

HOWTO zum Daemon "syslog" bzw. "syslog-ng" (System Logging)  
 (C) 2008 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>  
 OSTC GmbH, <http://www.ostc.de>

\$Id: syslog-HOWTO.txt,v 1.13 2010-05-26 13:56:46 tsbirn Exp \$

Diese Dokument beschreibt die Eigenschaften und Möglichkeiten des System-Logging-Daemons "syslog" und "syslog-ng" von Linux-Systemen.

## Inhaltsverzeichnis

- 1) Einleitung
- 2) Eigenschaften von "syslog"
- 3) Probleme mit "syslog"
- 4) Konfiguration
  - 4.1) Kategorie (facility)
  - 4.2) Priorität (priority)
  - 4.3) Aktion (action)
  - 4.4) Standard-Verzeichnis für Logdateien
  - 4.5) Beispiele für Einträge in "/etc/syslog.conf"
  - 4.6) Ablauf einer Umkonfiguration
- 5) Syslog-NG (Next Generation, Balabit)
  - 5.1) Objekte
  - 5.2) Objekt globale Optionen ("options")
  - 5.3) Objekt Quelle ("source")
  - 5.4) Objekt Ziel ("destination")
  - 5.5) Objekt Bedingung ("filter")
  - 5.6) Objekt Log ("log")
  - 5.7) Templates und Makros
  - 5.8) Beispielskript "log2mysql.sh"
- 6) Links

## 1) Einleitung

In jedem Betriebssystem laufen im Hintergrund viele sogenannte "Daemonen" (Dienste/Services), die nicht an ein Terminal gebunden sind. Ihre Meldungen (Debugging, Info, Warnung, Fehler) müssen aber trotzdem irgendwo ausgegeben oder abgelegt werden, um sie später auswerten zu können oder um Fehler zu suchen. Dafür gibt es unter Linux einen zentralen Dienst namens "syslog", an den die Daemonen ihre Meldungen schicken können.

Typische Anwendungen des syslog-Daemons sind:

- \* Protokollieren von Vorgängen
- \* Monitoring von Systemen
- \* Netzwerkweite Integration vieler Log-Quellen in zentrales Repository

Der Dämon "/usr/bin/syslogd" erlaubt das Aufzeichnen/Protokollieren der Meldungen vom Kernel oder anderen Programmen in Dateien ("Logging"). So können System-Fehler oder sonstige Ereignisse aufgehoben und später nachgesehen werden. Er nimmt die Meldungen anderer Dienste entgegen und verarbeitet sie. Was er damit tut, wird in einer Datei namens "/etc/syslog.conf" konfiguriert.

Jede Meldung, die "syslog" bekommt, wird vom Absender mit einer KATEGORIE (Facility/Quelle) und einer PRIORITÄT (Priority) klassifiziert. Weiterhin enthält Sie den NAMEN des Absenderprogramms und einen frei wählbaren TEXT. Der NAME wird auch TAG (String) genannt. Mit ihm identifiziert sich der Daemon oder die Anwendung (wird zum Präfix vor dem Logmeldungstext).

- \* Kategorie: kern, mail, auth(priv), cron, daemon, local0-9, lpr, user
- \* Priorität: panic/emerg, alert, crit, err(or), warn(ing), notice, info, debug

Auf Basis dieser Informationen (und evtl. anderer) steuert "syslog" die weitere Verarbeitung der Meldungen über BEDINGUNGEN. Trifft auf eine Meldung eine der Bedingung zu (es dürfen beliebig viele zutreffen), dann kann sie:

- \* In eine Datei geschrieben (angehängt) werden: /PFAD/ZUR/DATEI  
(Datei muss bereits existieren!)
- \* In eine Datei geschrieben werden ohne Puffern: -/PFAD/ZUR/DATEI  
(Datei muss bereits existieren!)
- \* Auf allen Terminals eines Users ausgegeben werden: USERNAME
- \* Auf allen Terminals ausgegeben werden: \*
- \* Einem Kommando übergeben werden: | KMDO
- \* Über das Netz an einen Rechner geschickt werden: @HOSTNAME  
(muss per Konfiguration Meldungen annehmen)

