



## CCcam loggen mit dem KIWI Syslog Daemon

In dieser Anleitung will ich euch erklären, wie ihr mittels KIWI Syslog euren Trace/Debug der CCcam zu jeder Zeit überwachen könnt. Ihr müsst dazu nicht die CCcam mittels `-dv` im Telnet starten. Das Verhalten der CCcam wird im laufenden Betrieb abgefragt.

Voraussetzungen:

- CCcamServer
- Receiver (z.B. Dreambox) mit einem Syslog-Daemon (z.B. Gemini, sonstiges CVS)
- KIWI Syslog Daemon auf dem PC (Laden könnt ihr diesen in bekannten Quellen oder hier <http://www.solarwinds.com>)

### Überprüfen der Dreambox

Als erstes müssen wir schauen, ob ein Syslogd auf dem Receiver läuft. Benutzer vom Gemini, LT-Image, iCVS und OoZooN brauchen es nicht testen, die haben es auf jeden Fall. Dazu verbinden wir uns mittels Telnet auf unsere Box und geben folgendes ein *busybox*.

```
192.168.178.57 - PuTTY
root@dm8000:~# busybox
BusyBox v1.01 (2009.01.27-20:17+0000) multi-call binary

Usage: busybox [function] [arguments]...
or: [function] [arguments]...

BusyBox is a multi-call binary that combines many common Unix
utilities into a single executable. Most people will create a
link to busybox for each function they wish to use and BusyBox
will act like whatever it was invoked as!

Currently defined functions:
[, addgroup, adduser, ar, arping, ash, awk, basename, bunzip2,
busybox, bzip2, cat, chgrp, chmod, chown, chroot, chvt, clear,
cp, cpio, crond, crontab, cut, date, dc, dd, deallocvt, delgroup,
deluser, df, dirname, dmesg, dos2unix, du, dumpkmap, echo, egrep,
env, expr, false, fbset, fgrep, find, free, ftpget, getkey, getty,
grep, gunzip, gzip, hdparm, head, hexdump, hostname, httpd, id,
ifconfig, ifdown, ifup, insmod, ip, kill, killall, klogd, ln,
loadfont, loadkmap, logger, login, logname, logread, losetup,
ls, lsmmod, md5sum, mkdir, mkfifo, mknod, mkswap, mktemp, modprobe,
more, mount, mv, nc, netstat, nslookup, od, openvt, passwd, patch,
pidof, ping, pivot_root, printf, ps, pwd, readlink, renice, reset,
rm, rmdir, rmmmod, route, run-parts, rx, sed, seq, sh, shasum,
sleep, sort, start-stop-daemon, strings, stty, su, sulogin, swapoff,
swapon, sync, sysctl, syslogd, tail, tar, tee, telnet, telnetd,
test, time, top, touch, tr, traceroute, true, tty, udhcpc, umount,
uname, uniq, unix2dos, unzip, uptime, uudecode, uuencode, vi,
vlock, watch, wc, wget, which, who, whoami, xargs, yes, zcat

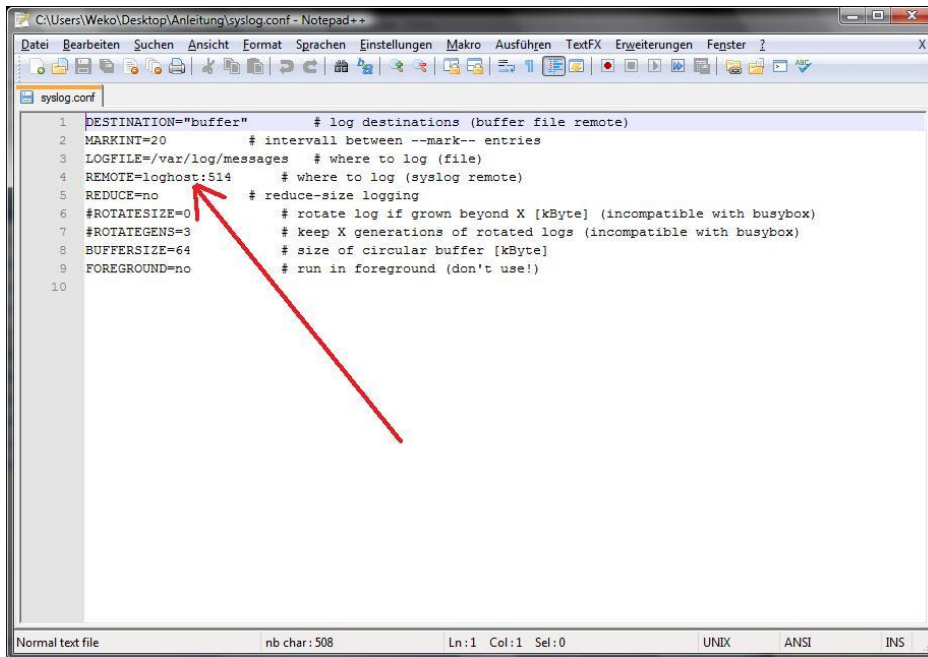
root@dm8000:~#
```

