosts...** from the main menu **System Management** opens a list of all hosts monitored by System Management.

For each host, its name, IP address, the names of the monitored profiles, the number of assigned unacknowledged events, the current status, and the time of the last status change are displayed.

The column **License Assigned** shows whether a System Management license is in use for the respective host. With the buttons **Revoke License** or **Assign License** located below the table, the licenses can be revoked or assigned for selected hosts.

**Important Note**:
If a host does not have an assigned license, the monitors assigned to it are set to *unset* and no evaluations are performed.

If the menu item **Monitored Hosts** is selected from the context menu of an agent object, a corresponding overview is displayed that is restricted to the hosts monitored by the current agent.

## 5.1.6 Control Center

If the Control Center plugin has been initialized, a number of Control Center overviews are provided for System Management objects. The main menu entry **Server->Control Center->Control Center - Overview...** can be used to open the Control Center. The information can then be found on the submap of the object „*System Management*“. Using the main menu entry **System Management->Control Center - Overview...** will also open the overview. In both cases the following chart windows are available:

- **Recent 10 Critical Parameters**: This chart window lists up to ten parameters whose status most recently changed to critical.
- **Top CPU Usage**: This chart window displays the objects with the highest CPU usage.
- **Top Memory Usage**: This window shows the objects with the highest memory usage.
- **Top Filesystem Usage**: This chart window shows the file systems with the highest percentage capacity utilization.
- **Top Network Out Usage**: This chart window depicts the objects with the highest outgoing network usage (in percent).
- **Top Network In Usage**: This chart window shows the objects with the highest incoming network usage (in percent).

More about the Control Center can be found in the *Control Center Plugin User Guide*.

## 5.2 Measurements, Status Evaluation and Events

The System Management Monitors trigger the measurement of parameters on all systems that are monitored by the respective monitoring profile. This can be activated manually or automatically in configured time intervals (see *Section 5.5.6*).

The result of a measurement is a parameter value that consists of the current value (e.g. numerical or `true` / `false`) and a short description. Based on the measured value the parameter its status is set. This will be either defined by the sensor that performed the measurement, or will be the result from the threshold configuration of the parameter (see *Section 5.5.1*).

If the short description of the parameter value or the status of the parameter changes after a measurement, a respective event is generated. This can be seen within the Event Browser.

The generation of events can be deactivated for individual parameters by using the entry **Disable Events** from their context menu. The entry **Enable Events** reverts this.

### 5.2.1 Status of Monitors

By default a System Management monitor inherits its status from its parameter with the most critical status.

Here the following exception exists: If an error occurs during the execution of the monitoring function and therefore the evaluation cannot be completed, the monitor receives the status *Unknown* and its parameters receive the status *Unset*. and a corresponding error message is generated.

For example, this is the case if the monitored host could not be reached.

If the Host or the Agent could not be reached, a respective *Reachability Status* will be added to the Host or the Monitors/Parameters.

The respective message can be seen by using the context menu entry **Show Values...** (see *Section 5.4*).

## 5.3 Analysis of Parameter Values

To allow a quick analysis of parameters, all System Management parameter objects offer the entry **Show Values...** in their context menu. This entry opens a detailed view about the parameter consisting of three tabs that are described in the following.

### 5.3.1 History

The tab **History** shows the historic development of the parameter value. In the case of a numerical value, by default the development will be visualized as a value/time graph (see *Appendix C*). For other value types, a table with the chronological order of the values will be displayed.

In the case of numerical values, the table view can also be selected by unchecking the checkbox **Show Chart**.

The table view contains one row per collected parameter value. For each measuring the exact date of the measurement, the measured value, the resulting status of the parameter and a short description are displayed. The button **Long Message** changes to the detailed view for all values that are selected in the table (this view is build like the view **Current Value** - see *Section 5.3.4*). Clicking the button **Delete History** deletes *all* values collected for the parameter irrevocably.

## 5.3.2 Trend Monitoring

From the perspective of resource and capacity planning, it is crucial to recognize upcoming bottlenecks or breakdowns before they actually happen.

The History values (see *Section 5.3.1*) and thresholds defined for these values (see *Section 5.5.1*) can be used to perform checks against the current state of the system (like the usage of hard disk space). These thresholds generate a warning when they are actually exceeded by measured values.

Example:
If it should be prevented that a parameter exceeds 90%, a threshold of 90% would not be sufficient. The warning would only be given when the violation has already happened. To avoid this, a lesser threshold (like 80%) has to be assigned. This might give ample warning that the 90% will be reached in the near future. But it might also generate a false warning, since the 90% might never be reached.

Based on the historic data, the Trend Monitoring predicts upcoming values. Alarms can be generated in advance of an incoming problem. Preventive actions for the problem can therefore be applied before the predicted error state appears.

The Trend Monitoring consists of two components:

1. On demand trend calculation of monitor parameters and their graphical representation (see below).
2. Automated early warnings by supporting threshold configurations for trend calculations (see *Section 5.5.2*).

**Trend Calculation:**

The Trend Calculation is provided for System Management monitors that return numeric values.

The calculation uses the last 360 values (or all values, if less than 360 values are stored) provided by the collected historic values (see *Section 5.3.1*).

Based on these values and on various prediction functions the trend calculation tries to identify a trend or a regular behavior by generating additive or multiplicative time series models. The time series which correlates best with the historic data will be used to predict the upcoming values. The number of sample points calculated for the future is limited to one quarter of the sample points taken from the historic values to perform the calculation. If e.g. the calculation is based on the data of a whole day, the prediction will be calculated for the upcoming 6 hours.

The historic values (shown in a red graph) together with the calculated values (shown in a blue graph) are displayed on the tab **Trend Analysis**.

**Trend Configuration:**

Two parameters of the Trend Monitoring can be reconfigured for each System Management Agent.

Both can be set on the tab **Configuration Properties** within the configuration of the respective agent.

- **trendanalysis_history_max_length**:
  This parameter defines the maximum number of values (default: 360) that are used to calculate the trends. Possible values are in the range of 180 to 720.
  (The time needed to perform the calculations rises with the third power of this value. Changes should therefore be handled with care).
- **trendanalysis_weight_max_multiplicator**:
  This parameter defines the responsiveness of the trend analysis in relation to the age of the collected values (default: 2). If e.g. the parameter is set to 10, then the error rate of the latest value is weighted with 10, the error rate of the oldest value is weighted with 1, and the values in between are weighted on a linear scale between 1 and 10.
  Setting this value higher leads to a trend analysis that weights newer values much higher then older values, which leads to trends that follow closer to the last received values. Values below 1 are not allowed.

## 5.3.3 Status History

The tab **Status History** graphically displays the parameter status over time. It has the same structure and usage as the tab **History** (see *Section 5.3.1*).

## 5.3.4 Current Value

This tab provides detailed information about the last measurement of the parameter.

The head of the page contains the name of the host for which the parameter was measured, as well as the name of the parameter and monitor that created the result. The date of the last measurement, the measured value and a short description follow. The resulting status of the parameters is highlighted by color.

The main part of the page consists of detailed information about the measurement. This can contain additional information and may provide insights about the composition of the measured value.

The bottom of the page contains a short explanation how the status of the parameter has been determined from the measurement. This can be a standard value or the result of the threshold configuration for the parameter (see *Section 5.5.1*).

## 5.3.5 Comparison of Parameters

If more than one parameter is selected in the object tree, the context menu entry **Show History...** is available. By this entry a comparison of the value history from the selected parameters will be opened.

For numerical values, the graph displays multiple curves and the table view displays an additional column for each parameter.

This allows to compare multiple parameters to identify potential correlations.

## 5.4 Analysis of Monitor Status

System Management Monitor objects also provide the **Show Values...** context menu entry. This opens a detailed view consisting of two tabs.

The tab **Current Value** displays the results of the last measurement and is constructed like the respective page for parameters (see *Section 5.3.4*). In the case of an error during the execution of the monitoring function, the respective error message will be displayed here.

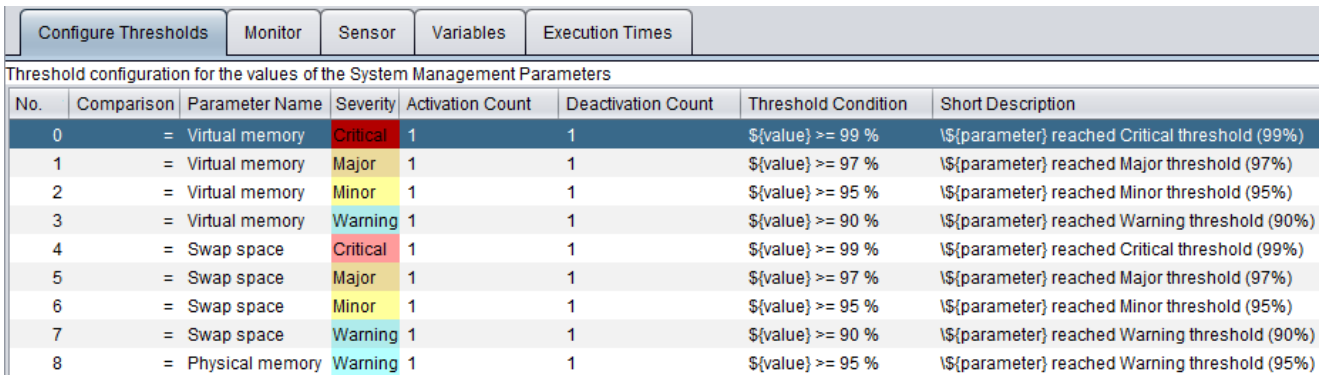
The tab **History** displays a table like the value history for parameters (see *Section 5.3.1*).

## 5.5 Monitor Configuration

The configuration dialog for System Management Monitors can be displayed by using the context menu entry **Configure...** and consists of five tabs that are explained in the following subsections.

### 5.5.1 Configure Thresholds

The page **Configure Thresholds** consists of a table that contains a threshold and a status assignment per row.


