

Nils Scharrnbeck

**Automatisierung von Monitoring-Aufgaben  
im Bereich des IT-Infrastruktur-Betrieb**

eingereicht als

DIPLOMARBEIT

an der

HOCHSCHULE MITTWEIDA

---

UNIVERSITY OF APPLIED SCIENCES

Mathematik, Physik, Informatik

Mittweida, 2009

Nils Scharnbeck

**Automatisierung von Monitoring-Aufgaben  
im Bereich des IT-Infrastruktur-Betrieb**

eingereicht als

DIPLOMARBEIT

an der

HOCHSCHULE MITTWEIDA

---

UNIVERSITY OF APPLIED SCIENCES

Mathematik, Physik, Informatik

Mittweida, 2009

Erstprüfer: Prof. Dr.-Ing. Mario Geißler

Zweitprüfer: Dipl.-Ing. Ulrich Clara

Vorgelegte Arbeit wurde verteidigt am:

Bibliographische Beschreibung:

Scharrnbeck, Nils:

Automatisierung von Monitoring-Aufgaben im Bereich des IT-Infrastruktur Betrieb -  
83 S. Mittweida, Hochschule Mittweida, Fachbereich Mathematik, Physik, Informatik  
Diplomarbeit, 2009

Referat

Ziel der Diplomarbeit ist es, eine generische Schnittstelle für das Monitoring von Anwendungen zu definieren und als Prototyp umzusetzen. Zuerst wird auf die Themen Monitoring und Simple Network Management Protocol (SNMP) eingegangen. Anschließend werden die für die Schnittstelle benötigten Dateien definiert. Im vorletzten Punkt erfolgen die Beschreibung der Umsetzung und die Dokumentation des erstellten Skriptes. Am Schluss dieser Arbeit wird kurz an einem Beispiel die Funktionalität der Schnittstelle nachgewiesen.

## **Inhaltsverzeichnis**

|  |           |
|--|-----------|
| <b>ABKÜRZUNGSVERZEICHNIS</b>   | <b>1</b>  |
| <b>ABBILDUNGSVERZEICHNIS</b>   | <b>2</b>  |
| <b>TABELLENVERZEICHNIS</b>   | <b>3</b>  |
| <b>VORWORT</b>   | <b>4</b>  |
| <b>AUFGABENSTELLUNG UND ABGRENZUNG DER DIPLOMARBEIT</b>                          | <b>5</b>  |
| <b>0 EINLEITUNG</b>  | <b>7</b>  |
| <b>1 EINFÜHRUNG IN DAS MONITORING</b>  | <b>8</b>  |
| <b>1.1 Definition und betrachtete Arten des Monitorings</b>                      | <b>8</b>  |
| 1.1.1 Definition Monitoring  | 8         |
| 1.1.2 Echtzeitmonitoring   | 8         |
| 1.1.3 Trendmonitoring  | 9         |
| <b>1.2 Definition und Bedeutung von Grenz- und Schwellwerten beim Monitoring</b> | <b>9</b>  |
| 1.2.1 Definition Grenzwerte  | 9         |
| 1.2.2 Definition Schwellwerte  | 9         |
| 1.2.3 Festlegung von Grenz- und Schwellwerten                                    | 10        |
| <b>2 DAS SIMPLE NETWORK MANAGEMENT PROTOCOL</b>                                  | <b>11</b> |
| <b>2.1 Überblick über das Simple Network Management Protocol</b>                 | <b>11</b> |
| <b>2.2 Beschreibung der wichtigsten SNMP-Versionen</b>                           | <b>12</b> |
| <b>2.3 Vergleich der SNMP-Versionen</b>  | <b>14</b> |
| <b>2.4 Management Information Base und Object Identifier</b>                     | <b>15</b> |
| 2.4.1 Managed Objects und Management Information Base                            | 15        |
| 2.4.2 Aufbau einer Management Information Base                                   | 16        |
| 2.4.3 Object Identifier  | 18        |

|              |  |           |
|--------------|--|-----------|
| <b>2.5</b>   | <b>SNMP-Operationen</b>  | <b>19</b> |
| 2.5.1        | Die get-Operation  | 20        |
| 2.5.2        | Die getnext-Operation  | 20        |
| 2.5.3        | Die getbulk-Operation  | 21        |
| 2.5.4        | Anmerkung  | 23        |
| <b>3</b>     | <b>DIE SCHNITTSTELLE</b>   | <b>24</b> |
| <b>3.1</b>   | <b>Vorstellung möglicher Lösungsansätze</b>                              | <b>24</b> |
| 3.1.1        | Eigener SNMP-basierter Lösungsansatz                                     | 24        |
| 3.1.2        | IBM Tivoli Monitoring Agent  | 25        |
| 3.1.3        | Nagios Remote Plugin Executor (NRPE)                                     | 26        |
| 3.1.4        | Vergleich der Lösungsansätze   | 28        |
| <b>3.2</b>   | <b>Definition der Schnittstelle</b>                                      | <b>29</b> |
| 3.2.1        | Bestandteile der Schnittstelle   | 29        |
| 3.2.2        | Namenskonventionen für die Konfigurations- und Messwertedatei            | 30        |
| 3.2.3        | Definition der Konfigurationsdatei                                       | 30        |
| 3.2.4        | Definition der Messwertedatei  | 39        |
| 3.2.5        | Leistungsumfang des Skriptes   | 40        |
| 3.2.6        | Beschreibung der Hilfsdateien „processdata.conf“ und „currentstatus.tmp“ | 40        |
| <b>3.3</b>   | <b>Abbildung von Messwerten auf OID's</b>                                | <b>42</b> |
| 3.3.1        | Abbildung von einzeiligen Skriptergebnissen mittels exec                 | 42        |
| 3.3.2        | Abbildung von mehrzeiligen Skriptergebnissen mittels exec                | 44        |
| 3.3.3        | Abbildung eines Skriptergebnisses mittels extend                         | 46        |
| 3.3.4        | Vergleich der Abbildungsmöglichkeiten                                    | 50        |
| <b>3.4</b>   | <b>Verwaltung und Struktur der zu verwendenden OID's</b>                 | <b>51</b> |
| <b>3.4.1</b> | <b>Verwaltung der OID's</b>  | <b>51</b> |
| <b>3.4.2</b> | <b>Struktur der OID's</b>  | <b>51</b> |

|            |  |           |
|------------|--|-----------|
| <b>4</b>   | <b>UMSETZUNG UND DOKUMENTATION DER SCHNITTSTELLE</b>                       | <b>56</b> |
| <b>4.1</b> | <b>Voraussetzungen für den Einsatz der Schnittstelle</b>                   | <b>56</b> |
| <b>4.2</b> | <b>Beschreibung der Funktionsweise der Schnittstelle</b>                   | <b>56</b> |
| <b>4.3</b> | <b>Beschreibung der Funktionsweise der einzelnen Skriptparameter</b>       | <b>59</b> |
| 4.3.1      | Parameter -h   | 59        |
| 4.3.2      | Parameter -l   | 59        |
| 4.3.3      | Parameter -a   | 60        |
| 4.3.4      | Parameter -d   | 60        |
| 4.3.5      | Parameter -r   | 60        |
| 4.3.6      | Parameter -u   | 62        |
| 4.3.7      | Parameter -o   | 63        |
| 4.3.8      | Parameter -i   | 63        |
| 4.3.9      | Parameter -c   | 64        |
| 4.3.10     | Parameter -e   | 65        |
| 4.3.11     | Parameter -v   | 65        |
| 4.3.12     | Hinweise zu den Parametern -r und -u                                       | 65        |
| <b>4.4</b> | <b>Fehlermeldungen</b>   | <b>66</b> |
| <b>5</b>   | <b>TEST DER SCHNITTSTELLE</b>  | <b>67</b> |
| <b>5.1</b> | <b>Test der einzelnen Skriptparameter des Verarbeitungsskriptes</b>        | <b>67</b> |
| <b>5.2</b> | <b>Test zur Verhinderung des gleichzeitigen Schreib- und Lesezugriffes</b> | <b>67</b> |
| <b>5.3</b> | <b>Nachweis der Schnittstellenfunktionalität</b>                           | <b>67</b> |
| <b>6</b>   | <b>ZUSAMMENFASSUNG UND AUSBLICK</b>  | <b>68</b> |

|   |           |
|---|-----------|
| <b>ANLAGEN</b>  | <b>69</b> |
| <b>Anlage A Wertebereiche und Beschreibung von SNMP-Datentypen</b>                    | <b>69</b> |
| <b>Anlage B Befehlsübersicht des Skriptes</b>   | <b>71</b> |
| <b>Anlage C Bestandteile des allgemeinen Teils einer Konfigurationsdatei</b>          | <b>72</b> |
| <b>Anlage D Bestandteile des messwertspezifischen Teils einer Konfigurationsdatei</b> | <b>74</b> |
| <b>Anlage E Zulässige Zeichen in der Schnittstelle</b>                                | <b>76</b> |
| <b>Anlage F Anleitung zur Schnittstellenerstkonfiguration</b>                         | <b>77</b> |
| <b>Anlage G Anleitung für den Anwendungsverantwortlichen</b>                          | <b>79</b> |
| <b>Anlage H Anleitung zur Fehlerbeseitigung in Konfigurationsdateien</b>              | <b>79</b> |
| <b>Anlage I Ergebnisse des Skripttestes</b>   | <b>80</b> |
| <b>LITERATURVERZEICHNIS</b>   | <b>82</b> |
| <b>ERKLÄRUNG ZUR SELBSTSTÄNDIGEN ANFERTIGUNG DER ARBEIT</b>                           | <b>83</b> |

## Abkürzungsverzeichnis

|         |  |
|---------|--|
| ASCII   | American Standard Code for Information Interchange |
| IANA    | Internet Assigned Numbers Authority                |
| IETF    | Internet Engineering Task Force                    |
| ITM     | IBM Tivoli Monitoring                              |
| MIB     | Management Information Base                        |
| MO      | Managed Object                                     |
| NRPE    | Nagios Remote Plugin Executor                      |
| OID     | Object Identifier                                  |
| RFC     | Requests for Comment                               |
| SNMP    | Simple Network Management Protocol                 |
| SNMPv1  | SNMP Version 1                                     |
| SNMPv2  | SNMP Version 2                                     |
| SNMPv2p | Party-Based SNMP Version 2                         |
| SNMPv2u | User-Based SNMP Version 2                          |
| SNMPv3  | SNMP Version 3                                     |



## Abbildungsverzeichnis

|  |    |
|--|----|
| Abbildung 1: Aufgabenstellung  | 5  |
| Abbildung 2: Umfeld der Schnittstelle  | 7  |
| Abbildung 3: graphische Darstellung der Grenz- und Schwellwerte, Reaktionszeit | 10 |
| Abbildung 4: SNMP allgemein  | 11 |
| Abbildung 5: Layout eines SNMP-MIB Moduls                                      | 16 |
| Abbildung 6: Zuordnung Zeichenkette $\Leftrightarrow$ Nummer einer OID         | 18 |
| Abbildung 7: wichtige Unterbäume der root-Node                                 | 19 |
| Abbildung 8: get-Operation   | 20 |
| Abbildung 9: getbulk-Operation   | 22 |
| Abbildung 10: eigener SNMP-basierter Lösungsansatz                             | 24 |
| Abbildung 11: NRPE allgemein   | 26 |
| Abbildung 12: Direkte Nutzung des NRPE-Plugin                                  | 27 |
| Abbildung 13: indirekte Nutzung des NRPE- Plugin                               | 28 |
| Abbildung 14: Skizze der Schnittstelle   | 58 |

## Tabellenverzeichnis

|   |    |
|---|----|
| Tabelle 1: Übersicht über RFC's, die SNMP- Versionen beschreiben                | 13 |
| Tabelle 2: Funktionsumfang der SNMP-Versionen                                   | 14 |
| Tabelle 3: Bestandteile der Skriptausgabe für ITM-Agenten                       | 26 |
| Tabelle 4: Vergleich der Lösungsansätze   | 29 |
| Tabelle 5: Gliederung der textbasierten Konfigurationsdatei                     | 31 |
| Tabelle 6: Optionen des Wertes Type   | 35 |
| Tabelle 7: Optionen des Wertes Displaytype                                      | 35 |
| Tabelle 8: erforderliche Werte des allgemeinen und messwertspezifischen Teils   | 37 |
| Tabelle 9: Übersicht über gespeicherte Werte in der Datei "currentstatus.tmp"   | 41 |
| Tabelle 10: MIB für Abbildung eines einzeiligen Skriptergebnisses mittels exec  | 43 |
| Tabelle 11: MIB für Abbildung eines mehrzeiligen Skriptergebnisses mittels exec | 45 |
| Tabelle 12: MIB für Abbildung eines Skriptergebnisses mittels extend            | 48 |
| Tabelle 13: Vor- und Nachteile der Abbildungsmöglichkeiten                      | 50 |
| Tabelle 14: Ausgabezeilen des allgemeinen Teils einer Konfigurationsdatei       | 52 |
| Tabelle 15: Ausgabezeilen der Messwertparameter                                 | 54 |
| Tabelle 16: Zuordnung ITM-Ausgabe   | 64 |
| Tabelle 17: Fehlermeldungen   | 66 |

## Vorwort

Diese Diplomarbeit wurde von November 2008 bis April 2009 bei der Firma MTU Aero Engines GmbH, Dachauer Straße 665, 80995 München in der Abteilung IT-Technik angefertigt.

Ich möchte mich ausdrücklich bei all denen bedanken, die an der Entstehung dieser Arbeit mitgewirkt haben, insbesondere bei Herrn Prof. Dr. -Ing. Mario Geißler, Herrn Dipl.-Ing. Ulrich Clara und allen Kollegen der Abteilung IT-Technik, die mich während der Erstellung der Diplomarbeit unterstützt haben.

## Aufgabenstellung und Abgrenzung der Diplomarbeit

Im Rahmen der Diplomarbeit ist für das Trendmonitoring eine generische, plattformübergreifende und dateibasierte Schnittstelle zu entwerfen und umzusetzen. Diese soll einem Anwendungsverantwortlichen ermöglichen, beliebige Messwerte durch ein zentrales Erfassungssystem (Monitoringsystem) darzustellen. Darüber hinaus muss die Schnittstelle so gestaltet werden, dass der Anwendungsverantwortliche in der Lage ist, Grenz- und Schwellwerte für die zu überwachenden Messwerte mitgeben zu können. Für das Abfragen der einzelnen Werte soll das Standardnetzwerkprotokoll SNMP verwendet werden. Dabei ist ein Skript zu erstellen, das eine automatisierte Konfiguration eines SNMP-Daemons ermöglicht. Diese soll mit Hilfe einer Konfigurations- und einer Messwertedatei erfolgen. Die dafür verwendeten Dateien sind textbasiert zu realisieren, um eine einfache Handhabung zu gewährleisten. Des Weiteren ist zu berücksichtigen, dass die Konfigurations- und Messwertedateien an beliebigen Stellen in einem Netzwerk (z.B. auf einem bestimmten Netzlaufwerk) oder lokal auf dem zu überwachenden Server liegen können.

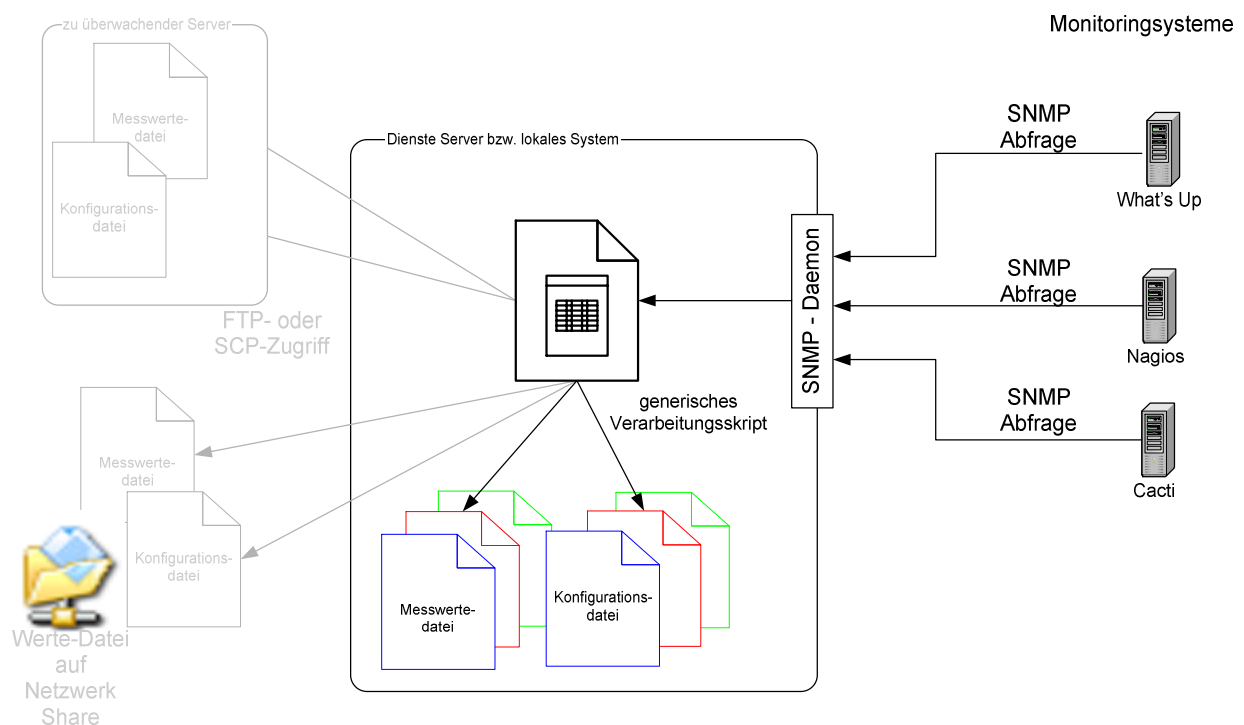


Abbildung 1: Aufgabenstellung

Um die Aufgabenstellung für diese Diplomarbeit einzugrenzen, wurden folgende Festlegungen vereinbart:

- Als Messwerte sind nur Zahlen zulässig.
- Es werden nur Echtzeitwerte als Messwerte zugelassen, d.h. die Messwertedatei enthält immer nur den aktuellen Messwert und keine Messwertreihen.
- Wegen der großen Verbreitung im Administrationsbereich und seiner Plattformunabhängigkeit ist als Programmiersprache für das Skript der Schnittstelle Perl zu verwenden.
- Das Skript ist so zu gestalten, dass es als lokale Schnittstelle und auf einem Diensteserver eingesetzt werden kann.
- In der ersten Version sollen nur lokale Konfigurations- und Messwertedateien durch die Schnittstelle berücksichtigt werden.
- Alle für das Schnittstellenskript einer Anwendung benötigten Dateien sollen in einem Verzeichnis liegen.
- Falls auf eine SNMP-Version Bezug genommen wird, ohne dass die Version angegeben wurde, ist immer die SNMP-Version 2c gemeint.
- Für die Umsetzung der Schnittstelle sind keine SNMP-Operationen (set- und trap-Operation) zu verwenden.
- Das Abrufen von SNMP-Standardmesswerten mittels dieser Schnittstelle wird nicht unterstützt.
- Wenn von plattformübergreifend die Rede ist, ist immer Windows und Linux gemeint, da diese Systeme am häufigsten bei der MTU Anwendung finden.
- Sämtliche Angaben zu Linux beziehen sich auf Debian Linux 5.0.

## 0 Einleitung

Im Laufe der Zeit haben sich bei der MTU im Rahmen des Monitorings von Basis-Infrastrukturkomponenten die Personengruppen der Anwendungsverantwortlichen sowie das Monitoringteam herausgebildet. Erstere sind für eine oder mehrere Anwendungen im Unternehmen zuständig, für die spezielle Kenntnisse nötig sind. Von den im Unternehmen verwendeten Monitoringtools haben sie jedoch keine oder nur geringe Kenntnisse. Im Gegensatz dazu führt das Monitoringteam das Monitoring von Anwendungen und Systemen im Unternehmen durch. Das Monitoringteam besitzt hierfür detaillierte Kenntnisse über die im Unternehmen eingesetzten Monitoringtools, jedoch keine Kenntnisse über die zu überwachenden Anwendungen. In größeren Unternehmen wird dieses Monitoring oft durch externe Dienstleister durchgeführt.

Zwischen diesen beiden Personengruppen tritt folgendes Problem auf: Ein Anwendungsverantwortlicher möchte Messwerte für einen Sachverhalt in einem Monitoringtool dargestellt haben. Er kennt jedoch nicht die Form, in der ein Monitoringtool diese Werte erwartet. Das Monitoringteam hingegen kennt zwar die Schnittstellen des Monitoringsystems, aber nicht die der zu überwachenden Anwendung. Meine Schnittstelle soll eine einheitliche Beschreibungsmöglichkeit bieten, mit der ein Anwendungsverantwortlicher, auf einfache Art, dem Monitoringteam seine Messwerte zur Verfügung stellen kann.