Logdateien liegen unter Linux fast ausschließlich unter "/var/log" bzw. Unterverzeichnissen darin und habe häufig die Endung "\*.log" oder einen Präfix "log.". Typische Namen von Standard-Logfiles sind:

```

* /var/log/allmessages          # ALLE Logmeldungen
* /var/log/apache2/access.log   # Webserver-Zugriffe
* /var/log/apache2/error.log    # Webserver-Fehler
* /var/log/auth.log             # An/Abmeldungen (erfolgreich/fehlgeschlagen)
* /var/log/cups/access_log      # Druckaufträge
* /var/log/kern.log             # Kernel-Meldungen
* /var/log/localmessages        # ALLE lokalen Logmeldungen
* /var/log/mail.(log|warn|err|info) # Mail-Logmeldungen (Fehler, Warnungen, ...)
* /var/log/messages            # ALLE Logmeldungen
* /var/log/samba/log.nmbd       # Samba Namensauflösung
* /var/log/samba/log.smbd       # Samba Sharezugriffe
* /var/log/syslog               # Systemmeldungen
* /var/log/user.log             # Benutzermeldungen

```

Folgende Meldungen werden nicht von "syslog" aufgezeichnet:

```

* Beim Booten des Systems vom Kernel generierte Meldungen werden NICHT
  mitgeloggt, dafür ist "dmesg" zuständig.
* Normale Loginvorgängen an einem Terminal werden NICHT mitgeloggt,
  dafür sind "last", "lastb", "lastlog" und "faillog" zuständig.
* Von Benutzer abgesetzte Kommandos werden NICHT mitgeloggt,
  dafür ist "lastcomm" zuständig.

```

## 2) Eigenschaften von "syslog"

```

* Im Zshg. mit Sendmail entstanden
* Ausgangsbasis: BSD-syslog (sehr alt: 1983)
* Sehr einfach gestrickt: UDP-Nachrichten im Klartext
* Protokoll-Definition: RFC 3154 + RFC 3195
* Verwendet Port 514
* Option "-r" (remote) -> Annahme beliebiger Nachrichten auf UDP-Port 514

```

```

                Nachricht (kurzer Text)
Client -----> Server/Daemon
Client -----> Relay -----> Server/Daemon

```

## 3) Probleme mit "syslog"

```

* Verlust von Paketen möglich (UDP garantiert Paket-Ankunft nicht)
* Überflutung mit Paketen möglich (DoS = Denial of Service)
* Gefälschte Pakete möglich (alles auf Port 514 wird akzeptiert)
* Nachrichten sind unverschlüsselt (liegen im Klartext vor)
* Unflexible Filter: nur Facility (Ursprungsdienst) + Priority (Dringlichkeit)

```

Lösung: syslog-ng (siehe weiter unten).

## 4) Konfiguration

Welche Meldungen wo aufgezeichnet werden sollen, ist in folgender Konfigurations-Datei einzutragen:

```
/etc/syslog.conf
```

Alle Meldungen werden zusätzlich standardmäßig auf Konsole "tty10" ausgegeben (mit "Strg-Alt-F10" erreichbar).

Jede Zeile in der Konfigurations-Datei enthält ein oder mehrere durch ";" getrennte "KATEGORIE.PRIORITÄT"-Paare (Facility/Quelle + Meldungspriorität) der aufzuzeichnenden Meldungen und dann durch TABs (keine Leerzeichen!) getrennt die beim Auftreten einer derartigen Meldung auszuführende Aktion "ACTION" (z.B. den Namen einer Datei für die Aufzeichnung der Meldungen).

```
KATEGORIE.PRIORITÄT;... TAB... ACTION
```

Mehrere solche Paare mit der gleichen Aktion sind durch ";" zu trennen (kein Leerzeichen dazwischen!). Mehrere Kategorien und/oder Prioritäten mit der gleichen Aktion sind durch "," zu trennen. Ein "=" vor einer Priorität verlangt GENAU diese Priorität, ein "!" vor einer Priorität trifft auf alle NIEDRIGEREN Prioritäten zu; ein "!=" vor einer Priorität trifft auf alle Prioritäten AUSSER dieser zu. Kommentarzeilen werden wie üblich durch ein führendes "#" gekennzeichnet, Leerzeilen sind ebenfalls erlaubt.

