HOWTO zum Daemon "syslog", "syslog-ng" und "rsyslog" (System Logging)

$Id: syslog-HOWTO.txt,v 1.30 2019/11/26 19:37:07 tsbirn Exp $

Diese Dokument beschreibt die Eigenschaften und MÃ¶glichkeiten der
System-Logging-Daemons "syslog", "syslog-ng" und "rsyslog" von Linux-Systemen.
FÃ¼r Informationen zur Logrotation "logrotate" siehe --> logrotate-HOWTO.txt

---

INHALTSVERZEICHNIS

---

1) Einleitung

In jedem Betriebssystem laufen im Hintergrund viele sogenannte "Daemonen"
(Dienste/Services), die nicht an ein Terminal gebunden sind. Ihre Meldungen
(Fehler, Warnung, Information, Debugging) mÃ¼ssen aber trotzdem ausgegeben
oder aufgezeichnet werden, um sie spÃ¤ter auswerten oder darin Fehler suchen
zu kÃ¶nnen. DafÃ¼r gibt es unter Linux einen zentralen Dienst namens "syslog",
an den die Daemonen ihre Meldungen schicken kÃ¶nnen.

Typische Anwendungen des syslog-Daemons sind:

  * Protokollieren von Ereignissen und AblÃ¤ufen
  * Monitoring von Systemen
  * Netzwerkweite Integration vieler Log-Quellen in zentrales Repository

Der DÃ¤mon "/usr/bin/syslogd" erlaubt das Aufzeichnen/Protokollieren der
Meldungen vom Kernel oder anderen Programmen in Dateien ("Logging"). So kÃ¶nnen
System-Fehler oder sonstige Ereignisse aufgezeichnet und spÃ¤ter analysiert
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/Ursprungsdienst) und einer PRIORITÃM-^DT (Priority/Severity/
Dringlichkeit) klassifiziert. Weiterhin enthÃ¤lt Sie den NAMEN des
Sendeprogramms und einen frei wÃ¤hlbaren TEXT. Der NAME wird auch TAG (String)
genannt. Mit ihm identifiziert sich der sendende Daemon oder die sendende
Anwendung (wird zum PrÃ¤fix vor dem Meldungs-TEXT).

  * Kategorie: kern, mail, authpriv, cron, daemon, local0-9, lpr, user
  * PrioritÃ¤t: emerg, alert, crit, error, warning, notice, info, debug

Auf Basis dieser Informationen (und evtl. anderer) steuert "syslog" die weitere
Verarbeitung der Meldungen Ã¼ber BEDINGUNGEN. Trifft auf eine Meldung eine der
Bedingung zu (es dÃ¼rfen beliebig viele zutreffen), dann wird eine dazugehÃ¶rige
AKTION ausgefÃ¼hrt:

  * In eine Datei schreiben (anhÃ¤ngen): /PFAD/ZUR/DATEI
    (Datei muss bereits existieren!)
  * In eine Datei schreiben (ohne Puffern): -/PFAD/ZUR/DATEI
    (Datei muss bereits existieren!)
  * Auf allen Terminals eines Users ausgeben: USERNAME

```
  * Auf allen Terminals ausgeben:                    *
  * Einem Kommando (ausfÃ¼hrbare Datei) Ã¼bergeben:   | KMDO
  * Ã¼ber das Netz an einen Rechner schicken:         @HOSTNAME
    (muss per Konfiguration Meldungen annehmen)

Logdateien liegen unter Linux unter "/var/log" bzw. Unterverzeichnissen darin
und habe hÃ¤ufig die Endung ".log" oder einen PrÃ¤fix "log." (falls sie
komprimiert sind auch die Endung ".log.gz". Typische 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             # DruckauftrÃ¤ge
  * /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 NamensauflÃ¶sung
  * /var/log/samba/log.smbd              # Samba Sharezugriffe
  * /var/log/syslog                      # Systemmeldungen
  * /var/log/user.log                    # Benutzermeldungen

Folgende Meldungen werden NICHT von "syslog" aufgezeichnet:

  * Kernel-Meldungen beim System-Boot landen in einem speziellen Kernel-Puffer
    (auflisten per "dmesg")
  * Lokale Terminal-Logins
    (auflisten durch "last", "lastb", "lastlog" und "faillog")
  * Von Benutzern abgesetzte Kommandos
    (auflisten durch "lastcomm" falls Benutzer-Accounting aktiv)


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"
------------------------
* Paket-Verlust mÃ¶glich (verbindungslose UDP-Ã¼bertragung ohne Ankunftsgarantie)
* Paket-Ã¼berflutung mÃ¶glich (DoS = Denial of Service)
* GefÃ¤lschte Pakete mÃ¶glich (keine Authentifizierung, alles auf Port 514 akzep.)
* Nachrichten unverschlÃ¼sselt Ã¼bertragen (liegen im Klartext vor)
* Unflexible Filter: Nur Kategorie (Quelle) + PrioritÃ¤t (Dringlichkeit)
* Kategorie + PrioritÃ¤t teilweise uneinheitlich verwendet
* UrsprÃ¼ngliche Quelle geht bei Weiterleitung (Relay) evtl. verloren
* Nur ein Logformat (fix)

LÃ¶sung: "syslog-NG" oder "rsyslog" (siehe weiter unten).


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

  /etc/syslog.conf

Alle Meldungen werden zusÃ¤tzlich standardmÃ¤Ã¼ig auf der Teext-Konsole "tty10"
ausgegeben (per "Strg-Alt-F10" erreichbar).

Jede Zeile in der Konfigurations-Datei enthÃ¤lt ein oder mehrere durch ";"
getrennte "KATEGORIE.PRIORITÃM-^DT"-Paare (Quelle + Dringlichkeit) der
aufzuzeichnenden Meldungen und dann durch TABs (keine Leerzeichen!) getrennt
die beim Auftreten einer derartigen Meldung auszufÃ¼hrende Aktion "ACTION" (z.B.
den Namen einer Datei fÃ¼r die Aufzeichnung der Meldungen).

  KATEGORIE.PRIORITÃM-^DT;...  TAB...   ACTION

Mehrere solche Paare mit der gleichen Aktion sind durch ";" zu trennen (kein
Leerzeichen dazwischen!). Mehrere Kategorien und/oder PrioritÃ¤ten mit der
gleichen Aktion sind durch "," zu trennen. Ein "=" vor einer PrioritÃ¤t verlangt
```

GENAU diese PrioritÃ¤t, ein "!" vor einer PrioritÃ¤t trifft auf alle NIEDRIGEREN PrioritÃ¤ten zu; ein "!=" vor einer PrioritÃ¤t trifft auf alle PrioritÃ¤ten AUSSER dieser zu. Kommentarzeilen werden wie Ã¼blich durch ein fÃ¼hrendes "#" gekennzeichnet, Leerzeilen sind ebenfalls erlaubt.

EinschrÃ¤nkungen bei Solaris (und alten syslog-Versionen):
* Nur TABS zur Trennung von KATEGORIE.PRIORITÃM-^DT und ACTION erlaubt
* "*" nur bei KATEGORIE erlaubt, nicht bei PRIORITÃM-^DT
  (Ersatz: "DEBUG" statt "*" verwenden)

4.1) Kategorie / Ursprungsdienst / Quelle (Facility)

Kann einer der folgenden Werte (oder "*" fÃ¼r alle) sein:

Wert	Nr	Bedeutung
kern	0	Kernel
user	1	Meldung von Benutzern
mail	2	Mailsystem
daemon	3	Daemon allgemein
auth	4	Anmeldung/Authentifizierung (frÃ¼her "security")
syslog	5	Meldung durch syslog selbst
lpr	6	Drucksystem
news	7	Newsgruppensystem
uucp	8	UUCP-Meldung (Unix to Unix Copy)
cron	9	cron-Daemon (clock)
authpriv	10	Anmeldung/kritische Sicherheitsmeldungen
ftp	11	FTP-Server
ntp	12	NTP-Subsystem
audit	13	?
console	14	?
cron2	15	cron-Daemon
local0..7	16-23	Frei verwendbare Meldungen (eigene Programme/Dienste)
mark		Markierung durch syslog selbst (Zeitstempel/timestamp)
*		Alle Kategorien

4.2) PrioritÃ¤t / Dringlichkeit (Priority/Severity)

Kann einen der folgenden Werte (oder "*" fÃ¼r alle) annehmen, die Dringlichkeit steigt dabei von oben nach unten. Jede PrioritÃ¤t umfasst zugleich die HÃM-^VHEREN PrioritÃ¤ten mit (d.h. "err" umfasst auch "crit", "alert", "emerg"):

