



IVD

INFINISTORE VirtualDisk

LogAnalyzer

Administrator Guide

GRAU
DATA STORAGE

Version 2.3

INFINISTORE VirtualDisk

Impressum

Dieses Handbuch wurde mit äußerster Sorgfalt erstellt. Dennoch können inhaltliche oder formale Fehler nicht ausgeschlossen werden.

Geschützte Warennamen sind in diesem Handbuch nicht als solche gekennzeichnet. Aus dem Fehlen einer derartigen Kennzeichnung kann also nicht geschlossen werden, daß es sich um einen freien Warennamen handelt.

Alle Rechte, auch für den Fall von Schutzanmeldungen, vorbehalten. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht, liegt beim Herausgeber.

Änderungen vorbehalten.

Dieses Handbuch bezieht sich auf die Version 2.3.1 (Build: 4)

Herausgeber: GRAU Data Storage AG,
Marie-Curie-Straße 19, 73529 Schwäbisch Gmünd
Deutschland

Redaktion: GRAU Data Storage AG,
Am Weinberg 20, 86732 Oettingen i. Bay.
Deutschland

Copyright © 2005 by GRAU Data Storage AG,
Marie-Curie-Straße 19, 73529 Schwäbisch Gmünd
Deutschland

6. überarbeitete Auflage, im Dezember 2005

INFINISTORE VirtualDisk

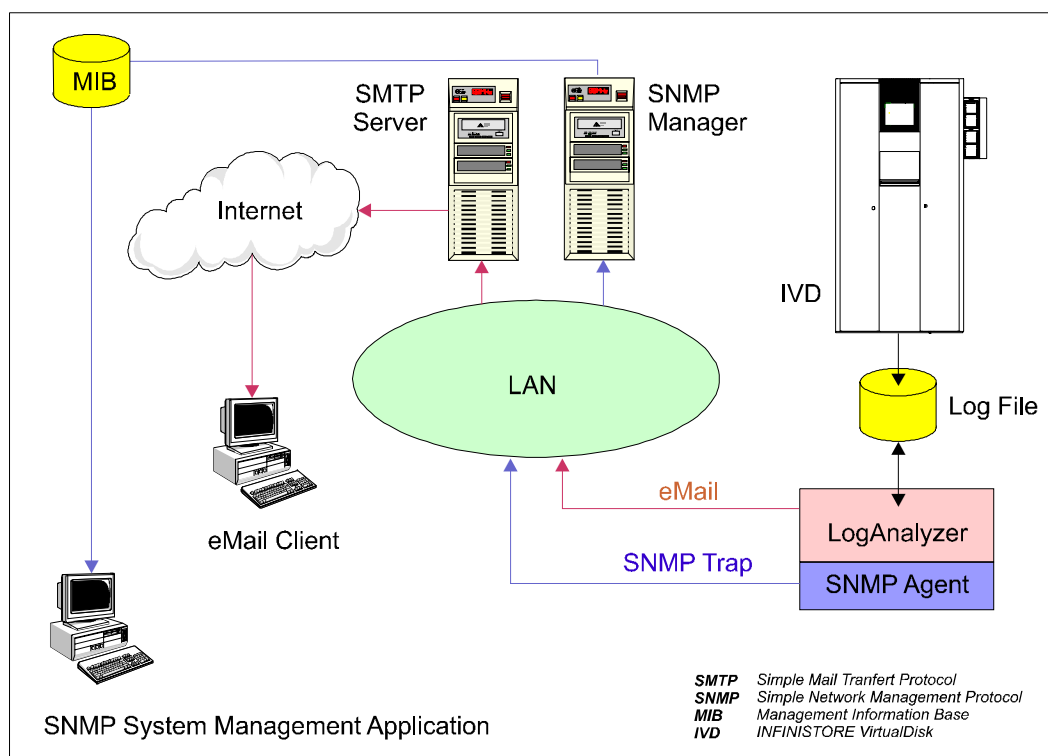
Inhalt

Impressum	3
Inhalt	5
1. Allgemein	7
2. Installation	8
2.1 Installation auf einem Windows-System	8
2.2 Installation auf einem Linux-System	9
3. Konfiguration	11
3.1 Die <i>LogAnalyzer.config</i> Konfigurations-Datei	12
3.2 Die <i>LogAnalyzerRules.py</i> Konfigurations-Datei	16
3.2.1 Beschreiben eines Filters	17
3.2.2 Aufruf von Aktionen	20
3.2.3 Beispiel einer <i>LogAnalyzerRules.py</i> Konfigurations-Datei	22
3.3 MIB-Dateien	23
4. Einsatz auf einem Linux-System	25
4.1 Die <i>LogAnalyzer_messages.config</i> Konfigurations-Datei	26
4.2 Die <i>LogAnalyzerRules_messages.py</i> Konfigurations-Datei	27
5. GRAU Service Center	29
6. Software-Lizenzvertrag	31

INFINISTORE VirtualDisk

1. Allgemein

Der *IVD-LogAnalyzer* dient der permanenten Auswertung einer frei definierbaren Log-Datei. In der Regel wird er für den Versand von eMails oder SNMP-Traps basierend auf Fehler- und Warnmeldungen, die aus der Log-Datei entnommen werden, eingesetzt.

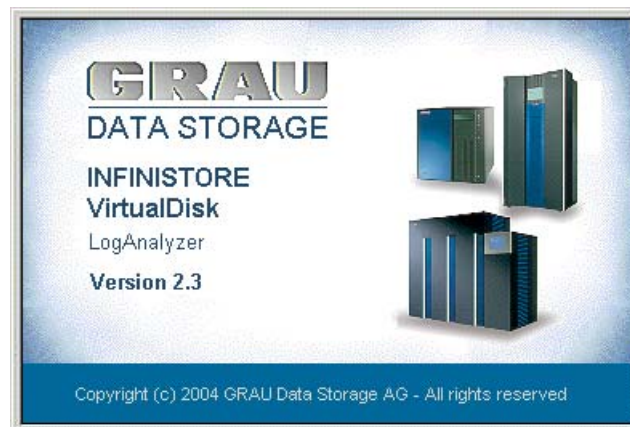


Die Meldungen werden hierbei in einem Paket gesammelt und in zeitlichen Intervallen an zuvor festgelegte Zieladressen verschickt. Abhängig vom Auftreten frei definierbarer Log-Meldungen werden eMails und/oder SNMP-Traps (Simple Network Management Protocol) generiert.