Einschränkungen bei Solaris (und alten syslog-Versionen):

```

* Nur TABs zur Trennung von FACILITY.PRIORITY und ACTION erlaubt
* "*" nur bei "facility" erlaubt, nicht bei "Priorität"
  (Ersatz: DEBUG statt "*" verwenden)

```

## 4.1) Kategorie (facility)


Kann einer der folgenden Werte (oder "\*" für alle) sein:

Wert	Bedeutung
auth	Anmeldung/Authentifizierung
authpriv	Anmeldung/kritische Sicherheitsmeldungen
cron	cron-Daemon
daemon	Daemon allgemein
ftp	FTP-Server
kern	Kernel
lpr	Drucksystem
mail	Mailsystem
mark	Markierung durch syslog selbst (Zeitstempel/timestamp)
news	Newsgruppensystem
security	Sicherheitsmeldung (VERALTET, nun "auth")
syslog	Meldung durch syslog selbst
user	Meldung von Benutzern
uucp	UUCP-Meldung (Unix to Unix Copy)
local0..7	Frei verwendbare Meldungen (eigene Programme/Dienste)
*	Alle Kategorien

## 4.2) Priorität (priority)

Kann einen der folgenden Werte (oder "\*" für alle) annehmen, die Dringlichkeit steigt dabei von oben nach unten. Jede Priorität umfasst die HÖHEREN Prioritäten mit (d.h. "err" umfasst auch "crit", "alert", "emerg"):

Wert	Bedeutung
debug	Debugmeldung
info	Information
notice	Mitteilung
warn(ing)	Warnung
err(or)	Fehler
crit	Kritischer Fehler
alert	Alarm
emerg	Notmeldung (Kernel!)
panic	Notmeldung (VERALTET, nun "emerg")
*	Alle Prioritäten (analog "debug")
none	Kategorie NICHT aufzeichnen
=info	NUR Info-Meldungen
!err	Alle Meldungen KLEINERER Priorität
!=alert	Alle Meldungen AUSSER "alert"


  
zunehmende Dringlichkeit

## 4.3) Aktion (action)

Kann einen der folgenden Werte annehmen. Wird der Dateiname mit einem "-" eingeleitet, so wird die Meldung sofort ("synchron") in die Datei geschrieben, d.h. NICHT zunächst im Speicher gepuffert. Stürzt der Rechner kurz nach dem Empfang einer ungepufferten Meldung ab, kann es daher NICHT passieren, dass die Meldung NICHT in der Logdatei steht:

Aktion	Bedeutung
FILE	An Datei FILE anhängen (gepuffert = asynchron, Pfadname)
-FILE	An Datei FILE anhängen (ungepuffert = synchron, Pfadname)
CMD	An Kommando "CMD" auf Stdin übergeben (Pfadname)
@HOST	An Rechner HOST schicken
USER,...	An Benutzer USER,... schicken (auf deren Terminals)
*	An alle Benutzer schicken (auf allen Terminals ausgeben)

## 4.4) Standard-Verzeichnis für Logdateien

Das Standard-Verzeichnis für die Logging-Dateien lautet:

/var/log

Die Log-Dateien darin müssen regelmäßig vom Systemverwalter kontrolliert und gekürzt werden, um Systemprobleme festzustellen und das Dateisystem nicht mit Log-Meldungen "vollzumüllen". Alternativ übernimmt auch der Daemon "logrotate" diese Aufgabe.