Configure Thresholds | Monitor | Sensor | Variables | Execution Times

Threshold configuration for the values of the System Management Parameters

| No. | Comparison | Parameter Name | Severity | Activation Count | Deactivation Count | Threshold Condition | Short Description |
|---|---|---|---|---|---|---|---|
| 0 | = | Virtual memory | Critical | 1 | 1 | ${value} >= 99 % | \${parameter} reached Critical threshold (99%) |
| 1 | = | Virtual memory | Major | 1 | 1 | ${value} >= 97 % | \${parameter} reached Major threshold (97%) |
| 2 | = | Virtual memory | Minor | 1 | 1 | ${value} >= 95 % | \${parameter} reached Minor threshold (95%) |
| 3 | = | Virtual memory | Warning | 1 | 1 | ${value} >= 90 % | \${parameter} reached Warning threshold (90%) |
| 4 | = | Swap space | Critical | 1 | 1 | ${value} >= 99 % | \${parameter} reached Critical threshold (99%) |
| 5 | = | Swap space | Major | 1 | 1 | ${value} >= 97 % | \${parameter} reached Major threshold (97%) |
| 6 | = | Swap space | Minor | 1 | 1 | ${value} >= 95 % | \${parameter} reached Minor threshold (95%) |
| 7 | = | Swap space | Warning | 1 | 1 | ${value} >= 90 % | \${parameter} reached Warning threshold (90%) |
| 8 | = | Physical memory | Warning | 1 | 1 | ${value} >= 95 % | \${parameter} reached Warning threshold (95%) |

*Figure 3 The Default Threshold Configuration for the Monitor „Memory Usage“*

If the monitor measures a new value for one of its parameters, the thresholds will be checked in their order (column **Number**) and it will be tested, if they match for the parameter and its value. If this is the case, the status of the parameter will be assigned as configured. The status determined by the sensor will be overwritten. In addition - if the status differs from the previous status of the parameter - an event with the configured message will be generated. The subsequent thresholds will not be checked for the current measurement.

The threshold consists of the following components:

- **Parameter Name**
  An expression that will be compared with the name of the measured parameter (see Comparison Operator).

- **Comparison**
  The method that should be used to compare the expression with the name of the parameter. The operators `=` and `!=` check character by character for equality or inequality between the expression and the parameter name. For the operators `~` and `!~` the expression will be treated as a regular expression, and it will be checked whether the name of the parameter matches with it.

- **Threshold Condition**
  A boolean expression using the StringFormatter Syntax (see *OpenScape FM Desktop User Guide*) which is used to evaluate the parameter value. The variable `${value}` is a placeholder for the measured parameter value.

  If the monitor already has parameters, the context menu of this textfield contains the submenu **Parameter specific**. For easier configuration, predefined threshold conditions that are respective to the current values of the parameters are provided.

- **Activation Count**
  Determines how often the threshold condition has to be met in immediate succession for the parameter until the threshold takes effect.

- **Deactivation Count**
  If the threshold fired for the parameter, the deactivation number determines how often the threshold condition must not be met in immediate succession before the threshold does stop taking effect.

If the monitor has been performed at least once, and parameters exist, a selection list with these parameters will be provided within the configuration field for the column **Parameter Name**.

**Important Note**:
The default threshold values usually apply to all parameters of a monitor (**Parameter Name**: `.*`).
If an individual parameter should be configured separately, using the **+** button new entries can be created in the table, the corresponding **Parameter Name** can be selected and the desired **Threshold Condition**s can be configured.
Since only the first matching condition is executed when evaluating the table, conditions for specific parameters should be moved ahead of the general conditions by using the ↑ button

The status assignment consists of:

- **Severity**
  The status that will be assigned to the parameter when the threshold takes effect.

- **Short Description**
  The text of the generated event when the status of the parameter changes.

**This leads to the following behavior**:

If the name of the measured parameter matches the *[Expression **Parameter Name**]* according to the *[Operator **Comparison**]* and the *[**Threshold Condition**]* matches for *[**Activation Count**]* times in a row, then the parameter gets the status *[**Severity**]*. If the parameter changes its status, an event will be generated with the message *[**Short Description**]*.

If, in the following, the *[**Threshold Condition**]* will not match *[**Deactivation Count**]* times in a row, the threshold no longer matches for the parameter, and the next threshold will be checked.

## 5.5.2 Configure Trend Evaluation Thresholds

Similar to the threshold definition for current monitor values (see *Section 5.5.1* above), thresholds can be defined to values that are predicted by the Trend Evaluation (see *Section 5.3.2*). These thresholds can be used to generate a warning if the trend evaluation forecasts an incoming problem.

General threshold definitions always check against the latest value that has been received for their parameter.

In contrast, trend evaluation predicts a whole set of values for the future. Therefore thresholds always have to be checked against all values that are currently predicted.

Out of this reason, and also to distinguish between actual monitor results and the predictions, new monitoring parameters for the predicted values can be defined. Thresholds can then be attached to these parameters which change the status of the new parameter and generate an event when they are exceeded. As for any other events, an ECE workflow can be used to trigger specific actions when such an event occurs.

**Creating an Additional Trend Evaluation Monitor Parameter:**

New trend evaluation monitor parameters can be created by using the menu entry **Configure...** from the context menu of the respective monitor symbol and switching to page **Forecasts** of the configuration window.

The buttons at the bottom of the window can be used to add (**+**) or remove (**-**) a parameter or to move the order of the parameters (**up** or **down arrow**).

The following values can be configured for each trend evaluation monitor parameter:

- **Name**: This is the name of the new parameter. It should be meaningful and unique to distinguish it from the other parameters of the monitor.
- **Base**: This value represents the parameter for which a prediction should be performed.
- **Function**: This defines how the predicted values should be evaluated against the thresholds. The function defines the result values of the new parameter which can either be the *Maximum*, *Minimum* or *Average* of all currently predicted values. The thresholds are checked against these results.
- **Interval**: This value defines how often a calculation should be performed. This can either be: *Hourly*, *Daily*, *Weekly*, *Monthly* or whenever a value is received for the original parameter (Raw).
- **Horizon**: This value defines for how long into the future a prediction should be made. Here a reasonable interval should be selected, since the prediction gets more vague the farther it reaches into the future. A good prediction interval might have a quarter of the length in relation to the historic values that are used for the prediction.

The new parameters are displayed in the same monitor container as the monitor parameters from which they are derived. As a matter of fact, they are usual parameters that only have a special method to collect their data.

**Creating Thresholds for Trend Evaluation Monitor Parameters:**

Since trend evaluation monitor parameters are handled like any other parameter that receives numerical values as its results, thresholds can be handled in the same fashion as described in *Section 5.5.1*.

It only has to be kept in mind, that these parameters do not use actual values, but trigger their thresholds based on guesses, and on *all* values that are calculated for a single guess.

**Important Note**:
Since the prediction may generate noticeable load to the OpenScape FM server, by default monitors that use the early warnings will only be performed on a daily basis.

## 5.5.3 Monitor

The tab **Monitor** can be used to configure the basic attributes of a monitor:

- **Key Path**: The identifier of the monitor. It has to be unique for each monitor. Monitors that have the same identifier may not be used to monitor the same IP node.
  The key also determines how the monitor is displayed within the OpenScape FM. If the key contains „/“ characters, the OpenScape FM will interpret the key like a pathname and will display the monitor as a hierarchical construct, with a subsymbol for every path element.
- **Description**: This parameter describes the intended function of the monitor.
- **History**: This parameter defines the amount of result values that will be stored within the OpenScape FM for each parameter of the monitor. If the amount is exceeded the oldest value will be removed and the new result will be stored instead. These results and the results of the average history values can be displayed as described in *Section 5.3.1*.
  By default it is set to 288, which corresponds to one day of data if the monitor is executed every 5 minutes (see *Section 5.5.6*).
- **History - hourly average**: The amount of calculated hourly average values that are stored within the OpenScape FM for each parameter of the monitor.
  By default it is set to 168, which corresponds to one week of data.
- **History - daily average**: The amount of calculated daily average values that are stored within the OpenScape FM for each parameter of the monitor.
  By default it is set to 90, which corresponds to three month of data.
- **History - weekly average**: The amount of calculated weekly average values that are stored within the OpenScape FM for each parameter of the monitor.
  By default it is set to 52, which corresponds to one year of data.
- **History - monthly average**: The amount of calculated monthly average values that are stored within the OpenScape FM for each parameter of the monitor.
  By default it is set to 72, which corresponds to six years of data.
- **Monitor History**: This parameter defines the amount of monitor return values that will be stored within the OpenScape FM for the monitor. These values can be displayed as described in *Section 5.3.1*.
  By default it is set to 288, which corresponds to one day of data if the monitor is executed every 5 minutes (see *Section 5.5.6*).
- **Target IP:** The IP node to which the monitor symbol will be attached. When the variable `${targets}` is used, the monitor symbol will be attached to the symbol of the monitored system.
- **Export to Database (Reporting)**
  This selection menu determines whether the monitor data should be exported, for example, for later evaluation in a report.

*diabled* or *enabled* switches the export permanently off or on.
If *filtered* is selected, the data is filtered according to the configuration of the Data Export Monitor.
The XML file of the data export monitor can be found under
`<installation_directory>\ssma\conf\Data Export.xml`.
In this file it can be configured in the parameters `cimtype` and `excludeKeyPath` under which conditions the values are to be exported.

## 5.5.4 Sensor

On this tab the sensor, the actual monitoring function of the monitor, will be defined. This includes the type of the sensor (e.g. JavaScript) and a number of parameters that should be provided (e.g. IP address of the monitored systems or the user account that is to be used).

If a sensor is modified, these modifications only become effective after the corresponding agent has been restarted (see *Section 5.7.5*).

The type of the sensor will be selected with the selection list **Sensor**. It has the following entries:

- **Script-Sensor**
  This sensor type is based on XML sensor scripts.

  Two parameters are needed to define the access for this type:

  - **Script:** The path to the script file that should be used. The macro `${agent.confdir}` can be used to define a path in relation to the configuration directory of the agent. Path elements can be subdivided by using `${file.seperator}`.