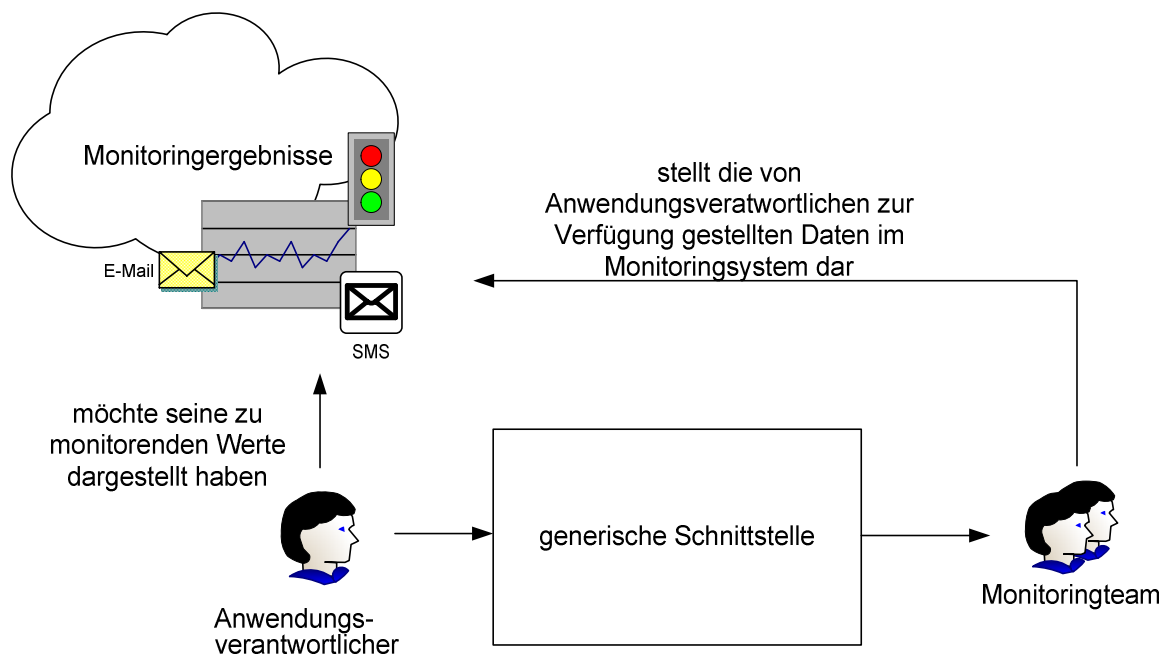


Abbildung 2: Umfeld der Schnittstelle

## 1 Einführung in das Monitoring

### 1.1 Definition und betrachtete Arten des Monitorings

#### 1.1.1 Definition Monitoring<sup>1</sup>

Beim Monitoring handelt es sich um eine systematische Erfassung, Protokollierung oder Beobachtung eines Vorgangs durch technische Hilfsmittel oder andere Beobachtungssysteme. Der Zweck des Monitorings besteht darin, in einen beobachteten Vorgang steuernd eingreifen zu können, falls definierte Grenz- und Schwellwerte über- bzw. unterschritten werden.

Für das Monitoring im Netzwerk angeschlossener Systeme bedeutet dies, dass man Netzwerke, deren Hardware (z.B. Server, Router, Drucker) oder Dienste (z.B. E-Mail, DNS, FTP) durch geeignete Hilfsmittel (z.B. SNMP) überwacht. Ziel dieser Überwachung soll es sein, zeitnah auf auftretende Probleme reagieren zu können und bzw. oder langfristige Trends von bestimmten Parametern, z.B. Last eines Netzwerkes, zu erkennen und ggf. gegensteuern zu können. Für diese Diplomarbeit werden folgende zwei relevante Arten des Monitorings unterschieden: Echtzeitmonitoring und Trendmonitoring.

#### 1.1.2 Echtzeitmonitoring

Unter Echtzeitmonitoring (zeitnahes Monitoring) versteht man das Abfragen von aktuellen Zuständen einzelner oder mehrerer Systemkomponenten, wobei der Fokus auf den Abfragen von Zustandswerten (z.B. CPU- Temperatur, CPU- Last, noch vorhandene Festplattenkapazität der root-Partition) liegt. Falls ein solcher zu überwachender Wert eine vorgegebene Grenze über- oder unterschreitet, ist es möglich, eine Meldung (z.B. Warnung, Fehler) auszulösen. Diese kann in Form einer graphischen Anzeige, E-Mail, SMS usw. erfolgen. Das Abfrageintervall der einzelnen Werte liegt bei diesem Monitoring in einem relativ kurzen Bereich (Minuten oder Sekunden), da hier der aktuelle Zustand von ausgewählten Netzwerkkomponenten im Vordergrund steht.

---

<sup>1</sup> vgl. <http://de.wikipedia.org/wiki/Monitoring> (06.11.2008)

### 1.1.3 Trendmonitoring

Beim Trendmonitoring (Langzeitmonitoring) geht es hingegen darum, über Wochen und Monate hinweg Monitoringwerte, wie die Lizenznutzung bestimmter Software oder die CPU- Auslastung, aufzuzeichnen und daraus Trends abzulesen, um gewissen Entwicklungen frühzeitig entgegen zu wirken oder sich abzeichnende Engpässe zu erkennen. Das Abfrageintervall der Werte liegt hierbei in größeren Bereichen (meist Minuten, Stunden) als beim Echtzeitmonitoring, da hier der Fokus auf der langfristigen Erfassung der Werte liegt und einzeln auftretende Extremwerte (Ausreißer) von geringerem Interesse sind. Die Darstellung der Werte erfolgt meist in Graphen.

## 1.2 Definition und Bedeutung von Grenz- und Schwellwerten beim Monitoring

### 1.2.1 Definition Grenzwerte

Unter Grenzwerten werden Messwerte verstanden, die nicht über- oder unterschritten werden dürfen. Beim Über- oder Unterschreiten dieser kritischen Grenze ist eine Alarmierung des Verantwortlichen für die Anwendungen bzw. Systeme erforderlich. Sie kann z.B. per E-Mail, Telefon, SMS oder in visueller Form erfolgen.

### 1.2.2 Definition Schwellwerte

Schwellwerte sind Messwerte, die über- oder unterschritten werden können. Beim Über- oder Unterschreiten dieser Werte ist lediglich eine Warnung (auch in abgestufter Form) auszulösen, z.B. per E-Mail, SMS. Diese macht den Anwendungsverantwortlichen darauf aufmerksam, dass sich ggf. ein Problem anbahnt. Es ist möglich, mehrere Schwellwerte zu definieren.



### 1.2.3 Festlegung von Grenz- und Schwellwerten

Die Grenz- und Schwellwerte müssen so gewählt werden, dass zwischen ihnen und den auftretenden Messwerten ein gewisser Sicherheitsabstand besteht. Dieser darf aber nicht zu groß gewählt werden, da es sonst vorkommen kann, dass kaum oder keine Fehler und Warnungen angezeigt werden. Grenzwerte sollten des Weiteren so definiert werden, dass sie unterhalb einer bestimmten Grenze liegen, um noch einen (zeitlichen) Puffer für ein eventuelles Gegensteuern durch den Anwendungsverantwortlichen zu haben. Die Ermittlung der optimalen Grenz- und Schwellwerte erfolgt durch Iteration. In diesem Fall passt ein Anwender die entsprechenden Grenz- und Schwellwerte den Gegebenheiten an.

Möglichkeiten für die Festlegung von Grenz- und Schwellwerten sind:

- vorgegebene Herstellergrenzwerte (z.B. CPU-Temperatur),
- physikalische und logische Grenzen (z.B. Festplattenkapazität),
- Mittelwerte (z.B. durchschnittliche Lizenznutzung).

Mitunter müssen Grenz- und Schwellwerte im Laufe der Zeit angepasst werden, weil sich die anfangs definierten Werte verändert haben.

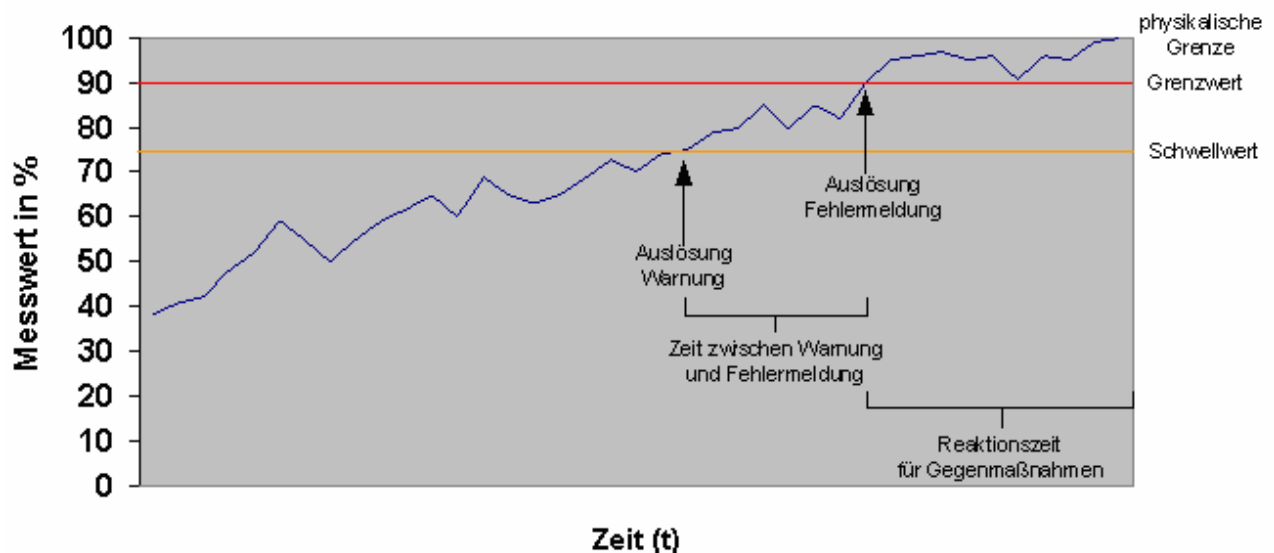
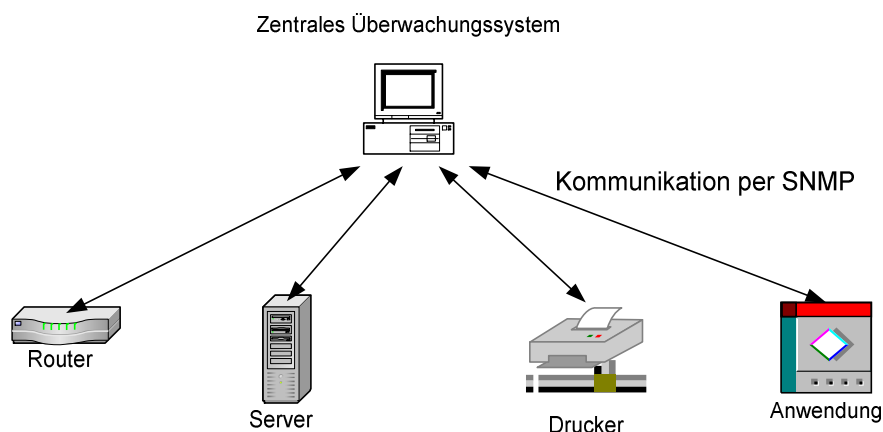


Abbildung 3: graphische Darstellung der Grenz- und Schwellwerte, Reaktionszeit

## 2 Das Simple Network Management Protocol

### 2.1 Überblick über das Simple Network Management Protocol

Bei SNMP handelt es sich um ein Netzwerkprotokoll, mit dessen Hilfe es möglich ist, an Netzwerken angeschlossene Systeme, z.B. Router, Server oder Drucker, und Anwendungen durch ein zentrales System zu steuern und zu überwachen<sup>2</sup>.



**Abbildung 4: SNMP allgemein**

Die Aufgabe dieses Protokolls besteht in der Regelung der Kommunikation zwischen den zu überwachenden Komponenten (Clients) und einem Monitoringsystem (Server). Um diese Kommunikation zu gewährleisten, werden in SNMP die Datenpakete und der Kommunikationsablauf beschrieben. Ebenso kann eine Fehlererkennung, Fehlerbenachrichtigung und Fernkonfiguration erfolgen. Für die Kommunikation stellt SNMP verschiedene Operationen zur Verfügung<sup>3</sup>. Diese können in zwei Kategorien unterteilt werden: Operationen, bei denen ein Monitoringsystem aktiv die Werte von einem System abfragt und Operationen, bei denen das Monitoringsystem passiv auf Aktionen eines zu überwachenden Systems reagiert. Als Beispiel für aktive SNMP-Operationen sind sämtliche get-Operationen zu nennen. Hingegen gibt es nur die trap-Operation, die passiv arbeitet. Für das eigentliche Überwachen der Clients setzt man Agenten auf den zu überwachenden Systemen ein. Ihre Aufgabe besteht in der Bereitstellung von Informationen (Messwerten) für das Monitoringsystem. Agenten sind von vielen Hardware-

<sup>2</sup> vgl. Mauro, Schmidt S. 1

<sup>3</sup> siehe Punkt 2.5

herstellern (z.B. Cisco Systems) oder in Betriebssystemen (z.B. Windows, Linux) bereits integriert<sup>4</sup>. Darüber hinaus existieren SNMP-Agenten von Drittanbietern, die bestehende Agenten erweitern oder ersetzen. Das bekannteste Beispiel hierfür ist NetSNMP<sup>5</sup>.

Da in SNMP lediglich die Kommunikation zwischen einem im Netzwerk angeschlossenen System und einem zentralen Monitoringsystem beschrieben wird, sind außerdem die Managed Objects (MO's) sowie die Management Information Base (MIB) für die Abfrage von Messwerten notwendig. Im Laufe der Zeit wurden drei Versionen des SNMP-Protokolls entworfen und von der Internet Engineering Task Force<sup>6</sup> (IETF) in Requests for Comments (RFC) beschrieben.

## 2.2 Beschreibung der wichtigsten SNMP-Versionen

Die erste Version (SNMPv1) bietet Funktionen zum Abfragen und Setzen von Werten. Die Authentifizierung erfolgt über so genannte Community-Strings. Es handelt sich dabei um Passwörter, die im Klartext angegeben werden. Wenn eine SNMP- basierte Anwendung dieses Passwort kennt, ist sie in der Lage, auf die Management Informationen eines Gerätes zuzugreifen. Als typische Community-Strings sind z.B. „public“, „read-only“, „read-write“ oder „trap“ zu nennen. Diese Version des SNMP-Protokolls ist veraltet, wird jedoch noch von vielen (Geräte)-Herstellern unterstützt<sup>7</sup>.

In der zweiten Version des SNMP-Protokolls (SNMPv2) wurde der Leistungsumfang des Protokolls um Funktionen und zusätzliche Datentypen erweitert<sup>8</sup>. Die Authentifizierung mittels Community-String, wie in SNMPv1, wurde jedoch in der SNMP-Version 2c (SNMPv2c) beibehalten. Um diese unsichere Authentifizierung zu beseitigen, wurden zusätzlich die Versionen Party-Based SNMP Version 2 (SNMPv2p) und User-Based SNMP Version 2 (SNMPv2u) durch die IETF verabschiedet. Dadurch ist eine Authentifizierung, beispielsweise mittels Passwort, möglich. Sie besitzen jedoch nur eine geringe Verbreitung, da in der dritten und zurzeit aktuellen Version (SNMPv3) die Authentifizierungsmöglichkeiten nochmals

---

<sup>4</sup> vgl. Mauro, Schmidt S. 3

<sup>5</sup> <http://www.net-snmp.org/> (11.02.2009)

<sup>6</sup> [www.ietf.org](http://www.ietf.org) (11.02.2009)

<sup>7</sup> vgl. Mauro, Schmidt S. 2

<sup>8</sup> vgl. Tabelle 2

verbessert und erweitert wurden. Zudem beinhaltet sie alle Möglichkeiten der Version SNMPv2. Trotz dieser Vorteile hat sich die Version SNMPv3 noch nicht genügend durchgesetzt.

Überblick über die zugrunde liegenden Spezifikationen der einzelnen SNMP-Versionen:

| SNMP- Version | zugrunde liegende IETF-Spezifikation <sup>9</sup> |
|---------------|---|
| SNMPv1        | RFC 1157  |
| SNMPv2        | RFC 3416 bis 3418                                 |
| SNMPv2p       | RFC 1441 bis 1447                                 |
| SNMPv2u       | RFC 1909<br>RFC 1910                              |
| SNMPv2c       | RFC 1901, RFC 1905, RFC 1906                      |
| SNMPv3        | RFC 3410 bis 3418<br>RFC 2576                     |

**Tabelle 1: Übersicht über RFC's, die SNMP- Versionen beschreiben**

Zum Abfragen von Messwerten einzelner zu überwachender Systeme stehen in SNMP verschiedene Operationen zur Verfügung. Die nachfolgende Tabelle stellt dar, in welcher SNMP-Version welche Operation verwendet werden kann. Eine nähere Beschreibung der für die Diplomarbeit wichtigen SNMP-Operationen erfolgt im Punkt 2.5.

<sup>9</sup> alle hier aufgelisteten RFC's sind unter <http://www.ietf.org> (11.02.2009) abrufbar

| SNMP- Operation | SNMP-Version |                |
|-----------------|--------------|----------------|
|                 | SNMPv1       | SNMPv2/ SNMPv3 |
| get             | x            | x              |
| getnext         | x            | x              |
| getbulk         |              | x              |
| set             | x            | x              |
| getresponse     | x            | x              |
| trap            | x            | x              |
| notification    |              | x              |
| inform          |              | x              |
| report          |              | x              |

**Tabelle 2: Funktionsumfang der SNMP-Versionen**

### 2.3 Vergleich der SNMP-Versionen

Angesichts der im Punkt 2.2 beschriebenen Eigenschaften der einzelnen SNMP-Versionen kommt die SNMP-Version 1 nicht in Betracht, da sie lediglich Community-Strings als Authentifizierungsmöglichkeit bietet. Des Weiteren fehlt eine Unterstützung von 64-Bit-Werten sowie eine Möglichkeit der Abfrage von mehreren Werten<sup>10</sup>. Für die SNMP-Version 2 spricht, dass sie den gleichen Operationsumfang wie die Version 3 besitzt. Nachteilig ist hierbei das Fehlen einer sicheren und zuverlässigen Authentifizierung in der Version SNMPv2c. Die Versionen SNMPv2p und SNMPv2u werden aufgrund ihrer geringen Verbreitung für das Abfragen nicht in Betracht gezogen. Die Verwendung der SNMP-Version 3 zum Abfragen der Messwerte wird in einer ersten Version der Schnittstelle nicht genutzt, da die Authentifizierung, welche die SNMP-Version 2c bietet, für diese einfachen Aufgaben ausreicht. Weiter spricht gegen die Verwendung der Version SNMPv3 ihre geringe Verbreitung.

---

<sup>10</sup> vgl. Eckel

## 2.4 Management Information Base und Object Identifier

### 2.4.1 Managed Objects und Management Information Base

Da das SNMP-Protokoll nur die Kommunikation zwischen den im Netzwerk angeschlossenen Systemen und einem Monitoringsystem beschreibt, sind demzufolge die MO's und die MIB notwendig, um Messwerte abfragen zu können. MO's beinhalten die einzelnen Messwerte (z.B. CPU -Temperatur, Anzahl der eingeloggten Nutzer) eines zu überwachenden Systems und werden durch einen Namen (Object Identifier), eine Nummer und einen Namensstring definiert<sup>11</sup>.

Die einzelnen MO's werden mit Hilfe einer Beschreibungssprache, die auf Abstract Syntax Notation One (ASN.1) basiert, in einer MIB beschrieben.

Dabei stehen folgende Datentypen für die Definition von MO's zu Verfügung:

- Integer/ Integer32
- Gauge/ Gauge32
- Counter /Counter32 /Counter64
- Timeticks/ Octet String
- Unsigned32
- Object Identifier
- IpAdress/ Networkadress
- Opaque/ Bits
- Sequence/ Sequence of

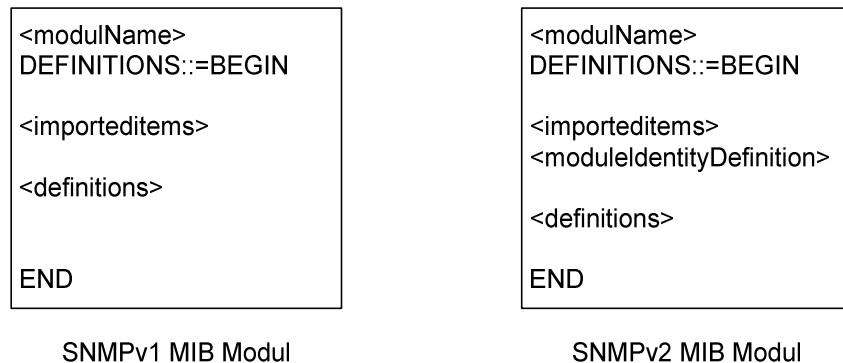
Detaillierte Informationen zu den Datentypen in SNMP-MIB's können u.a. in Perkins, McGinnis, S. 38ff. sowie in der Anlage A nachgelesen werden.

---

<sup>11</sup> vgl. Michler S. 8ff

## 2.4.2 Aufbau einer Management Information Base

Mit den im Punkt 2.4.1 genannten Datentypen ist es möglich, MO's bzw. MIB's zu definieren. Dabei ist folgender Aufbau einer MIB zu beachten:



**Abbildung 5: Layout eines SNMP-MIB Moduls<sup>12</sup>**

Ein MIB-Modul besteht aus einer äußeren Hülle (wrapper), die dieses gegenüber anderen MIB-Modulen abgrenzt. Die Abgrenzung erfolgt durch BEGIN und END. In dieser Hülle befindet sich der Abschnitt IMPORTS<sup>13</sup>, mit dessen Hilfe Elemente aus anderen MIB-Modulen einbezogen und verwendet werden können. Nach dem IMPORTS-Abschnitt, aber noch vor der Definition von anderen Elementen, erfordert ein SNMPv2 MIB-Modul ein so genanntes MODULEIDENTITY-Konstrukt. Dieses enthält Informationen zum Autor und Verwendungszweck der MIB. Danach folgen die Definitionen der einzelnen Elemente, die die Werte der MIB enthalten.