## 4.5) Beispiele für Einträge in "/etc/syslog.conf"

```
-----
mail.*                /var/log/mail.log      # Alle Meldungen aus Mailsystem
mail.debug            /var/log/mail.log      #   (analog)
mail,news.*          /var/log/mailnews.log  # Alle Meldungen von mail+news
auth.warning         /var/log/auth.log      # Sicherheits-M. ab "warning"
*.warn;*.err        /dev/tty10             # Warnungen+Fehler an Terminal tty10
*.crit               | lpr -P loglp        # Kritische Meldungen drucken
*.emerg;*.panic     *                       # Emergency+Panic an alle Term.
*.                  -/var/log/allmessages  # Alle Meldungen aufzeichnen
*.debug             -/var/log/allmessages  #   (analog)
*.=info             /var/log/info.log      # NUR Info-Meldungen
*.!err              /var/log/belowerr.log  # Alle KLEINER Error-Meldungen
*.!=alert           /var/log/noalert.log   # Alle AUSSER Alert-Meldungen
-----
```

## 4.6) Ablauf einer Umkonfiguration

Um Änderungen am Log-Verhalten durchzuführen, müssen folgende Schritte in der angegebenen Reihenfolge durchgeführt werden:

- 1) Datei "/etc/syslog.conf" editieren (z.B. auskommentierte Zeile einkom.).
- 2) Neue Log-Dateien ANLEGEN, falls sie noch nicht existieren (z.B. per "touch /var/log/allmessages", unbedingt notwendig!).
- 3a) syslog-Dämon mitteilen, dass er seine Konfigurationsdatei neu lesen soll. Dazu ist seine Prozessnummer zu ermitteln und dem Prozess das Signal SIGHUP (1 = Hangup) zu schicken.

```
ps ax | grep syslogd      # => PID
kill -1 PID              # oder
kill -HUP PID           # oder
kill -SIGHUP PID        # oder
killall -HUP syslogd
```

- 3b) Alternativ kann der syslog-Dämon auch beendet und neu gestartet werden, da er bei jedem Start seine Konfigurationsdatei liest, oder er kann aufgefordert werden, seine Konfigurationsdatei neu zu lesen:

```
/etc/init.d/syslog restart
rcsyslog reload          # rc... nur bei SuSE-Linux
```

- 3c) Alternativ kann das System auch heruntergefahren und wieder neu gebootet werden, da der syslog-Dämon bei jedem Systemstart hochfährt und seine Konfigurationsdatei liest:

```
shutdown -r now
reboot
init 6
```

- 4) Eine ständig aktualisierte Anzeige aller Systemmeldungen auf einer Konsole wird z.B. durch Einloggen als root auf dieser Konsole und Absetzen des folgenden Kommandos erreicht (-f=follow):

```
tail -f /var/log/allmessages
```

- 5) Testen durch Erzeugen von Meldungen auf der Kommandozeile (Shell-Skript): (Default-Kategorie+Priorität "user.info" und Default Tag "logger")

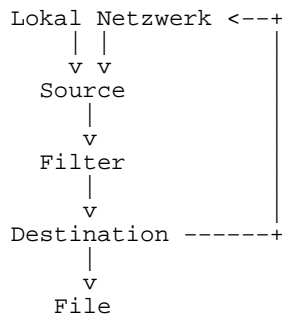
```
logger -p FACILITY.PRIORITY -t TAG "TEXT"
logger "Hallo hier bin ich"
logger -p kern.panic "Nur ein Scherz..."
logger -p kern.panic -t TEST "Nur ein Test..."
```