Dort müsste unter "Currently defined functions" **syslogd** stehen. Sollte das nicht der Fall sein, könnt ihr jetzt aufhören und in den Boards suchen, ob es eine Möglichkeit gibt Syslogd auf eure Box zu bekommen.

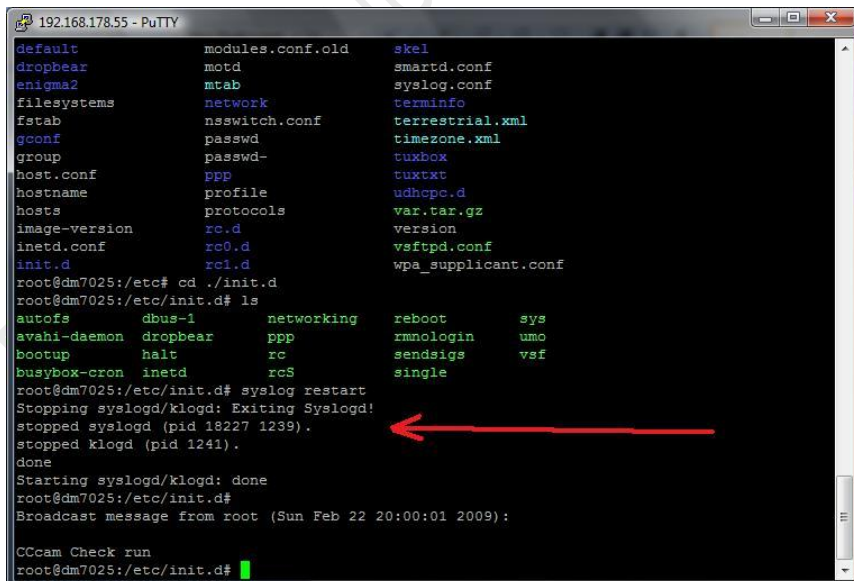
Steht bei euch Syslogd geht es jetzt weiter. Ich erkläre es für CVS-Image und fürs Gemini.

### Syslogd einrichten CVS-Images:

Verbindet euch mittels FTP auf eure Box und ladet euch die **syslog.conf**. Öffnet diese mit einem linuxfähigen Editor und editiert diese im Bereich **REMOTE** (roter Pfeil). Tragt dort die IP eures PC ein, wo wir später KiwiSysLogd installieren. Den Port lasst bitte auf 514.