Beispiel<sup>14</sup>:

```
HAUS-MIB DEFINITIONS ::= BEGIN
IMPORTS
    enterprises
        FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC-1212
    DisplayString
        FROM RFC-1213;
```

<sup>12</sup> Quelle: Perkins, McGinnis S.59 Abb.3-2

<sup>13</sup> vgl. Beispiel

<sup>14</sup> eigenes Beispiel, in Anlehnung an Michler S. 11ff.

```

-- WICHTIG: Die MIB ist nur eine Beispiel-MIB sie existiert real NICHT
-- enterprise OID fuer "hausverwaltung" ==> 987654321

hausverwaltung OBJECT IDENTIFIER ::= {enterprises 987654321}
haus OBJECT IDENTIFIER ::= {hausverwaltung 1}
-- Einordnung von HAUS in den MIB-Baum von Hausverwaltung

baujahr OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Baujahr des Hauses"
 ::= {haus 1}

anschrift OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Postanschrift"
 ::= {haus 2}

schluessel OBJECT-TYPE
    SYNTAX INTEGER (0..25)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Anzahl der vorhandenen Schluessel"
 ::= {haus 3}

mieteranzahl OBJECT-TYPE
    SYNTAX INTEGER {0..20}
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Anzahl der Mietparteien"
 ::= {haus 4}

verwalter OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Name des Hausverwalters"
 ::= {haus 5}

stromverbrauch OBJECT-TYPE
    SYNTAX COUNTER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Stromverbrauch fuer Keller und Treppenhaus"
 ::= {haus 6}

END

```

Die einzelnen MO's einer MIB werden per Object Identifier (OID) angesprochen.



### 2.4.3 Object Identifier<sup>15</sup>

Bei einer OID handelt es sich um ein Namensschema, das in Baumform organisiert ist. Es stellt sicher, dass einem Messwert (z.B. CPU-Temperatur eines x86-PC's) ein eindeutiger Name zugewiesen wird. Dieser Name eines jeden Knotens und Blattes des MIB-Baumes kann durch eine Nummer oder Zeichenkette dargestellt werden. Bei der Schreibweise mittels Zeichenkette ist jedem Element einer OID eine Nummer zugeordnet.

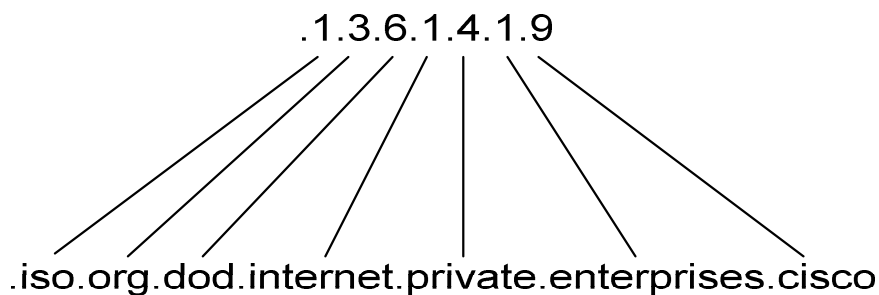
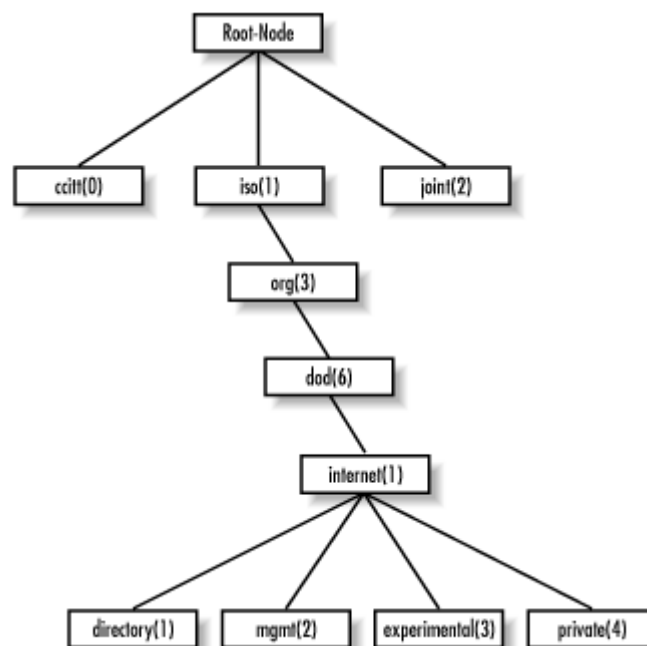


Abbildung 6: Zuordnung Zeichenkette <=> Nummer einer OID

Die Baumstruktur enthält alle bei der Internet Assigned Numbers Authority<sup>16</sup> (IANA) registrierten MIB's. Die Adressierung des Baumes erfolgt, indem jede OID durch einen Punkt (.) getrennt wird. Dabei wird mit der Adressierung beim root-Element begonnen.



<sup>15</sup> vgl. Mauro, Schmidt S. 23ff., Michler S. 10ff.

<sup>16</sup> <http://www.iana.org/> (21.11.2008)

### Abbildung 7: wichtige Unterbäume der root-Node<sup>17</sup>

Beschreibung der relevanten Unterbäume der root-Node:

- ccitt: vom Comité Consultatif International Téléphonique et Télégraphique (CCITT), (heute: ITU Telecommunication Standardization Sector), administrierter Unterbaum
- joint: von ISO und CCITT gemeinsam administrierter Unterbaum
- dod: (Department of Defence), Relikt aus der militärischen Vergangenheit des ARPANET; erst unterhalb der Internet-Node stehen für SNMP relevante Informationen
- directory: wird nicht benutzt
- mgmnt: Standard-Set von Internet-Management-Objekten
- private: unter dieser Node stehende OID's werden von Institutionen und Unternehmen verwendet

Wenn ein Unternehmen ein Gerät entwickelt hat und dazu eine MIB bereitstellt, kann es bei der IANA eine OID für diese MIB beantragen<sup>18</sup>. Die IANA vergibt eine so genannte Enterprise-Nummer, unter der die MIB des Unternehmens veröffentlicht werden kann.

## 2.5 SNMP-Operationen

*Alle angeführten Beispiele wurden mit dem Tool „net-snmp“<sup>19</sup> auf einem Debian Linux System durchgeführt. Die abgerufenen Werte stammen von einem Windowsserver.*

Sie realisieren den Datenaustausch zwischen einem lokalen SNMP-Agenten und einem Monitoringsystem. Dabei stellt der SNMP-Agent einem Monitoringsystem die Werte zur Verfügung.

---

<sup>17</sup> Quelle: Mauro, Schmidt S.24 Abbildung 2-2

<sup>18</sup> <http://pen.iana.org/pen/PenApplication.page> (21.11.2008)

<sup>19</sup> Download unter <http://net-snmp.sourceforge.net> (21.11.2008)

## 2.5.1 Die get-Operation<sup>20</sup>

Das Monitoringsystem sendet eine get-Abfrage an den SNMP-Agenten des zu überwachenden Systems, um einen Wert abzufragen (z.B. Name des Systems). Der SNMP-Agent antwortet mit einer get-Antwort auf diese Abfrage. Als Antwort erhält das Monitoringsystem den angeforderten Messwert. Mit dieser Operation kann immer nur ein Wert abgefragt werden.

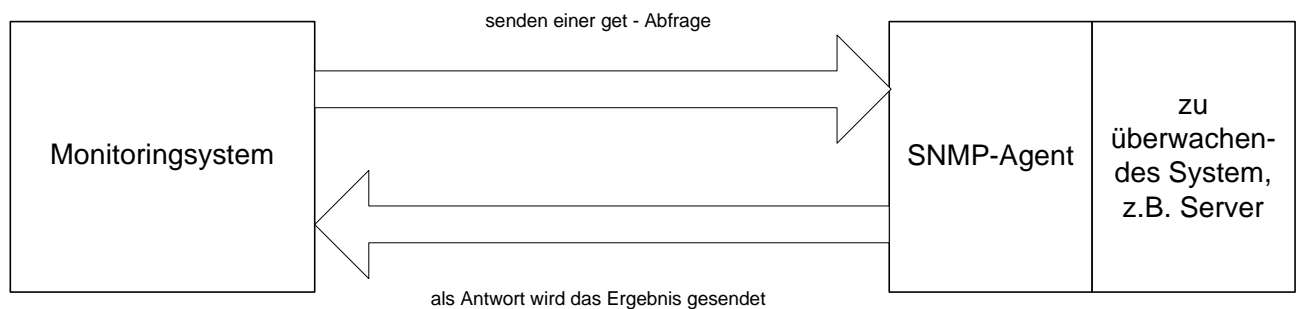


Abbildung 8: get-Operation<sup>21</sup>

Allgemeine Syntax:

```
snmpget -v <SNMPVersion> -c <Communitystring> Zieladresse22 OID
```

Beispiel 1: Ausgabe des Inhalts eines Wertes mittels numerischer OID

```
$ snmpget -v 2c -On -c public 192.168.0.25 1.3.6.1.2.1.1.5.0
.1.3.6.1.2.1.1.5.0 = STRING: testserver01
```

Beispiel 2: Ausgabe des Inhalts eines Wertes mittels Angabe des MIB-Namens

```
$ snmpget -v 2c -c public 192.168.0.25 1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: testserver01
```

## 2.5.2 Die getnext-Operation<sup>23</sup>

Sie leitet eine Reihe von get-Operationen ein. Mit ihrer Hilfe ist es möglich, mehrere Messwerte (OID's) abzufragen. Die getnext-Operation durchläuft dabei alle MO's ab

<sup>20</sup> vgl. Mauro, Schmidt, S.38ff.

<sup>21</sup> eigene Darstellung in Anlehnung an Mauro, Schmidt S. 38 Abb. 2-5

<sup>22</sup> kann IP-Adresse, Systemname oder fully qualified domain name sein

<sup>23</sup> vgl. Mauro, Schmidt, S. 43ff.

der angegebenen Stelle im MIB-Baum. Für die Automatisierung dieser Operation gibt es den Befehl `snmpwalk`.

Allgemeine Syntax:

```
snmpwalk -v <SNMPVersion> -c <Communitystring> Zieladresse Start-OID
```

Beispiel 1: Ausgabe des Inhalts der Werte mittels numerischer OID

```
$ snmpwalk -v 2c -On -c public 192.168.0.25 1.3.6.1.2.1.1
.1.3.6.1.2.1.1.1.0 = STRING: Hardware: x86 Family 15 Model 65 Stepping 3 AT/AT
COMPATIBLE-Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)
.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.1.4.1.311.1.1.3.1.2
.1.3.6.1.2.1.1.3.0 = Timeticks: (3687716367) 426 days, 19:39:23.67
.1.3.6.1.2.1.1.4.0 = STRING: Scharrnbeck
.1.3.6.1.2.1.1.5.0 = STRING: testserver01
.1.3.6.1.2.1.1.6.0 = STRING: 044/308
.1.3.6.1.2.1.1.7.0 = INTEGER: 79
```

Beispiel 2: Ausgabe des Inhalts der Werte mittels Angabe des MIB-Namens

```
$ snmpwalk -v 2c -c public 192.168.0.25 1.3.6.1.2.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 15 Model 65 Stepping 3 AT/AT
COMPATIBLE-Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.311.1.1.3.1.2
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (3687717494) 426 days, 19:39:34.94
SNMPv2-MIB::sysContact.0 = STRING: Scharrnbeck
SNMPv2-MIB::sysName.0 = STRING: testserver01
SNMPv2-MIB::sysLocation.0 = STRING: 044/308
SNMPv2-MIB::sysServices.0 = INTEGER: 79
```

### 2.5.3 Die getbulk-Operation<sup>24</sup>

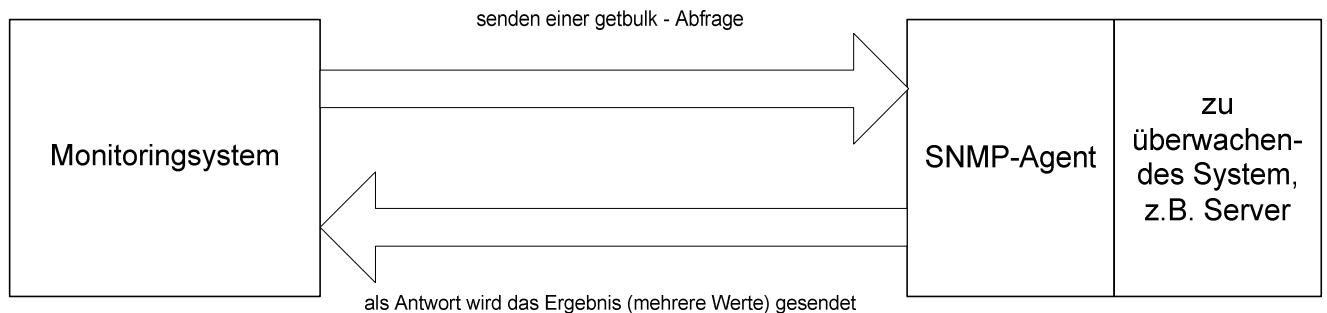
Mit dieser Operation ist es ebenfalls möglich, mehrere SNMP-Werte abzurufen. Das Monitoringsystem sendet eine getbulk-Abfrage an den SNMP-Agenten, um Werte abzufragen (z.B. Systembeschreibung, Messwerte zum Netzwerkinterface). Anschließend antwortet der SNMP-Agent mit einem get-response. Die beim Monitoringssystem eingehende SNMP-Nachricht enthält alle abzufragenden Werte. Um diese Operation nutzen zu können, ist die Angabe zweier Variablen notwendig: `non-repeaters` und `max-repetitions`. `Non-repeaters` beinhaltet die Anzahl der Variablen, die nicht den Wiederholungen unterliegen. Für diese Variablen wird nur der erste Wert mit einer get-Operation geholt. Hingegen gibt `max-repetitions` an, wie oft die `getnext`-Operation für die restlichen Variablen wiederholt werden soll<sup>25</sup>. Die ange-

---

<sup>24</sup> vgl. Mauro, Schmidt S. 53ff.

<sup>25</sup> vgl. Perkins, McGinnis S.180ff.

fürten Beispiele dienen der näheren Erläuterung. Die getbulk-Operation ist ab SNMPv2 verfügbar.



**Abbildung 9: getbulk-Operation<sup>26</sup>**

Allgemeine Syntax:

```
snmpbulkget -v <SNMPVersion> -c <Communitystring> Zieladresse -non-repeaters -  
max-repetitions Variable1, ..., Variable n
```

Beispiel 1:

```
$ snmpbulkget -v 2c -c public 192.168.0.25 -Cn1 -Cr3 sysDescr ifInOctets  
ifOutOctets
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 15 Model 65 Stepping 3 AT/AT  
COMPATIBLE-Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)  
IF-MIB::ifInOctets.1 = Counter32: 255445  
IF-MIB::ifOutOctets.1 = Counter32: 255445  
IF-MIB::ifInOctets.65539 = Counter32: 2050097051  
IF-MIB::ifOutOctets.65539 = Counter32: 3179769697  
IF-MIB::ifInUcastPkts.1 = Counter32: 3081  
IF-MIB::ifOutUcastPkts.1 = Counter32: 3081
```

Beispiel 2:

```
$ snmpbulkget -v 2c -c public 192.168.0.25 -Cn2 -Cr3 sysDescr ifInOctets  
ifOutOctets
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 15 Model 65 Stepping 3 AT/AT  
COMPATIBLE-Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)  
IF-MIB::ifInOctets.1 = Counter32: 523813  
IF-MIB::ifOutOctets.1 = Counter32: 523813  
IF-MIB::ifOutOctets.65539 = Counter32: 1521202364  
IF-MIB::ifOutUcastPkts.1 = Counter32: 6131
```

<sup>26</sup> eigene Darstellung in Anlehnung an Mauro, Schmidt S. 54 Abb. 2-7

### Beispiel 3:

```
$ snmpbulkget -v 2c -c public 192.168.0.25 -Cn0 -Cr3 sysDescr ifInOctets  
ifOutOctets
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 15 Model 65 Stepping 3 AT/AT  
COMPATIBLE-Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)  
IF-MIB::ifInOctets.1 = Counter32: 523813  
IF-MIB::ifOutOctets.1 = Counter32: 523813  
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.311.1.1.3.1.2  
IF-MIB::ifInOctets.65539 = Counter32: 4160281036  
IF-MIB::ifOutOctets.65539 = Counter32: 1521442196  
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (3807790092) 440 days, 17:11:40.92  
IF-MIB::ifInUcastPkts.1 = Counter32: 6131  
IF-MIB::ifOutUcastPkts.1 = Counter32: 6131
```

#### 2.5.4 Anmerkung

Neben den bereits beschriebenen existieren weiterhin die set-, trap- getresponse -, notification -, inform-und report-Operation, die alle ab SNMPv2 verfügbar sind. Da diese für meine Diplomarbeit von geringer Bedeutung sind, werden sie nur genannt. Detaillierte Informationen zu ihrer Beschreibung können z.B. unter Perkins, McGinnis, S.171ff. nachgelesen werden.

### 3 Die Schnittstelle

#### 3.1 Vorstellung möglicher Lösungsansätze

##### 3.1.1 Eigener SNMP-basierter Lösungsansatz

Der eigene zu entwickelnde Lösungsansatz basiert darauf, dass ein Skript die zu überwachenden Messwerte der entsprechenden Anwendung ermittelt. Diese Messwerte werden in einer standardisierten Messwertedatei gespeichert, deren Beschreibung mittels einer Konfigurationsdatei erfolgt. Ein weiteres Skript verarbeitet die Daten der Konfigurations- und Messwertedatei so, dass die Messwerte per SNMP abrufbar sind. Der Aufruf dieses Skriptes erfolgt hierbei durch den SNMP-Daemon und über einen Scheduler (z. B. cron). Dabei werden die einzelnen Messwerte durch Skriptaufrufe in der Datei `snmpd.conf` unter der in der Konfigurationsdatei definierten OID verfügbar gemacht. Dieses Skript kann zusammen mit der Konfigurations- und Messwertedatei lokal auf dem zu überwachenden System liegen. Alternativ können diese Dateien auch durch einen zentralen Diensteserver verarbeitet werden. Falls die Lösungsvariante Diensteserver verwendet werden soll, ist anzumerken, dass ein einziges Skript die Verarbeitung von mehreren Konfigurations- und Messwertedateien realisiert. Der SNMP-Daemon stellt anschließend einem beliebigen Monitoringsystem den abgefragten Messwert zur Verfügung.

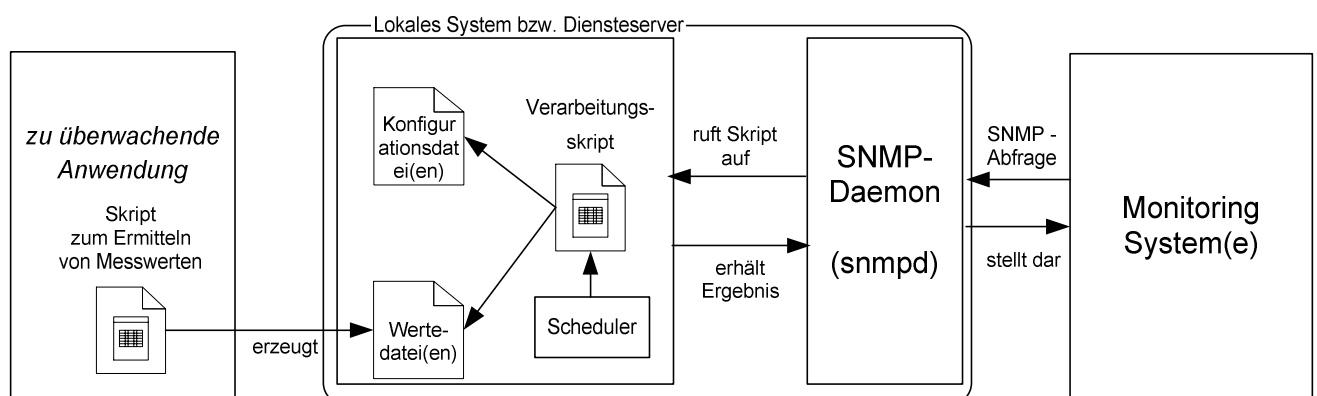


Abbildung 10: eigener SNMP-basierter Lösungsansatz

### 3.1.2 IBM Tivoli Monitoring Agent

Hierbei handelt es sich um einen Monitoring-Agenten für das ITM6 System von IBM, der lokal auf dem zu überwachenden System installiert ist. Dieser ermittelt die zu überwachenden Informationen und sendet sie an das IBM Tivoli Monitoring<sup>27</sup>. Mit Hilfe des Universal Agent Builders kann ein Agent so angepasst werden, dass er den individuellen Anforderungen entspricht.