## 5) Syslog-NG (Next Generation, Balabit)

- ```
-----
```
- \* Verbesserungen gegenüber "syslog"
    - + Netzwerkfähigkeiten wesentlich erweitert
      - Auch per TCP (gesicherte Übertragung, verschlüsselte Tunnels)
    - + Filter wesentlich erweitert
    - + Logformat über "Templates" definierbar
    - + Meldungsteile in Makros -> Automatismen
    - + Feingranulare Logdateien
    - + Innerhalb "file"-Destinations werden Makros expandiert
      - > automatische Host/Datums-Hierarchie von Verz./Dateien möglich
  - \* Neue Version 3.0
    - + OSE (Open Source Edition)

- SSL/TLS-Verschlüsselung der Kommunikation
- SSPMF-Support (Syslog Standard Protocol and Message Format)
- Trennung in Name+Wert-Paare
- Makros
- Log-Meldungsteile umschreiben/modifizieren
- Ablage in Datenbanken möglich (MySQL, MS-SQL, Oracle, PostgreSQL, SQLite)
- + PE (Premium Edition)
  - Kommerzielle Variante von Balabit (kostenpflichtig)
  - Sichere Ablage auf Platte (verschlüsselt, signiert)
  - Binäre und komprimierte Ablage auf Platte
  - Zeitstempel hochgenau und von externen Zeitquellen
  - Syslog-Agent für Windows
  - Windows-GUI für Nachrichtenkonfiguration
  - Nachrichten-Puffer auf Festplatte falls Logserver nicht erreichbar
  - Hochlastfähig
  - Viele Plattformen (Linux, Solaris, Windows, Open/Free/NetBSD, IBM AIX, HP-UX)

\* Ablauf-Diagramm des syslog-ng:



- \* Konfigurationsdateien (entweder A oder B)
- /etc/syslog-ng/syslog-ng.conf # A) Für alle Dienste gemeinsam
  - /etc/syslog-ng/syslog-ng.d/\* # B) Pro Dienst eine Datei

### 5.1) Objekte

Syslog-NG basiert auf einem "Baukasten" bestehend aus benannten "Objekten", die erst bei der Verknüpfung zu einem "Log-Objekt" aktiviert werden (auf Klammerung jedes Objekts durch "{...}" und Abschluss durch ";" achten).

| Objektyp         | Definitionssyntax                                                                  |
|------------------|------------------------------------------------------------------------------------|
| Option           | options { OPTION1(PARAMS); ... };                                                  |
| Quelle/Source    | source IDENT { DRIVER(PARAMS) };                                                   |
| Ziel/Destination | destination IDENT { DRIVER(PARAMS); ... };                                         |
| Filter           | filter IDENT { EXPRESSION; ... };                                                  |
| Log/Aktion       | log { source(ID); ...; filter(ID); ...; destination(ID); ...; flags(FLAG; ...); }; |

### 5.2) Objekt globale Optionen ("options")

Legt globale Einstellungen des "syslog-ng"-Daemons fest.

| Option + Parameter         | Bedeutung                                    |
|----------------------------|----------------------------------------------|
| sync(N)                    | Anz. Pufferzeilen bevor auf Platte schreiben |
| log_fifo_size(N)           | Anz. Ausgabepufferzeilen                     |
| chain_hostnames(Y/N)       | Eigenen Hostnamen an generierenden anhängen  |
| keep_hostnames(Y/N)        | Umschreiben des Hostnamen aktivieren         |
| check_hostnames(Y/N)       | Hostnamen auf gültige Buchstaben prüfen      |
| use_dns(Y/N)               | DNS zur Hostnamenauflös. nutzen (blockiert!) |
| dns_ache(Y/N)              | DNS-Antworten zwischenspeichern              |
| dns_cache_size(N)          | Anz. gepufferte DNS-Antworten                |
| dns_cache_expire(N)        | Anz. Sek die DNS-Antwort gepuffert wird      |
| dns_cache_expire_failed(N) | Anz. Sek die fehlgeschlagene DNS-Antwort ... |
| use_fqdn(Y/N)              | Vollständigen Hostnamen (FQHN) verwenden     |
| stats(N)                   | Anz. Sek zwischen Statistiken                |