Wert	Nr	Bedeutung	
debug	7	Debugmeldung	
info	6	Information	
notice	5	Mitteilung	
warn/warning	4	Warnung	zunehmende
err/error	3	Fehler	Dringlichkeit!
crit	2	Kritischer Fehler	
alert	1	Alarm	
emerg/panic	0	Notmeldung (Kernel, panic VERALTET)	v
*		Alle PrioritÃ¤ten (analog "debug")	
none		Kategorie NICHT aufzeichnen	
=info		Meldungen GENAU dieser PrioritÃ¤t	
!err		Meldungen KLEINERER PrioritÃ¤t	
!=alert		Meldungen AUSSER PrioritÃ¤t "alert"	

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 zunÃ¤chst im Speicher gepuffert. StÃ¼rzt 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 anhÃ¤ngen (gepuffert = asynchron, Pfadname)
-FILE	An Datei FILE anhÃ¤ngen (ungepuffert = synchron, Pfadname)

```
   | | CMD     | An Kommando "CMD" auf Stdin Ã¼bergeben (Pfadname)          |
   | @HOST     | An Rechner HOST schicken                                  |
   | USER,...  | An Benutzer USER,... schicken (auf seine Terminals)       |
   | *         | An alle Benutzer schicken (auf alle Terminals)            |
   +-----------+-----------------------------------------------------------+
```

4.4) Standard-Verzeichnis fÃ¼r Logdateien

Das Standard-Verzeichnis fÃ¼r die Logging-Dateien lautet:

```
  /var/log
```

Die Log-Dateien darin mÃ¼ssen regelmÃ¤Ã-^_ig vom Systemverwalter kontrolliert und gekÃ¼rzt werden, um Systemprobleme festzustellen und das Dateisystem nicht mit Log-Meldungen "vollzumÃ¼llen". Alternativ Ã¼bernimmt auch der Daemon "logrotate" diese Aufgabe (siehe --> logrotate-HOWTO.txt).

4.5) Beispiele fÃ¼r EintrÃ¤ge 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 ttyp10
  *.crit          | lpr -P loglp          # Kritische Meldungen drucken
  *.emerg         *                        # Emergency an alle Terminals
  *.*             -/var/log/allmessages    # Alle Meldungen aufzeichnen
  *.debug         -/var/log/allmessages    #   (analog)
  *.=info         /var/log/info.log        # NUR Info-Meldungen
  *.!=alert       /var/log/noalert.log     # Alle AUSSER Alert-Meldungen
  *.err           /var/log/err+above.log   # Alle GROESSER+GLEICH Error-Meldungen
  *.=err          /var/log/err+above.log   # NUR Error-Meldungen
  *.!err          /var/log/belowerr.log    # Alle KLEINER Error-Meldungen
  *.!=err         /var/log/noalert.log     # Alle AUSSER Error-Meldungen
```

4.6) Ablauf einer Konfiguration

Um Ã-^Dnderungen am Log-Verhalten durchzufÃ¼hren, mÃ¼ssen folgende Schritte in der angegebenen Reihenfolge durchgefÃ¼hrt werden:

1) Konfigurationsdatei 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) Logging-DÃ¤mon mitteilen, dass er seine Konfigurationsdatei(en) neu lesen soll. Dazu ist seine Prozessnummer zu ermitteln und dem Prozess das Signal SIGHUP (1 = Hangup) zu schicken.