2. Installation

2.1 Installation auf einem Windows-System

Das Setup-Programm des *IVD-LogAnalyzer* befindet sich auf der Installations-CD im Verzeichnis *\Win32\Tools\LogAnalyzer*. Starten Sie von dort aus das Programm *setup.exe*, um den *IVD-LogAnalyzer* zu installieren. Folgen Sie dabei den Anweisungen auf dem Bildschirm bis die Installation abgeschlossen ist.



Nach vollständiger Anpassung der Konfigurationsdateien *LogAnalyzerRules.py* und *LogAnalyzer.config* (siehe Seite 11) ist der *IVD-LogAnalyzer* einsatzbereit.

Deinstallation

- ☒ Stoppen Sie den Service *GRAU IVD-LogAnalyzer* falls der *IVD-LogAnalyzer* als Service gestartet wurde.
- ☒ Entfernen Sie das Produkt *GRAU IVD-LogAnalyzer* über den Software-Manager von Microsoft Windows.

2.2 Installation auf einem Linux-System

Für die Installation des *IVD-LogAnalyzers* müssen die unten aufgeführten Packages installiert sein. Eine entsprechende Überprüfung wird durch *rpm* während der Installation des *IVD-LogAnalyzers* durchgeführt.

☒ compat version 2002.8.15

Die rpm-Packages für die Installation des *IVD-LogAnalyzers* auf einem Linux-System finden Sie in folgenden Verzeichnissen der Installations-CD:

SuSE: */sles8/ia32/tools/LogAnalyzer*

RedHat: */rhel3/ia32/tools/LogAnalyzer*

Aufruf:

```
rpm -i LogAnalyzer-2.3-X.i386.rpm
```

x = Revisionsnummer des *IVD-LogAnalyzers*

Beispiel:

```
rpm -i /usr/src/packages/RPMS/i386/LogAnalyzer-2.3-1.i386.rpm
```

Prüfen der Installation:

```
rpm -qa | grep LogAnalyzer
```

Eine positiv verlaufende Prüfung zeigt das installierte rpm-Package des *IVD-LogAnalyzers* an (siehe Beispiel).

```
LogAnalyzer-2.3-1
```

Durch die Installation des *IVD-LogAnalyzers* wurden folgende Dateien angelegt:

- /opt/ivd/sbin/loganalyzer	- /usr/local/lib/_random.so
- /opt/ivd/sbin/loganalyzer_messages	- /usr/local/lib/array.so
	- /usr/local/lib/cStringIO.so
- /opt/ivd/doc/LogAnalyzer23DE.pdf	- /usr/local/lib/fcntl.so
- /opt/ivd/doc/LogAnalyzer23UK.pdf	- /usr/local/lib/math.so
	- /usr/local/lib/pwd.so
- /etc/opt/ivd/loganalyzerrules.py	- /usr/local/lib/struct.so
- /etc/opt/ivd/loganalyzer.config	- /usr/local/lib/time.so
- /etc/opt/ivd/loganalyzerrules_messages.py	- /usr/local/lib/_socket.so
- /etc/opt/ivd/loganalyzer_messages.config	- /usr/local/lib/binascii.so
- /etc/opt/ivd/gds.mib	- /usr/local/lib/collections.so
- /etc/opt/ivd/gds_ivd.mib	- /usr/local/lib/itertools.so
	- /usr/local/lib/md5.so
- /etc/init.d/loganalyzer	- /usr/local/lib/strop.so
- /etc/init.d/loganalyzer_messages	- /usr/local/lib/termios.so
- /usr/local/lib/operator.so	- /usr/local/lib/zlib.so

Nach vollständiger Anpassung der Konfigurationsdateien *loganalyzerrules.py* und *loganalyzer.config* (siehe Seite 11) ist der *IVD-LogAnalyzer* einsatzbereit.

Die Dateien *loganalyzer_messages*, *loganalyzerrules_messages.py* und *loganalyzer_messages.config* dienen der Überwachung einer zweiten Log-Datei (bsplw. der Linux Log-Datei */var/log/messages*). Der Aufbau dieser Dateien ist identisch dem Aufbau der Dateien *loganalyzer*, *loganalyzerrules.py* bzw. *loganalyzer.config* (siehe Seite 11). Weitere Informationen finden Sie ab Seite 25.

Deinstallation

Verwenden Sie folgendes rpm-Kommando, um den *IVD-LogAnalyzer* zu entfernen.

```
rpm -e LogAnalyzer-2.3-1
```

Start als Prozess

Für den Fall, daß der *IVD-LogAnalyzer* als Prozess gestartet werden soll, muß zuvor die Umgebungsvariable *LD_LIBRARY_PATH* wie unten beschrieben gesetzt werden.

```
export LD_LIBRARY_PATH=/usr/local/lib
```

Start als Deamon

```
usage: /etc/init.d/loganalyzer {start|stop|status|restart}
usage: /etc/init.d/loganalyzer_messages {start|stop|status|restart}
```

3. Konfiguration

Für den *IVD-LogAnalyzer* ist vor seiner Nutzung eine individuelle Anpassung in folgenden Konfigurationsdateien vorzunehmen:

☒ **LogAnalyzer.config**

Dient der Konfiguration des Mail- bzw. SNMP-Versandes

☒ **LogAnalyzerRules.py**

Dient der Konfiguration der Log-Analyse

Unabhängig von der Plattform (Windows oder Linux) kann der *IVD-LogAnalyzer* als Prozess von der Kommando-Zeile gestartet werden. Als Parameter muß hierbei der Dateiname der Konfigurationsdatei angegeben werden.

```
LogAnalyzer LogAnalyzer.config
```