    **Important Note**:
    For security reasons, scripts or programs can only be started from within the following directory or its sub directories:
    `<OpenScape FM Server Installation Directory>\ssma`
    If this restriction should be deactivated, e.g. for the internal agent within the folder
    `<OpenScape FM Server Installation Directory>\startup\conf`
    the file `OpenScapeFM.properties` has to contain the entry
    `ssma.restrict.programstart=false`.

  - **Timeout (Sec.)**: The time in seconds after which the execution of the script will be canceled.

    If `0` or `-1` is entered here, the script is not aborted regardless of the runtime.
    If a termination is intended, the minimum waiting time is 2 seconds. The default is 60 seconds.

3. **JavaScript-Sensor**:
   This sensor type uses Java script to collect the data.

   Two parameters are needed to define the access to the sensor:

   - **Script:** This parameter defines the path to the Java script that should be executed. The variable `${dir.sys.mgmt.templates.scripts}` can be used to reference the default directory for JavaScripts within the OpenScape FM installation. Path elements can be subdivided with `${file.seperator}`.

   - **Exceeding Time (Sec.)**: The time in seconds after which the execution of the script will be canceled.

4. **Method-Sensor**:
   This sensor type activates the selected Java method.

   The parameters are used to select the **Classpath**, the Java **Class** and the **Method** of the class, that are used by the monitor to trigger this sensor. The elements of the classpath can be separated with `${file.seperator}`.

5. **Log File-Sensor**:
   This sensor checks a selected log file.

   Two parameters are necessary to define this type:

   - **Log File**: Defines the path to the monitored file.
   - **Configuration File**: Defines the path to the file that contains the rules that are used to check the *Log File*.

The parameter values are listed in a table in the lower part or the window.

For the parameters *Username* and *Password* an arbitrary character string can be delivered.

Alternatively the macros `$_user{}` or `$_password{}` can be used, to reference user accounts that are configured like described in *Section 5.7.1*. The macros reference user ids of the type `*`. If a specific account type should be used, it can be assigned in curly brackets (e.g. for the type `ssh`: `$_user{ssh}` or `$_password{ssh}`) respectively.

## 5.5.5 Variables

The tab **Variables** is used to define the variables that are introduced by one or more sensors of the monitoring profile to which the monitor belongs. The tab is only available, if at least on variable of this type is declared within the profile.

All variables displayed here can be used in the sensor configuration of all sensors of the same profile (see *Section 5.5.4*).

The meaning of the respective variable is described by tooltips.

**Important Note:**
The variable contents are valid for the *Profile* that contains the current monitor. If other monitors contained in the current profile use the same variable, it will also be set to the configured value.

## 5.5.6 Execution Times

The tab **Execution Times** can be used to define the execution intervals or execution times for the monitor.

The pull down menu **Execution time (type)** defines the execution type of the monitor. The following types can be selected:

- **Interval (sec.):** The monitor will be executed once every X seconds, where X is the number entered in the field **Interval (sec.)**.

- **Cron time:** This selection will start the monitor automatically at defined points in time that correspond to the entry in the field **Cron time**. The expected input string is based on the UNIX cron time format. Five parameters separated by spaces are expected. The parameters are either a `*` (as a wildcard) or an integer. In order the parameters stand for **Minute**, **Hour**, **Day**, **Month**, **Weekday** (0=7=sunday, 1=monday, ...). The monitor will be started at times where *Minute*, *Hour*, *Month*, *Day* and *Weekday* match. When *Day* **and** *Weekday* are defined (not `*`) it is sufficient when either *Day* or *Weekday* match.
  Examples:
  - „`15 * * * *`“
    will execute every hour at 15 minutes past the hour (0:15, 1:15, ...)
  - „`0 0 * * *`“
    will execute every day at midnight
  - „`30 3 * * 1`“
    will execute every monday at 3:30am.
- **Single:** The monitor will be started automatically. But only once, when the respective agent is started. This option should be selected, when the monitor‘s sensor runs continuously.
- **Passiv:** The monitor will never execute its sensor automatically. The monitor will only be executed, if it is started manually (see *Section 5.5.7*).

### 5.5.7 Manual Execution

Monitors (and their sensors) will be executed automatically according to their configured execution times (see *Section 5.5.6*).

In addition they can be started manually by using the entry **Execute Monitor...** from their context menu. This entry will start a single execution of the monitor.

If there are monitors inside a Parameter Container (see *Section 4.2.4*), all monitors inside the container can be executed simultaneously using the entry **Execute all Monitors** from the context menu of the container.

## 5.6 Monitoring Profiles

The following describes the creation and management of monitoring profiles and how IP nodes are added to them.

### 5.6.1 Overview about the Monitoring Profiles of an Agent

Within the context menu of a System Management Agent object the entry **Configure** can be found. The page **Monitoring Profiles** of the configuration dialogue opened by this entry provides an overview of the System Management Agent‘s active monitoring profiles.

By using the buttons within the page:

- **Create**
  a new profile can be created (see *Section 5.6.2*).
- **Show**
  the configuration file of a selected profile can be shown (see *Section 5.6.6*).
- **Edit**
  a selected profile can be modified (see *Section 5.6.3*).
- **Delete**
  a selected profile can be deleted (see *Section 5.6.4*).

This view can alternatively be opened by using the main menu entry **Show Agents...** as described in *Section 5.1.2*.

## 5.6.2 Creating a Monitoring Profile

By using the main menu entry **System Management->Monitoring Profil**e the wizard to create a Monitoring Profile can be started.

On the starting page of the wizard, first the System Management Agent has to be selected for which the profile should be created.

Then in four steps the range of functions (*what* will be monitored - see *Section 5.6.2.1* and *Section 5.6.2.2*), the systems to be monitored (*who will be monitored*- see *Section 5.6.2.3*) and some meta information, like e.g the name of the profile (see *Section 5.6.2.4*), can be configured.

For a selected Agent the creation of a Monitoring Profile can be started by two additional methods:

- Using the list of System Management Agents (see *Section 5.1.2*).
- Using the Monitoring Profile overview of an Agent (see *Section 5.6.1*).

The configuration is then carried out using the same four configuration steps mentioned above.

### 5.6.2.1 Selection

The first tab provides a categorized selection of profile templates (left tree view). Generally these templates will be treated as individual modules that provide a single specific monitoring function (monitor). These are functions like e.g. the measurement of the CPU usage, of the main memory usage or a database ping.

In contrast, the category *Profiles* contains pre-built profiles which consist of several profiles. Clicking on a template displays a short description below the selection. An overview about the available templates can be found in *Appendix D*.

Selected templates can be moved into the list on the right side by using the **>>** button. All templates in the right list will finally be included into the monitoring profile. By clicking the button **<<** selected templates can be removed from the profile.

#### 5.6.2.2 Monitors

This page lists all monitors that are defined in the selected profiles.

It is possible to edit the name and description of the monitors. They can also be removed from the monitoring profile (the box **Include** has to be unchecked).

#### 5.6.2.3 Target IP Address

This page is used to specify the target systems. These are the systems that should be monitored by the monitoring profile. Located on the left is the object tree which displays only IP nodes on the lowest level. By selecting one or more IP nodes and pressing the button **>>** these nodes can be included to the monitoring. The button **<<** can be used to exclude IP nodes from the target system list of the profile.

By default the OpenScape FM system itself is already added.

#### 5.6.2.4 Creation

This tab provides the option to assign a name, a short description and an annotation to the created monitoring profile. These are used for informative purposes only.

By clicking the button **Save & Activate**, the created monitoring profile will be activated by the System Management Agent. After this, the monitors of the profile will start immediately with the measurement of the parameters on the system.

It has to be kept in mind, that probably access data for SSH or WMI connections to the selected target systems has to be provided, before the sensor function can be executed. More about this can be found in *Section 5.7.1*.

By clicking the button **Save as Template** no monitoring profile will be created and activated. Instead, a new template will be created. This can be used in the future for new monitoring profiles and can be found in the category *Custom* on the tab **Selection**.

### 5.6.3 Changing a Monitoring Profile

By clicking the button **Edit** within the overview of an agent‘s monitoring profiles (see *Section 5.6.1*) the wizard to edit a monitoring profile will be started. It has the same structure as the wizard to create a monitoring profile (see *Section 5.6.2*).

Within the tab **Selection** now by default the profile itself is selected as a **Template**. By adding additional templates, the profile can be expanded by further monitoring functions.

### 5.6.4 Deleting a Monitoring Profile

Clicking the button **Delete** within the overview of the monitoring profiles of an agent (see *Section 5.6.1*) deactivates and deletes the selected monitoring profiles.

This action cannot be revoked!

## 5.6.5 Adding and Removing IP Nodes

The container object of a System Management Agent contains the submap **Managed Monitoring Profiles** which contains an object for every monitoring profile that is active on the agent.

These monitoring profile objects contain the submap *targets* which contains an object for every IP node that is monitored by the respective profile. Data export modules contain the submaps *export_target* or *sm_export_target* instead.

While the **Managed Monitoring Profiles** container provides an overview about the monitoring profiles that are active on the agent, the **Remote** container, located on the same submap, provides an overview about the systems that are monitored by the agent.

The *Remote* container displays a container object for each system that is monitored by the agent. This object contains entries for the individual active Monitoring Profiles monitored by the agent on the respective system.

When a new target IP address is added to a monitoring profile, an object will be added to the related target container within the *Managed Monitoring Profiles* container, and entries within the *Remote* container are created.

**Important Note:**
An IP node that is not yet monitored by System Management can only be added if a respective System Management license is still available.

If the added IP address is represented by an IP node object within the OpenScape FM, the respective System Management objects will be added to the submap of the IP node object (see *Section 4.2.3*).

If the added IP address is *not* already represented by an IP node object within the OpenScape FM, by default the OpenScape FM tries to create this object (see *IP Manager Plugin User Guide*). If the creation is possible, the System Management information will also be added to the submap of this new object.