Dabei werden vom Agenten folgende Betriebssysteme unterstützt:

- Aix (32- und 64- Bit-Version)
- HP-UX (32- und 64- Bit-Version, Itanium 64)
- Linux, Linux26, LinuxRH21
- Linux z-Series (32- und 64- Bit-Version)
- Linux Power PC (64- Bit-Version)
- Linux (32- und 64- Bit -Version)
- Solaris Sparc (32- und 64- Bit-Version)

Der eingesetzte Agent ist so konfiguriert, dass er die Ausgabe eines generischen Skriptes an IBM Tivoli weiterleitet. Die Konfiguration des Agenten erfolgt mittels einer dazugehörigen Konfigurationsdatei, welche die auszuführenden Skripte enthält. Es ist zu beachten, dass die Ausgabe der konfigurierten Skripte aus folgenden 14 Bestandteilen bestehen muss:

| Bestandteil | Datentyp     | Beschreibung  |
|-------------|--------------|---|
| Date        | Timestamp    | Zeitstempel, der durch das Skript erstellt wurde            |
| ScriptName  | String (64)  | Name des Skriptes, maximal 128 Zeichen                      |
| Status      | Integer      | Statusflag  |
| Text1       | String (128) | Beschreibung des Bestandteils „Value1“, maximal 128 Zeichen |
| Value1      | Integer      | Wert 1  |
| Text2       | String (128) | Beschreibung des Bestandteils „Value2“, maximal 128 Zeichen |
| Value2      | Integer      | Wert 2  |

<sup>27</sup> <http://www-01.ibm.com/software/de/tivoli> (20.02.2009)



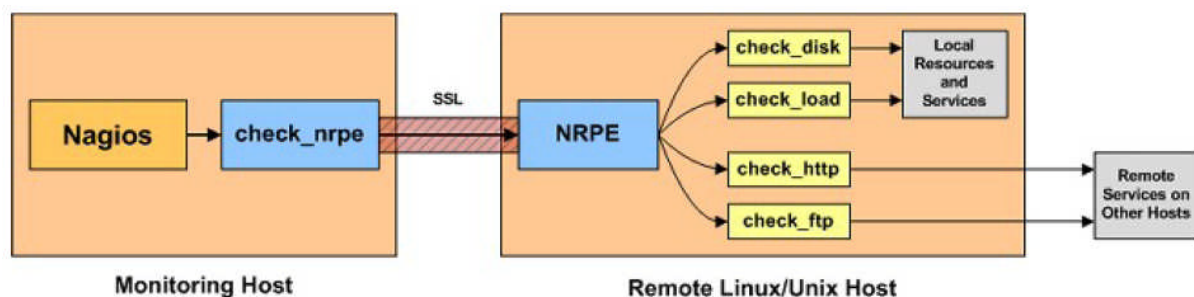
| Bestandteil | Datentyp      | Beschreibung  |
|-------------|---------------|---|
| Text3       | String (128)  | Beschreibung des Bestandteils „Value3“, maximal 128 Zeichen |
| Value3      | Integer       | Wert 3  |
| Text4       | String (128)  | Beschreibung des Bestandteils „Value4“, maximal 128 Zeichen |
| Value4      | Integer       | Wert 4  |
| Text5       | String (128)  | Beschreibung des Bestandteils „Value5“, maximal 128 Zeichen |
| Value5      | Integer       | Wert 5  |
| StatusText  | String (1024) | Status text, maximal 1024 Zeichen                           |

**Tabelle 3: Bestandteile der Skriptausgabe für ITM-Agenten**

Die Trennung dieser Bestandteile erfolgt mittels Semikolon. Es ist nicht notwendig, dass alle Bestandteile angegeben werden müssen. Die Ausgabe des Skriptes kann eine oder mehrere Zeilen umfassen.

### 3.1.3 Nagios Remote Plugin Executor (NRPE)<sup>28</sup>

Bei NRPE handelt es sich um ein Addon für das Monitoringsystem Nagios<sup>29</sup>, das es ermöglicht, Nagios-Plugins auf entfernten Linux-Systemen auszuführen. Da lokale Ressourcen (z.B. CPU -Auslastung) eines entfernten Linux-Systems nicht direkt mit Nagios abgefragt werden können, ist der Einsatz eines solchen „Agenten“ notwendig.



**Abbildung 11: NRPE allgemein<sup>30</sup>**

<sup>28</sup> NRPE Documentation S. 2 ff.

<sup>29</sup> <http://www.nagios.org> (18.02.2009)

<sup>30</sup> Quelle: NRPE Documentation S.2

Dieses Addon besteht aus folgenden Teilen:

- dem `check_nrpe` Plugin auf dem Monitoring Host,
- dem NRPE-Daemon auf entferntem Linux Host.

Falls mit NRPE eine lokale Ressource auf einem entfernten Host abgefragt werden soll, wird das `check_nrpe`-Plugin auf dem Nagios-Server (Monitoring Host) gestartet. Dieses kontaktiert anschließend den NRPE-Daemon auf dem abzufragenden Host. Dabei kann optional eine SSL- oder SSH- Verschlüsselung für die Verbindung zwischen dem Nagios-Server (Monitoring Host) und dem zu überwachenden Host verwendet werden. Die SSH-Verschlüsselung kann mittels SSH-Tunnel oder dem Plugin `check_by_ssh` durchgeführt werden. Anschließend führt der lokale NRPE-Daemon das entsprechende Plugin auf dem zu überwachenden Host aus. Das Ergebnis wird über den NRPE-Daemon und das `check_nrpe` Plugin zurück an den Monitoring Host gesendet. Das NRPE-Plugin kann direkt oder indirekt eingesetzt werden. Bei der direkten Nutzung werden die Messwerte, z.B. CPU-Temperatur oder Speicherauslastung, unmittelbar vom entsprechenden Host abgefragt.

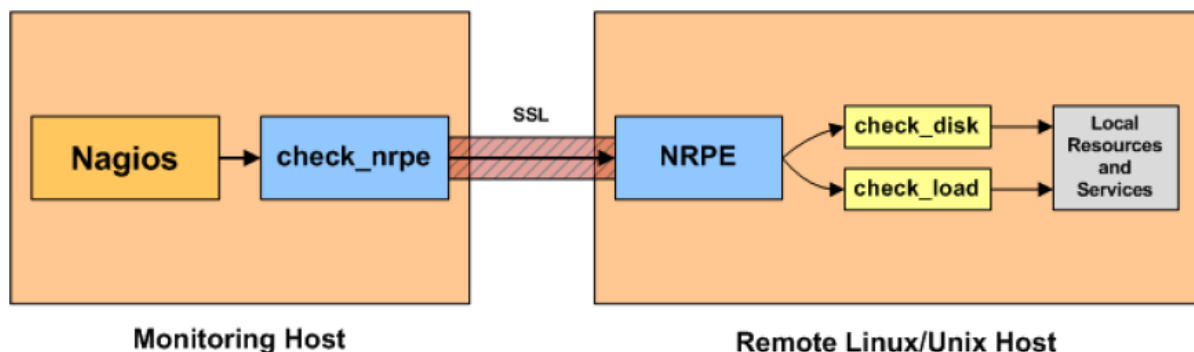


Abbildung 12: Direkte Nutzung des NRPE-Plugin<sup>31</sup>

Falls ein System nicht direkt durch ein Monitoring Host zu überwachen ist, gibt es die Möglichkeit, den NRPE-Daemon so zu konfigurieren, dass dieser als Proxy fungiert und die Messwerte indirekt abfragt. Eine Einsatzmöglichkeit für diese Art der Erfassung wäre, wenn z.B. aus Sicherheitsgründen keine zusätzlichen Agenten oder Plugins auf dem zu überwachenden System installiert sein dürfen.

<sup>31</sup> Quelle NRPE Documentation S. 3, oben

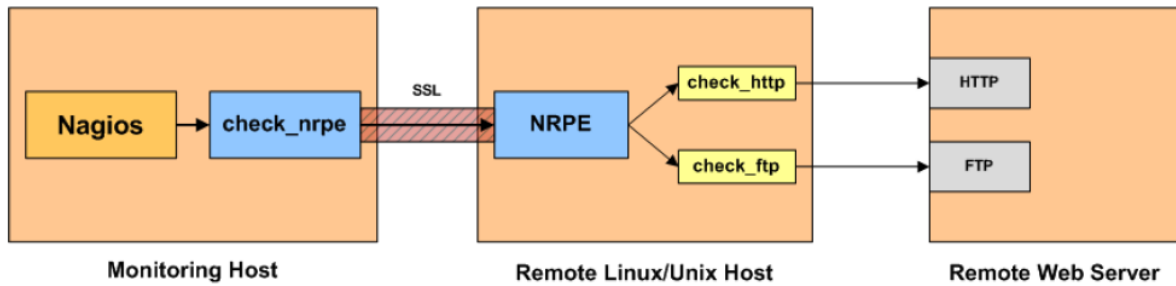


Abbildung 13: indirekte Nutzung des NRPE- Plugin<sup>32</sup>

### 3.1.4 Vergleich der Lösungsansätze

| Lösungsansatz                             | Vorteile   | Nachteile  |
|---|--|--|
| eigener Lösungsansatz                     | <ul style="list-style-type: none"> <li>- Messwerte können von allen Monitoring-systemen dargestellt werden, die SNMP-Abfragen unterstützen</li> <li>- einfach zu konfigurieren</li> </ul>                          | <ul style="list-style-type: none"> <li>- Beschränkung auf SNMP als Abfragemöglichkeit der Messwerte</li> <li>- bei der Verwendung von sehr vielen Messwerten kann es u.U. zu einem Timeout des SNMP-Daemons kommen</li> <li>- die SNMPD-Konfiguration bei SUSE Linux Enterprise Server muss mit „root“-Rechten erfolgen</li> </ul> |
| IBM Tivoli Monitoring Agent <sup>33</sup> | <ul style="list-style-type: none"> <li>- kompletter IBM Tivoli - Leistungsumfang kann genutzt werden (z. B. Alarmierungsmöglichkeiten)</li> <li>- individuelle Anpassung des Agenten an die Bedürfnisse</li> </ul> | <ul style="list-style-type: none"> <li>- nur IBM Tivoli kann als Monitoringsystem verwendet werden</li> <li>- keine SNMP - Unterstützung</li> </ul>  |

<sup>32</sup> Quelle NRPE Documentation S. 3, unten

<sup>33</sup> es wird nur die unter 3.1.2 beschriebene Konfiguration des ITM-Agenten betrachtet

| Lösungsansatz      | Vorteile  | Nachteile   |
|--------------------|---|---|
| Nagios-Plugin NRPE | <ul style="list-style-type: none"> <li>- ist auf allen Serversystemen der MTU vorhanden</li> <li>- Verschlüsselung des Datenverkehrs zwischen Monitoringsystem und zu überwachendem System ist bereits implementiert</li> <li>- indirekte Abfrage von Messwerten möglich</li> </ul> | <ul style="list-style-type: none"> <li>- nur Nagios kann als Monitoringsystem verwendet werden</li> <li>- keine SNMP - Unterstützung</li> </ul> |

**Tabelle 4: Vergleich der Lösungsansätze**

Da bei der MTU IBM Tivoli das Standardmonitoringtool im Bereich des IT-Betriebes ist, wurde entschieden, die zu entwerfende Schnittstelle so zu erweitern, dass die Messwerte zusätzlich durch den angepassten ITM-Agenten abrufbar sind. Eine Erweiterung der Schnittstelle hinsichtlich einer Anbindung an NRPE wird nicht in Betracht gezogen, da NRPE zurzeit nicht verwendet wird.

### 3.2 Definition der Schnittstelle

#### 3.2.1 Bestandteile der Schnittstelle

Die Schnittstelle umfasst folgende Elemente:

- ein zentrales Verarbeitungsskript,
- eine Skriptkonfigurationsdatei (processdata.conf),
- eine oder mehrere Konfigurationsdateien (eine für jede zu überwachende Anwendung),
- jeweils eine dazugehörige Messwertedatei,
- eine Datei, welche die aktuelle Konfiguration der Schnittstelle enthält (currentstatus.tmp).

Die Konfigurationsdatei enthält sämtliche Konfigurationsinformationen, die für die Messwerte notwendig sind. Die eigentlichen Messwerte befinden sich in einer dazu-

gehörigen Messwertedatei. Dabei wurde für einen ersten Prototyp der Schnittstelle festgelegt, dass eine Konfigurationsdatei genau eine Messwertedatei beschreibt und diese Dateien im gleichen Verzeichnis liegen. Das Skript hat die Aufgabe, die Messwerte so aufzubereiten, dass sie per SNMP abrufbar sind. Der SNMP-Daemon wird dabei nicht direkt durch das Skript konfiguriert. Für diese Aufgaben stehen dem Skript die Hilfsdateien „processdata.conf“ und „currentstatus.tmp“ zur Verfügung.

### 3.2.2 Namenskonventionen für die Konfigurations- und Messwertedatei

Für die Bildung der Dateinamen sind ausschließlich Zahlen, Großbuchstaben, Kleinbuchstaben und der Unterstrich zulässig. Als Dateierweiterung für die Konfigurationsdatei „.conf“ zu verwenden. Eine Messwertedatei hat die Dateierweiterung „.data“. Um eine Zuordnung der beiden Dateien zu gewährleisten, muss der Dateiname der Konfigurations- mit dem der Messwertedatei übereinstimmen. Hierbei ist bei Linux-Systemen auf die Groß- und Kleinschreibung der Dateinamen zu achten. Die Länge der Dateinamen wird durch die maximal zulässige Länge von Dateinamen des zugrunde liegenden Betriebssystems bestimmt.

### 3.2.3 Definition der Konfigurationsdatei

Die Konfigurationsdatei ist in einen allgemeinen und einen messwertspezifischen Teil gegliedert. Der allgemeine Teil hat eine definierte (fixe) Länge und dient zum Festlegen von allgemeinen Konfigurationswerten (z.B. Autor, Erstellungsdatum, Trennzeichen). Die Länge des messwertspezifischen Teils ist variabel und definiert die einzelnen Messwerte, für deren Anzahl es keine Begrenzung gibt. Die nachfolgende Tabelle zeigt die Werte, die im allgemeinen und messwertspezifischen Teil angesiedelt sind.

| allgemeiner Teil    | messwertspezifischer Teil |
|---------------------|---------------------------|
| ABOUT               | INDEX                     |
| AUTHOR              | NAME                      |
| CONTACT             | DESCRIPTION               |
| DATE OF CREATION    | TYPE                      |
| DATE OF LAST CHANGE | DISPLAYTYPE               |

| allgemeiner Teil        | messwertspezifischer Teil |
|-------------------------|---------------------------|
| SEPARATOR               | DISPLAYART                |
| SEPARATOR DECIMAL VALUE | ERROR-LEVEL               |
| DATAFILE                | ALARMERROR-LEVEL          |
| BASE-OID                | TEXTERROR-LEVEL           |
| EMAIL1                  | WARNING-LEVEL             |
| EMAIL2                  | ALARMWARNING-LEVEL        |
| EMAIL3                  | TEXTWARNING-LEVEL         |
| SMS1                    |                           |
| SMS2                    |                           |
| SMS3                    |                           |
| TEXT1                   |                           |
| TEXT2                   |                           |
| TEXT3                   |                           |

**Tabelle 5: Gliederung der textbasierten Konfigurationsdatei**

Jeder Konfigurationswert des allgemeinen Teils ist in einer neuen Zeile der Konfigurationsdatei gespeichert. Die Trennung zwischen der Wertebeschreibung und des eigentlichen Konfigurationswertes erfolgt durch das Gleichheitszeichen. Der messwertspezifische Teil ist in Zeilenform aufgebaut. Die einzelnen Elemente werden durch ein festgelegtes Trennzeichen (z.B. Semikolon) voneinander abgeteilt. Dieses wird im allgemeinen Teil der Konfigurationsdatei im Konfigurationswert „SEPARATOR“ festgelegt. Für jeden zu definierenden Messwert ist eine separate Zeile zu verwenden.

Für alle Zeilen der Konfigurationsdatei gilt:

- Leerzeilen und Zeilen, die mit dem Doppelkreuz beginnen, werden von der Skriptverarbeitung ausgeschlossen,
- zusätzlich werden alle Zeichen einer Zeile, die sich rechts des Doppelkreuzes befinden, ignoriert.

In der Tabelle 5 ist die Reihenfolge der Konfigurationswerte des allgemeinen und messwertspezifischen Teils festgelegt. Um den Anfang und das Ende der Konfigurationsdatei anzugeben, werden alle Elemente in eine Anfangs- (-BEGIN-) und Endmarkierung (-END-) eingefasst.

Die in der Konfigurationsdatei enthaltenen Werte haben folgende Bedeutung:

#### ABOUT:

- kurze Angaben zu den Messwerten, die mit Hilfe der Konfigurationsdatei überwacht werden
- Inhalt dieses Wertes ist auf eine Zeile begrenzt
- Beschreibungen über mehrere Zeilen können als Kommentar in die Konfigurationsdatei aufgenommen werden

#### AUTHOR:

- gibt den Autor der Konfigurationsdatei an
- zugleich Ansprechpartner und Verantwortlicher für die Konfigurationsdatei und die dazugehörige Messwertedatei

#### CONTACT:

- Angabe der E-Mail-Adresse oder Telefonnummer des Autors der Konfigurationsdatei als Kontakt (z.B. für Rückfragen)

#### DATE OF CREATION:

- beinhaltet das Datum der Erstellung der Konfigurationsdatei als String
- zu verwendendes Datumsformat: YYYY.MM.DD

#### DATE OF LAST CHANGE:

- beinhaltet das Datum der letzten Änderung der Konfigurationsdatei als String
- zu verwendendes Datumsformat: YYYY.MM.DD

#### SEPARATOR:

- legt das Trennzeichen in der Messwertedatei und für die einzelnen Konfigurationswerte des messwertspezifischen Teils fest
- Verwendung verschiedener Trennzeichen für Konfigurations- und Messwertedatei nicht möglich
- nur Kommas, Doppelpunkte oder Semikolons zulässig
- Unterscheidung vom Trennzeichen für Dezimalwerte

#### SEPARATOR DECIMAL VALUE:

- legt Trennzeichen der Dezimalwerte fest
- nur Kommas oder Punkte zulässig
- Unterscheidung vom Trennzeichen für die Konfigurationswerte eines Messwertes

#### DATAFILE:

- gibt die zugehörige Messwertedatei an
- Angabe des vollständigen Dateipfades

#### BASE-OID:

- gibt die Basis-OID an, unter der die einzelnen Messwerte einer Konfigurationsdatei abrufbar sind
- nur OID's in numerischer Form zulässig (z.B. 1.3.4.5.11.6.2.1)
- muss immer mit einer Zahl beginnen
- darf in allen durch das Skript zu verarbeitenden Konfigurationsdateien nur einmal vorkommen

#### EMAIL1:

- gibt die E-Mail-Adresse an, an die eine Alarmierung bzw. Warnung versendet werden soll
- gilt ebenso für die Werte EMAIL2 und EMAIL3



#### SMS1:

- gibt die Telefonnummer an, an die eine Alarmierung bzw. Warnung versendet werden soll
- gilt ebenso für die Werte SMS2 und SMS3

#### TEXT1:

- gibt den Benachrichtigungstext an, der bei der Alarmierung bzw. Warnung in einer SMS oder E-Mail verwendet werden soll
- Länge des Textes ist auf eine Zeile begrenzt
- gilt ebenso für die Werte TEXT2 und TEXT3

#### INDEX:

- wird für die Bildung der OID des Messwertes benötigt
- OID eines Messwertes ergibt sich aus der Basis-OID für die Konfigurationsdatei und dem vergebenen Index
- besteht aus einer Zahl, die größer als Null ist
- darf in einer Konfigurationsdatei nur einmal vorkommen

#### NAME:

- gibt den Namen des konfigurierten Messwertes an
- kann aus Zahlen und/oder Buchstaben bestehen
- darf keine Leerzeichen enthalten
- kurze und aussagekräftige Namen sind zu wählen

#### DESCRIPTION:

- beschreibt den zu konfigurierenden Messwert

#### TYPE:

- gibt den Datentyp des Messwertes an
- dabei ist folgendes definiert:

| Option  | Beschreibung   |
|---------|--|
| INTEGER | bei dem Messwert handelt es sich um eine 64-Bit Ganzzahl |
| FLOAT   | bei dem Messwert handelt es sich um eine Dezimalzahl     |

**Tabelle 6: Optionen des Wertes Type**

**DISPLAYTYPE:**

- legt fest, wie der Messwert von einem Monitoringsystem interpretiert werden soll
- dieser Wert kann je nach Monitoringsystem unterschiedlich sein
- folgende Festlegungen gelten:

| Option     | Darstellung der Messwerte als  |
|------------|--------------------------------|
| ABSOLUT    | absoluter Wert                 |
| COUNTER    | 32-Bit Zähler (SNMPv1)         |
| COUNTER32  | 32-Bit Zähler (SNMPv2)         |
| COUNTER64  | 64-Bit Zähler                  |
| GAUGE      | 32-Bit Ganzzahl (SNMPv1)       |
| GAUGE32    | 32-Bit Ganzzahl (SNMPv2)       |
| GAUGE64    | 64-Bit Ganzzahl                |
| INTEGER    | 32-Bit Ganzzahl (SNMPv1)       |
| INTEGER32  | 32-Bit Ganzzahl (SNMPv2)       |
| UNSIGNED32 | vorzeichenlose 32-Bit Ganzzahl |

**Tabelle 7: Optionen des Wertes Displaytype**

**DISPLAYART:**

- legt fest, wie der Messwert durch das Monitoringsystem dargestellt werden soll
- in der ersten Version der Schnittstelle sind nur Graphen möglich, dies ist mit „GRAPH“ in der Konfigurationsdatei anzugeben

**ERROR-LEVEL:**

- definiert den Grenzwert, bei dem eine Alarmierung ausgelöst werden soll

#### ALARMERROR-LEVEL:

- beschreibt, welche E-Mail-Adresse oder SMS-Nummer im Alarmierungsfall verwendet werden soll
- mehrere Einträge sind zulässig und durch ein Leerzeichen zu trennen
- an dieser Stelle ist nicht die Zieladresse anzugeben, sondern die im allgemeinen Teil verwendeten Kürzel (z.B. EMAIL1, SMS3)