Dies ist hilfreich, wenn der *IVD-LogAnalyzer* situationsbedingt mit unterschiedlichen Konfigurations-Dateien gestartet werden soll oder wenn mehrere Log-Dateien überwacht werden müssen (siehe auch Seite 25). Wird der *IVD-LogAnalyzer* ohne Parameter als Prozess aufgerufen, erscheint folgender Hilfe-Text:

```
LogAnalyzer
Version 2.3.1.0004 (C) Copyright GRAU DataStorage AG, 2001-2005.
All rights reserved.
usage: LogAnalyzer [-h|--help|-d|--debug|-s|smtp_debug|-v] <ConfigFile>
-h | --help           shows the usage information
-d | --debug          enables the debug mode
-s | --smtp_debug     enables smtp debug mode
-v                   shows the version information
```

Zudem wird der *IVD-LogAnalyzer* auf einem Windows-System als Service eingerichtet, welcher, nach Abschluß der Konfiguration, über den Service Manager gestartet werden kann. Um den *IVD-LogAnalyzer* nach jedem Systemstart automatisch zu aktivieren, ändern Sie die Startart des *IVD-LogAnalyzer* im Service-Manager auf *automatic*.

Bei Linux-Systemen der *IVD-LogAnalyzer* als Daemon gestartet. Informationen um einen Daemon starten bzw. stoppen zu können finden Sie in Ihrer Linux-Dokumentation.



Wird der *IVD-LogAnalyzer* mit der Option *-d* oder *--debug* gestartet, werden alle gestarteten Aktionen mit den für die Ausführung relevanten Rules am Bildschirm angezeigt.

Weitere Informationen über Rules finden Sie ab Seite 16.

3.1 Die LogAnalyzer.config Konfigurations-Datei

Das Installationsprogramm legt eine Standard-Konfigurationsdatei an. Die Standardwerte sind entsprechend den gegebenen Anforderungen anzupassen (siehe Beispiel).

```
# =====
# Template for the configurationfile of the LogAnalyzer Version 2.3.1
# Created: 25.10.05
# Author: GDS Development
# =====

# === USED TERMS ===
# AnalysisFile: The file which analyzes the LogAnalyzer
# Package:      The file, produced by the LogAnalyzer.
#               This is the file which will be attached to the email.

[LogAnalyzer]
# =====
# This section describes internal settings of the LogAnalyzer
# =====

# --- SleepTime ---
# After "SleepTime" the LogAnalyzer checks the AnalysisFile
# again for new Messages
SleepTime = 0.1

# --- Debug ---
# Debug = 0      ==> No Debuginformation is shown (Default)
# Debug = 1      ==> Shows Debuginformation ( Rules, Actions, etc.)
Debug = 0

# --- ProcessFileFromBeginning ---
# ProcessFileFromBeginning = 0 ==> Processing the AnalysisFile at the end
#                               ( Default, Waiting for new messages )
# ProcessFileFromBeginning = 1 ==> Processing the AnalysisFile from the
#                               beginning. ( For debugging purposes )
ProcessFileFromBeginning = 0

# --- RulesFile ---
# The rules, which the LogAnalyzer will load on startup
# NOTE: Must be fully qualified pathname !
RulesFile = C:\Program Files\GDS\LogAnalyzer\LogAnalyzerRules.py

# --- TmpDir ---
# Directory where temporary files are stored.
# If not set, the System-Temporary-Directory will be used.
# NOTE: Must be fully qualified pathname !
TmpDir = C:\Program Files\GDS\LogAnalyzer\LogTmp

# --- ActivityLog ---
# In this file the LogAnalyzer logs its activities.
# NOTE: Must be fully qualified pathname !
ActivityLog = C:\Program Files\GDS\LogAnalyzer\Analyzer.log

# --- AnalysisFile ---
# The file which is analyzed by the LogAnalyzer
AnalysisFile = C:\Program Files\GDS\LogAnalyzer\logfile2.txt
```

```

[Mailer]
# =====
# This section describes the settings for the MAIL-features of the LogAnalyzer
# =====

# --- ServerName ---
# Name of the SMTP-Server, which will be used to send the mails
ServerName = mailgds1.graadatastorage.de

# --- SMTP Authentication ---
UseSMTPAuthentication = 0
#Username = mustermann
#Password = bux91

# --- Text ---
# This text will be used in the body of the mail
# NOTE: For multiple lines the continuation-lines must start with a whitespace!
MailText =
    This is a package from your LogAnalyzer
    -----
    ...
    Your LogAnalyzer
    -----

# --- Subject ---
# Subject-Part of the mailheader
Subject = IVD-Mail

# --- Subject Prefix---
# Used for MailCurrentLogMessages("Sample Subject")
# If used, this text will be prepended to the text "Sample Subject"
SubjectPrefix = IVD001

# --- From ---
# From-Part of the mailheader
From = IVD001@company.com

# --- To ---
# To-Part of the mailheader
To = service@GrauDataStorage.de mueller@company.com maier@company.com

# --- AttachmentPrefix ---
# The Packagefilename is composed by: <AttachmentPrefix>_<YYMMDD_HHMMSS>.log.zip
# Example : IVD_001_040517_174320.log.zip
AttachmentPrefix = IVD_001_

# --- Zip Attachment ---
ZipAttachment = 1

# --- Slot ---
# Time interval to send the package (in minutes)
Slot = 1

```

```
[SNMP]
# =====
# This section describes the settings for the SNMP-features of the LogAnalyzer
# =====

# --- DestinationIP ---
# The IP-Address of the management unit
DestinationIP = 127.0.0.1

# CommunityName
# The community name of the machine
CommunityName = public

# --- SenderOID ---
# The SenderOID. This value depends on the product, that will be used
SenderOID = 1.3.6.1.4.1.18945.1.1.0
```