The automatic creation of IP nodes can be disabled for an agent by unchecking the Checkbox **Create IP nodes for systems automatically** on the tab **Agent Parameters** of the System Management Agent configuration. This can be opened by using the entry **Configure...** from the context menu of the System Management Agent object.

There are four methods to add IP nodes as target systems to a monitoring profile:

1. Direct handling of the target IP address list (see *Section 5.1.1*).
2. Using the wizard to handle the monitoring profile (see *Section 5.6.3*).
3. Directly adding a target IP address to a monitoring profile by using the entry **Add Target IP** from the context menu of the target container for the respective monitoring profile.
4. Target IP addresses can be added automatically by the System Management itself. For example, when the VMware monitoring detects a new VMware client (see *Section 7.2*).

IP nodes can be removed from the list of targets for a monitoring template by deleting the respective object within the *Managed Monitoring Profiles* or *Remote* container or by one of the wizards mentioned above.

### 5.6.6 Displaying the Configuration File of a Monitoring Profile

By clicking the button **Show** within the overview of an agent‘s monitoring profiles (see *Section 5.6.1*) the XML file will be opened that contains the configuration of the profile. It can be edited here.

It has to be kept in mind, that the System Management Agent cannot work with an invalid profile configuration. Manual changes of the configuration file are therefore not recommended. More about the structure of this file can be found in *Appendix E*.

## 5.7 Agents

The following describes the configuration and the management of System Management Agents by an OpenScape FM installation.

### 5.7.1 User Access for Monitored Systems

The context menu of a System Management Agent object contains the **Configure...** entry. This opens a configuration dialog with the tab **System Management->Monitored System Accounts** which allows the entry of user accounts for systems that should be monitored by profiles of the System Management Agent.

A large part of the System Management‘s monitoring functions use remote accesses to the monitored systems, using protocols like SSH or WMI, or direct data connections. In such cases it is necessary to provide the System Management Agent with the data of the respective user account. For SSH connections e.g. a Unix user account is needed, for WMI requests a Windows user account and for databases a database user account.

The dialog consists of a table in which each row represents a user account. When a system is monitored by the System Management Agent, and a user account is needed for this, a row with the respective target system will be automatically added to the table. The fields **User** and **Password/Key** are not filled automatically.

The field **Agent** defines the systems on which the user account is available. The fields **User** and **Password/Key** contain the user data that should be used. The password will be saved encrypted and will only be displayed when it is initially entered.

The field **Type** allows the specification of more than one user account for this system. If e.g. an SSH account and a database account are needed for the same system, they can be distinguished by using different types (e.g. `ssh` and `db`). To inform the agent which account should be used for which monitoring functions, additional configuration steps might be necessary. These are described in *Section 5.5.4*.

A standard value `*` for the type of an account shows, that this account will be used as the default account for this system.

## 5.7.2 Configuration Properties

The context menu of a System Management Agent object contains the **Configure...** entry. This entry opens a configuration dialog. Its tab **Configuration Properties** can be used to define variables for the configuration of sensors (see *Section 5.5.4*).

The dialog consists of a table in which each row represents the assignment of a value to a key-type pair. Generally the **Key** represents a reference object, while **Type** describes a property of the object. For example, if a respective service port should be assigned individually to a number of IP nodes, it suggests itself to use the IP address as the key and an identifier like `port` as the type.

Within the sensor configuration the macro `$_property` can be used to reference configuration properties. For example, the value of an argument can be set to `$_property{port, ${monitor.targetip}}`. During the execution of the sensor, it will replaced with the value of the configuration property that has the type `port` and its key which is the IP address of the respective monitored IP node.

If a referenced configuration property does not already exist, it will be automatically generated with an empty value.

## 5.7.3 Password Protection of the Agent

Using the factory settings, a System Management Agent can be managed and configured by every OpenScape FM that detects the agent. In distributed environments it may be useful to provide authentication data for specific agents, to prevent this behavior.

For this, the program **setAgentPassword** can be used. (Windows: `setAgentPassword.exe`, Linux: `setAgentPassword`). It can be found in the installation directory of the OpenScape FM (internal agent) or of the System Management Agent (external agent). After entering the current user name and password, an new user name and password can be entered. This change gets active after a restart of the agent (see *Section 5.7.5*).

The factory settings of an agent are:

User name: `ssma`
Password: `materna`

If the OpenScape FM detects a System Management Agent, it tries to authenticate itself on the agent by using the account name and password. If the access does not succeed, the agent cannot be managed.

The context menu of a System Management Agent object contains the **Configure...** entry. This opens a configuration dialog that contains the tab **Agent Account** which provides the means to change the user identification that will be used to authenticate on this agent.

Alternatively the main menu entry **System Management->Agent Account...** can be used to enter a global identification. This identification will be used as a default for newly detected System Management Agents.

## 5.7.4 Managing and Unmanaging of the Agents

If the OpenScape FM identifies a System Management Agent on a managed IP node, automatically the agent will also be managed.

This leads to the visibility of the complete hierarchy of the monitoring profiles, monitors and parameters, as well as the monitored systems of the agent within the object tree. In addition, the OpenScape FM receives events and status messages from this agent.

The managing of a System Management Agent is relevant in respect of the available licenses. The number of managed System Management Agents is limited by the number of licensed System Management Agents. If the managing of a System Management Agent would access the number of licensed System Management Parameters, the agent cannot be managed.

By not managing the agent, the mechanisms described are deactivated for this agent. This means that the object hierarchy of the agent stays invisible and events and status messages will not be received.

## 5.7.5 Restarting an Agent

The restart of an agents causes the rereading of the profile configurations (XML files).

Every configuration of monitoring profiles - or of other properties of a System Management Agent - by using the OpenScape FM user interface, leads to an automatic restart of this agent. This will be performed in the background.

If manual changes - e.g. on profile configurations (XML files) (see *Section 5.5.4*) or on login names of the agent (see *Section 5.7.3*) - should be activated, a manual restart of the agent has to be requested. This can be done by selecting the entry **Restart Agent** from the context menu of the System Management Agent object.

## 5.7.6 System Management Agent as a Service

During the installation of an external agent, the agent will be registered to the operating system as a service. Therefore it will be started automatically when the system is started.

On Windows systems the service OpenScape System Management will be registered. Stopping or starting the service, deactivates or activates the System Management Agent.

On Linux systems the installation directory of the agent contains the executable programs `stopAgent` to deactivate the agent, and `startAgent` to activate the agent.

Internal agents of an OpenScape FM cannot be deactivated independently of the OpenScape FM.

## 5.7.7 Backup Monitor

The System Management extents the OpenScape FM Backup Management by a mechanism that backups the configuration of System Management Agents (see *OpenScape FM Desktop User Guide*).

The backup service consists of a backup client that registers itself to the Backup Management Service of the OpenScape FM.

Individual System Management Agents can be added to the automatic backup process. Manual backups or restores can also be performed.

Generally no manuel configuration is necessary. If an OpenScape FM discovers a System Management Agent, it will be automatically added to the Backup Process of the OpenScape FM and the Backup Monitor of the System Management Agent will be configured accordingly.

The Backup Monitor of a System Management Agent can be deactivated to remove the respective agent from the backup process.

A manual configuration (see below) of a Backup Monitor might be necessary, if more than one OpenScape FM Server monitors its host system.

**Important Note**:
The Backup Monitor stores the configuration of the System Management Agents. It does NOT store the data collected by the agents.

### Configuring the Backup Monitor for an Agent

The backups are performed by a special monitor that is running on each agent that should be added to the backup process. A Backup Monitor therefore has to be configured individually for each respective System Management Agent.

The backup monitor template is named *Agent Backup* and can be found at the location
`Templates->Applications->Agent->Agent Backup`
when a new monitor is added to an agent. The monitor is added as usual (see *Section 5.6.2*). On the creation tab **Target IPs** the IP node of the OpenScape FM server has to be selected. The current agent will then be added to the Backup Management of the selected OpenScape FM.

To enable the backup monitor to connect to the OpenScape FM to which it should be added, the respective connection parameters have to be entered. This can be done on the page **Monitored System Accounts** within the agent‘s configuration (see *Section 5.7.1*). Here the entry for the respective OpenScape FM server with the type *backup* has to be configured. The entry will be automatically displayed if the monitor has been run at least once.

### Using the Backup Monitor for an Agent

When a backup monitor is configured and registered for an agent, a respective parameter symbol will be added to the monitor. The values of this parameter represent the results of the various monitor actions (registration, backup, restore).

Agent configuration backups are performed whenever an automatic or manual backup is performed for the OpenScape FM server (see *OpenScape FM Desktop User Guide*).

The listed backups for the OpenScape FM contain separate entries for the System Management Agents. These entries can be used to perform a manual restore for a selected entry.

**Important Note**:
Only backups of System Management Agents that are currently running and connected/registered to the OpenScape FM are displayed.

## 5.7.8 Log File of External Agents

The log file `agent.log` of an external stand-alone agent can be viewed from within the client.

To do this, the **Log File** entry must be selected from the context menu of the agent object or the **System Management** container object below the IP node object of the agent host.

# 6 Data Export

This chapter contains information about the data export of the System Management Agents.

## 6.1 Export Database

The OpenScape FM uses a **Export** database that stores the parameter data and the events that are gathered or triggered by managed System Management Agents. This database is the data source for the reports listed in *Section 6.2*. PostgreSQL (Postgres) and MySQL are supported.

To avoid an ‚Overflow' of the used database, by default, this data is kept for 30 days and an appropriate deletion process is started daily.

By using the central server settings
(main menu entry: **Server->Administration->Server Properties**)
this default settings can be configured differently on the page **Data Export**.

How long the data will be stored (**Clean Up Interval (days)**) and how often such a deletion should be performed (**Cleanup Check Interval (days)**) can be individually configured for stored events, status values and parameter data using the respective sub pages (**Event Export**, **Status Export** and **Parameter Export**).

For events and status values, this configuration concerns the internal database of the OpenScape FM server. For parameter values, it applies to the database assigned to the System Management Agent that is collecting the respective data.