#### TEXTERROR-LEVEL:

- beschreibt den Text, der bei einer Benachrichtigung verwendet werden soll
- mehrere Einträge sind zulässig und durch ein Leerzeichen zu trennen
- an dieser Stelle ist nicht der Text anzugeben, sondern die im allgemeinen Teil verwendeten Kürzel (z.B. TEXT1)

#### WARNING-LEVEL:

- definiert den Schwellwert, bei dem eine Warnung ausgelöst werden soll

#### ALARMWARNING-LEVEL:

- beschreibt, welche E-Mail-Adresse oder SMS-Nummer im Warnungsfall verwendet werden soll
- mehrere Einträge sind zulässig und durch ein Leerzeichen zu trennen
- an dieser Stelle ist nicht die Zieladresse anzugeben, sondern die im allgemeinen Teil verwendeten Kürzel (z.B. EMAIL1, SMS3)

#### TEXTWARNING-LEVEL:

- beschreibt den Text, der bei einer Benachrichtigung verwendet werden soll
- mehrere Einträge sind zulässig und durch ein Leerzeichen zu trennen
- an dieser Stelle ist nicht der Text anzugeben, sondern die im allgemeinen Teil verwendeten Kürzel (z.B. TEXT1, TEXT2)

Falls ein Konfigurationswert nicht gesetzt wird, ist dieser freizulassen. Folgende Konfigurationswerte müssen gesetzt werden:

| allgemeiner Teil        | messwertspezifischer Teil |
|-------------------------|---------------------------|
| ABOUT                   | INDEX                     |
| AUTHOR                  | NAME                      |
| CONTACT                 | TYPE                      |
| DATE OF CREATION        | DISPLAYTYPE               |
| DATE OF LAST CHANGE     | DISPLAYART                |
| SEPERATOR               |                           |
| SEPERATOR DECIMAL VALUE |                           |
| DATAFILE                |                           |
| BASE-OID                |                           |

**Tabelle 8: erforderliche Werte des allgemeinen und messwertspezifischen Teils**

## Beispiel:

```
-BEGIN-
#general part/allgemeiner Teil
ABOUT=ueberwachung der anzahl aus-/ eingehender e-mails
AUTHOR=Nils Scharrnbeck #Praktikant 03/2008-08/2008
CONTACT=Nils.Scharrnbeck@test.de
DATE OF CREATION=2008.11.26
DATE OF LAST CHANGE=2008.11.27
SEPARATOR=;
SEPARATOR DECIMAL VALUE=,
DATAFILE=C:\Scripttest\files\Konfigurationsdateitemplate1.data
BASE-OID=1.3.6.1.4.1.2021.50.1.100

#EMAIL
EMAIL1=Nils.Scharrnbeck@test.de #E-Mail des Autors
EMAIL2=helpdesk@test.de # E-Mail Hotline
EMAIL3=Nils.Scharrnbeck@privat.de #alternative E-Mail des Autors

#SMS
SMS1=089-380-01 #Telefon Scharrnbeck
SMS2=089-380-110 #Telefohn Hotline
SMS3=0196-123456789 #alternative Telefonnummer Scharrnbeck

#TEXT
TEXT1=Schwellwert ueberschritten
TEXT2=Grenzwert ueberschritten
TEXT3=Vorgesetzten informieren

#special part/ messwertspezifischer Teil
#INDEX/NAME OF MEASURED VALUE;DESCRIPTION;TYPE;DISPLAYART;ERROR-LEVEL;ALARMERROR-LEVEL;TEXTERROR-LEVEL;
#WARNING-LEVEL;ALARMWARNING-LEVEL;TEXTWARNING-LEVEL
1;mailsinlh;eingehende Mails der letzten h;INTEGER;ABSOLUT;GRAPH;900;EMAIL1 SMS1;TEXT2;800;; #unkritisch
2;mailoutlh;eingehende Mails der letzten h;INTEGER;ABSOLUT;GRAPH;900;EMAIL1 SMS1;TEXT2;800;; #unkritisch
3;mailsin6h;eingehende Mails der letzten 6h;INTEGER;ABSOLUT;GRAPH;5400;EMAIL1 SMS1;TEXT2;4800;SMS1 SMS3;TEXT1
4;mailout6h;ausgehende Mails der letzten 6h;INTEGER;ABSOLUT;GRAPH;5400;EMAIL1 SMS1;TEXT2;4800;SMS1 SMS3;TEXT1
5;mailsin24h;eingehende Mails der letzten 24h;INTEGER;ABSOLUT;GRAPH;22500;EMAIL1 EMAIL2 SMS3;TEXT2 TEXT3;20000;EMAIL1;TEXT1
#kritisch
6;mailout24h;ausgehende Mails der letzten 24h;INTEGER;ABSOLUT;GRAPH;22500;EMAIL1 EMAIL2 SMS3;TEXT2 TEXT3;20000;EMAIL1;TEXT1
#kritisch
-END-
```

### 3.2.4 Definition der Messwertedatei

Sie enthält die Messwerte, die durch die Konfigurationsdatei beschrieben werden. Dabei ist festgelegt, dass in jeder Zeile ein Messwert gespeichert wird. Wegen der besseren Zuordnung steht vor jedem Messwert sein Name, den er laut Konfigurationsdatei besitzt. Die Trennung des Namens vom Messwert erfolgt durch das in der dazugehörigen Konfigurationsdatei definierte Trennzeichen (SEPARATOR). Der angegebene Name des Messwertes muss mit dem Namen in der Konfigurationsdatei übereinstimmen. Dabei wird auf Groß- und Kleinschreibung der Messwertnamen geachtet. Die Reihenfolge der einzelnen Messwerte in der Messwertedatei muss nicht mit der Reihenfolge übereinstimmen, wie sie in der Konfigurationsdatei definiert wurde. Die Verwendung von Kommentar- und Leerzeilen ist analog zu den Konfigurationsdateien möglich. Als Zeichen für den Zeilenumbruch muss „\n“ verwendet werden. Die Verwendung von dezimalen Messwerten erfolgt in Dezimalschreibweise. Weiterhin dürfen die Messwerte nur den Wert und keine Zusatzinformation (z.B. Maßeinheiten, Prozentzeichen) beinhalten. Die Messwertedateien werden vom Anwendungsverantwortlichen erstellt (z.B. durch eigene Skripte oder im Einzelfall manuell) und zur Verfügung gestellt. Um einen gleichzeitigen Schreib- und Lesezugriff auf die Messwertedatei zu verhindern, muss dieses Skript, welches die Messwerte in die Datei schreibt, vor dem Schreibzugriff eine Datei mit der Endung „.LOCK“ erstellen. Für den Dateinamen ist der gleiche Name wie für die Messwertedatei zu verwenden. Nach Beendigung des Schreibzugriffes ist diese Datei durch das Skript wieder zu löschen.

#### Beispiel:

```
mailsin24h;15 #kritisch  
mailsout24h;0 #kritisch
```

```
mailsin1h;61  
mailsout1h;98
```

```
mailsin6h;70  
mailsout6h;27
```

### 3.2.5 Leistungsumfang des Skriptes

Innerhalb der Schnittstelle bildet das Skript das zentrale (verbindende) Element. Dessen Aufgabe ist es, eine durch den Anwendungsverantwortlichen erstellte Messwertedatei so aufzubereiten, dass die gespeicherten Messwerte per SNMP oder generischem ITM-Agenten abrufbar sind. Weiterhin stehen eine Funktion zum Validieren einer Konfigurationsdatei, zum automatischen Erstellen der Konfigurationszeilen für die Datei snmpd.conf sowie ein Log-Mechanismus zur Verfügung. Eine detaillierte Beschreibung des Skriptes erfolgt unter Punkt 4.

### 3.2.6 Beschreibung der Hilfsdateien „processdata.conf“ und „currentstatus.tmp“

Die Hilfsdatei „processdata.conf“ enthält die vom Skript zu verarbeitenden Konfigurationsdateien. Sie ist so aufgebaut, dass der Dateipfad jeder Konfigurationsdatei in einer separaten Zeile angegeben wird. Diese Angabe muss als absoluter Dateipfad erfolgen, da sich die Konfigurationsdateien nicht im gleichen Verzeichnis befinden können wie das Verarbeitungsskript.

Die Datei „currentstatus.tmp“ enthält hingegen die aktuell gültigen Werte der Schnittstelle. Dabei werden alle notwendigen Werte des allgemeinen und messwert-spezifischen Teils jeder Konfigurationsdatei zeilenweise zwischengespeichert:

| Werte des allgemeinen Teils | Werte des messwertspezifischen Teils  |
|-----------------------------|---------------------------------------|
| „cp“                        | NAME                                  |
| ABOUT                       | DESCRIPTION                           |
| BASE-OID                    | OID des Messwertes                    |
| AUTHOR                      | TYPE                                  |
| CONTACT                     | DISPLAYTYPE                           |
| DATE OF CREATION            | DISPLAYART                            |
| DATE OF LAST CHANGE         | Messwert                              |
| DATAFILE                    | ERROR-LEVEL                           |
| EMAIL1                      | zu verwendende Alarmierungsadresse(n) |
| EMAIL2                      | beim Über- bzw. Unterschreiten des    |
| EMAIL3                      | Grenzwertes                           |

| Werte des allgemeinen Teils | Werte des messwertspezifischen Teils  |
|-----------------------------|---------------------------------------|
| SMS1                        | zu verwendender Benachrichtigungstext |
| SMS2                        | beim Über- bzw. Unterschreiten des    |
| SMS3                        | Grenzwertes                           |
| TEXT1                       | WARNING-LEVEL                         |
| TEXT2                       | zu verwendende Alarmierungsadresse(n) |
| TEXT3                       | beim Über- bzw. Unterschreiten des    |
|                             | Schwellwertes                         |
|                             | zu verwendender Benachrichtigungstext |
|                             | beim Über- bzw. Unterschreiten des    |
|                             | Schwellwertes                         |

**Tabelle 9: Übersicht über gespeicherte Werte in der Datei "currentstatus.tmp"**

Die Trennung dieser Werte erfolgt mittels Semikolon. Diese Datei wird deshalb verwendet, da ein Timeout des SNMP-Daemons verhindert werden soll. Dieser tritt auf, wenn der SNMP-Daemon zu lange auf einen Ergebniswert (z.B. eines Skriptes) warten muss. Die Datei „currentstatus.tmp“ wird vom Skript während des neuen Einlesens der Schnittstellenkonfiguration automatisch erstellt.

Dagegen muss die Datei „processdata.conf“ mit Hilfe von Skriptparametern editiert werden. Ein manuelles Bearbeiten der beiden Dateien ist nicht gestattet. Diese Hilfsdateien befinden sich im selben Verzeichnis wie das Skript.



### 3.3 Abbildung von Messwerten auf OID's

Um Abbildungen eines Skriptergebnisses auf eine OID durchführen zu können, ist das Paket NetSNMP notwendig. Dabei wird der darin enthaltene SNMP-Daemon so erweitert, dass er diese Messwerte strukturiert darstellen kann.

#### 3.3.1 Abbildung von einzeiligen Skriptergebnissen mittels exec

Für die Abbildung eines Skriptergebnisses, das aus einer Ergebniszeile besteht, kann der SNMP-Daemon mittels exec erweitert werden, indem in der Konfigurationsdatei snmpd.conf eine exec-Zeile hinzugefügt wird. Sollen mehrere einzeilige Skriptergebnisse dargestellt werden, kann die Konfigurationsdatei snmpd.conf um beliebig viele Konfigurationszeilen ergänzt werden<sup>34</sup>.

Eine solche Konfigurationszeile hat folgenden Aufbau:

```
exec Skriptname vollständiger_Pfad_zum_Skript [Skriptparameter]
```

Die Zuordnung der OID zu den Skriptergebnissen erfolgt durch den SNMP-Daemon. Mit dieser Abbildungsvariante ist es nicht möglich, die OID für das Skriptergebnis selbst festzulegen. Die Abbildung des Wertes der Ergebniszeile erfolgt dabei in dem Unterbaum „extTable“ (OID: 1.3.6.1.4.1.2021.8) der MIB „UCD-SNMP-MIB“ (OID: 1.3.6.1.4.1.2021). Der Wert des Skriptergebnisses kann unter der OID „1.3.6.1.4.1.2021.8.1.101.x“ abgerufen werden.

---

<sup>34</sup> vgl. Beispiel

Dieser Unterbaum hat folgenden Aufbau:

| Objektname   | OID                      | Beschreibung  |
|--------------|--------------------------|---|
| extTable     | 1.3.6.1.4.1.2021.8       |   |
| extEntry     | 1.3.6.1.4.1.2021.8.1     | root des Unterbaums   |
| extIndex     | 1.3.6.1.4.1.2021.8.1.1   | Index   |
| extNames     | 1.3.6.1.4.1.2021.8.1.2   | kurze, aus einem Wort bestehende Beschreibung des auszuführenden Skriptes bzw. Kommandos        |
| extCommand   | 1.3.6.1.4.1.2021.8.1.3   | gibt das ausgeführte Skript bzw. Kommando an  |
| extResult    | 1.3.6.1.4.1.2021.8.1.100 | Exitstatus der Skriptausführung   |
| extOutput    | 1.3.6.1.4.1.2021.8.1.101 | erste Zeile des Skriptergebnisses als String  |
| extErrFix    | 1.3.6.1.4.1.2021.8.1.102 | tritt während der Skript- oder Kommando- ausführung ein Fehler auf, wird der Wert auf 1 gesetzt |
| extErrFixCmd | 1.3.6.1.4.1.2021.8.1.103 | dient der Festlegung von Aktionen, die durchgeführt werden sollen, falls extErrFix auf 1 steht  |

**Tabelle 10: MIB für Abbildung eines einzeiligen Skriptergebnisses mittels exec**

Beispiel:

Konfiguration in snmpd.conf:

```
exec script1 C:\test\getdata.pl -firstValue
exec script2 C:\test\getdata.pl -secondValue
exec script3 C:\test\getdata.pl -thirdValue
```

SNMP-Abfrage:

```
$ snmpwalk -c public -On -v2c 192.168.0.25 1.3.6.1.4.1.2021.8.1.101
```

Ergebnis:

```
.1.3.6.1.4.1.2021.8.1.101.1 = STRING: 64
.1.3.6.1.4.1.2021.8.1.101.2 = STRING: 69
.1.3.6.1.4.1.2021.8.1.101.3 = STRING: 45
```

### 3.3.2 Abbildung von mehrzeiligen Skriptergebnissen mittels exec

Gibt ein Skript mehr als eine Ergebniszeile zurück, kann mittels folgender Konfigurationszeile jede dieser Ergebniszeilen auf eine separate OID abgebildet werden.

```
exec ZielOID Skriptname vollständiger_Pfad_zum_Skript [Skriptparameter]
```

Das Skriptergebnis wird auf eine Pseudo- MIB abgebildet. Dabei werden die Skriptergebnisse auf eine OID abgebildet, die in der Konfigurationszeile unter „ZielOID“ angegeben wird.

Dieser Lösungsansatz kann auch dazu verwendet werden, ein einzeiliges Skriptergebnis auf eine bestimmte OID mittels „exec“ abzubilden. Hierbei ist jedoch zu beachten, dass die ZielOID nicht durch einen anderen Messwert belegt ist.

Die Pseudo- MIB besitzt folgende Struktur:

| Name                | OID           | Bedeutung  |
|---------------------|---------------|--|
| extIndex            | ZielOID.1.1   | Index  |
| extNames            | ZielOID.2.1   | kurze, aus einem Wort bestehende Beschreibung des auszuführenden Skriptes bzw. Kommandos       |
| extCommand          | ZielOID.3.1   | gibt das ausgeführte Skript bzw. Kommando an   |
| extResult           | ZielOID.100.1 | Exitstatus der Skript- oder Kommandoausführung   |
| extOutput (1.Zeile) | ZielOID.101.1 | Ergebnis 1.Zeile als String  |
| extOutput (2.Zeile) | ZielOID.101.2 | Ergebnis 2.Zeile als String  |
| extOutput (3.Zeile) | ZielOID.101.3 | Ergebnis 3.Zeile als String  |
| extOutput (n.Zeile) | ZielOID.101.n | Ergebnis n-te Zeile als String   |
| extErrFix           | ZielOID.102.1 | tritt während der Skript- oder Kommandoausführung ein Fehler auf, wird der Wert auf 1 gesetzt  |
| extErrFixCmd        | ZielOID.103.1 | dient der Festlegung von Aktionen, die durchgeführt werden sollen, falls extErrFix auf 1 steht |

**Tabelle 11: MIB für Abbildung eines mehrzeiligen Skriptergebnisses mittels exec**

Beispiel:

Konfiguration in snmpd.conf:

```
exec 1.3.6.1.4.1.2021.50.1 script1 C:\test\getdata.pl -all
```

SNMP-Abfrage:

```
$ snmpwalk -c public -v 2c -On 192.168.0.25 1.3.6.1.4.1.2021.50.1
```

Ergebnis:

```
.1.3.6.1.4.1.2021.50.1.1.1 = INTEGER: 1
.1.3.6.1.4.1.2021.50.1.2.1 = STRING: "script1"
.1.3.6.1.4.1.2021.50.1.3.1 = STRING: "C:\test\getdata.pl -all"
.1.3.6.1.4.1.2021.50.1.100.1 = INTEGER: 0
.1.3.6.1.4.1.2021.50.1.101.1 = STRING: "System"
.1.3.6.1.4.1.2021.50.1.101.2 = STRING: "13458"
.1.3.6.1.4.1.2021.50.1.101.3 = STRING: "77"
.1.3.6.1.4.1.2021.50.1.101.4 = STRING: "111"
.1.3.6.1.4.1.2021.50.1.101.5 = STRING: "25"
.1.3.6.1.4.1.2021.50.1.101.6 = STRING: "1173"
.1.3.6.1.4.1.2021.50.1.101.7 = STRING: "0"
.1.3.6.1.4.1.2021.50.1.101.8 = STRING: "6"
.1.3.6.1.4.1.2021.50.1.101.9 = STRING: "0"
.1.3.6.1.4.1.2021.50.1.101.10 = STRING: "229"
.1.3.6.1.4.1.2021.50.1.101.11 = STRING: "1"
.1.3.6.1.4.1.2021.50.1.101.12 = STRING: "9"
.1.3.6.1.4.1.2021.50.1.101.13 = STRING: "4"
.1.3.6.1.4.1.2021.50.1.102.1 = INTEGER: 0
.1.3.6.1.4.1.2021.50.1.103.1 = ""
```

Hinweis:

Dieser Lösungsansatz ist als deprecated (veraltet) gekennzeichnet und soll nicht verwendet werden. Es wird empfohlen, „extend“ zu verwenden<sup>35</sup>.

### 3.3.3 Abbildung eines Skriptergebnisses mittels extend

Die Abbildung von ein- oder mehrzeiligen Skriptergebnissen auf eine OID kann auch unter Verwendung des extend-Befehls erfolgen. Dabei ist es möglich, für die Abbildung eine eigene OID zu verwenden.

Für die Konfiguration ist folgende Zeile in der Datei „snmpd.conf“ notwendig:

```
extend Skriptname vollständiger_Pfad_zum_Skript [Skriptparameter]
```

Wenn das Ergebnis auf eine bestimmte OID abgebildet werden soll, verändert sich die Zeile folgendermaßen:

---

<sup>35</sup> vgl. NetSNMP

```
extend .ZielOID Skriptname vollständiger_Pfad_zum_Skript [Skriptparameter]
```

Die hierfür verwendete MIB wird in Tabelle 12 beschrieben<sup>36</sup>.