Anpassen folgender Angaben in den Sektionen innerhalb der Konfigurationsdatei:

Schlüssel	Überprüfung/Anpassung
SleepTime	Wartezeit in Sekunden zwischen den Analyse-Zyklen.
Debug	Debug Modus 0 = Debug-Mode aus (Default), 1 = Debug-Mode an.
ProcessFileFromBeginning	Analyse-Modus 0 = Analyse der neu hinzugekommenen Log-Meldungen (Default) 1 = Analyse der gesamten Log-Datei (Debug-Zwecke)
RulesFile	Vollqualifizierter Pfadname der Datei, welche die Analyse-regeln enthält.
TmpDir	Vollqualifizierte Verzeichnisangabe für die Ablage temporärer LogAnalyzer-Dateien. (Default: System Temp-Verzeichnis). Achten Sie darauf, daß dieses Verzeichnis existiert, ggf. muß dieses Verzeichnis vor dem ersten Start erzeugt werden.
ActivityLog	Vollqualifizierter Pfadname der Log-Datei des <i>IVD-Log-Analyzers</i> .
AnalysisFile	Vollqualifizierter Pfadname der Log-Datei, die analysiert werden soll.
ServerName	Name oder IP-Adresse des eigenen SMTP Servers.

Schlüssel	Überprüfung/Anpassung
UseSMTPAuthentication	Authentifizierungs-Modus 0 = Inaktiviert die Authentifikation (Default) 1 = Aktiviert die Authentifikation Falls die Authentifikation aktiviert wird, werden für folgende Schlüssel Einträge benötigt: <i>Username</i> <i>Password</i> Entfernen Sie in diesem Fall das Kommentarzeichen.
MailText	Text, welcher im Mail enthalten ist. Bei mehrzeiligem Text müssen die Folgezeilen mit einem Leerzeichen beginnen.
Subject	Betreff-Zeile der Mail.
SubjectPrefix	Der in diesem Schlüssel definierte Text, wird bei einer <i>MailCurrentLogMessage</i> -Aktion (siehe Seite 20) dem Betreff-Text vorangestellt.
From	Mail-Adresse des Absenders.
To	Eine oder mehrere Mail-Adressen der Empfänger durch Leerzeichen getrennt.
AttachmentPrefix	Präfix für die angehängte Datei (Analyse-Ergebnisse) damit diese beim Empfänger nicht fortlaufend überschrieben wird.
ZipAttachment	Kompressions-Modus 0 = Inaktiviert die Kompression für Anhänge 1 = Aktiviert die Kompression für Anhänge (Default)
Slot	Zeit-Intervall für den Versand von Meldungs-Paketen (alle ausgefilterten Log-Meldungen) in Minuten.
DestinationIP	Die IP-Adresse des SNMP-Servers.
CommunityName	Ein zum Gruppieren von SNMP-Hosts verwendeter Name. Standardgemäß gehören alle Hosts zur Community <i>public</i> , dem Standardnamen für die gemeinsame Community aller SNMP-Hosts.
SenderOID	Objekt ID (OID) des Sender-Systems. Die OID ist abhängig vom jeweiligen GRAU-Produkt: IVD =1.3.6.1.4.1.18945.1.1.0 ITL =1.3.6.1.4.1.18945.1.2.0 IAF =1.3.6.1.4.1.18945.1.3.0 FMA =1.3.6.1.4.1.18945.1.4.0 FSR =1.3.6.1.4.1.18945.1.5.0

3.2 Die *LogAnalyzerRules.py* Konfigurations-Datei

Die Datei *LogAnalyzerRules.py* beinhaltet die Regeln wonach der *IVD-LogAnalyzer* die definierte Log-Datei untersucht. Diese Regeln werden in Form von regulären Ausdrücken formuliert und als Rules bezeichnet. Vereinfacht dargestellt stellen diese Rules Suchmuster dar, welche auf den String (Zeichenkette) einer Log-Meldung angewandt werden.

Passen die Suchmuster (Matches) auf den String einer Log-Meldung leitet der *IVD-LogAnalyzer* eine entsprechende Aktion ein, indem eine vordefinierte Prozedur gestartet wird. Somit besteht eine Rule aus einer Komponente, die den String einer Log-Meldung analysiert (filter) und einer Komponente, welche die auszuführende Prozedur definiert (siehe Seite 20).

Syntax eines Rules:

```
RULES = \  
[  
    {  
        'filter': STRING,  
        'action': PROCEDURE  
    }  
]
```

Beispiel:

```
RULES = \  
[  
    {  
        'filter': ".*errors.*",  
        'action': "MailLogMessages() "  
    }  
]
```


3.2.1 Beschreiben eines Filters

Im zuvor dargestellten Beispiel wird beim Auftreten einer Log-Meldung, die den String „errors“ enthält die Aktion *MailLogMessages()* gestartet, welche den Versand einer Mail verursacht. Im Filter steht vor und nach dem eigentlichen Suchstring „errors“ die Zeichenkombination „.*“ (Wildcard). Diese Wildcard ist stellvertretend für eine beliebige Zeichenkette.

Für jede Filter-Konstellation muß eine eigene Rule erzeugt werden. Im folgenden Beispiel soll immer eine Mail versandt werden, wenn die Begriffe „error“ oder „failer“ in einer Log-Meldung auftauchen

```
RULES = \
[
  {
    'filter': ".*error.*",
    'action': "MailLogMessages() "
  },
  {
    'filter': ".*failer.*",
    'action': "MailLogMessages() "
  }
]
```

Mit Hilfe der Exclude-Anweisung ist es möglich, die Suchabfragen in den Rules so zu definieren, daß man bestimmte Meldungen ausschließt, obwohl die Filter-Bedingung für diese Log-Meldung erfüllt ist.