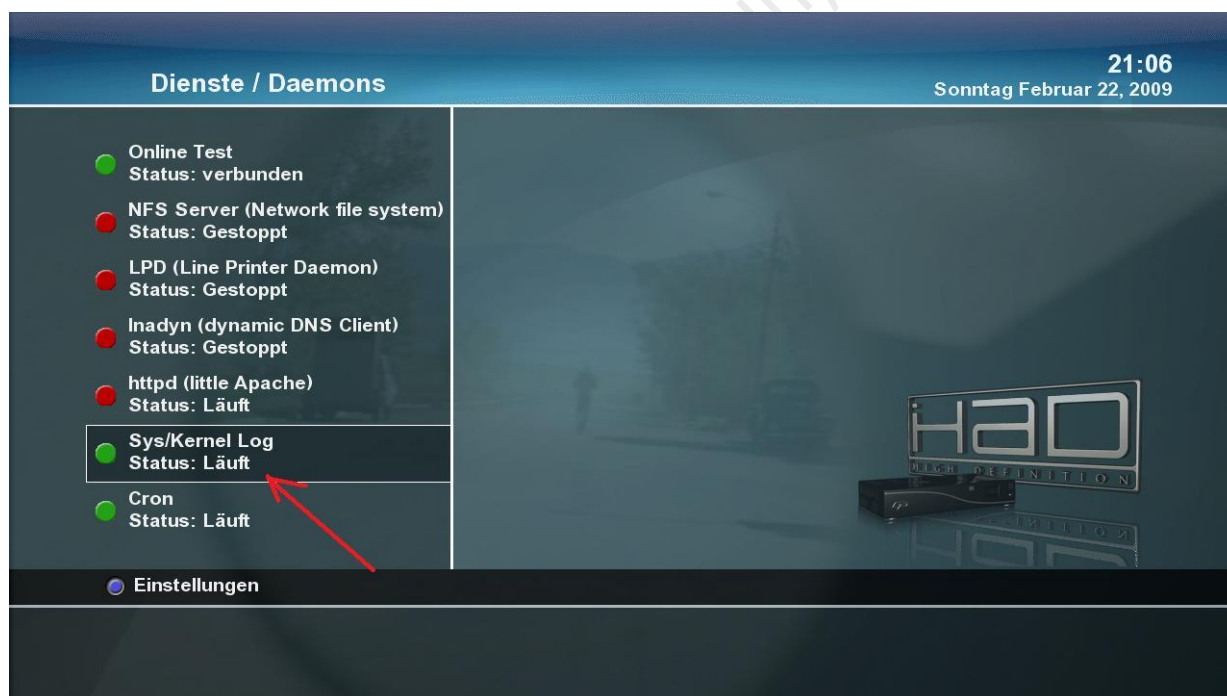
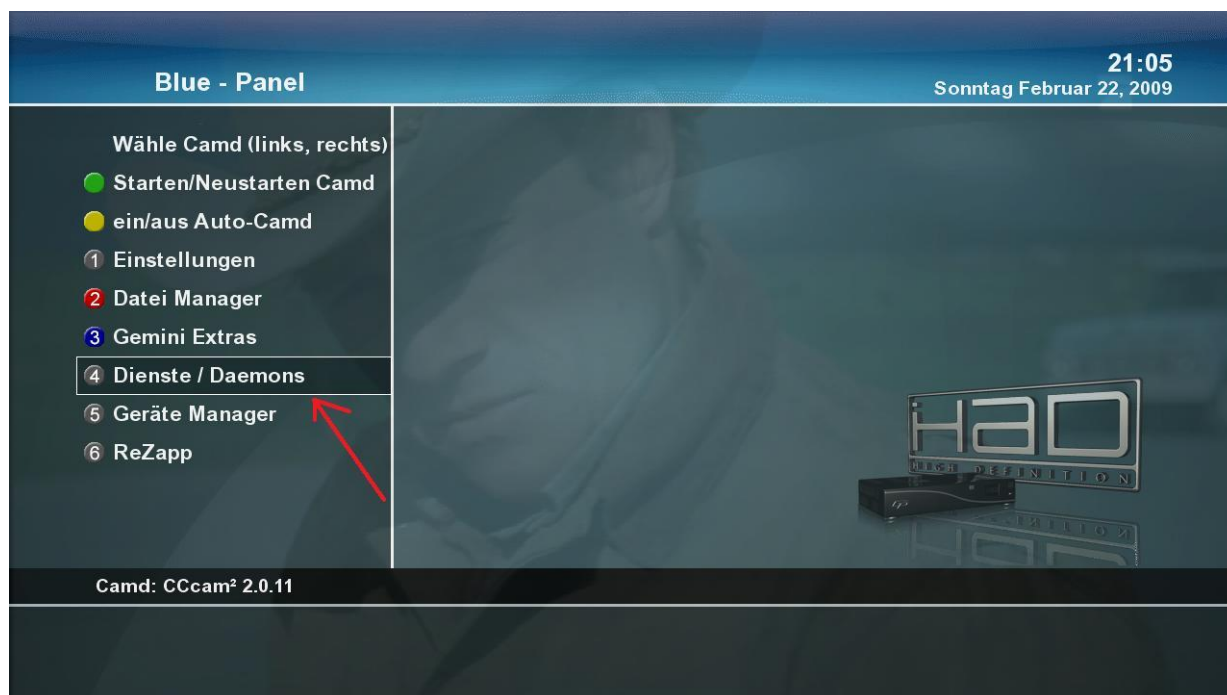
Ladet die Datei wieder auf eure Box und startet den Syslog Daemon neu, damit die Config neu eingelesen wird. Bei CVS verbindet ihr euch mittels Telnet nach **/etc/init.d** und gebt folgendes ein **syslog restart**. Jetzt sollte bei euch der Restart (Bild) durchlaufen.