For Parameter Data, it can also be determined for which of the system management agents such data should be exported at all (List: **Export Parameter Data for Agent**).

Only Parameter Data of agents that are marked in this list and whose symbol is *not* currently in the state '*Unmanaged*' will be exported.

**Important Note**:
If a deletion check is performed, all data of the respective type that is older than the deletion interval corresponding to the type is irrevocably deleted!

## 6.2 Reports

The following Reports contain data provided by System Management Monitors:

- CPU Usage (Monitor *CPU Usage**)
- File System (Monitor *File System**)
- Network Performance (Monitor *Network Performance*)
- Network Interfaces (Monitor *Network Usage**)
- Memory Usage (Monitor *Memory Usage**)

The monitors marked with * are by default automatically activated with the profile *Basic Monitoring* (see *Section 7.1.1*).

The reports mentioned above can be generated with the *Report Center*. More about this can be found in the *Report Manager Plugin User Guide*.

# 7 Special Monitoring Functions

This chapter introduces a number of profiles and monitors that contain basic and often used monitoring functions.

## 7.1 Pre-Installed Monitoring Profiles

The following monitoring profiles are activated automatically during the installation of the OpenScape FM or of System Management Agents.

### 7.1.1 Basic Monitoring

This profile contains monitors that observe basic performance data of the system:

- **CPU Usage***
  Monitors the usage of the system processors.
- **File System***
  Monitors the relative usage of the detected hard disc partitions of the system.
- **IO Statistics**
  Monitors the throughput in MB per second (write, read), the average wait time per IO request in milliseconds (write, read) and the requests in progress (queue length) for read/write combined.
- **Memory Usage***
  Monitors the usage of the main memory in percent and of the pagefiles (Windows) or Swaps (Linux).
- **Network Usage***
  Monitors the throughput, the error rate and the relative usage of all identified network interfaces of the system (with the exception of loopback interfaces).
- **Process Top**
  By default, identifies the 10 running processes that have the highest CPU usage.
- **System Info**
  Collects basic information about the system like the operating system, architecture, system time, processors, available main memory, available hard disk space and network interfaces.

The monitors marked with an * collect data for the reports mentioned in *Section 6.2*.

### 7.1.2 Internal Monitoring

This profile is only available for Internal Agents of the OpenScape FM. It contains monitors that control performance numbers of the running OpenScape FM installation:

- **Active Users**
  Monitors the number of signed on users for the OpenScape FM.

- **Disk Usage**
  Monitors the size of the OpenScape FM installation on the hard disk (including the Server database).
- **IP Polls**
  Monitors the average number of IP status and configuration updates per hour that are performed by the IP Manager.
- **Logfile**
  Monitors the Logfile of the server in respect of error messages.
- **Memory Usage**
  Monitors the main memory usage of the Java VM, in which the OpenScape FM is running.
- **SNMP Traps**
  Monitors the average number of SNMP traps per hour that are received by the IP Manager.

## 7.1.3 Network Monitoring

- **SipPing**
  Sometimes IP phones do not accept calls, even when they are still reachable over the network and have not reported any problem via SNMP. Attempts to establish a SIP session can be used to recognize such faults. If these do not succeed, the phone most-likely has a problem and an alarm should be generated.

  The SipPing monitor tries to perform a SIP INVITE or OPTIONS request to check a remote SIP stack.

  The status of the monitor is set to *Critical* and an event with a respective message is created, if the SIP stack does not respond or responds with a critical error. Else the status will be set to *Normal*.

  **Configuration**

  The *Target IP Address* is the IP node that represents the IP phone that should be checked.

  The *Target Call Number* is the phone number of the IP phone. This number is only needed when the phone is configured to check for its number during SIP requests.

  **Notes**:

  The monitor does not work with SIP-TLS (Transport Layer Security).

  On Windows systems the OpenScape FM service might need to be started with higher privileges.

## 7.1.4 Mail Monitoring

The Mail Monitor Profile checks whether the mail traffic between two servers runs smoothly. The configured SMTP server sends test mails with different attachment sizes to a configured IMAP server and notifies their runtime.

The profile contains an SMTP Monitor which sends test mails and monitors their delivery, and an IMAP monitor which monitors the traffic of the received mails.

The SMTP Monitor gets notified, if a message is received by the IMAP server. The monitor can therefore calculate the time the transfer of the individual mail has taken. This runtime can be monitored by individual thresholds (see *Section 5.5.1*).

To get the monitor profile to work, the mail address of the sender and recipient as well as the servers (SMTP and IMAP including ports) have to be configured. Also the Login Credentials for the individual mail accounts have to be configured.

To add or configure parameters, the **Sensor** tab within the configuration of the respective monitor can be used. Here, for example, the sending of further mails with different sized attachments can be handled.

### 7.1.5 Performance Management

This profile contains monitors that are based on the functionality of the *Performance Management*. More about this can be found in the *Performance Management User Guide*.

## 7.2 VMware Monitoring

The VMware Monitoring offers a simple but capable solution for the monitoring of virtual devices.

It provides the user with a number of overviews and status updates for VMware Servers, specific virtual machines and host systems.

### 7.2.1 VMware Status Monitor

The VMware Status Monitor can be used to monitor status critical parameters of virtual machines like e.g. CPU-, Memory- or Disk-Usage and the critical parameters Overall-Status, CPU- and Memory-Usage of host systems.

After the creation of the monitor (see *Section 7.2.2*), the monitor will automatically discover all virtual machines and host systems available to the configured user of the monitored VMware Server.

Therefore the monitor needs the login credentials and the appropriate rights on the monitored VMware Servers. The discovered virtual machines will be created as IP Nodes named with their respective hostnames. The same will happen to the host systems.

The status of these IP Nodes will change according to the status of the monitored parameters, which are located as child objects of the IP Nodes, and according to the configured status propagation.

Virtual machines and host systems of the VMware Server, which do not give information about their IP address, but which return a MAC address unknown to the OpenScape FM, will be shown in the system management folder of the agent.

The monitor can be executed manually on each virtual machine or host system by using the menu item **Execute Monitor…** in the context menu of the VMware Status Monitor on the specific IP Node. In addition the monitor will be executed regularly, depending on the configured time interval. By using the menu item **Show Values…** in the context menu of the VMware Status Monitor on a specific IP Node, system information will be displayed.

Most values will not change in a frequency higher than all five minutes, because the VM Server cannot provide values in shorter intervals.

## 7.2.2 Creation of a VMware Status Monitor

To create a VMware Status Monitor, the tab **Profiles** can be opened within the configuration window of the System Management Agent. The button **Create** starts the creation of new System Management monitors.

Within the **Selection**, the template for the VMware Status Monitor ("`templates->Applications->VMware->VMware Status`") can be chosen from the template list on left side of the dialog. The buttons **Back** and **Next** can be used to switch between the steps of the wizard.

The step **Monitors** provides configuration options for the **Short Description**, the **Annotation** of the selected monitors and for the property **VMware: Monitored counters**. This property contains a configurable list of performance counters (e.g. System Parameter) that should be monitored on the VMs (besides these are parameters that are always monitored). A list of Performance Counters known by the VMware Servers can be displayed by using the VMware Performance Monitor. But not all VMware Server support all Performance Counters. The VMware Status Monitor does not display the unsupported Performance Counters, even when they have been configured.

These are the only properties that should be changed in this step. The other properties are configured by the template and the monitor and should not be changed by a user.

The step **Target IPs** contains an instance of the navigation tree called **Object tree**, which contains the IP Nodes discovered by the OpenScape FM. The monitored VMware Servers have to be selected in the **Object tree** on the left side of the tab and added to the list **Selected objects** (by using the button **>>)**.

This means that VMware Servers that should be monitored have to be previously added to the OpenScape FM as an IP node. No IP nodes besides the VMware Servers should be selected.

In the step **Finish** a **Name**, a **Short Description** and a **Comment** can be chosen for the monitor. Then either the monitor can be created with the button **Save and Activate** or a custom template for VMware Status Monitors can be created with the button **Save as Template**. After the activation of the monitor the agent will be restarted automatically.

**Note:**
After a restart of the agent a connection to the VMware Server is added to the tab **Monitored System Accounts** of the System Management Agent. The **Target** of the connection equals the host name of the VMware Server, while the connection **Type** is "vmware-webservice". The login credentials for this connection have to be entered in the fields **User** and **Password** and saved afterwards.

## 7.2.3 VMware Performance Monitor

The VMware Performance Monitor can be used to monitor the performance counters (i.e. system parameters) of specific virtual machines of VMware Servers.

After the creation of the monitor (see *Section 7.2.4*), all known Performance Counters will be displayed in the Counter Overview.

The specified virtual machines on the monitored VMware Servers, for which the monitor's login credentials have the appropriate rights, will be listed under **Virtual Machine**.

The Monitor is located under "`System Management Agent->Remote->System Management` (Hostname of the VMware Server)" and "`IP Node of the VMware Server->System Management (Hostname of the Agent machine)`".

The structure of the VMware Performance Monitor consists of the name of the monitor with the child object **VMware Performance**, which will be created after the first execution of the VMware Performance Monitor. The object VMware Performance contains the **Counters Overview** and the object **Virtual Machine**. The tab **Current Value** of the Counters Overview shows all known counters of the VMware Server. The monitored virtual machines are all displayed as child objects of Virtual Machine and contain an overall summary as well as specific monitored parameters.

The monitor can be executed manually using the menu item **Execute Monitor** in the context menu of the VMware Performance Monitor. In addition the monitor will be executed regularly, depending on the configured time interval.

## 7.2.4 Creation of a VMware Performance Monitor

To create a VMware Performance Monitor, the tab **Monitoring Profiles** within the configuration window of the System Management Agent can be used. The button **Create** starts the creation of new System Management Monitors.

Within the **Selection** step, the template for the VMware Performance Monitor ("`templates->Applications->VMware->VMware Performance`") can be chosen from the template list on left side of the dialog. The buttons **Back** and **Next** can be used to switch between the steps of the wizard.