Ein Beispiel macht den Einsatz der Exclude-Anweisung deutlich. Beim Auffinden des Strings „errors“ in einer Log-Meldung soll wieder eine Mail generiert werden. Dies soll jedoch nicht geschehen, wenn es sich um die Meldung „No errors found“ handelt.

```
RULES = \
[
  {
    'filter': ".*error.*",
    'exclude': ".*No errors found.*",
    'action': "MailLogMessages() "
  }
]
```

Alle Log-Meldungen von IVD verfügen über denselben Aufbau wie die Meldung im folgenden Beispiel (Error-Nummer 15001).

```
[2004/03/12 23:12:15, infinistore, LAS, GTL-340i, 20040312000002, 000021]
ERROR: Move: D=1 -> S=3 FAILED (15001: Invalid barcode in slot or drive.
'Drive # 1 bc: expected: 000021')
```

Weitere Informationen über den Aufbau und die Error-Codes der Log-Meldungen entnehmen Sie bitte dem *IVD User's Guide*.

Möchten Sie den *IVD-LogAnalyzer* veranlassen, nach dem String „[E]“ zu suchen, welcher eine Log-Meldung als Error-Meldung kennzeichnet, haben Sie das Problem, geschützte Zeichen (hier „[“ und „]“) im Filter definieren zu müssen. Bei der Angabe von geschützten Zeichen in einem Filter, wird den geschützten Zeichen ein Backslash vorangestellt (siehe Beispiel).

```
RULES = \
[
  {
    'filter': ".*\[E\].*",
    'action': "MailLogMessages()"
  }
]
```

So wird aus dem Suchstring „[E]“ der Suchstring „\[E]“.

Das Wildcard „.*“, welches für eine beliebige Zeichenkette steht, wurde ja bereits an anderer Stelle erläutert.

Schutz-Zeichen:

Zeichen	wird zu ...
[\[
]	\]
{	\{
}	\}
'	\'
"	\"
\	\\

Meta-Zeichen:

Zeichen	ersetzt ...
. *	kein, oder mehrmaliges Auftreten beliebiger Zeichen.
.	ein einzelnes Zeichen.
^	Das Zirkumflex steht für den Anfang eines Strings.
\$	steht für das Ende eines Strings.
	Trennt Alternativen (A B = Zeichen 'A' oder 'B').

In jedem Suchstring können *Regular Expressions* (RE) eingesetzt werden. Dies bedeutet, daß der Filter greift, sobald ein über *Regular Expressions* definierter Substring mit der jeweiligen Log-Meldung übereinstimmt. Das folgende Beispiel soll die Wirkungsweise der *Regular Expressions* verdeutlichen.

Log-Meldungen:

```
[FSC1111][E][2003/12/16 - 18:39:36,292][00000001][0448:061c] ML2-Partition ...  
[FSC1111][I][2003/12/16 - 18:39:36,292][00000001][0448:061c] Switch to Part. [E] ...
```

Rule mit RE

```
RULES = \  
[  
  {  
    'filter': "^\\[. {7} \\] \\[E\\]",  
    'action': "MailLogMessages() "  
  }  
]
```

Die oben dargestellte Rule filtert nur die erste der beiden Log-Meldungen, da sich zwischen den eckigen Klammern am String-Anfang (^\\[) exakt sieben Zeichen (. {7}) befinden bevor ein [E] (\\[E\\]) auftritt.

Weitere Informationen zu *Regular Expressions* entnehmen Sie bitte der einschlägigen Fachliteratur.

3.2.2 Aufruf von Aktionen

Als Aktionen stehen folgende Prozeduren zur Verfügung:

Mail Versand und Speichern von Log-Meldungen

Prozedur	Beschreibung
<code>MailLogMessages()</code>	Die gefilterten Log-Meldungen werden als Mail an die in der Datei <i>LogAnalyzer.config</i> definierte Adresse versandt.
<code>MailLogMessages('btr')</code>	Die gefilterten Log-Meldungen werden als Mail an die in der Datei <i>LogAnalyzer.config</i> definierte Adresse versandt. Als Betreff-Text (Subject) wird allerdings der Inhalt von <i>btr</i> eingesetzt (siehe Beispiel unten).
<code>MailCurrentLogMessage('btr')</code>	Die aktuell gefilterte Log-Meldung wird als Mail an die in der Datei <i>LogAnalyzer.config</i> definierte Adresse versandt. Als Betreff-Text (Subject) wird der Inhalt von <i>btr</i> eingesetzt.
<code>SaveCurrentLogMessage()</code>	Die aktuell gefilterte Log-Meldung wird vom LogAnalyzer in einer temporären Datei gespeichert.
<code>DisableActions()</code>	Nach dem Aufruf dieser Prozedur werden alle folgenden Prozedur-Aufrufe unterdrückt.
<code>EnableActions()</code>	Nach dem Aufruf dieser Prozedur wird die Prozedur-Unterdrückung (siehe <i>DisableActions</i>) aufgehoben.

Beispiel:

```

RULES = \
[
  {
    'filter': "^.*\[E\].*",
    'action': "MailLogMessages('Error Message from system IVD001')"
  }
]

```

Senden von SNMP-Traps

Prozedur	Beschreibung
<code>SendTrap (Trap)</code>	Die gefundene Log-Meldung wird abhängig vom übergebenen Trap-Bezeichner <i>Trap</i> als SNMP-Trap verschickt.

Trap	Kategorie	Beschreibung
I	Info	Ein Info SNMP-Trap wird gesendet.
W	Warning	Ein Warning SNMP-Trap wird gesendet.
E	Error	Ein Error SNMP-Trap wird gesendet.
C	Critical	Ein Critical SNMP-Trap wird gesendet.

Beispiel:

```

RULES = \
[
  {
    'filter': ".*\[E\].*",
    'action': "SendTrap('E')"
  }
]

```

Zudem ist es möglich, mehr als eine Prozedur pro Aktion aufzurufen. In diesem Fall werden die Aktionen durch ein Komma getrennt.