Für diese Darstellung gilt folgendes: In „x“ ist die Länge des unter „Skriptname“ in der Datei snmpd.conf hinterlegten Namens abgelegt. Unter „y“ sind die ASCII-Werte der einzelnen Zeichen aus dem angegebenen Skriptnamen dargestellt. Dabei wird jeder Buchstabe durch einen Punkt voneinander getrennt.

| Name                | OID                                | Bedeutung   |
|---------------------|------------------------------------|---|
| nsExtendNumEntries  | .1.3.6.1.4.1.8072.1.3.2.1.0        | Anzahl der erwarteten Einträge (Entries)                |
| nsExtendCommand     | .1.3.6.1.4.1.8072.1.3.2.2.1.2.x.y  | Pfad des ausgeführten Skriptes oder Kommandos           |
| nsExtendArgs        | 1.3.6.1.4.1.8072.1.3.2.2.1.3.x.y   | Argument für die Ausführung des Skriptes oder Kommandos |
| nsExtendInput       | .1.3.6.1.4.1.8072.1.3.2.2.1.4.x.y  | zusätzliche Eingabe für das Skript                      |
| nsExtendCacheTime   | .1.3.6.1.4.1.8072.1.3.2.2.1.5.x.y  | CacheTime   |
| nsExtendExecType    | .1.3.6.1.4.1.8072.1.3.2.2.1.6.x.y  | Ausführungstyp  |
| nsExtendRunType     | .1.3.6.1.4.1.8072.1.3.2.2.1.7.x.y  | Runtyp  |
| nsExtendStorage     | .1.3.6.1.4.1.8072.1.3.2.2.1.20.x.y | Art der Speicherung                                     |
| nsExtendStatus      | .1.3.6.1.4.1.8072.1.3.2.2.1.21.x.y | Status  |
| nsExtendOutput1Line | .1.3.6.1.4.1.8072.1.3.2.3.1.1.x.y  | 1. Zeile der Ausgabe                                    |
| nsExtendOutputFull  | .1.3.6.1.4.1.8072.1.3.2.3.1.2.x.y  | komplette Ausgabe des                                   |

<sup>36</sup> vgl. <http://csn.cs.psu.ac.th/~kui/doc/ns-extend-tree.pdf> (12.01.2009)

| Name                         | OID                                 | Bedeutung  |
|------------------------------|-------------------------------------|--|
| nsExtendOutNumLines          | .1.3.6.1.4.1.8072.1.3.2.3.1.3.x.y   | Skriptergebnisses<br>Anzahl der<br>Ausgabezeilen des<br>Skriptes oder<br>Kommandos |
| nsExtendResult               | .1.3.6.1.4.1.8072.1.3.2.3.1.4.x.y   | Exitstatus der Skript-<br>oder Kommandoaus-<br>führung                             |
| nsExtendOutLine<br>(1.Zeile) | .1.3.6.1.4.1.8072.1.3.2.4.1.2.x.y.1 | Ergebnis 1.Zeile als<br>String   |
| nsExtendOutLine<br>(2.Zeile) | .1.3.6.1.4.1.8072.1.3.2.4.1.2.x.y.2 | Ergebnis 2.Zeile als<br>String   |
| nsExtendOutLine<br>(3.Zeile) | .1.3.6.1.4.1.8072.1.3.2.4.1.2.x.y.3 | Ergebnis 3.Zeile als<br>String   |
| nsExtendOutLine<br>(n.Zeile) | .1.3.6.1.4.1.8072.1.3.2.4.1.2.x.y.n | Ergebnis n-te Zeile als<br>String  |

**Tabelle 12: MIB für Abbildung eines Skriptergebnisses mittels extend**

Beispiel:

Konfiguration in snmpd.conf:

```
extend System perl C:\test\getdata.pl -all
```

SNMP-Abfrage

```
$ snmpwalk -v 2c -c public -On 192.168.0.25 .1.3.6.1.4.1.8072.1.3.2
```

## Ergebnis:

```
.1.3.6.1.4.1.8072.1.3.2.1.0 = INTEGER: 1
.1.3.6.1.4.1.8072.1.3.2.2.1.2.6.83.121.115.116.101.109 = STRING: "perl"
.1.3.6.1.4.1.8072.1.3.2.2.1.3.6.83.121.115.116.101.109 = STRING:
"C:\test\getdata.pl -all"
.1.3.6.1.4.1.8072.1.3.2.2.1.4.6.83.121.115.116.101.109 = ""
.1.3.6.1.4.1.8072.1.3.2.2.1.5.6.83.121.115.116.101.109 = INTEGER: 5
.1.3.6.1.4.1.8072.1.3.2.2.1.6.6.83.121.115.116.101.109 = INTEGER: 1
.1.3.6.1.4.1.8072.1.3.2.2.1.7.6.83.121.115.116.101.109 = INTEGER: 1
.1.3.6.1.4.1.8072.1.3.2.2.1.20.6.83.121.115.116.101.109 = INTEGER: 4
.1.3.6.1.4.1.8072.1.3.2.2.1.21.6.83.121.115.116.101.109 = INTEGER: 1
.1.3.6.1.4.1.8072.1.3.2.3.1.1.6.83.121.115.116.101.109 = STRING: "System"
.1.3.6.1.4.1.8072.1.3.2.3.1.2.6.83.121.115.116.101.109 = STRING: "System
13458
77
111
25
1173
0
6
0
229
1
9
4"
.1.3.6.1.4.1.8072.1.3.2.3.1.3.6.83.121.115.116.101.109 = INTEGER: 13
.1.3.6.1.4.1.8072.1.3.2.3.1.4.6.83.121.115.116.101.109 = INTEGER: 0
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.1 = STRING: "System"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.2 = STRING: "13458"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.3 = STRING: "77"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.4 = STRING: "111"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.5 = STRING: "25"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.6 = STRING: "1173"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.7 = STRING: "0"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.8 = STRING: "6"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.9 = STRING: "0"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.10 = STRING: "229"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.11 = STRING: "1"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.12 = STRING: "9"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.6.83.121.115.116.101.109.13 = STRING: "4"
```

Länge des Skriptnamens „System“: 6 Zeichen

„OID“ für den Skriptnamen: .83.121.115.116.101.109

Wenn eine eigene OID verwendet werden soll, wird der Teil der OID „.1.3.6.1.4.1.8072. 1.3.2“ durch die angegebene ZielOID ersetzt. Wichtig hierbei ist, dass eine OID genutzt werden soll, die nicht durch einen anderen Wert belegt ist.



### 3.3.4 Vergleich der Abbildungsmöglichkeiten

| Lösungsansatz  | Vorteile   | Nachteile   |
|--|--|---|
| Abbildung eines<br>einzeiligen Skript-<br>ergebnisses mittels exec | <ul style="list-style-type: none"> <li>- eigene MIB für die<br/>Abbildung der<br/>Messwerte</li> </ul>   | <ul style="list-style-type: none"> <li>- Ergebniswerte können<br/>nicht auf eigene OID's<br/>abgebildet werden</li> <li>- nur der erste Wert des<br/>Skriptes ist auf eine OID<br/>abbildbar</li> </ul> |
| Abbildung von<br>mehrzeiligen Skript-<br>ergebnissen mittels exec  | <ul style="list-style-type: none"> <li>- Abbildung auf „eigene“<br/>OID's möglich</li> <li>- mehrere Messwerte<br/>können auf eine OID<br/>abgebildet werden</li> </ul>  | <ul style="list-style-type: none"> <li>- keine eigene MIB im<br/>Hintergrund (nur<br/>Pseudo- MIB)</li> <li>- Ansatz ist laut<br/>NetSNMP-<br/>Dokumentation veraltet</li> </ul>                        |
| Abbildung eines Skript-<br>ergebnisses mittels extend              | <ul style="list-style-type: none"> <li>- Abbildung auf „eigene“<br/>OID's möglich</li> <li>- eigene MIB für die<br/>Abbildung der Mess-<br/>werte im Hintergrund</li> <li>- mehrere Messwerte<br/>können auf eine OID<br/>abgebildet werden</li> </ul> | <ul style="list-style-type: none"> <li>- Verwendung<br/>komplizierter, langer und<br/>unübersichtlicher OID's</li> </ul>  |

**Tabelle 13: Vor- und Nachteile der Abbildungsmöglichkeiten**

Aufgrund der Vor- und Nachteile, die sich aus der Tabelle 13 für die einzelnen Abbildungsarten ergeben, habe ich für die Diplomarbeit festgelegt, dass der unter Punkt 3.3.3 beschriebene Ansatz verwendet wird.

### 3.4 Verwaltung und Struktur der zu verwendenden OID's

#### 3.4.1 Verwaltung der OID's

Die Notwendigkeit besteht darin, einen Überblick über bereits vergebene und noch freie OID's zu schaffen, um Doppelbelegungen zu vermeiden. Außerdem muss eine Dokumentation existieren, aus der hervorgeht, welche Messwerte welcher OID zugeordnet sind. Diese kann in Form einer Exceltabelle geschrieben werden, die folgende Spalten beinhaltet:

- Datum,
- OID des Messwertes,
- Messwertname,
- System,
- Beschreibung,
- Konfigurationsdatei,
- Verantwortlicher,
- Kommentar.

Die Verwaltung der OID's muss durch eine zentrale Stelle in einem Unternehmen erfolgen.

#### 3.4.2 Struktur der OID's

Für die Struktur einer zu verwendenden OID wurde festgelegt, dass für jede Konfigurationsdatei eine Basis-OID genutzt wird. In der Regel werden in einer Konfigurationsdatei die Messwerte einer Anwendung definiert. Hierbei erfolgt eine Zuordnung zwischen einer Basis-OID und der zu überwachenden Anwendung. Im Anschluss an die Basis-OID folgt ein Index, der für jeden einzelnen Messwert dieser Konfigurationsdatei unterschiedlich sein muss. Dieser wird vom Anwender festgelegt. Weiter gilt, dass alle allgemeinen Informationen der Konfigurationsdatei (z.B. Autor, Erstellungsdatum) unter dem Index „0“ abrufbar sind. Die OID eines solchen einzelnen allgemeinen Konfigurationswertes setzt sich aus der Basis- OID, dem Index „0“, einem festen Anteil „.4.1.2.2.99.112“ und der Nummer, unter der der Wert abrufbar ist, zusammen.

Dabei ist folgendes für diese Nummer (n) definiert:

| Nummer (n) | Beschreibung   |
|------------|--|
| 1          | „cp“ (wird vom Skript für jede Konfigurationsdatei vergeben) |
| 2          | ABOUT  |
| 3          | BASE- <b>OID</b>   |
| 4          | AUTHOR   |
| 5          | CONTACT  |
| 6          | DATE OF CREATION   |
| 7          | DATE OF LAST CHANGE  |
| 8          | DATAFILE   |
| 9          | EMAIL1   |
| 10         | EMAIL2   |
| 11         | EMAIL3   |
| 12         | SMS1   |
| 13         | SMS2   |
| 14         | SMS3   |
| 15         | TEXT1  |
| 16         | TEXT2  |
| 17         | TEXT3  |

**Tabelle 14: Ausgabezeilen des allgemeinen Teils einer Konfigurationsdatei**

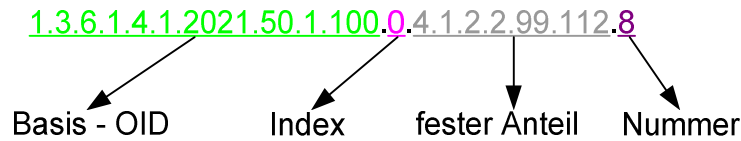
Für den Fall, dass einige dieser Werte nicht gesetzt sind, bleibt das Ergebnisfeld der Ausgabe leer.

Beispiel: „Abrufen der zugehörigen Messwertedatei“

Basis-**OID**: 1.3.6.1.4.1.2021.50.1.100

Nummer der Ausgabezeile, die die Messwertedatei enthält: 8

Für das Beispiel ergibt sich folgende OID, die mit snmpget abgerufen werden kann:



```
$ snmpget -c public -v2c -On 192.168.0.25
1.3.6.1.4.1.2021.50.1.100.0.4.1.2.2.99.112.8

.1.3.6.1.4.1.2021.50.1.100.0.4.1.2.2.99.112.8 = STRING: "Konfigurationsdateitemplate1.data"
```

Um eine einzelne Information zu einem bestimmten Messwert (z.B. dem aktuellen Wert) abzurufen, muss eine OID verwendet werden, die sich aus der Basis-OID und dem Index, den der Anwender diesem Messwert vergeben hat, ergibt. Diese beiden Bestandteile ergeben zusammen die OID für den Messwert. Im Anschluss folgt der fixe Anteil „.4.1.2“. Danach werden die Länge des vergebenen Messwertnamens und der Messwertname als OID angefügt. Schließlich folgt die Nummer, unter der die abzufragende Information des Messwertes abrufbar ist.



Für die Nummer (m) ist folgendes definiert:

| Nummer (m) | Beschreibung   |
|------------|--|
| 1          | NAME   |
| 2          | DESCRIPTION  |
| 3          | OID des Messwertes   |
| 4          | TYPE   |
| 5          | DISPLAYTYPE  |
| 6          | DISPLAYART   |
| 7          | Messwert   |
| 8          | ERROR-LEVEL  |
| 9          | zu verwendende Alarmierungsadresse(n) beim Über- bzw. Unterschreiten des Grenzwertes |

| Nummer (m) | Beschreibung   |
|------------|--|
| 10         | zu verwendender Benachrichtigungstext beim Über- bzw. Unterschreiten des Grenzwertes   |
| 11         | WARNING-LEVEL  |
| 12         | zu verwendende Alarmierungsadresse(n) beim Über- bzw. Unterschreiten des Schwellwertes |
| 13         | zu verwendender Benachrichtigungstext beim Über- bzw. Unterschreiten des Schwellwertes |

**Tabelle 15: Ausgabezeilen der Messwertparameter**

Für den Fall, dass einige dieser Werte nicht gesetzt sind, bleibt das Ergebnisfeld der Ausgabe leer.

Beispiel: Abrufen des aktuellen Wertes des Messwertes „mailout24h“

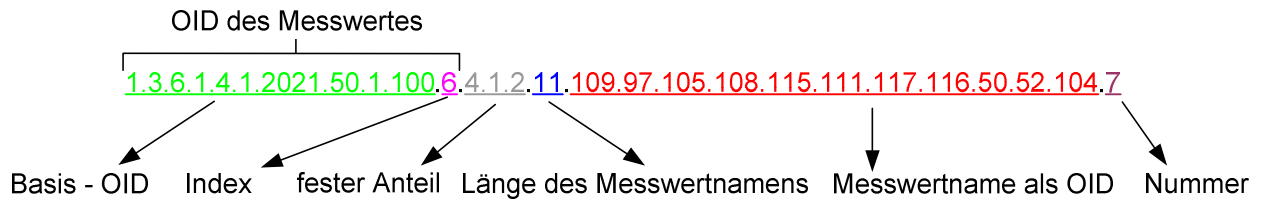
Basis-OID: **1.3.6.1.4.1.2021.50.1.100**  
Index des Messwertes: **6**  
Messwertname: mailout24h  
OID des Messwertes: **1.3.6.1.4.1.2021.50.1.100.6**  
Länge des Messwertnamens: **11**  
Messwertname als OID: **109.97.105.108.115.111.117.116.50.52.104**

Für den Messwertnamen „mailout24h“ ergibt sich folgende Zuordnung zwischen den einzelnen Zeichen (oben) und den dazugehörigen ASCII-Zeichen-Werten (unten):

|     |    |     |     |     |     |     |     |    |    |     |
|-----|----|-----|-----|-----|-----|-----|-----|----|----|-----|
| m   | a  | i   | l   | s   | o   | u   | t   | 2  | 4  | h   |
| 109 | 97 | 105 | 108 | 115 | 111 | 117 | 116 | 50 | 52 | 104 |

Ausgabezeile für den Messwert: **7**

Für das Beispiel ergibt sich folgende OID, die mit snmpget abgerufen werden kann:



```
$ snmpget -c public -v2c -On 192.168.0.25
```

```
1.3.6.1.4.1.2021.50.1.100.6.4.1.2.11.109.97.105.108.115.111.117.116.50.52.104.7
```

```
1.3.6.1.4.1.2021.50.1.100.6.4.1.2.11.109.97.105.108.115.111.117.116.50.52.104.7 = STRING: "25"
```

## 4 Umsetzung und Dokumentation der Schnittstelle

### 4.1 Voraussetzungen für den Einsatz der Schnittstelle

Um das Skript ausführen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Betriebssystem: Windows XP Service Pack 2 oder  
Windows Server 2003 Service Pack 2 oder  
Linux
- Perl: ab Version: 5.8.8 Build 822
- benötigte Perlmodule: Switch-2.14<sup>37</sup>  
Getopt-Long 2.37\_02<sup>38</sup>
- NetSNMP: ab Version: 5.4.1

Zusätzlich wird das Schreib- und Ausführungsrecht in jenem Verzeichnis benötigt, in dem sich das Skript befindet. Weiter wird unter Linux das Recht benötigt, den SNMP-Daemon neu zu starten.

### 4.2 Beschreibung der Funktionsweise der Schnittstelle

Das Skript ist das zentrale Element der Schnittstelle, welches in der Lage ist, eine oder mehrere Konfigurationsdateien zu verarbeiten, die sich auf dem gleichen System wie das Skript befinden müssen. Die Verarbeitung von Konfigurationsdateien, die sich auf anderen Systemen befinden, wird in dieser Version nicht unterstützt. Die Messwertedateien werden in bestimmten Abständen durch ein beliebiges Skript aktualisiert. Um diese aktuellen Messwerte in die Schnittstelle übernehmen zu können, muss eine Aktualisierung der Schnittstelle durchgeführt werden. Hierfür bietet das Skript verschiedene Möglichkeiten. Bei der ersten wird die gesamte Schnittstelle neu eingelesen. Dabei werden alle in der Hilfsdatei „processdata.conf“ vorhandenen Konfigurationsdateien abgerufen und die dort definierten Messwerte so aufbereitet, dass das Skript in der Lage ist, die Parameterwerte und die zugehörigen Messwerte eines definierten Messwertes per SNMP oder ITM-Agenten abzurufen. Das Ergebnis dieser Operation ist die Erstellung der Datei „currentstatus.tmp“. Vorher werden die Konfigurationsdateien überprüft, ob sie der Definition ent-

---

<sup>37</sup> Download unter: <http://search.cpan.org/~rgarcia/Switch-2.14> (26.02.2009)

<sup>38</sup> Download unter: <http://search.cpan.org/dist/Getopt-Long> (26.02.2009)

sprechen. Bei eventuell auftretenden Fehlern wird entweder die gesamte fehlerhafte Konfigurationsdatei oder der fehlerhafte Messwert nicht in die Konfiguration übernommen. Die gefundenen Fehler können optional in einer Log-Datei gespeichert werden. Die zweite Möglichkeit besteht darin, dass von allen definierten Messwerten nur der Wert aktualisiert wird. Bei der dritten Variante können nur Messwerte einer bestimmten Konfigurationsdatei aktualisiert werden. Die Abbildung der Messwerte auf bestimmte OID's erfolgt, indem das Skript durch einen erweiterten SNMP-Daemon mit den entsprechenden Parametern aufgerufen wird. Dabei verarbeitet der SNMP-Daemon die Ausgabezeilen des Skriptes so, dass jede Zeile auf eine eigene OID abgebildet wird. Die Daten für die einzelnen Messwerte werden aus der Datei „currentstatus.tmp“ abgefragt. Falls die Darstellung mittels ITM-Agenten durchgeführt werden soll, ist das Skript in der Lage, den Inhalt der Datei „currentstatus.tmp“ so auszugeben, dass der ITM-Agent sie verarbeiten kann. Außerdem kann das Skript einzelne Konfigurationsdateien auf ihre Fehlerfreiheit testen. Das Hinzufügen und Entfernen von Dateinamen zu bzw. aus der Hilfsdatei „processdata.conf“ erfolgt mit Hilfe von Skriptparametern.



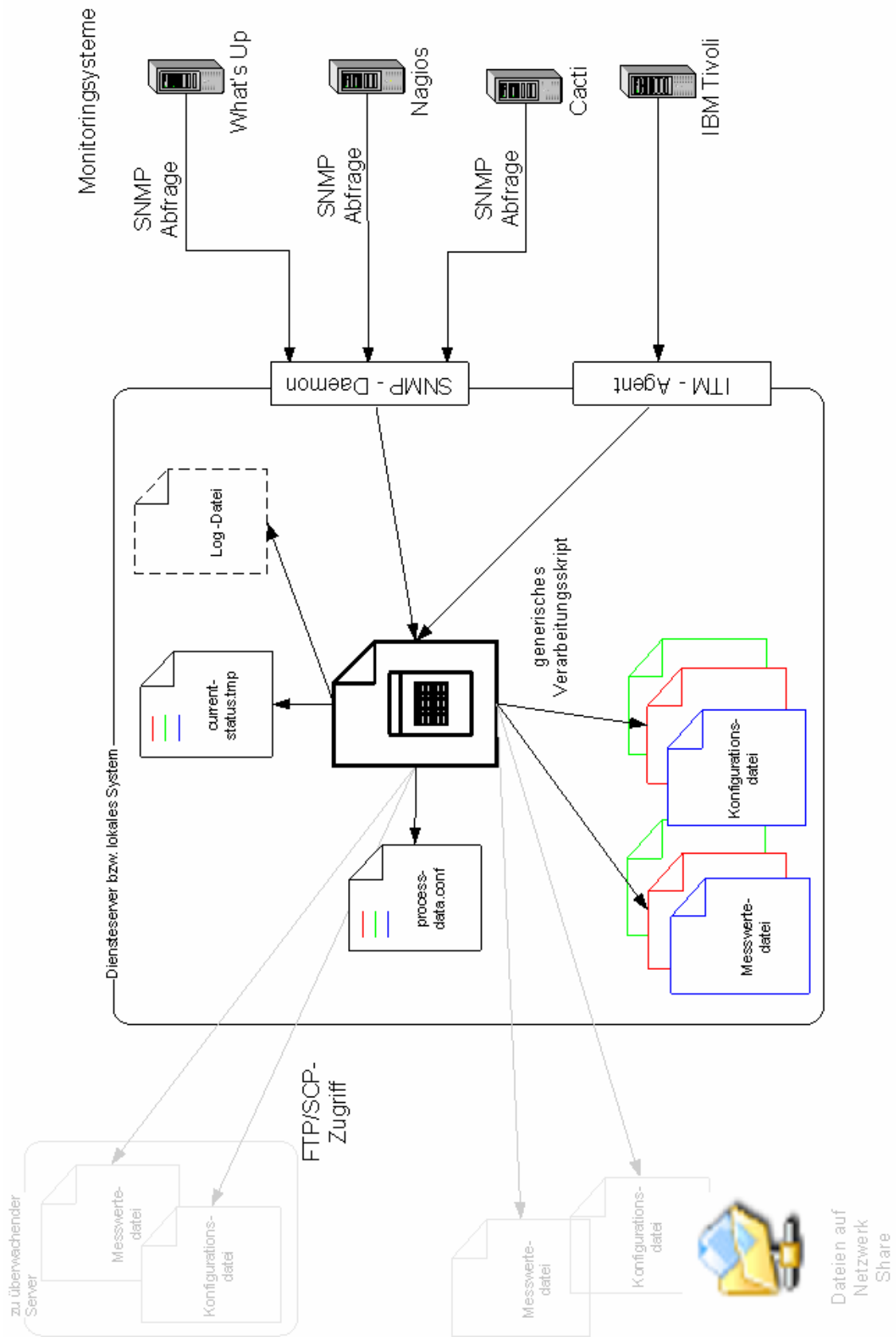


Abbildung 14: Skizze der Schnittstelle

### 4.3 Beschreibung der Funktionsweise der einzelnen Skriptparameter<sup>39</sup>

Die Auswertung der Parameter erfolgt durch das Perlmodul „Getopt-Long“. Die Parameter können in beliebiger Reihenfolge und anstelle von „-Parameter“ oder „--Parameter“ verwendet werden. Zwischen Groß- und Kleinschreibung der Parameter wird nicht unterschieden.