```
  ps ax | grep syslogd        # --> PID
  ps ax | grep syslog-ng      # --> PID
  ps ax | grep rsyslogd       # --> PID
  kill -1 PID                 # per PID
  kill -HUP PID               # per PID
  kill -SIGHUP PID            # per PID
  killall -HUP syslogd        # per Name
  killall -HUP syslog-ng      # per Name
  killall -HUP rsyslogd       # per Name
  service syslog reload       # Dienst-Steuerung
  /etc/init.d/syslog reload   # Dienst-Steuerung
  rcsyslog reload             # rc... nur bei SuSE-Linux
```

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

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

3c) Alternativ kann das System auch heruntergefahren und wieder neu gebootet werden, da der syslog-DÃ¤mon bei jedem Systemstart hochfÃ¤hrt und seine Konfigurationsdatei liest:

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

4) Eine stÃ¤ndig 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 ...        # -f=follow (mehr als eine Datei erlaubt)
       less /var/log/allmessages ...           # mit "f/F" auch in Follow-Modus schalten

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

        logger -p KATEGORIE.PRIORITÃM-^DT -t TEST "TEXT"         # Marke TEST (Std: logger)
        logger "Hallo hier bin ich"                          # user.notice + logger
        logger -p kern.emerg "Nur ein Scherz..."             # Priority kern.emerg
        logger -p kern.emerg -t TEST "Nur ein Test..."       # Priority kern.emerg
        logger -i -p kern.emerg -t TEST "Nur ein Test..."    # PID loggen


5) Syslog-NG (Next Generation, Balabit)
---------------------------------------
* Verbesserungen gegenÃ¼ber "syslog"
  + NetzwerkfÃ¤higkeiten wesentlich erweitert
    - Auch per TCP (gesicherte Ã¼bertragung, verschlÃ¼sselte Tunnels)
  + Filter wesentlich erweitert
  + Logformat Ã¼ber "Templates" definierbar
  + Meldungsteile in Makros --> Automatismen
  + Feingranulare Logdateien
  + Innerhalb "file"-Destinations werden Makros expandiert
    --> automatische Host/Datums-Hierarchie von Verz./Dateien mÃ¶glich

* Ab Version 3.0
  + OSE (Open Source Edition)
    - SSL/TLS-VerschlÃ¼sselung der Kommunikation
    - SSPMF-Support (Syslog Standard Protocol and Message Format)
    - Trennung in Name+Wert-Paare
    - Makros
    - Log-Meldungsteile umschreiben/modifizieren
    - Ablage in Datenbanken mÃ¶glich (MySQL, MS-SQL, Oracle, PostgreSQL, SQLite)
  + PE (Premium Edition)
    - Kommerzielle Variante von Balabit (kostenpflichtig)
    - Sichere Ablage auf Platte (verschlÃ¼sselt, signiert)
    - BinÃ¤re und komprimierte Ablage auf Platte
    - Zeitstempel hochgenau und von externen Zeitquellen
    - Syslog-Agent fÃ¼r Windows
    - Windows-GUI fÃ¼r Nachrichtenkonfiguration
    - Nachrichten-Puffer auf Festplatte falls Logserver nicht erreichbar
    - HochlastfÃ¤hig
    - Viele Plattformen (Linux, Solaris, Windows, Open/Free/NetBSD, IBM AIX, HP-UX)

* Ablauf-Diagramm des syslog-ng fÃ¼r die Verarbeitung einer Nachricht:

           Lokal Netzwerk <--+
               | |           |
               v v           |
             Source          |
               |             | Netzwerk
               v             |
             Filter          |
               |             |
               v             |
           Destination ------+
               |
               v
              File

* Konfigurationsdateien (entweder A oder B):
    /etc/syslog-ng/syslog-ng.conf   # A) FÃ¼r alle Dienste gemeinsam
    /etc/syslog-ng/syslog-ng.d/*    # B) Pro Dienst eine Datei
    /etc/sysconfig/syslog           # SuSE: Einstellungen fÃ¼r syslog-ng

* Kommandozeilen-Optionen:

    +----------------+-----------------------------------------------+
    | Option         | Beschreibung                                  |
    +----------------+-----------------------------------------------+
    | -d             | Debug Modus (kein Daemon)                     |
    | -F             | Im Vordergrund laufen (kein Daemon)           |
    | -s             | Nur Syntax-PrÃ¼fung durchfÃ¼hren (kein Daemon) |
    | -v             | Verbose Modus (kein Daemon)                   |
    | -V             | Version ausgeben                              |
    | -f FILE        | Konfigurationsdatei (mehrfach erlaubt)        |
    | --cfgfile=FILE |   (STD: /etc/syslog-ng/syslog-ng.conf)        |
    | -p FILE        | Datei fÃ¼r PID                                 |
    | --pidfile=FILE |   (STD: /var/run/syslog-ng.pid)               |
```

-C DIR	Chroot Verz.
-u UID	Prozess-Benutzer
-g GID	Prozess-Gruppe
-a SOCKET	Weitere Log-Socket analog /dev/log (Suse!)