The step **Monitors** contains the **Short Description** and the **Comment** for the specific monitor as well as the following **Global Properties** for the VMware Performance Monitor:

- **VMware: Monitored counters** contains the list of system parameters to be monitored. All values that are displayed in the Counters Overview can be configured as a comma separated list. Usually not all displayed counters are supported by the VMware Server.
- **VMware: Monitored entities** contains the list of VMs to be monitored. The virtual machines are configured as a comma separated list of their names.
- **VMware: Sampling interval** contains the interval in which the monitor should be executed.

These are the only properties that should be changed in this step. All other properties are filled by the template and the monitor itself and should not be changed by the user.

The step **Target IPs** contains an instance of the navigation tree called **Object tree**, which contains the IP Nodes discovered by the OpenScape FM. The monitored VMware Servers have to be selected in the **Object tree** on the left side of the tab and added to the list **Selected objects** (by using the Button **>>)**.

This means that VMware Servers that are to be monitored have to be previously added to the OpenScape FM as an IP node. No IP nodes besided the VMware Servers should be selected.

In the step **Finish** a **Name**, a **Short Description** and a **Comment** can be set for the combined monitor. Then the monitor can either be created with the button **Save and Activate** or a custom template for VMware Performance Monitors can be created with the button **Save as Template**. After the activation of the monitor the agent will be restarted automatically.

**Note:**
After a restart of the agent a connection to the VMware Server is added to the tab **Monitored System Accounts** of the System Management Agent. The **Target** of the connection equals the host name of the VMware Server, while the connection **Type** is "vmware-webservice". The login credentials for this connection have to be entered in the fields **User** and **Password** and saved afterwards.

### 7.2.5 Creation of a Combined VMware Monitor

To create a combined VMware Monitor, consisting of a VMware Status and a VMware Performance Monitor, the tab **Monitoring Profiles** within the configuration window of the System Management Agent can be used. The button **Create** starts the creation of new System Management Monitors.

Within the **Selection**, templates for the VMware Performance and VMware Status Monitor ("`templates->Applications->VMware->VMware Performance/VMware Status`") can be chosen from the template list on the left side of the dialog. The buttons **Back** and **Next** can be used to switch between the steps of the wizard.

The step **Monitors** will then include the properties of both monitor templates explained in *Section 7.2.2* and *Section 7.2.4*. The only notable difference consists of the **property VMware: Monitored counters** being applied for both monitors.

The step **Target IPs** contains an instance of the navigation tree called **Object tree**, which contains the IP Nodes discovered by the OpenScape FM. The VMware Servers to be monitored have to be selected in the **Object tree** on the left side of the tab and added to the list **Selected objects** (by using the Button **>>)**.

This means that VMware Servers that should be monitored have to be previously added to the OpenScape FM as an IP node. No IP nodes besides the VMware Servers should be selected.

In the step **Finish** a **Name**, a **Short Description** and a **Comment** can be set for the combined monitor. Then the monitor can either be created with the button **Save and Activate** or a custom template for the combined VMware Monitor can be created with the button **Save as Module**. After the activation of the monitor the agent will be restarted automatically.

**Note:**
After a restart of the agent a connection to the VMware Server is added to the tab **Monitored System Accounts** of the System Management Agent. The **Target** of the connection equals the host name of the VMware Server, while the connection **Type** is "vmware-webservice". The login credentials for this connection have to be entered in the fields **User** and **Password** and saved afterwards.

## 7.3 Warm Standby Monitor

Warm Standby functionality of the OpenScape FM is realized by a distinct monitoring profile for the internal System Management Agent, which is delivered with each OpenScape FM. This monitoring profile has to run on the OpenScape FM server that should replace the monitored OpenScape FM in case of problems.

When the *WarmStandby Monitor* profile is activated, it will remotely monitor, backup, replace and restore another management server. The local management server (OpenScape FM slave) runs in **standby mode** and monitors the given remote management server (OpenScape FM master).

In order to use the monitoring profile, an administrative account of the remote master OpenScape FM has to be defined. This account is used to access and download the database of the master OpenScape FM server. It can be set within the configuration of the internal agent (see *Section 5.7.1*).

The configuration is simply done by activating the *WarmStandby Monitor* in the slave OpenScape FM, defining the host name/IP of the master OpenScape FM server and entering the administrator credentials (**User** and **Password**) of the master OpenScape FM server.

In standby mode, the slave OpenScape FM performs backups of the data of the master OpenScape FM every 5 minutes as a default. This intervall can be configured on the tab **Execution Times** within the configuration of the included **Backup** monitor.

Whether the master management server can be reached is checked by the included **Observer** monitor. By default this monitor checks the connection every 5 minutes. This intervall can be configured on the tab **Execution Times** of this monitor.

In case of a connection failure to the master management server the slave management server switches to **active mode**. It backups its own current configuration data and installs the backed up configuration of the master server onto itself and restarts. After the restart the slave management server will then carry out the tasks of the master management server.

In case the master OpenScape FM server is detected to be up again, the slave OpenScape FM switches back to standby mode. The behavior during the switch back can be configured by the following auto restore modes:

- **Auto Restore On**: If the slave management server gets connection to the master server again, using the **Restore** monitor, the slave server restores the current configuration of itself to the master management server and restarts it (**Auto Restore**).
- **Auto Restore Off**: The restore of the master server can be triggered manually by executing the **Restore** monitor on the slave.

Two scripts can be defined to perform actions during the transition phases:

- The script `standby2active` will be executed after the slave management server has switched from standby to active.
- The script `active2standby` will be executed, if the slave has switched back from active to standby mode.

Two further scripts can be defined that will be run before (`preScript`) and after (`postScript`) the backup of the master OpenScape FM server.

The tab **Configuration Properties** within the configuration of the respective agent of the slave OpenScape FM server can be used to define the actual files that will be executed when scripts are triggered. The individual possible scripts are identified by the host IP of the slave server (column **Key**) and the name of the individual script (column **Type**). The matching **Value** field should contain a path to the file that should be executed (including the file name itself).

# A 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 tooltips for the corresponding right symbols (tree or submap).

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

# B Hardware and Software Requirements

The System Management is a part of the OpenScape FM installation. The system requirements can be found in the *OpenScape FM Desktop User Guide*.

The respective Release Notes should also be considered.

The monitoring of parameters on Windows systems is performed using WMI. It has to be kept in mind that, due to technical reasons, WMI requests can only be transmitted from Windows systems. As a result, Windows systems can only be monitored by System Management Agents that themselves are installed on a Windows system. In addition, the service WMI Performance Adapter must run on the monitored system, and rights to perform WMI requests have to be granted.

# C Usage of Value Graphs

By default, histories of numerical parameter values (see *Section 5.3.1*) are visualized as value graphs:

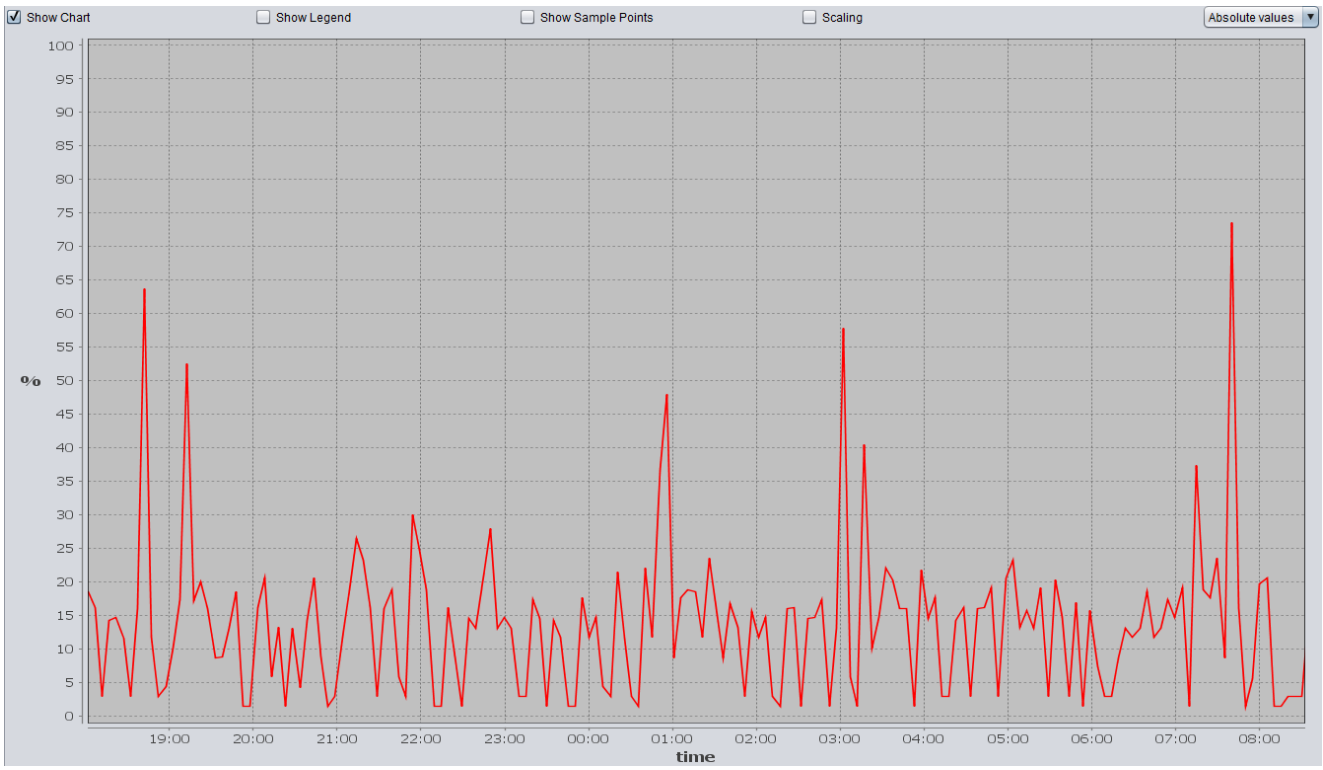



*Figure 4 Value Graph*

The historical values of the parameter value (y-axis) will be displayed against the respective measurement times (x-axis).