### 5.3) Objekt Quelle ("source")

Definiert, welche Quelle von Meldungen relevant ist.

| Quelle      | Beispiel                                          |
|-------------|---------------------------------------------------|
| unix-stream | unix-stream("/dev/log")                           |
| internal    | internal() WICHTIG!                               |
| file        | file("/proc/kmsg" log_prefix("kernel: "))         |
| pipe/fifo   | pipe("/dev/xconsole")                             |
| usertty     | usertty("root")                                   |
| udp         | udp(ip(0.0.0.0) port(514))                        |
| tcp         | tcp(ip(192.168.1.1) port(514) maxconnections(10)) |

#### 5.4) Objekt Ziel ("destination")

Definiert, was mit einer Meldung passieren soll (Speichern auf Datei, Übergeben an ein Skript/Programm, Weiterleiten an einen anderen Rechner).

| Ziel    | Beispiel                                                      |
|---------|---------------------------------------------------------------|
| file    | file("/var/log/syslog" owner("root") group("adm") perm(0755)) |
| udp     | udp("192.168.1.12") destport "514" spoof_source no)           |
| tcp     | tcp("192.168.1.12") destport "514" spoof_source no)           |
| program | program("/usr/local/bin/log2mysql.sh")                        |

#### 5.5) Objekt Bedingung ("filter")

Filtert über Bedingungen aus den eintreffenden Meldungen die relevanten heraus. Filter sind verschachtelbar, d.h. ein Filter kann als Element andere Filter verwenden (per "filter(FILTERNAME)").

Kombination folgender Funktionen:

```
+ facility(FAC1, ...)      # Quell-Kategorie
+ level(PRI01, ...)       # Quell-Priorität
+ program(REGEXP)         # Quell-Programm
+ host(REGEXP)A           # Quell-Rechner
+ netmask(IP/MASK)        # Quell-IP/Netzwerkmaske
+ match(REGEXP)           # Vergleich Meldungstext mit REGEX
+ filter(NAME)            # Subfilter-Name (Schachtelung von Filtern)
```

Kombination obiger Funktionen per:

```
+ Klammerung:             ( )
+ Logische Operatoren: and, or not
```

Reguläre Ausdrücke zum Matchen der Nachrichtentexte per "match" möglich:

```
+ . # EIN beliebiges Zeichen
+ [abc] # EIN Zeichen "a", "b" oder "c" (Zeichenklasse)
+ [a-c] # EIN Zeichen "a" ... "z" (Zeichenklasse)
+ [^abc] # EIN Zeichen AUSSER "a", "b" oder "c" (Zeichenklasse)
+ [^a-c] # EIN Zeichen AUSSER "a" ... "z" (Zeichenklasse)
+ C* # 0-N mal Zeichen C
+ C? # 0-1 mal Zeichen C
+ C+ # 1-N mal Zeichen C
```

Beispiele

```
filter f_cnews { level(err,crit) and facility(news); };
filter f_iptables { facility(kern) and match("IN=") and match("OUT="); };
filter f_messages { not facility(news,mail) and not filter(f_iptables); };
```

#### 5.6) Objekt Log ("log")

```
+ Quelle + Filter + Ziel zusammenfassen
+ Mehrere Quellen, Filter und Ziele erlaubt
+ Zusätzlich Optionen erlaubt (Flags)
+ Beispiele
log { source(src); filter(f_syslog); destination(syslog); };
log { source(src); filter(f_daemon); destination(daemon); };
log { source(src); filter(f_lpr); destination(lpr); };
log { source(src); filter(f_mail); destination(mail); };
log { source(src); filter(f_user); destination(user); };
log { source(src); filter(f_all); destination(sql); };
```

#### 5.7) Templates und Makros

Templates definieren Ausgabeformate von Meldungen, es sind darin Makros der Form \$XXX erlaubt. Die Option "template\_escape" sorgt dafür, dass Anführungszeichen maskiert werden.