Somit wären wir auf der Box soweit fertig. Weiter geht es mit dem Einrichten vom KIWI SYSLOG DAEMON am PC.

## Syslog einrichten Gemini-Image:

Drückt bei eurer Dream auf den blauen Button. Geht in die Dienste/Daemons und sucht nach Syslog.

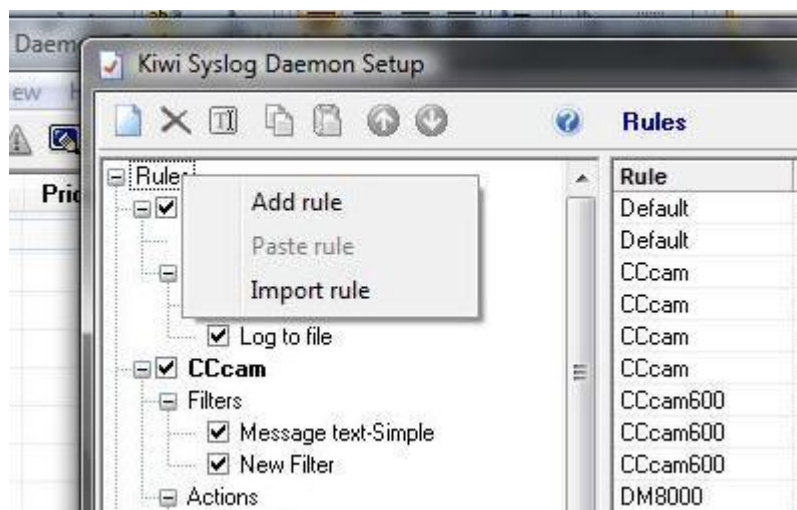


Markiert **Sys/Kernel Log** und drückt den blauen Button um in die Einstellungen zu kommen. Dort geht ihr auf **remote logging** und schaltet es an. Es kommen zwei neue Felder zum Vorschein. Bei Host tragt die IP eures PC, wo ihr KIWI SYSLOG Daemon laufen lassen wollt, ein. Beim Port bitte nichts ändern.