# C.1 Display Options

Above the graph, the following options can be found:

- **Show Chart**
  Switches between the graph and the table view.
- **Show Legend**
  Activates or deactivates a legend below the graph. A legend might be useful when the history of several parameters is displayed (see *Section 5.3.5*).
- **Show Sample Points**
  Activates or deactivates the display of measuring points within the graph.
- **Scaling**
  Shows or hides extended controls that allow the adjustment (scaling and movement) of the graph.

## C.1.1 Using the Mouse

By dragging the mouse, an area can be selected within the graph. After releasing the mouse button, the view will be enlarged for the selected area. This can be reverted by using the entries of the context menu **Zoom Out**.

## C.1.2 Context Menu

Right clicking the graph area opens a context menu with the following entries:

- **Show Series**
  Allows the selection of the displayed graphs.
- **Print Separator**
  Enables the display of a separator to mark days, weeks or months on the time axis.
- **Properties...**
  Opens a dialog for the detailed configuration of the display (like header, colors).
- **Copy**
  Copies the graph as a graphic with alpha channel (32 bit) to the clipboard. This can be pasted into popular picture editing programs.
- **Save As...**
  Stores the graph as a PNG graphic with alpha channel (32 bit).
- **Print...**
  Prints the graphic.
- **Zoom In / Zoom Out / Auto Range**
  Can be used to adjust the graph.

## C.2 Aggregation of Values

Above the graph, on the right side, a selection list can be found that provides different methods to aggregate the values.

*Absolute Values* is selected by default. This will display the graph as raw data without any aggregation. An aggregation based on intervals like *Hourly Average* aggregates the data as averages for the respective interval.

# D Available Monitoring Profiles and Monitors

The following monitoring profiles will be automatically activated during the initialization of the System Management Plugin.

**Note**:
Additional Monitors are described in the *Web User Guide*.

## D.1 Basic Monitoring

The following basic monitoring functions are functions that can monitor the local system as well as an arbitrary number of remote systems (Windows and Linux) when the necessary login data is given.

The monitoring of Unix systems uses SSH, while the monitoring of Windows systems (Windows Vista or newer) uses WMI.

### D.1.1 CPU Usage

Monitoring of the average processor load of all processes on the target systems during configurable time intervals.

### D.1.2 Memory Usage

Monitoring of the current usage of the physical memory (RAM), the swaps or page files and the collective virtual memory of the target systems.

### D.1.3 File System

Monitoring of the current usage and the total capacity of the physical and virtual file system of the target systems.

### D.1.4 Network Usage

Monitoring of the throughput, the load and the relative error rate of network interfaces for the target systems during configurable time intervals.

### D.1.5 Process Top

Listing of the processes that currently use the most CPU time and monitoring of the currently highest CPU usage for a single process.

### D.1.6 System Info

Listing of basic information about the target system (hardware, operating system).

## D.2 Internal Performance Monitoring

### D.2.1 Active Users

Listing of the users currently logged into the OpenScape FM.

### D.2.2 Disk Usage

Hard disc space required by the OpenScape FM installation.

### D.2.3 IP Polls

Number of performed status and configuration updates since the last monitoring of the average number per hour.

### D.2.4 Logfile

Monitoring of the server logfile in respect of occurred error messages.

### D.2.5 Memory Usage

Current memory usage of the Java Virtual Machine (JVM) on which the server is running.

### D.2.6 SNMP Traps

The number of contained SNMP traps since the last check of the average number per hour.

# D.3 Technology Specific Monitoring

## D.3.1 Service Workbench

This profile monitors the status of a Service Workbench installation as well as its application server. For new OpenScape FM installations, which include the Service Workbench, this profile will be activated by default.

## D.3.1 OpenScape Voice

The following profiles are provided by the System Management to collect and monitor OpenScape Voice specific data:

- **SIP Statistics**
  Provides SIP Statistic data for OpenScape Voice environments.
- **OpenScape Voice - Call Admission Control**
  Collects data about used bandwidth and the number of concurrent calls for CAC Groups.
- **OpenScape Branch - Registered Subscribers**
  Checks the number of registered subscribers for OpenScape Branches.
- **Mediatrix Monitoring**
  Checks the number of used ports and general information for Mediatrix devices.
- **MediaServer**
  Collects data about the current number of connections, the number of conference endpoints and the number of G711 and G729 streams for UC media servers.
- **UC Backend**
  The UC backend cluster is based on a Linux high availability solution which can be monitored by the profile „OS/Linux/Linux Cluster“. The profile monitors cluster nodes, resources and their status.
- **Oracle SBC**
  Below „Telephony“, there is a monitoring profile for an Oracle (ACME) session border controller. It is based on SNMP and shows important system parameters like CPU usage, memory usage etc.

These profiles are described in detail within the *OpenScape Voice User Guide*.

## D.3.2 Microsoft Exchange Server

This profile monitors the status of the services required by a Microsoft Exchange Server.

### D.3.3 Active Directory

This profile monitors the status of services required by an Active Directory server, as well as the status of its LDAP database.

### D.3.4 Citrix Environment

This profile consists of specific monitors for the components Delivery Controller, Provisioning Services and Database Servers of the Citrix environment.

The profile monitors the status of TCP ports and the Windows services of the components, e.g. CitrixBrokerService, CitrixHostService, CitrixMachineCreationService (DDC), BNPXE, SoapServer and StreamService (PVS).

## D.4 WarmStandby Monitoring

The WarmStandby Monitoring remotely monitors, backups, replaces and restores another management server. The local management server (OpenScape FM slave) runs in *standby mode* and monitors the given remote management server (OpenScape FM master).

If the monitored management server fails, the monitoring management server takes over until the monitored management server is back and running.

A detailed description can be found in *Section 7.3*.

### D.4.1 Backup

This monitor regularly backups the data of the monitored OpenScape FM server.

### D.4.2 Observer

This monitor regularly checks whether the monitored OpenScape FM is working.

If a failure is detected, the data from the last backup (see *Section D.4.1*) is installed on the local OpenScape FM, and the local OpenScape FM is restarted replacing the failed OpenScape FM server.

### D.4.3 Restore

This monitor is manually or automatically used to copy the local data to the monitored OpenScape FM server, if this server has recovered from a failure. The monitored OpenScape FM server then resumes its tasks.

# E New Monitoring Functions

The System Management (SM) plugin includes monitoring capabilities for many common systems and applications. These monitoring functions (so-called *Monitors*) can be configured easily (in terms of their behavior, e.g. polling intervals). The monitors are executed by the *SM-Agent*. Monitors are used to e.g. query the CPU, hard drive, memory and network usage of Windows- and Unix-based PCs,

The existing monitoring capabilities can be extended by new custom monitors quite easily.

This attachment explains how to create an SM-Monitor in form of a script and how to integrate it into the SM. At first, two basic monitor scripts for Unix- and Windows platforms are presented. Subsequently the configuration of the System Management Agent will be extended to include the scripts into the SM.

## E.1 Overview

Each SM-Monitor is assigned to an IP address. In the simplest case, this is the IP address of the system on which the SM-Agent is installed. However, a SM-Monitor can also be assigned to an external IP address in case it does not monitor parameters of the local system, but of the system addressed by the external IP-address.

In the client, the SM-Monitors are displayed within a container named like the monitoring agent. These SM-Monitors include one or more SM-Parameter objects, which represent the results of each monitoring script.

For example, an SM-Monitor to query the disk usage on a PC contains one SM-Parameter for each disk drive. This parameter represents the result of the query and the System Management state (e.g. *normal* or *critical*) of the monitored drive. It offers the possibility to view the values determined by the script (e.g. `disk usage = 26.3%`) as well.

## E.2 User Defined Monitoring Functions

Monitors can be included in form of any executable file, which prints its result in a presumed output format into the *Standard Error Channel*.

For example: On Unix based systems, arbitrary shell scripts can be included. On Windows based systems, Visual Basic Scripts can be included.

### E.2.1 Output Format to Generate SM-Parameters

The output to generate an SM-Parameter is interpreted by the System Management Agent and must therefore follow a fixed format:

```
#datatype|name|status|value|shortmessage|longmessage
```

The format begins with a hash ("`#`"), followed by six data fields separated by the pipe symbol ("`|`").

The six data fields have the following meaning:

1. **datatype**

   The parameter's data type. Allowed values are `Boolean`, `Long`, `Double` and `String`. In addition the data types `DoubleMap` and `LongMap` can handle multiple combined values for each parameter.

2. **name**

   The name of the parameter and the label of the parameter symbol.

3. **status**

   The status of the parameter. This value can be set in relation to the result provided by the script. The most common values are: *Unset*, *Unkown*, *Normal*, *Warning*, *Minor*, *Major* and *Critical*. The System Management generates an event each time the status of a parameter changes.

4. **value**

   The value of the output parameter. This value must be compatible with the specified data type. In addition, numerical values (of data type `Long` or `Double`) will be displayed as a graph within the result view of the parameter.

5. **shortmessage**

   A short textual output of the parameter/script. This value is optional. The System Management generates an event each time the shortmessage of a parameter changes.

6. **longmessage**

A detailed textual output of the parameter/script. This value is optional.

## E.3 Sample Scripts

The following section presents two simple examples for Unix and Windows.

The scripts consist of three parts:

- The **Initialization block** which initializes common variables and sets required environment variables.
- The **Implementation block** which implements the logic and functionality of the script logic.
- The **Output block** which exports the calculated / queried values and transfers them to the System Management Agent.

### E.3.1 Sample Script for Linux / Unix

The following example shows how to extend the SM-Agent by a user defined shell script (`.sh`) on a Unix-based platform. Shell scripts can make use of common shell commands and operations as well as of all tools available for the system like e.g. `awk`.

The script shown in `example.sh` generates a simple, static SM-Parameter named *DemoParameter*.

The initialization block is not needed in this example and remains empty.

Inside the implementation block 6 variables are assigned with values. Furthermore, the variable `shortmsg` is assigned with the first command line argument that gets passed to the script (line 14).