Beispiel:

```

RULES = \
[
  {
    'filter': ".*\[E\].*",
    'action': "MailCurrentLogMessage('Critical Message'), SendTrap('C')"
  }
]

```

3.2.3 Beispiel einer *LogAnalyzerRules.py* Konfigurations-Datei

```
# =====
# Sample Rulesfile for LogAnalyzer Version 2.3.1
# Created: 25.10.05
# Author: GDS Development
# =====

RULES = \
[
    {
        'filter': "Execution stack dump",
        'action': "DisableActions()"
    },

    {
        'filter': "End of stack dump",
        'action': "EnableActions()"
    },

    {
        'filter': "^\\[{7}\\]\\[E\\]",
        'action': "SaveCurrentLogMessage()"
    },

    {
        'filter': "^\\[{7}\\]\\[E\\]",
        'action': "SendTrap('E')"
    },

    {
        'filter': ".*",
        'exclude': "(^\\[{7}\\]\\[E\\]|End of stack dump|^$)",
        'action': "SaveCurrentLogMessage()"
    },

    {
        'filter': ".*So what.*",
        'action': "MailLogMessages()"
    },

    {
        'filter': "Critical",
        'action': "MailCurrentLogMessage('Critical message found')"
    },

    {
        'filter': "Critical",
        'action': "SendTrap('C')"
    }
]
```

3.3 MIB-Dateien

MIB-Dateien dienen dazu, ein Software-Produkt in ein SNMP Management System einzubinden. Die für den *IVD-LogAnalyzer* relevanten MIB-Dateien lauten:

- ☒ *GDS.MIB*
- ☒ *GDS_IVD.MIB*

Diese MIB-Dateien sind nur erforderlich, wenn die vom *IVD-LogAnalyzer* erzeugten SNMP-Traps durch das SNMP Management System verarbeitet werden sollen. Sie werden dann vom SNMP Management System importiert.

Durch die Installation des *IVD-LogAnalyzers* werden die oben aufgeführten MIB-Dateien in folgenden Verzeichnissen abgelegt:

Windows:	Installationsverzeichnis des <i>IVD-LogAnalyzers</i> Standard: <i>C:\Program Files\GDS\LogAnalyzer</i>
Linux:	<i>/etc/opt/ivd</i>

Weitere Informationen zu MIB-Dateien entnehmen Sie der Dokumentation Ihres SNMP Management Systems.



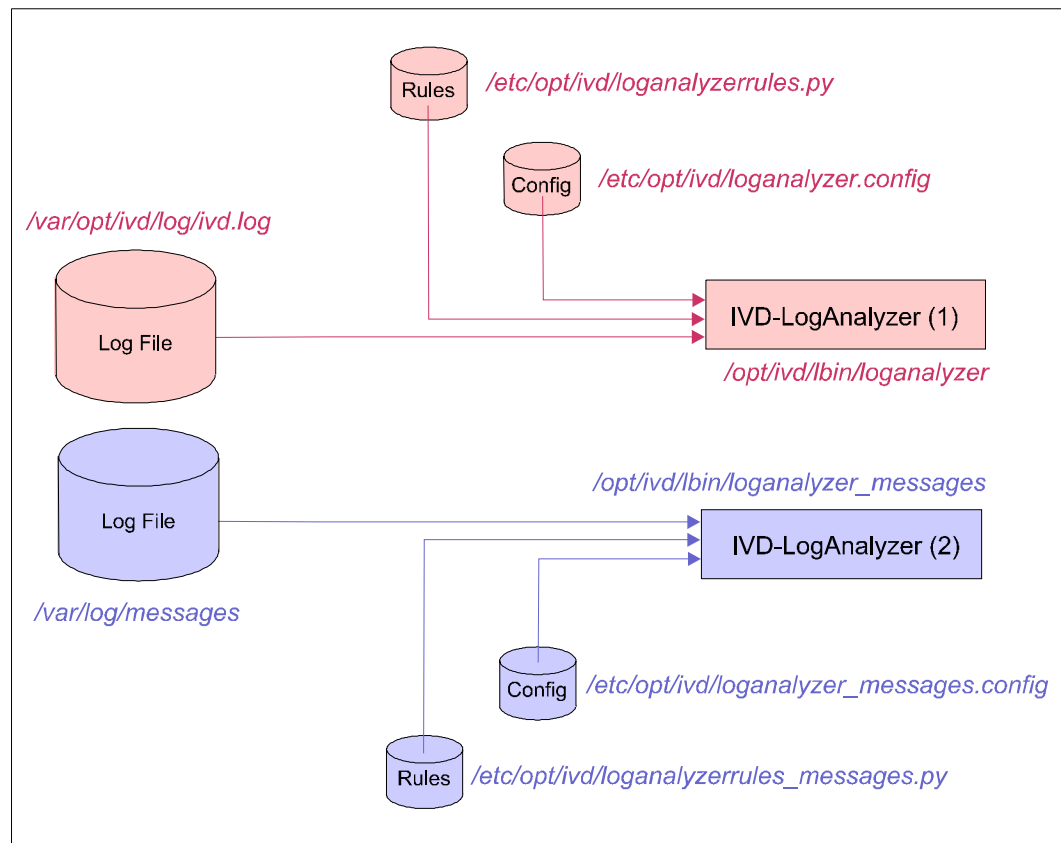
Benutzen Sie einen *SNMP Trap Watcher* um Ihrer SNMP-Konfiguration zu testen. Einen geeigneten *SNMP Trap Watcher* finden Sie im Internet auf der Seite

<http://www.bttsoftware.co.uk/snmptrap.html>

INFINISTORE Virtualdisk

4. Einsatz auf einem Linux-System

Auf einem Linux-System ist es wünschenswert, daß neben der IVD Log-Datei `/var/opt/ivd/log/ivd.log` auch die Linux Log-Datei `/var/log/messages` überwacht wird. Der *IVD-LogAnalyzer* muß deshalb zweimal mit unterschiedlichen Konfigurations-Dateien gestartet werden.