#### 4.3.1 Parameter -h

Mit diesem Parameter können alle verfügbaren Skriptparameter angezeigt werden. Bei seiner Gestaltung wurde sich an dem unter Linux üblichen Aufbau einer Hilfe orientiert.

Skriptaufruf:

```
ProcessScript.pl -h
```

#### 4.3.2 Parameter -l

Der Zweck dieses Parameters besteht darin, dem Verantwortlichen der Schnittstelle die Möglichkeit zu geben, um auftretende Fehler in der Form „Timestamp: Nachricht“ in einer Log-Datei abzuspeichern.

Der Timestamp hat folgendes Format:

```
YYYY.MM.DD HH:MM:SS UTC:
```

Seine Generierung erfolgt zu dem Zeitpunkt, an dem ein Fehler erkannt wurde. Als Basiszeit für die Ermittlung des Auftretens des Fehlers wird die Universal Time Coordinated (UTC) des zu überwachenden Systems zu Grunde gelegt. Der Teil „Nachricht“ enthält die Fehlerbeschreibung. Dieser Parameter kann nur mit den Parametern „-c“ und „-r“ verwendet werden.

---

<sup>39</sup> Das Skript befindet sich auf dem beigefügten Datenträger

#### 4.3.3 Parameter -a

Hiermit wird der Hilfsdatei „processdata.conf“ eine neue zu verarbeitende Konfigurationsdatei hinzugefügt. Nach dem „-a“ - Parameter wird der Dateipfad der hinzuzufügenden Konfigurationsdatei erwartet. Vor der Aufnahme erfolgt eine Prüfung, ob es sich bei der angegebenen Datei um eine Konfigurationsdatei handelt, die noch nicht in der Hilfsdatei vorhanden ist. Wenn beides zutrifft, wird diese Datei in die Hilfsdatei aufgenommen. Anderenfalls erhält der Anwender eine Fehlermeldung.

Skriptaufruf:

```
ProcessScript.pl -a dateiname.conf
```

#### 4.3.4 Parameter -d

Er entfernt die angegebene Datei aus der Datei „processdata.conf“. Hierbei muss es sich bei der zu entfernenden Datei um eine Konfigurationsdatei handeln. Bei ihrer Angabe ist auf die Schreibweise des Dateipfades zu achten. Der Aufruf des Skriptes erfolgt in der Form:

```
ProcessScript.pl -d dateiname.conf
```

Ist eine angegebene Datei nicht in der Datei „processdata.conf“ enthalten, erfolgt keine Fehlermeldung.

#### 4.3.5 Parameter -r

Wenn die gesamte Konfiguration der Schnittstelle überprüft und neu eingelesen werden soll, ist dies mit dem Parameter „-r“ durchzuführen.

Zunächst wird geprüft, ob eine zu verarbeitende Konfigurationsdatei an der angegebenen Stelle existiert. Ist dies nicht der Fall, erfolgt ein Eintrag in die Log-Datei und mit der nächsten Konfigurationsdatei, die in „processdata.conf“ enthalten ist, wird fortgefahren. Wenn sie vorhanden ist, wird sie geladen und weiteren Überprüfungen unterzogen. Zuerst wird kontrolliert, ob der allgemeine Teil der Konfigurationsdatei den Vorgaben entspricht. Im Einzelnen wird folgendes geprüft:

- Sind die aufgeführten Werte angegeben?
  - o ABOUT, AUTHOR, CONTACT,
  - o DATE OF CREATION, DATE OF LAST CHANGE, SEPARATOR,
  - o SEPARATOR DECIMAL VALUE, DATAFILE, BASE-OID
- Entsprechen die Werte (SEPARATOR, SEPARATOR DECIMAL VALUE, BASE-OID) den definierten Vorgaben?
- Existiert die Messwertdatei an der angegebenen Stelle?
- Unterscheiden sich die Werte für „SEPARATOR“ und „SEPARATOR DECIMAL VALUE“?
- Sind die Namen der Konfigurationsdatei und der dazugehörigen Messwertdatei identisch?
- Ist die BASE-OID der Konfigurationsdatei nur einmal vergeben?

Wenn bei einer dieser Prüfungen ein Fehler auftritt, werden alle Messwerte, die diese Konfigurationsdatei enthält, nicht in die Schnittstellenkonfiguration übernommen. Auftretende Fehler werden auf der Standardausgabe angezeigt. Optional können diese Fehler in eine Log-Datei geschrieben werden. Wurden keine Fehler gefunden, werden alle allgemeinen Informationen in die Schnittstellenkonfiguration übernommen, außer „SEPERATOR“ und „SEPERATOR DECIMAL VALUE“. Danach geschieht die Überprüfung der einzelnen Messwerte wie folgt:

- Sind die Werte (INDEX, NAME, TYPE, DISPLAYTYPE, DISPLAYART) der Messwertdefinition angegeben?
- Ist der Wert für „INDEX“ eine Zahl und größer als Null?
- Kommt der angegebene Index in der Konfigurationsdatei nur einmal vor?
- Entsprechen die Werte (TYPE, DISPLAYTYPE, DISPLAYART) den definierten Vorgaben?

Falls die Überprüfung einen Fehler ergeben hat, wird der entsprechende Messwert nicht in die Schnittstellenkonfiguration übernommen. Es erfolgt wiederum die Ausgabe des Fehlers auf der Standardausgabe sowie eine optionale Speicherung in der Log-Datei. Anderenfalls werden alle in Tabelle 15 aufgeführten Werte in einer Zeile, durch Semikolon getrennt, in die Hilfsdatei „currentstatus.tmp“ geschrieben.

#### 4.3.6 Parameter -u

Um das komplette Neueinlesen und Überprüfen der gesamten Schnittstellenkonfiguration zu vermeiden, ist es hiermit möglich, nur die geänderten Messwerte einzulesen. Dabei stehen zwei Varianten zur Verfügung: Neueinlesen der Messwerte aller Konfigurationsdateien oder Neueinlesen der Messwerte einer Konfigurationsdatei. Für den ersten Fall muss das Skript folgendermaßen aufgerufen werden:

```
ProcessScript.pl -u all
```

Nach dem Skriptaufruf werden alle in der Konfiguration enthaltenen Messwerte aktualisiert. Dies erfolgt durch einen Abgleich der einzelnen Messwerte zwischen der aktuellen Schnittstellenkonfiguration und den Messwertedateien. Abschließend wird die aktuelle Konfiguration der Schnittstelle gespeichert, wobei die alte Konfiguration überschrieben wird. Dieser Skriptaufruf kommt dann zum Einsatz, wenn alle Messwertedateien zu einem bestimmten Zeitpunkt oder in einem bestimmten Intervall aktualisiert werden sollen. Es ist nicht möglich, dass einzelne Messwerte oder Konfigurationsdateien von der Aktualisierung ausgeschlossen werden können.

Für den zweiten Fall der Verwendung lautet der Skriptaufruf:

```
ProcessScript.pl -u Name_der_Konfigurationsdatei.conf
```

Er wird angewendet, falls die Messwertedateien zu unterschiedlichen Zeitpunkten oder Intervallen erfasst werden sollen. Für den Fall, dass eine in der Konfigurationsdatei angegebene Messwertedatei nicht gefunden werden kann, erfolgt keine Aktualisierung der entsprechenden Messwerte.

Wenn dieser Parameter mit fehlerhaften Konfigurations- und Messwertedateien verwendet wird, kann dies zu Fehlern und Fehlfunktionen führen. Deshalb sollte er nur zum Einsatz kommen, wenn Konfigurationsdateien mittels „-c“ überprüft wurden oder ein Neueinlesen der Schnittstelle erfolgt ist.

#### 4.3.7 Parameter -o

Das Abrufen einer Zeile aus der aktuellen Schnittstellenkonfiguration erfolgt durch diesen Parameter. Der Aufruf des Skriptes lautet:

ProcessScript.pl -o OID

Für „OID“ ist die OID anzugeben, unter der der allgemeine Teil einer Konfigurationsdatei oder ein Messwert abrufbar ist. Anschließend wird die gesamte aktuelle Konfiguration nach dieser „OID“ durchsucht und der entsprechende Wert mit den dazugehörigen Informationen auf der Standardausgabe zurückgegeben.

Die Ausgabe der einzelnen Werte der Messwerte geschieht in der Reihenfolge, wie sie in Tabelle 15 angegeben wurde. Für allgemeine Konfigurationswerte einer Konfigurationsdatei erfolgt die Ausgabe, wie sie in Tabelle 14 dargestellt wurde. Wird eine angegebene OID nicht in der Konfiguration gefunden, erfolgt keine Ausgabe.

#### 4.3.8 Parameter -i

Damit wird die Anbindung der Schnittstelle an den ITM-Agenten realisiert. Die gesamte aktuelle Konfiguration der Schnittstelle wird so aufbereitet, dass sie vom ITM-Agenten verarbeitet werden kann. Dabei werden die Bestandteile, die der ITM-Agent erwartet, denen zugeordnet, die in der Datei „currentstatus.tmp“ enthalten sind.

| Bestandteil, den der ITM-Agent erwartet | Bestandteil der Datei „currentstatus.tmp“ |
|---|---|
| Date                                    | Timestamp                                 |
| ScriptName                              | Skriptname                                |
| Status                                  | Status                                    |
| Text1                                   | „NAME“                                    |
| Value1                                  | Name des Messwertes                       |
| Text2                                   | „VALUE“                                   |
| Value2                                  | Wert des Messwertes                       |

| Bestandteil, den der ITM-Agent erwartet | Bestandteil der Datei „currentstatus.tmp“  |
|---|--|
| Text3                                   | „ERROR-LEVEL“  |
| Value3                                  | Grenzwert des Messwertes   |
| Text4                                   | „WARNING-LEVEL“  |
| Value4                                  | Schwellwert der Messwertes   |
| Text5                                   | „DISPLAYART“   |
| Value5                                  | Darstellungsart des Messwertes   |
| StatusText                              | Beschreibung des Messwertes, OID des Messwertes, Datentyp des Messwertes, Darstellungstyp des Messwertes, Zieladresse(n) für Alarmierung beim Überschreiten des Grenzwertes, zu verwendender Benachrichtigungstext beim Über- bzw. Unterschreiten des Grenzwertes, Zieladresse(n) für Alarmierung beim Überschreiten des Schwellwertes, zu verwendender Benachrichtigungstext beim Über- bzw. Unterschreiten des Schwellwertes |

**Tabelle 16: Zuordnung ITM-Ausgabe**

Für jeden Messwert wird eine separate Ausgabezeile verwendet. Für den Fall, dass ein Wert (z.B. ERROR-LEVEL) nicht in der Konfigurationsdatei angegeben wurde, bleibt der entsprechende Bestandteil der ITM-Ausgabe frei.

#### 4.3.9 Parameter -c

Seine Funktionsweise entspricht im Wesentlichen der des Parameters „-r“, nur die Konfiguration der Schnittstelle wird nicht erneuert. Der Anwender hat damit die Möglichkeit, einzelne Konfigurationsdateien auf ihre Fehlerfreiheit zu überprüfen, ohne diese in die Schnittstellenkonfiguration übernehmen zu müssen. Alle auftretenden Fehler werden auf der Standardausgabe angezeigt. Der für diesen Parameter benötigte Skriptaufruf lautet:

```
ProcessScript.pl -c dateiname.conf
```

#### 4.3.10 Parameter -e

Mit seiner Hilfe wird die Datei „export.txt“ erstellt. Sie enthält für jeden definierten Messwert der Schnittstelle eine snmpd.conf -konforme Konfigurationszeile, die den SNMP-Daemon erweitert. Sie besitzt folgenden Aufbau:

```
extend .OID Messwertname perl Skriptpfad\ProcessScript.pl -o OID
```

Hierbei wird für „OID“ die OID verwendet, unter der der Messwert laut Konfigurationsdatei abgerufen werden kann. Für „Skriptpfad“ wird der in der Skriptvariablen „\$path“ hinterlegte Pfad des Skriptes verwendet. Da die OID, unter der die Elemente des jeweiligen Messwertes abrufbar sind, unter Umständen sehr unübersichtlich werden kann, wurde entschieden, dass eine Kommentarzeile unterhalb der Konfigurationszeile die OID der ersten Zeile der Ausgabe enthält. Der Skriptaufruf unter Windows lautet:

```
ProcessScript.pl -e w
```

Um eine Linux - kompatible snmpd.conf- konforme Konfigurationszeile zu erhalten, muss das Skript mit „-e u“ gestartet werden.

#### 4.3.11 Parameter -v

Er zeigt die aktuelle Version des zentralen Verarbeitungsskriptes an, lässt sich aber nicht mit anderen Parametern kombinieren. Um die Version des Skriptes anzuzeigen, muss es folgendermaßen aufgerufen werden:

```
ProcessScript.pl -v
```

#### 4.3.12 Hinweise zu den Parametern -r und -u

Bei der Verwendung dieser Parameter kann es vorkommen, dass eine Messwertdatei gleichzeitig durch verschiedene Skripte geschrieben und gelesen werden kann. Deshalb wird vor deren Aufruf geprüft, ob eine Datei mit dem gleichen Dateinamen



und der Erweiterung „.LOCK“ existiert. Wenn dies nicht der Fall ist, wird der Parameter ausgeführt, ansonsten nicht.

#### 4.4 Fehlermeldungen

| Fehlermeldung  | Ursache  |
|--|--|
| error: only *.conf-files are allowed                                   | die angegebene Datei ist keine Konfigurationsdatei   |
| error: could not find 'datei01.conf'                                   | die Datei kann nicht an der angegebenen Stelle gefunden werden   |
| error: invalid parameter!  | das Skript wurde mit einem fehlerhaften Parameter gestartet  |
| error: no configfiles found in 'processdata.conf'                      | die Datei „processdata.conf“ enthält keine zu verarbeitenden Konfigurationsdateien   |
| error: can not find configfile: 'datei01.conf'                         | die angegebene Konfigurationsdatei kann nicht gefunden werden  |
| error: abort reload measured value: 'Messwert01'                       | der angegebene Messwert kann nicht in die Konfiguration übernommen werden, da er fehlerhaft ist  |
| error: abort reload configfile: "datei01.conf"                         | die angegebene Konfigurationsdatei kann nicht in die Schnittstellenkonfiguration übernommen werden, da im allgemeinen Teil mindestens ein Fehler enthalten ist |
| error: invalid option 'x'  | tritt auf, wenn der Parameter -e mit einer ungültigen Option gestartet wurde   |
| error: configfile 'datei01.conf' exist                                 | die angegebene Konfigurationsdatei befindet sich bereits in der Schnittstellenkonfiguration  |
| error: file 'Werte01.data' in use-reload or update of configfile abort | tritt auf, wenn versucht wird, eine Messwertedatei zu lesen, die jedoch noch aufgrund eines Schreibvorganges gesperrt ist                                      |

**Tabelle 17: Fehlermeldungen**

Für die Beseitigung von Fehlern in Konfigurationsdateien müssen die Schritte unternommen werden, die in der Anlage H beschrieben sind.

## 5 Test der Schnittstelle

### 5.1 Test der einzelnen Skriptparameter des Verarbeitungsskriptes

Für die Durchführung dieser Tests wurde das Skript jeweils mit den entsprechenden Parametern unter Windows und Linux gestartet. Die Ergebnisse hierzu können in der Anlage I nachgelesen werden.

### 5.2 Test zur Verhinderung des gleichzeitigen Schreib- und Lesezugriffes

Dies wurde getestet, indem eine Datei mit dem Namen einer vorhandenen Messwertedatei und der Endung „.LOCK“ im gleichen Verzeichnis wie die Konfigurationsdatei erstellt wurde. Danach wurde das Skript mit dem Parameter -r gestartet. Der Test hat ergeben, dass die mittels Lock-Datei gesperrte Konfigurationsdatei nicht in die Konfiguration der Schnittstelle übernommen wurde. Es erschien eine Fehlermeldung. Für den Fall, dass nur die einzelnen Messwerte aktualisiert werden sollen und eine Messwertedatei gesperrt ist, wurden die entsprechenden Messwerte der Konfigurationsdatei nicht aktualisiert. Es erfolgte wiederum eine Fehlermeldung.

### 5.3 Nachweis der Schnittstellenfunktionalität

Dieser Nachweis wurde an einem Lasersoft-Printserver erbracht. Dabei ermittelte ein Shellskript die Anzahl der Druckjobs aus SAP, die Anzahl der Druckjobs aus dem Mainframe-System, die Anzahl der genutzten Lizenzen sowie den aktuellen Speicher-verbrauch des LPD-Prozesses in MB und schrieb dies in eine Messwertedatei. Die Ermittlung dieser Messwerte erfolgte alle fünf Minuten. Danach wurde das Verarbeitungsskript so konfiguriert, dass die Messwerte alle zehn Minuten in die Schnittstelle übernommen wurden. Die Darstellung der Werte erfolgte im Monitoring-system Cacti<sup>40</sup>.

---

<sup>40</sup> [www.cacti.net](http://www.cacti.net) (27.02.2009)

## 6 Zusammenfassung und Ausblick

Ziel der Diplomarbeit war die Entwicklung einer generischen Schnittstelle für das Monitoring von Anwendungen mittels SNMP und deren Umsetzung als Prototyp. Für die erfolgreiche Erfüllung dieser Aufgabe musste ich mich zuerst in die Grundlagen von SNMP und Monitoring einarbeiten. Im Anschluss daran erfolgte die Definition der Schnittstelle. Hierbei habe ich mich mit bereits vorhandenen Lösungsmöglichkeiten beschäftigt, die eine Ähnlichkeit mit der von mir umzusetzenden Schnittstelle aufwiesen. Das Ergebnis war, dass die zu entwickelnde Schnittstelle um eine Anbindung an den IBM Tivoli Agenten erweitert wurde. Danach wurden die Struktur der Konfigurations- und Messwertedatei sowie der Funktionsumfang des zu erstellenden Verarbeitungsskriptes definiert. Nachfolgend habe ich mich mit verschiedenen Möglichkeiten beschäftigt, wie ein SNMP-Daemon erweitert werden kann. Das Resultat hiervon ist die Festlegung der OID-Struktur für die einzelnen Werte. Als nächstes erfolgten die Umsetzung des Skriptes für Windows und Linux sowie der Funktionstest für die einzelnen Skriptparameter. Am Ende der Diplomarbeit wurde die Funktion der gesamten Schnittstelle an einem praktischen Beispiel durch einen Anwendungsverantwortlichen nachgewiesen.