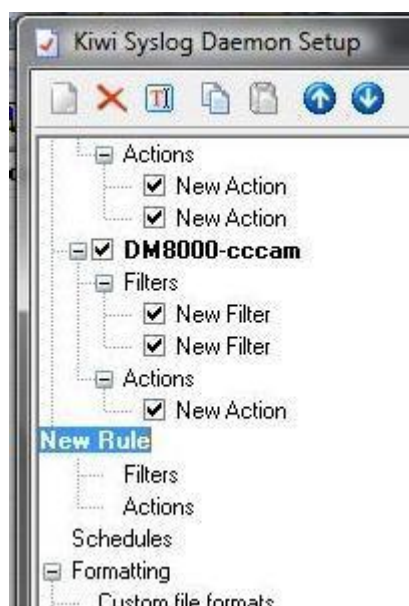




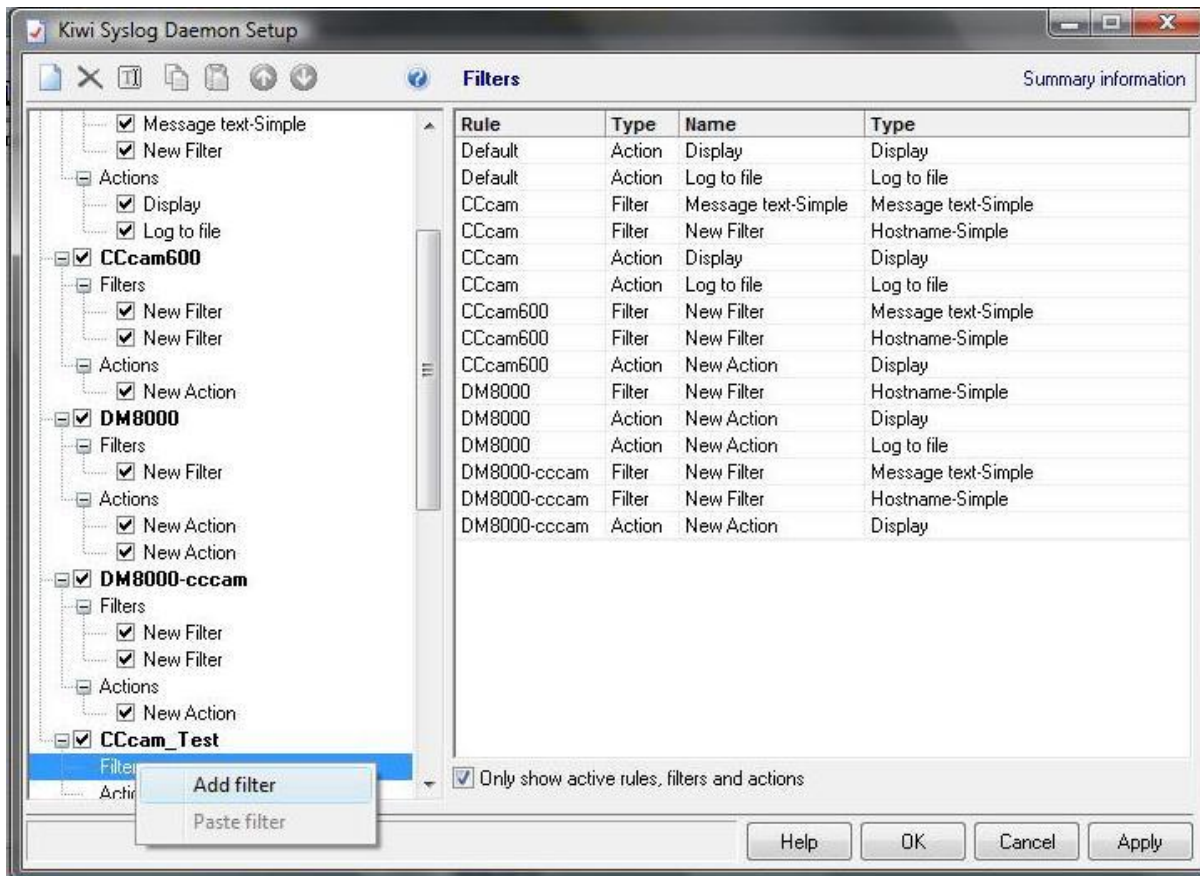
Klickt mit der rechten Maus auf **Rules** und wählt **Add rule** .



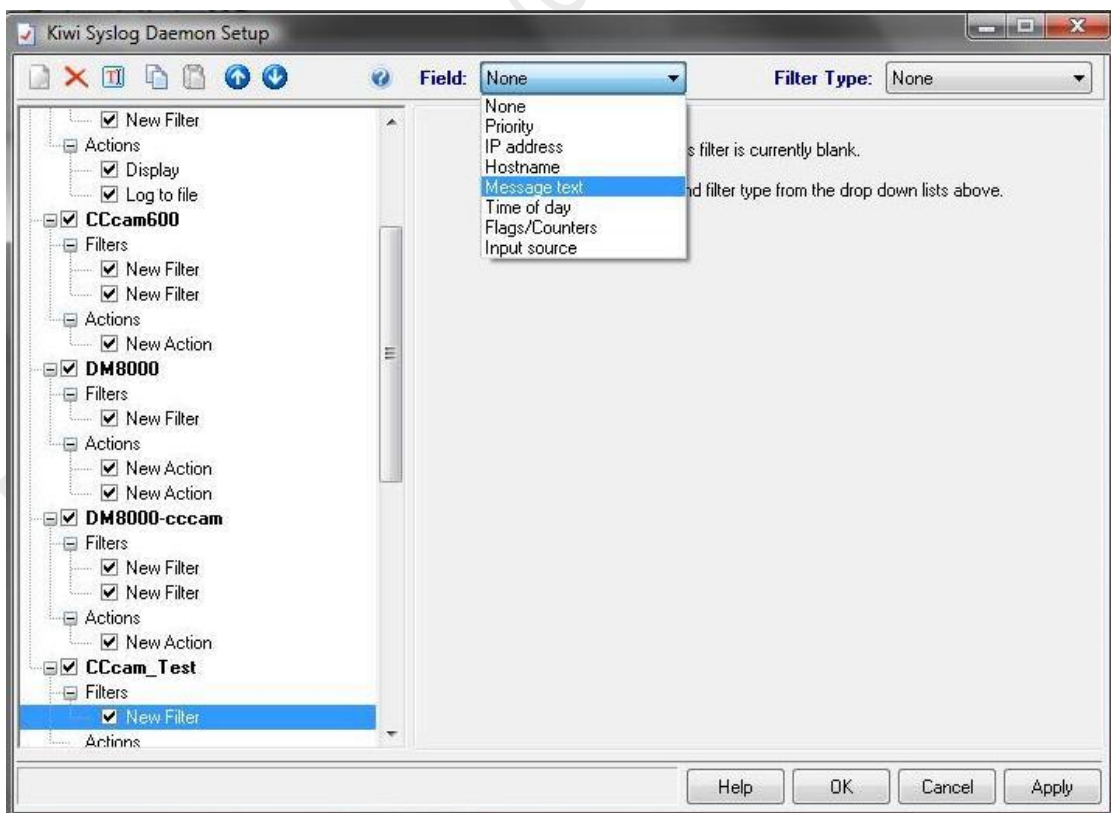
Dadurch wird eine neue Regel angelegt, benennt diese wie ihr wollt.