Die entsprechenden Templates wurden während der Installation angelegt (siehe Seite 10).

4.1 Die *LogAnalyzer_messages.config* Konfigurations-Datei

Zur Überwachung einer zweiten Log-Datei (hier */var/log/messages*) legt das Installationsprogramm eine zweite Standard-Konfigurationsdatei (*LogAnalyzer_messages.config*) an. Die Standardwerte sind entsprechend den gegebenen Anforderungen anzupassen (siehe Beispiel).

```
.
.
# --- RulesFile ---
# Path to the rules the LogAnalyzer will load on startup
# NOTE: Must be full qualified!
RulesFile = /etc/opt/ivd/loganalyzerrules_messages.py

# --- TmpDir ---
# Directory to write temporary files.
# If not set, the System-Temporary-Directory will be used.
# NOTE: Must be full qualified!
# TmpDir = d:\MyTmp\LogTmp

# --- ActivityLog ---
# In this file the LogAnalyzer writes down his own activities.
# NOTE: Must be full qualified!
ActivityLog = /var/opt/ivd/log/loganalyzer_messages.log

# --- AnalysisFile ---
# The file to be analyzed by the LogAnalyzer
AnalysisFile = /var/log/messages
.
.
```

Der Aufbau ist identisch zum Aufbau der Datei *LogAnalyzer.config*. Weitere Informationen über den Aufbau dieser Konfigurations-Datei finden Sie ab der Seite 12.

4.2 Die *LogAnalyzerRules_messages.py* Konfigurations-Datei

Zur Überwachung der Linux Log-Datei */var/log/messages* legt das Installationsprogramm eine Standard-Konfigurationsdatei (*LogAnalyzerRules_messages.py*) an. Der Aufbau ist identisch zum Aufbau der Datei *LogAnalyzerRules.py* (siehe Seite 16).

```
.  
.   
RULES = \  
[  
    {  
        'filter': "kernel.*error",  
        'action': "SaveCurrentLogMessage()"   
    },  
    {  
        'filter': "CRITICAL",  
        'action': "MailCurrentLogMessage('Critical')"  
    },  
]  
.  
.
```

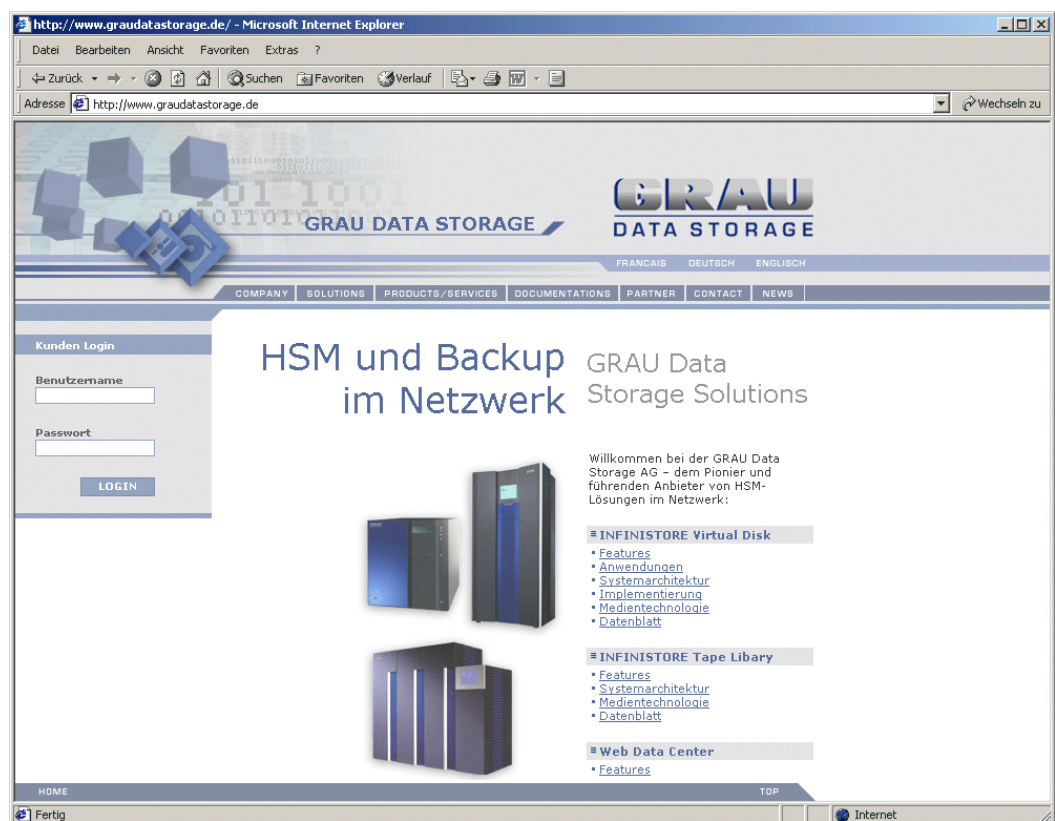
Weitere Informationen über den Aufbau dieser Konfigurations-Datei finden Sie ab der Seite 12.

INFINISTORE Virtualdisk

5. GRAU Service Center

Technischer Support	
Telefon:	+49 (0)7171 / 187-333
Telefax:	+49 (0)7171 / 187-250
E-mail:	support@graudatastorage.de
Internet	www.graadatastorage.de

Sales & Marketing	
Telefon:	+49 (0)7171 / 187-212
Telefax:	+49 (0)7171 / 187-250
E-mail:	info@graudatastorage.de
Internet	www.graadatastorage.de



INFINISTORE VirtualDisk

6. Software-Lizenzvertrag

Zur Beachtung:

Die Verwendung dieser Software unterliegt dem beigefügten Lizenzvertrag der GRAU Data Storage AG. Durch die Verwendung der Software stimmen Sie den Bestimmungen der nachstehenden Lizenz zu.