Da der von mir entwickelte Lösungsansatz lediglich einen Prototyp darstellt, sehe ich folgende Erweiterungs- bzw. Verbesserungsmöglichkeiten:

- Anpassung der Schnittstelle an die Erfordernisse des Betriebs
- Entwicklung einer eigenen MIB für die abzubildenden Werte
- Anbindung der Schnittstelle an das Monitoringsystem Nagios mittels NRPE oder einem eigenen Plugin
- Verwendung des betriebssystemeigenen Logmechanismus für die Speicherung der Log-Einträge
- Verwendung von Semaphoren zum Verhindern des gleichzeitigen Schreib- und Lesezugriffes von Messwertedateien

## Anlagen

### Anlage A Wertebereiche und Beschreibung von SNMP-Datentypen<sup>41</sup>

| ASN.1-Datentyp | Beschreibung  | Wertebereich             | verfügbar in     |
|----------------|---|--------------------------|------------------|
| Integer        | 32- Bit-Ganzzahl  | $-2^{31}$ bis $2^{31}-1$ | SNMPv1           |
| Integer32      | 32- Bit-Ganzzahl  | $-2^{31}$ bis $2^{31}-1$ | SNMPv2<br>SNMPv3 |
| Unsigned32     | vorzeichenlose<br>32- Bit-Ganzzahl  | 0 bis $2^{32}-1$         | SNMPv2<br>SNMPv3 |
| Gauge          | 32-Bit-Ganzzahl, die<br>inkrementiert und dekrementiert<br>wird;<br>Wertebereich kann nicht über-<br>schritten werden | 0 bis $2^{32}-1$         | SNMPv1           |
| Gauge32        | 32-Bit-Ganzzahl, die<br>inkrementiert und dekrementiert<br>wird;<br>Wertebereich kann nicht über-<br>schritten werden | 0 bis $2^{32}-1$         | SNMPv2<br>SNMPv3 |
| Counter        | 32- Bit-Zähler;<br>falls der maximale Wertebereich<br>überschritten wird, erfolgt eine<br>Zurücksetzung auf Null      | 0 bis $2^{32}-1$         | SNMPv1           |
| Counter32      | 32- Bit-Zähler;<br>falls der maximale Wertebereich<br>überschritten wird, erfolgt eine<br>Zurücksetzung auf Null      | 0 bis $2^{32}-1$         | SNMPv2<br>SNMPv3 |
| Counter64      | 64- Bit-Zähler;<br>falls der maximale Wertebereich<br>überschritten wird, erfolgt eine<br>Zurücksetzung auf Null      | 0 bis $2^{64}-1$         | SNMPv2<br>SNMPv3 |

<sup>41</sup> vgl. Perkins, McGinnis S.38 Tabelle 2-3 und Mauro, Schmidt S.26 ff. Tabelle 2-1

| ASN.1-Datentyp    | Beschreibung  | Wertebereich       | verfügbar in               |
|-------------------|---|--------------------|----------------------------|
| Timeticks         | Messung der Zeit im 100stel-Sekunden- Bereich                   | 0 bis $2^{32} - 1$ | SNMPv1<br>SNMPv2<br>SNMPv3 |
| Octet String      | Kette von null oder mehreren Octets oder Bytes                  |                    | SNMPv1<br>SNMPv2<br>SNMPv3 |
| Object Identifier | OID   |                    | SNMPv1<br>SNMPv2<br>SNMPv3 |
| IpAddress         | IP Version 4-Adresse  |                    | SNMPv1<br>SNMPv2<br>SNMPv3 |
| Networkadress     | Vereinigung von Netzwerk-adressen                               |                    | SNMPv1                     |
| Opaque            | ermöglicht, ASN.1-Codierungen in einen „Octet String“ zu packen |                    | SNMPv1<br>SNMPv2<br>SNMPv3 |
| Bits              | markierte Bits  |                    | SNMPv2<br>SNMPv3           |
| Sequence          | Liste, die aus null oder mehreren ASN.1- Datentypen besteht     |                    | SNMPv1<br>SNMPv2<br>SNMPv3 |
| Sequence of       | besteht aus mehreren Elementen des Datentyps Sequence           |                    | SNMPv1<br>SNMPv2<br>SNMPv3 |

## Anlage B Befehlsübersicht des Skriptes

| Parameter           | Beschreibung   | Beispiel  |
|---------------------|--|---|
| -a Datei.conf       | fügt die angegebene Konfigurationsdatei in die Datei „processdata.conf“ ein              | ProcessScript.pl -a datei01.conf                            |
| -c Datei.conf -l    | überprüft, ob die in „Datei.conf“ angegebene Konfigurationsdatei den Vorgaben entspricht | ProcessScript.pl -c datei01.conf -l                         |
| -d Datei.conf       | entfernt die angegebene Konfigurationsdatei aus der Datei „processdata.conf“             | ProcessScript.pl -d datei01.conf                            |
| -e w   u            | erzeugt eine neue Datei „export.txt“   | ProcessScript.pl -s w<br>ProcessScript.pl -s u              |
| -h                  | zeigt die Hilfe an   | ProcessScript.pl -h   |
| -i                  | erzeugt eine Ausgabe, die vom ITM-Agenten verarbeitet werden kann                        | ProcessScript.pl -i   |
| -l                  | schaltet den Log-Mechanismus für die Parameter -r und -c ein                             | ProcessScript.pl -r -l                                      |
| -o OID              | gibt die Werte zurück, die unter dieser angegebenen OID gespeichert sind                 | ProcessScript.pl -o 1.2.3.6.8                               |
| -r [-l]             | führt komplettes Neueinlesen der Schnittstelle durch                                     | ProcessScript.pl -r -l                                      |
| -u all   Datei.conf | aktualisiert die Messwerte der angegebenen Konfigurationsdatei oder aller Konf.-dateien  | ProcessScript.pl -u datei01.conf<br>ProcessScript.pl -u all |
| -v                  | zeigt die aktuelle Version des Skriptes an   | ProcessScript.pl -v   |

## Anlage C Bestandteile des allgemeinen Teils einer Konfigurationsdatei

| Konfigurationswert       | Beschreibung <sup>42</sup>  | zulässige Werte   |
|--------------------------|---|---|
| ABOUT*                   | kurze Angaben zu den Messwerten, die mittels der Konfigurationsdatei überwacht werden;<br>ist auf eine Zeile begrenzt       | beliebiger Text   |
| AUTHOR*                  | gibt den Autor der Konfigurationsdatei an   | beliebiger Text   |
| CONTACT*                 | Angabe der E-Mail-Adresse oder Telefonnummer des Autors der Konfigurations-datei als Kontakt                                | entsprechende Kontaktinformationen des Autors der Konfigurationsdatei (z.B. E-Mail) |
| DATE OF CREATION*        | beinhaltet das Erstellungsdatum der Konfigurationsdatei   | Datum im Format:<br>YYYY.MM.DD  |
| DATE OF LAST CHANGE*     | beinhaltet das Datum der letzten Änderung der Konfigurationsdatei   | Datum im Format:<br>YYYY.MM.DD  |
| SEPERATOR*               | legt das Trennzeichen in der Messwertedatei und für die einzelnen Konfigura-tionswerte des messwert-spezifischen Teils fest | nur Kommas,<br>Doppelpunkte und<br>Semikolons                                       |
| SEPERATOR DECIMAL VALUE* | legt Trennzeichen der Dezimalwerte fest   | nur Punkte und Kommas   |
| DATAFILE*                | gibt die zugehörige Messwertedatei an   | Pfad zur Messwertedatei   |

<sup>42</sup> detaillierte Beschreibung der einzelnen Werte sind im Punkt 3.2.3 nachzulesen

| Konfigurationswert | Beschreibung <sup>42</sup>   | zulässige Werte                         |
|--------------------|--|---|
| BASE-OID*          | gibt die OID an, unter der die einzelnen Messwerte einer Konfigurationsdatei abrufbar sind | nur OID's in der Form<br>1.3.5.8.6.11.4 |
| EMAIL1             | gibt die E-Mail-Adresse an,  | jeweils eine gültige E-Mail-Adresse     |
| EMAIL2             | an die eine Alarmierung  |   |
| EMAIL3             | bzw. Warnung versendet werden soll   |   |
| SMS1               | gibt die Telefonnummer   | jeweils eine gültige Telefonnummer      |
| SMS2               | an, an die eine  |   |
| SMS3               | Alarmierung bzw. Warnung versendet werden soll   |   |
| TEXT1              | gibt den   | jeweils ein beliebiger Text             |
| TEXT2              | Benachrichtigungstext an,  |   |
| TEXT3              | der bei der Alarmierung bzw. Warnung in einer SMS oder E-Mail stehen soll                  |   |

Die mit \* gekennzeichneten Konfigurationswerte müssen angegeben werden.



Anlage D Bestandteile des messwertspezifischen Teils einer Konfigurationsdatei

| Konfigurationswert | Beschreibung <sup>43</sup>  | zulässige Werte  |
|--------------------|---|--|
| INDEX*             | wird für die Bildung der<br>OID des Messwertes<br>benötigt                                  | ganze Zahl, die größer als<br>Null ist   |
| NAME*              | gibt den Namen des<br>konfigurierten Messwertes<br>an                                       | beliebiger, möglichst<br>kurzer Name   |
| DESCRIPTION        | beschreibt den zu<br>konfigurierenden Messwert  | beliebiger Text  |
| TYPE*              | gibt den Datentyp des<br>Messwertes an  | INTEGER oder FLOAT   |
| DISPLAYTYPE*       | legt fest, wie der Messwert<br>von einem Monitoring-<br>system interpretiert werden<br>soll | „ABSOLUT“ oder<br>„COUNTER“ oder<br>„COUNTER32“ oder<br>„COUNTER64“ oder<br>„GAUGE“ oder<br>„GAUGE32“ oder<br>„GAUGE64“ oder<br>„INTEGER“ oder<br>„INTEGER32“ oder<br>„UNSIGNED32“ |
| DISPLAYART*        | legt fest, wie der Messwert<br>durch das Monitoring-<br>system dargestellt werden<br>soll   | „GRAPH“  |
| ERROR-LEVEL        | definiert den Grenzwert,<br>bei dem eine Alarmierung<br>ausgelöst werden soll               | Zahl, die den Grenzwert<br>angibt  |

<sup>43</sup> detaillierte Beschreibung der einzelnen Werte sind im Punkt 3.2.3 nachzulesen

| Konfigurationswert | Beschreibung <sup>43</sup>  | zulässige Werte  |
|--------------------|---|--|
| ALARMERROR-LEVEL   | beschreibt, welche E-Mail-Adresse oder SMS-Nummer im Alarmierungsfall verwendet werden soll | „EMAIL1“, „EMAIL2“, „EMAIL3“, „SMS1“, „SMS2“, „SMS3“ ;<br>mehrere Werte sind durch ein Leerzeichen abzutrennen |
| TEXTERROR-LEVEL    | beschreibt den Text, der bei einer Benachrichtigung verwendet werden soll                   | „TEXT1“, „TEXT2“, „TEXT3“ ;<br>mehrere Werte sind durch ein Leerzeichen abzutrennen                            |
| WARNING-LEVEL      | definiert den Schwellwert, bei dem eine Warnung ausgelöst werden soll                       | Zahl, die den Schwellwert angibt   |
| ALARMWARNING-LEVEL | beschreibt, welche E-Mail-Adresse oder SMS-Nummer im Warnungsfall verwendet werden soll     | „EMAIL1“, „EMAIL2“, „EMAIL3“, „SMS1“, „SMS2“, „SMS3“ ;<br>mehrere Werte sind durch ein Leerzeichen abzutrennen |
| TEXTWARNING-LEVEL  | beschreibt den Text, der bei einer Benachrichtigung verwendet werden soll                   | „TEXT1“, „TEXT2“, „TEXT3“ ;<br>mehrere Werte sind durch ein Leerzeichen abzutrennen                            |

Die mit \* gekennzeichneten Konfigurationswerte müssen angegeben werden.

## Anlage E Zulässige Zeichen in der Schnittstelle

Die nachfolgende ASCII-Tabelle zeigt eine Übersicht über alle zulässigen Zeichen der Schnittstelle.

| ASCII-<br>Nummer | ASCII-<br>Zeichen | ASCII-<br>Nummer | ASCII-<br>Zeichen | ASCII-<br>Nummer | ASCII-<br>Zeichen | ASCII-<br>Nummer | ASCII-<br>Zeichen |
|------------------|-------------------|------------------|-------------------|------------------|-------------------|------------------|-------------------|
| 1                |                   | 33               | !                 | 65               | A                 | 97               | a                 |
| 2                |                   | 34               | "                 | 66               | B                 | 98               | b                 |
| 3                |                   | 35               | #                 | 67               | C                 | 99               | c                 |
| 4                |                   | 36               | \$                | 68               | D                 | 100              | d                 |
| 5                |                   | 37               | %                 | 69               | E                 | 101              | e                 |
| 6                |                   | 38               | &                 | 70               | F                 | 102              | f                 |
| 7                |                   | 39               | '                 | 71               | G                 | 103              | g                 |
| 8                |                   | 40               | (                 | 72               | H                 | 104              | h                 |
| 9                |                   | 41               | )                 | 73               | I                 | 105              | i                 |
| 10               |                   | 42               | *                 | 74               | J                 | 106              | j                 |
| 11               |                   | 43               | +                 | 75               | K                 | 107              | k                 |
| 12               |                   | 44               | ,                 | 76               | L                 | 108              | l                 |
| 13               |                   | 45               | -                 | 77               | M                 | 109              | m                 |
| 14               |                   | 46               | .                 | 78               | N                 | 110              | n                 |
| 15               |                   | 47               | /                 | 79               | O                 | 111              | o                 |
| 16               |                   | 48               | 0                 | 80               | P                 | 112              | p                 |
| 17               |                   | 49               | 1                 | 81               | Q                 | 113              | q                 |
| 18               |                   | 50               | 2                 | 82               | R                 | 114              | r                 |
| 19               |                   | 51               | 3                 | 83               | S                 | 115              | s                 |
| 20               |                   | 52               | 4                 | 84               | T                 | 116              | t                 |
| 21               |                   | 53               | 5                 | 85               | U                 | 117              | u                 |
| 22               |                   | 54               | 6                 | 86               | V                 | 118              | v                 |
| 23               |                   | 55               | 7                 | 87               | W                 | 119              | w                 |
| 24               |                   | 56               | 8                 | 88               | X                 | 120              | x                 |
| 25               |                   | 57               | 9                 | 89               | Y                 | 121              | y                 |
| 26               |                   | 58               | :                 | 90               | Z                 | 122              | z                 |
| 27               |                   | 59               | ;                 | 91               | [                 | 123              | {                 |
| 28               |                   | 60               | <                 | 92               | \                 | 124              |                   |
| 29               |                   | 61               | =                 | 93               | ]                 | 125              | }                 |
| 30               |                   | 62               | >                 | 94               | ^                 | 126              | ~                 |
| 31               |                   | 63               | ?                 | 95               | _                 | 127              | •                 |
| 32               |                   | 64               | @                 | 96               | `                 | 128              | €                 |

## Anlage F Anleitung zur Schnittstellenerstkonfiguration

1. Prüfen, ob die Voraussetzungen und benötigten Konfigurations- und Messwertedateien für die Verwendung der Schnittstelle erfüllt bzw. vorhanden sind.
2. Kopieren des Skriptes in einen beliebigen Ordner auf dem System  
Wenn dieser Kopiervorgang mit Hilfe eines FTP-Clients durchgeführt wird, muss als Transfermodus der ASCII-Modus verwendet werden, um Veränderungen an den Dateien zu vermeiden.
3. Falls das Skript auf einem Linux-System zu Einsatz kommt, muss gegebenenfalls das Ausführungsrecht für das Skript gesetzt werden.
4. Anlegen der Datei „processdata.conf“, falls diese noch nicht vorhanden ist. Sie muss sich im gleichen Verzeichnis wie das Skript befinden.
5. Anpassung der Variablen „\$path“ im Skript  
Hierbei muss der Pfad angegeben werden, in welchem sich das Skript befindet. Bei Pfadangaben im Windowsumfeld ist anstelle des einzelnen Backslashes „\“ der doppelte Backslash „\\“ zu verwenden.
6. Hinzufügen aller zu verarbeitenden Konfigurationsdateien zur Datei „processdata.conf“ mit Hilfe des Parameters -a.
7. Neueinlesen der Schnittstelle mit dem Parameter -r -l
8. Überprüfen der aktuellen Schnittstellenkonfiguration mit Hilfe der Log- Datei
9. Anlegen von Tasks (z.B. mittels cron), die die Schnittstelle in regelmäßigen Abständen aktualisieren
10. Konfigurieren des SNMP-Daemons
- 10a: Konfiguration des SNMP-Daemons unter Windows  
Die Konfiguration erfolgt mit Hilfe der Datei „snmpd.conf“, die sich im Verzeichnis „C:\usr\etc\snmp“ befindet. Um den SNMP-Daemon zu konfigurieren, muss die Zeile „rocommunity public“ erstellt werden.

## 10b: Konfiguration des SNMP-Daemons unter Debian Linux 5.0 (Lenny)

Zuerst muss in der Datei `/etc/default/snmpd` die Zeile:

```
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p  
/var/run/snmpd.pid 127.0.0.1'
```

in:

```
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p  
/var/run/snmpd.pid'
```

geändert werden. Diese Änderung bewirkt, dass der SNMP-Daemon nicht mehr nur auf das interne Loopback-Interface zugreift. Anschließend müssen folgende Zeilen der Datei `etc/snmp/snmpd.conf` geändert werden:

```
#      sec.name  source      community  
com2sec  paranoid  default     public  
#com2sec  readonly  default     public  
#com2sec  readwrite  default     private
```

in:

```
#      sec.name  source      community  
#com2sec  paranoid  default     public  
com2sec  readonly  default     public  
#com2sec  readwrite  default     private
```

Der "paranoia"-Modus, den der SNMP-Daemon nach der Standardinstallation besaß, wird in den „readonly“-Modus geändert.

### 11. Neustart des SNMP-Daemons

### 12. Überprüfen der Funktionalität des SNMP-Daemons

### 13. Hinzufügen der extend-Konfigurationszeilen für die einzelnen Messwerte in die Datei `snmpd.conf`.

Es ist möglich, sich die Konfigurationszeilen mit Hilfe des Parameters `-e` erstellen zu lassen, die dann noch in die Datei `snmpd.conf` kopiert werden müssen.

### 14. Neustart des SNMP-Daemons

### 15. Überprüfen der SNMP-Konfiguration mittels `snmpget` bzw. `snmpwalk`

### 16. Dokumentation der verwendeten OID's

## Anlage G Anleitung für den Anwendungsverantwortlichen

1. Beantragung einer OID für die zu überwachende Anwendung beim Schnittstellenverantwortlichen
2. Erstellung des Skriptes, das ihm die Messwerte liefert und sie in der Messwertedatei speichert
3. Erstellung der Konfigurationsdatei
4. Mitteilung an den Schnittstellenverantwortlichen, wo sich die Konfigurations- und Messwertedatei befindet, so dass er diese in die Schnittstellenkonfiguration aufnehmen kann

## Anlage H Anleitung zur Fehlerbeseitigung in Konfigurationsdateien

1. Öffnen der Log-Datei „ProcessScriptLog.log“, welche die Fehler enthält  
Sie befindet sich im gleichen Verzeichnis wie das Skript
2. Beseitigung der fehlerhaften Definition(en)
3. Erneutes Neueinlesen der Konfiguration
4. Wenn weitere Fehler gefunden werden, ist mit Schritt eins wieder zu beginnen

## Anlage I Ergebnisse des Skripttestes

| Test   | Ergebnis(se)   |
|--|--|
| Starten des Skriptes mit dem Parameter -a datei.conf   | Hinzufügen einer neuen Zeile in der Datei „processdata.conf“ mit dem Dateinamen datei.conf   |
| Starten des Skriptes mit dem Parameter -c datei.conf -l  | Durchführung der Überprüfung der angegebenen Konfigurationsdatei;<br>Speichern eventuell auftretender Fehler in der Log-Datei und Anzeigen auf der Standardausgabe   |
| Starten des Skriptes mit dem Parameter -d datei.conf   | Entfernen der angegebenen Datei „datei.conf“ aus „processdata.conf“  |
| Starten des Skriptes mit dem Parameter -h  | Anzeigen aller verfügbaren Parameter mit kurzer Erklärung  |
| Starten des Skriptes mit dem Parameter -i  | zeilenweise Ausgabe aller Messwerte mit den dazugehörigen Informationen in der Form, wie sie der ITM-Agent benötigt  |
| Starten des Skriptes mit dem Parameter -o OID  | zeilenweise Rückgabe aller verfügbaren Informationen des Messwertes, unter dem er mittels dieser OID abrufbar ist  |
| Starten des Skriptes mit dem Parameter -o  | keine Rückgabe von Informationen   |
| Starten des Skriptes mit dem Parameter -r [-l]<br>(Konfigurationsdateien sind fehlerfrei)                  | (Neu)erstellen der Datei „currentstaus.tmp“  |
| Starten des Skriptes mit dem Parameter -r [-l]<br>(Konfigurationsdateien enthalten eine BASE-OID mehrfach) | entsprechende Konfigurationsdatei, welche die mehrfach vergebene BASE-OID enthält, wird nicht in die Schnittstellenkonfiguration übernommen;<br>Ausgabe einer Fehlermeldung;<br>gegebenenfalls erfolgt ein Log-Eintrag |

| Test  | Ergebnis(se)   |
|---|--|
| Starten des Skriptes mit dem Parameter -r [-l]<br>(Konfigurationsdatei enthält fehlerhafte Angaben im allgemeinen Teil) | alle enthaltenen Messwerte der entsprechenden Konfigurationsdatei werden nicht in die Schnittstellenkonfiguration übernommen;<br>Ausgabe einer Fehlermeldung;<br>gegebenenfalls erfolgt ein Log-Eintrag                                    |
| Starten des Skriptes mit dem Parameter -r [-l]<br>(Konfigurationsdatei enthält fehlerhaften Messwert)                   | betreffender Messwert wird nicht in die Schnittstellenkonfiguration übernommen;<br>Ausgabe einer Fehlermeldung;<br>gegebenenfalls erfolgt ein Log-Eintrag  |
| Starten des Skriptes mit dem Parameter -e   | Erstellen der Datei „export.txt“ im Verzeichnis, in dem sich das Skript befindet;<br>„export.txt“ enthält jeweils eine Konfigurationszeile für den SNMP-Daemon und eine Kommentarzeile für jeden in der Schnittstelle definierten Messwert |
| Starten des Skriptes mit dem Parameter -u all   | Aktualisierung der Messwerte für alle in der Konfiguration enthaltenen Konfigurationsdateien   |
| Starten des Skriptes mit dem Parameter -u datei.conf  | Aktualisierung der Messwerte der angegebenen Konfigurationsdatei   |
| Starten des Skriptes mit dem Parameter -v   | Anzeige der aktuellen Skriptversion  |



## Literaturverzeichnis

### Mauro, Schmidt

Douglas R. Mauro; Kevin J. Schmidt: Essential SNMP, Second Edition O'Reilly 2005  
-442 S.

### Perkins, McGinnis

David Perkins; Evan McGinnis: Understanding SNMP MIB's, Prentice Hall PTR 1997  
- 510 S.

### Michler

Markus Michler; Netzwerkmanagement mit SNMP 2008-24 S.  
Berlin, Humboldt-Universität, Institut für Informatik, Lehrstuhl für  
Rechnerkommunikation Begleitmaterial zum Halbkurs „Spezielle Techniken der  
Rechnerkommunikation“, SS 08

### Eckel

Eckel, Peter: Stille Helfer unter Beschuss SNMP: eine kleine Bestandsaufnahme  
In: c't-Magazin Hannover: Heinz Heise Verlag-05/2002, S.46  
oder <http://www.heise.de/ct/02/05/046/> (17.12.2008)

### NRPE Documentation

Nagios NRPE Documentation  
Copyright (c) 1999-2007 Ethan Galstad-18 S.

### NetSnmpp (18.12.2008)

<http://www.net-snmp.org/docs/man/snmpd.conf.html>  
[net-snmp-users@lists.sourceforge.net](mailto:net-snmp-users@lists.sourceforge.net)

## Erklärung zur selbstständigen Anfertigung der Arbeit

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

München, den 29.04.2009

---