| Makro | Bedeutung |
|-------|-----------|
|       |           |

Mai 27, 10 3:00

**syslog-HOWTO.txt**

Page 7/7

|            |                                                                                                                 |
|------------|-----------------------------------------------------------------------------------------------------------------|
| \$FACILITY | Quelle: auth(priv),cron,daemon,ftp,kern,lpr,mail,mark,news<br>security(=auth, nicht!),syslog,user,uucp,local0-7 |
| \$PRIORITY | Priorität: debug,info,notice,warn(ing),err(or),crit,<br>alert,emerg/panic                                       |
| \$LEVEL    | "                                                                                                               |
| \$TAG      | Quelle+Priorität als 2-stellige Hexzahl                                                                         |
| \$DATE     | Datum im Standardformat                                                                                         |
| \$FULLDATE | Datum im Standardformat                                                                                         |
| \$ISODATE  | Datum im Standardformat                                                                                         |
| \$YEAR     | Jahr (4-stellig)                                                                                                |
| \$MONTH    | Monat (2-stellig)                                                                                               |
| \$DAY      | Tag (2-stellig)                                                                                                 |
| \$WEEKDAY  | Wochentag (3-stellig: Mon, Tue, ..., Sat, Sun)                                                                  |
| \$HOUR     | Stunde (2-stellig)                                                                                              |
| \$MIN      | Minute (2-stellig)                                                                                              |
| \$SEC      | Sekunde (2-stellig)                                                                                             |
| \$TZ       | Zeitzone (3-stellig)                                                                                            |
| \$TZOFFSET | Zeitzone-Differenz zu GMT (5-stellig)                                                                           |
| \$FULLHOST | Hostname (FQHN, mit Domain)                                                                                     |
| \$HOST     | Hostname (ohne Domain)                                                                                          |
| \$PROGRAM  | Programm von dem Meldung stammt                                                                                 |
| \$MESSAGE  | Eigentlicher Nachrichteninhalt (Text inkl. Programmname + PID)                                                  |
| \$MSG      | "                                                                                                               |
| \$MSGONLY  | Nur Nachrichtentext                                                                                             |

## Beispiele

```
template("[ $YEAR/$MONTH/$DAY $HOUR:$MIN:$SEC ] $PRIORITY $FACILITY $MESSAGE\n")

destination sql {
    program("/usr/local/sbin/log2mysql.sh"
    template("$HOUR $MIN $HOST $MSG")
    template_escape(yes)
};
```

## 5.8) Beispielskript "log2mysql.sh"

Beispiel für ein Programm "log2mysql.sh" zur Verarbeitung von Log-Meldungen (Ablage der Meldungen in einer MySQL-Datenbanktabelle):

```
#!/bin/sh
#-----
# log2mysql.sh
#-----
# destination sql {
#     program("/usr/local/sbin/log2mysql.sh"
#     template("$HOUR $MIN $HOST $MSG")
#     template_escape(yes)
# };
#-----
while read HOUR MIN HOST MSG
do
    echo "INSERT INTO logbook(hour,minute,host,message)
        VALUES ('$HOUR', '$MIN', '$HOST', '$MSG') |"
    mysql --user=USER --password=GEHEIM logs
done
```

## 6) Links

```
* http://syslog-win32.sourceforge.net Syslog-Implementierung für Window
s
* http://www.balabit.com/network-security/syslog-ng/ syslog-ng (Logging System)
* http://www.rsyslog.com/ rsyslog (alternatives Logprogramm)
)
* http://www.phplogcon.org/ phpLogCon (Webinterface zu Syslog)
) (Webinterface zu Syslog)
* http://www.syslog.org/ Syslog-Wiki und Forum
* http://content.hccfl.edu/pollock/AUnix2/Logging.htm Logging, Log File Rotation, and S
yslog Tutorial
* http://www.loganalysis.org/sections/syslog/syslog-replacements/ Syslog Replacements
* http://www.loganalysis.org/ Loganalysis (The System Log: Logg
ing News and Information)
```