5.1) Objekte

Syslog-NG basiert auf einem "Baukasten" bestehend aus benannten "Objekten", die erst bei der VerknÃ¼pfung zu einem "Log-Objekt" aktiviert werden (auf Klammerung jedes Objekts durch "{...}" und Abschluss durch ";" achten!).

Objekttyp	Definitionssyntax
Option	options { OPTION1(PARAMS); ... };
Quelle/Source	source NAME { DRIVER(PARAMS) };
Ziel/Destination	destination NAME { DRIVER(PARAMS); ... };
Filter	filter NAME { EXPRESSION; ... };
Log/Aktion	log { source(NAME); ...; filter(NAME); ...; destination(NAME); ...; flags(FLAG; ...); };

5.2) Globale Objekt-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 anhÃ¤ngen
keep_hostnames(Y/N)	Umschreiben des Hostnamen aktivieren
check_hostnames(Y/N)	Hostnamen auf gÃ¼ltige Buchstaben prÃ¼fen
use_dns(Y/N)	DNS zur HostnamenauflÃ¶s. nutzen (blockiert!)
dns_cache(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)	VollstÃ¤ndigen Hostnamen (FQHN) verwenden
stats(N)	Anz. Sek zwischen Statistiken

5.3) Objekt-Quelle ("source")

Definiert, aus welcher Quelle Meldungen stamme (Socket, eigene interne Dienste, Datei, Pipe, Terminal, TCP/UDP-Netzwerk-Port).

Quelle	Typ	Beispiel
unix-stream	Socket	unix-stream("/dev/log")
internal	Dienst	internal() WICHTIG!
file	Datei	file("/proc/kmsg" log_prefix("kernel: "))
pipe/fifo	Pipe	pipe("/dev/xconsole")
usertty	Terminal	usertty("root")
udp	Netzwerk	udp(ip(0.0.0.0) port(514))
tcp	Netzwerk	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, ÃM-^\bergeben an ein Skript/Programm, Weiterleiten an einen anderen Rechner).

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

5.5) Objekt-Bedingung ("filter")

Filtert Ã¼ber eine logische Kombination von Bedingungen per "and", "or", "not"
und Klammerung "(..")" aus den eintreffenden Meldungen die passenden heraus.
Filter sind verschachtelbar, d.h. ein Filter kann als Element andere Filter
verwenden (per "filter(FILTERNAME)").

Folgende Filter-Elemente gibt es:

Filter	Bedeutung
facility(FAC1, ...)	Quell-Kategorie(n)
level(P1, ...)	Quell-PrioritÃ¤t(en) (Liste)
level(P1 .. Pn)	Quell-PrioritÃ¤t(en) (Bereich von .. bis)
priority(P1, ...)	(analog)
priority(P1 .. Pn)	(analog)
program(REGEX)	Quell-Programm (per REGEX-Vergleich)
host(REGEX)	Quell-Rechner (per REGEX-Vergleich)
netmask(IP/MASK)	Quell-IP/Netzwerkmaske
match(REGEX)	Vergleich Meldungstext per REGEX
filter(NAME)	Subfilter-Name (Schachtelung von Filtern)

Bei "priority(...) oder level(...) ist eine durch "," (Komma) getrennte Liste
von EinzelprioritÃ¤ten oder ein Bereiche der Form P1..Pn (zwei Punkte) mÃ¶glich.

Logische Kombination der Filter-Elemente erfolgt per:

Komb.	Bedeutung
and	Logischer UND-Operator
or	Logischer ODER-Operator
not	Logischer NICHT-Operator
(...)	Klammerung

RegulÃ¤re AusdrÃ¼cke zum Matchen der Nachrichtentexte in "match":

Regex	Bedeutung
^	Meldungsanfang
$	Meldungsende
.	EIN beliebiges Zeichen
[abc]	EIN Zeichen "a", "b" oder "c" (Zeichenklasse)
[a-c]	EIN Zeichen "a" ... "c" (Zeichenklasse)
[^abc]	EIN Zeichen AUSSER "a", "b" oder "c" (Zeichenklasse)
[^a-c]	EIN Zeichen AUSSER "a" ... "c" (Zeichenklasse)
R1|R2	Regexp R1 ODER Regexp R2
C*	0-N mal Zeichen C
C?	0-1 mal Zeichen C
C+	1-N mal Zeichen C
\C	Metazeichen C ist normales Zeichen

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")