In the output block, the variable assignments are printed out (line 22).

The command `echo 1>&2` prints the output into the S*tandard Error Channel* and passes the result to the SM-Agent, that will then generate the SM-Parameter from these values.

**example.sh**

```
01 #!/bin/sh
02 # ***************
03 # initialization
04 # ***************
05
06 # ******************************
07 # user defined status calculation
08 # ******************************
09
10 name="DemoParameter"
11 status="Normal"
12 datatype="Long"
13 value=263
14 shortmsg="first argument: $1"
15 longmsg="shell script example"
16
17
18 # ***********************
19 # write output to stdErr
20 # ***********************
21
22 echo 1>&2 "#$datatype|$name|$status|$value|$shortmsg|$longmsg";
```

## E.3.2 Sample Script for Windows

A System Management Agent that is installed on a Windows platform can execute monitors which are (among others) programmed in *Visual Basic Script* (`.vbs`).

The following sample script `example.vbs` corresponds to the logic of the Unix shell script shown above. It generates an SM-Parameter named *DemoParameter*. This parameter has the data type `Long`, the constant value of `263` and the status `Normal`.

If the script is executed with a command line argument, this argument will be printed in the `shortmsg` of the SM-Parameter.

The initialization block contains two variable assignments. The variable `out` refers to the S*tandard Error Channel*. By using the command `out.write`, the calculated values can be transmitted to the SM-Agent (line 25).

The variable `argsNamed` refers to the command line arguments which were passed to the script.

The lines 17-19 check whether an argument has been passed. If this is the case, the first argument is appended to the variable `shortmsg`.

**example.vbs**

```
01 ' ***************
02 ' initialization
03 ' ***************
04 Dim out: Set out              = WScript.stdErr
05 Dim argsNamed: Set argsNamed  = WScript.Arguments
06
07 ' ******************************
08 ' user defined status calculation
09 ' ******************************
10 Dim name:     name     = "DemoParameter"
11 Dim status:   status   = "Normal"
12 Dim datatype: datatype = "Long"
13 Dim value:    value    = 263
14 Dim shortmsg: shortmsg = "first argument: "
15 Dim longmsg:  longmsg  = "visual basic script example"
16
17 If argsNamed.Count > 0 Then
18     shortmsg = shortmsg & argsNamed.Item(0)
19 End If
20
21 ' ***********************
22 ' write output to stdErr
23 ' ***********************
24
25 out.write "#" & datatype & "|" & name & "|" & status & "|" _
26 & value & "|" &  shortmsg & "|" & longmsg
```

## E.4 Integration of Custom Scripts into the Agent Configuration

The System Management Agent is configured via XML files which can be created and expanded by hand. These configuration files are placed inside the installation directory in the subfolder `/ssma/conf`. The format of the configuration files is defined by the DTD `AgentConfiguration.dtd`.

New custom scripts and their configuration files can be placed inside a new subdirectory of this folder.

For this example, a subfolder called `howto` is created, and the new script files (`example.sh` or `example.vbs`) will be copied into this folder.

**Note**:
On Unix-based systems, the executable flag for the script `example.sh` has to be set. It can be set by using the command „`chmod a+x example.sh`“.

Within the new directory (`ssma/conf/howto`) a new file with the name `AgentConfig.xml` is created.

This file defines which scripts will be executed. It also defines the intervals between the script launches and which command line arguments are used. The file `EmptyAgentConfig.xml` from the directory `ssma/conf` can be used as a template for the configuration file.

The following scripts show the SM-Agent configurations for the two examples described above.

**AgentConfig.xml (Unix Shell Skript)**

```
01 <?xml version="1.0" encoding="iso-8859-1"?>
02 <!DOCTYPE monitors SYSTEM 'AgentConfiguration.dtd'>
03
04 <monitors>
05    <monitor key_path="SH Monitor" description="my shell monitor"
       history="20" target_ip="local">
06       <script_sensor script="${agent.confdir}${file.separator}example.sh">
07          <argument>eins</argument>
08       </script_sensor>
09       <interval interval="180"/>
10    </monitor>
11 </monitors>
```

**AgentConfig.xml (Windows VBS Skript)**

```
01 <?xml version="1.0" encoding="iso-8859-1"?>
02 <!DOCTYPE monitors SYSTEM 'AgentConfiguration.dtd'>
03
04 <monitors>
05    <monitor key_path="VBS Monitor" description="my vbs monitor"
       history="20" target_ip="local">
06       <script_sensor script="cscript">
07          <argument> ${agent.confdir}${file.separator}example.vbs </argument>
08          <argument>eins</argument>
09       </script_sensor>
10       <interval interval="180"/>
11    </monitor>
12 </monitors>
```

The actual configuration of the SM-Agent starts with line 5 in both cases. Within the XML tag `monitors` (line 4) any number of monitor configurations can be defined.

A monitor configuration (line 5) consists of the following attributes:

- `key_path`: The name which is used to represent the new SM-Monitor.
- `description`: A short description of the monitor.
- `history`: The number of the last request results that are to be stored
- `target_ip`: The IP-address of the node to which the SM-Monitor is assigned. The value `local` represents the system on which the SM-Agent is installed.

The script that should be executed by the Agent is defined differently for VBS and shell scripts

For shell scripts, the script interpreter is specified in the first line of the script (*example.sh*, line 1). Therefore shell scripts can be directly executed by the SM-Agent.

The path of the script is set by the XML tag `<script_sensor>` and the attribute "`script =...`" (*AgentConfig.xml (Unix Shell Skript)* line 6). Via the XML tags `<argument>` any number of command line arguments can be passed to the shell script (line 7).

By default, Visual Basic scripts are assigned to the script interpreter `wscript.exe`, which is not able to generate an output. Therefore these scripts need to be explicitly executed via the interpreter `cscript.exe`. For this, the attribute "`script`" of the XML tag <`script_sensor`> has to be set to "`cscript`" (*AgentConfig.xml (Windows VBS Skript)*, line 6). The path of the script file is then specified via the first argument (line 7). Every additional argument will be passed to the script as a command line argument (line 8).

The XML tag `<interval interval="180"/>` defines the execution interval of the monitor in seconds (*AgentConfig.xml (Unix Shell Skript)*, line 9). With these configurations, the example scripts will be executed every three minutes and the result will be stored as a history value.

After all files have been created within the folder `ssma/conf/howto` as described, the SM-Agent has to be restarted. The new configuration files will then be evaluated by the Agent and the new SM-Monitors will be created within the SM. *Figure 5* shows the newly created *SH Monitor* from *AgentConfig.xml (Unix Shell Skript)* on a Linux machine. The corresponding shell script from *example.sh* created the SM-Parameter *DemoParameter*.

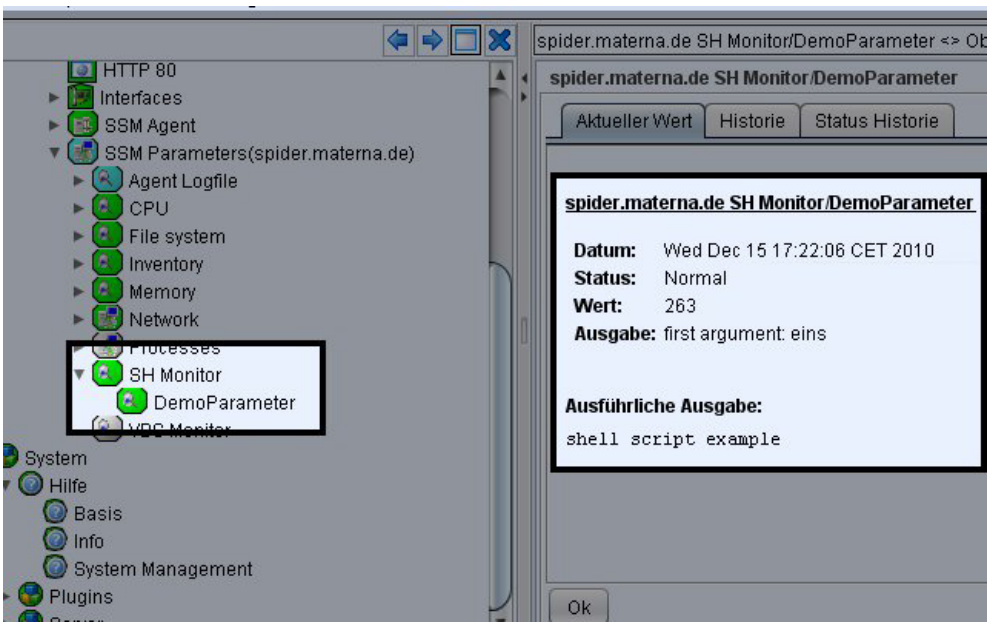



*Figure 5 New Monitor and Parameter*

# F Glossary

**OpenScape FM:** OpenScape Fault Management, the Unify Umbrella Management Platform for the OpenScape Enterprise Convergence Architecture. OpenScape FM is a web based client/server application for the management of heterogenous networks consisting of Unify telecommunication systems, IP systems and VoIP devices.

**System Management for OpenScape FM:** The System Management for OpenScape FM provides a variety of system management functions. It is a web-based client server software for the visualization and management of hierarchically structured networks and network topologies. It includes agent software which is used to monitor resources of remote systems. The agent software is written in java and runs on almost any platform for which a JVM exists. The monitoring functions used by the agent can be implemented in the language of choice for the target platform, e.g. Visual Basic on Windows or c-shell on UNIX systems.

**System Management Agent:** The System Management Agent enables the OpenScape FM to monitor various system resources (e.g. disk usage, process states) of network devices. The System Management Agent can be extended by functions (e.g. scripts) performing automatic or manual initiated system specific management tasks (e.g. starting backups, removing temporary files/logfiles).

**System Management Plugin:** Component which enables the OpenScape FM to communicate with System Management Agents.

**JVM:** Java Virtual Machine, the operating system specific software which is required on a computer in order to run Java programs. The System Management is implemented in Java and can therefore be run on virtually any platform for which a JVM exists.

# Index

## T



## U



## V



## W