Falls Sie mit der GRAU Data Storage AG keinen weiteren Vertrag unterzeichnet haben, werden die Benutzerrechte für diese Software durch die folgenden Lizenzbestimmungen festgelegt.

LIZENZERTEILUNG

GRAU Data Storage AG gewährt Ihnen die nicht ausschließliche, nicht übertragbare Lizenz und das Recht zur Verwendung der Software. Es ist nicht gestattet, die Software abzuändern oder die Lizenz- oder Steuerungsfunktionen zu deaktivieren. Falls Sie eine Mehrfachbenutzerlizenz besitzen, ist die gleichzeitige Verwendung der Software auf die maximale Anzahl berechtigter Benutzer beschränkt. Der Lizenznehmer ist für die Wahl des Programms zur Erreichung der beabsichtigten Ergebnisse und für die Installation, die Verwendung und die durch das Programm erlangten Ergebnisse verantwortlich. Alle durch diese Lizenz nicht speziell gewährten Rechte sind ausdrücklich der GRAU Data Storage AG vorbehalten.

EIGENTUMSRECHT

Sie stimmen zu, dass diese Software im Eigentum der GRAU Data Storage AG verbleibt, und dass Sie nur eine Lizenz zur Verwendung dieser Software besitzen. GRAU Data Storage AG behält das Eigentumsrecht an der gesamten Software, an allen Softwarekopien und zugehörigen Materialien.

KOPIEN UND ANPASSUNGEN

Es ist gestattet, eine Sicherungskopie der Software anzufertigen und zu behalten, sofern diese ausschließlich zu Sicherungs- und Archivierungszwecken verwendet wird und die Sicherungskopie in Ihrem Besitz bleibt. Zusätzlich sind alle auf den Etiketten (Label) der Originaldisk angezeigten Informationen (einschließlich der Bezeichnung der Software, der Seriennummer und aller Urheberrechtsvermerke) auf die Etiketten (Label) der Sicherungsdisk zu übertragen. Es ist nicht gestattet, die Software auf ein öffentliches Netzwerk zu kopieren oder die Software oder das Handbuch zu vermieten, zu verleihen, in Unterlizenz weiter zu geben, zeitlich festgelegt gemeinsam zu nutzen oder zu verkaufen.

VERBOT DES REVERSE ENGINEERING

Die Änderung, das Reverse Engineering, Reverse Compiling oder die Disassemblierung der Software ist ohne die schriftliche Genehmigung der GRAU Data Storage AG ausdrücklich verboten. In einigen Ländern ist eine begrenzte Disassemblierung oder ein beschränktes Reverse Compiling ohne Genehmigung gestattet. Für weitergehende diesbezügliche Informationen bitten wir Sie, sich an die Rechtsabteilung von GRAU Data Storage AG zu wenden. Es ist nicht gestattet, die Software zu dekodieren, sofern die Dekodierung nicht Bestandteil der Softwarefunktionen ist.

ÜBERTRAGUNG

Ohne die vorausgehende schriftliche Zustimmung der GRAU Data Storage AG ist es nicht gestattet, die Software und/oder diesen Lizenzvertrag an andere Personen zu übereignen oder zu übertragen. Falls eine solche Zustimmung erteilt wird und Sie die Software und/oder diesen Lizenzvertrag übereignen oder abtreten, müssen Sie derselben Person gleichzeitig jede Kopie der Software sowie die Begleitdokumentation übertragen, und der Empfänger muss diesem Lizenzvertrag zustimmen.

BEENDIGUNG

Dieser Lizenzvertrag endet automatisch bei Verletzung einer der Bedingungen dieses Lizenzvertrages. Falls Ihr Nutzungsrecht für die Software aus irgendwelchen Gründen endet, müssen Sie alle Kopien der Software und die gesamte, Ihnen bereitgestellte Dokumentation entfernen und vernichten.

EXPORTBESCHRÄNKUNGEN

Sie stimmen zu, die Software oder Begleitdokumentation oder alle Produkte, auf denen die Software eingesetzt ist oder die Dokumentation, die gegen anwendbare Gesetze oder Vorschriften verstoßen, nicht zu exportieren oder zu reexportieren. Die Verletzung einer Bestimmung dieses Vertrages führt automatisch zur Aufhebung Ihrer Lizenz und kann dazu führen, dass Sie gegenüber der GRAU Data Storage AG für rechtmäßige Schadensersatzforderungen haftbar sind.

**BESCHRÄNKTE GEWÄHRLEISTUNG UND
BESCHRÄNKUNG VON RECHTSMITTELN**

Die Software wird "in der bestehenden Form" ohne jederlei Gewährleistung ausdrücklicher oder konkludenter Art, einschließlich, ohne sich jedoch darauf zu beschränken, stillschweigende Gewährleistung oder die Zusicherung der Marktgängigkeit und Eignung für einen bestimmten Zweck bereitgestellt. Unbeschadet des Vorstehenden wird keine Gewährleistung dafür übernommen, dass die Software Ihre Anforderungen erfüllt oder mit anderer Software oder Hardware kompatibel ist. GRAU Data Storage AG haftet unter keinen Umständen für entgangenen Gewinn, Geschäftsunterbrechung, Verlust von Geschäftsinformationen oder Vermögensschaden, einschließlich, ohne sich jedoch darauf zu beschränken, Ersatz immateriellen Schadens, Schadensersatz für Aufwendungen bei Vertragserfüllung, Folgeschaden oder sonstige Schadensersatzforderungen. In einigen Ländern ist der Ausschluss einer Haftungsbeschränkung oder von Schadensersatz für Aufwendungen bei Vertragserfüllung oder von Folgeschäden nicht zulässig, so dass diese Beschränkung für Sie in diesem Fall eventuell keine Geltung besitzt.

Die Software wurde ausschließlich auf private Kosten entwickelt. Die Software und die Dokumentation wird als "Kommerzielle Computer Software" bezeichnet und lizenziert.

INFINISTORE ist ein eingetragenes Warenzeichen der GRAU Data Storage AG.