Fasst Quelle + Filter + Ziel zusammen. Meldungen aus einer Quelle, die den
Filter erfÃ¼llen werden dem Ziel Ã¼bergeben.
* Mehrere Quellen, Filter und Ziele erlaubt
* ZusÃ¤tzlich 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 dafÃ¼r, dass
AnfÃ¼hrungszeichen maskiert werden.
```

Makro	Bedeutung
$FACILITY	Quelle: auth(priv),cron,daemon,ftp,kern,lpr,mail,mark,news security(=auth, nicht!),syslog,user,uucp,local0-7
$PRIORITY $LEVEL	PrioritÃ¤t: debug,info,notice,warn(ing),err(or),crit, alert,emerg/panic
$TAG	Quelle+PrioritÃ¤t als 2-stellige Hexzahl
$DATE	Datum im Standardformat
$FULLDATE	Datum im Standardformat
$ISODATE	Datum im Format YYYY-mm-dd HH:MM:SS
$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	Zeitzonen-Differenz zu GMT (5-stellig)
$FULLHOST	Hostname (FQHN, mit Domain)
$HOST	Hostname (ohne Domain)
$PROGRAM	Programm von dem Meldung stammt (Tag)
$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 fÃ¼r 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) rsyslog (Rainer Gerhards)
----------------------------

6.1) Eigenschaften
------------------
* Konsistente Erweiterung von "syslog" (alles bekannte bleibt)
* Flexiber konfigurierbar
* Filter allgemeiner Art mÃ¶glich
* TCP-fÃ¤hig (verlÃ¤ÃM-^_licher Weiterversand)
* GSSAPI/TLS-fÃ¤hig (SSL-VerschlÃ¼sselung beim Weiterversand)
```

```
* Multi-Threaded
* Lokale Pufferung der Nachrichten falls EmpfÃ¤nger noch nicht bereit
* Weg Ã¼ber Relays nachvollziehbar
* Datenbank-Anbindung mÃ¶glich (MySQL, PostgreSQL)
* ISO-8601-Zeitstempel mit Unterscheidung von Millisekunden und Zeitzoneninfo
* PHPLogCon (in DB gespeicherte Logdaten mit Web-Browser sichten)
* Templates mÃ¶glich (Ausgabeformat der Meldung anpassen)
  + Drei verschiedene Ausgabeformate fest verdrahtet
  + Default: traditionelles Ausgabeformat von "syslog"
      TraditionalFormatWithPRI
      $template TraditionalFormatWithPRI,"%PRItext%:%timegenerated% %HOSTNAME% %%syslogtag%%msg::::d
rop-last-lf%\n"
      $template MyFormat,"%programname% meldet am %timegenerated%: %msg::::drop-last-lf% mit Priorit
Ã¤t %%syskigseveruty%\n"
      *.* -/var/log/syslog;TraditionalFormatWithPRI


6.2) Konfiguration
------------------
  /etc/rsyslog.conf                    # Hauptdatei
  /etc/rsyslogd.d/*.conf               # Include
  /etc/rsyslogd.d/00_common.conf       # Include
  /etc/rsyslogd.d/01_mysql.conf        # Include
  /etc/rsyslogd.d/01_postgresql.conf   # Include

  $IncludeConfig


7) Links
--------
* http://syslog-win32.sourceforge.net                                Syslog-Implementierung fÃ¼r Windo
ws
* http://www.balabit.com/network-security/syslog-ng                  Syslog-NG (Next Generation)
* http://www.rsyslog.com                                             Rsyslog (Alternative von Rainer G
erhards)
* http://www.phplogcon.org                                           phpLogCon (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                                         Loganalysis (The System Log: Logg
ing News and Information)
* http://www.loganalysis.org/sections/syslog/syslog-replacements    Syslog Replacements
* http://www.wikidorf.de/reintechnisch/Inhalt/SyslogNGEinfuehrung   EinfÃ¼hrung in Syslog-NG
* http://home.datacomm.ch/prutishauser/texte/syslog-ng-de.txt       Syslog-NG Reference manual (deu)
* http://www.campin.net/syslog-ng/faq.html                          Syslog-NG FAQ
* http://www.campin.net/syslog-ng/syslog.html                       What is Syslog-NG?
* http://code.google.com/p/eventlog-to-syslog                       Windows Log to Syslog
* http://www.intersectalliance.com/projects                         Windows Log to Syslog
```