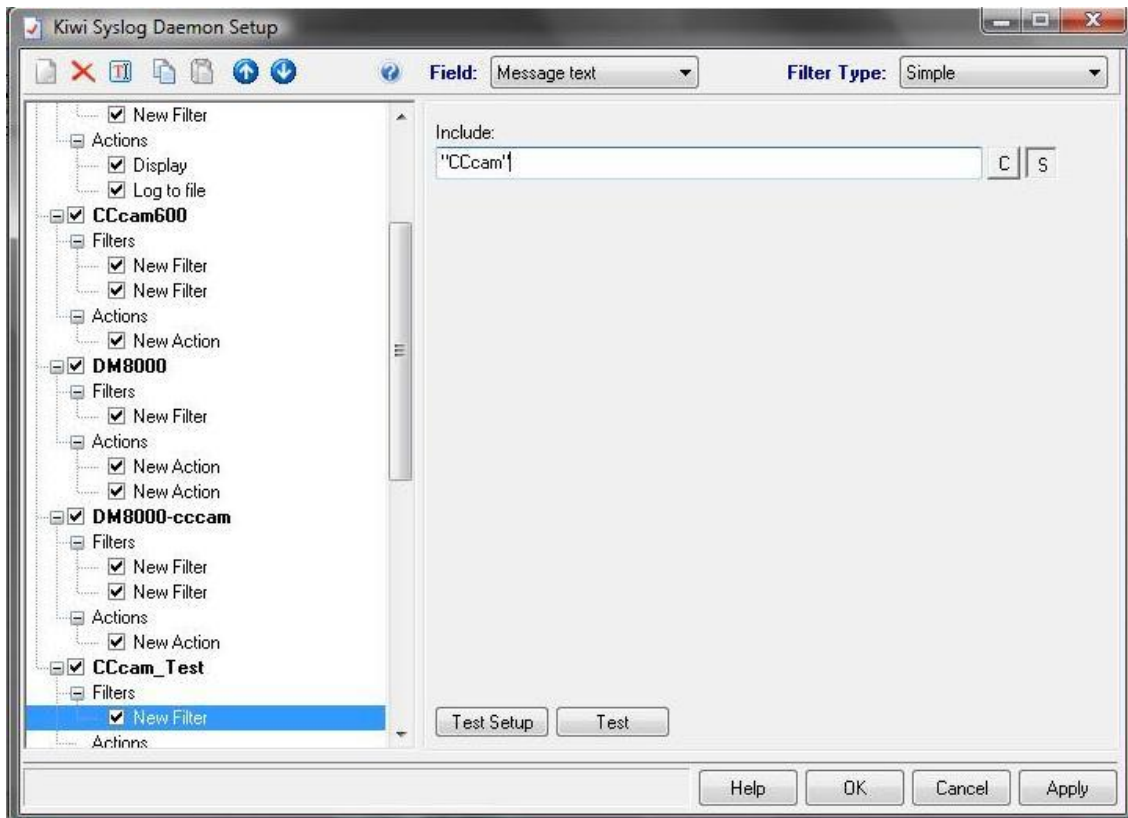


Jetzt kommen wir zum Einschränken der Ausgabe, dazu klickt ihr mit rechts auf Filters und fügt eine neue Regel dazu.

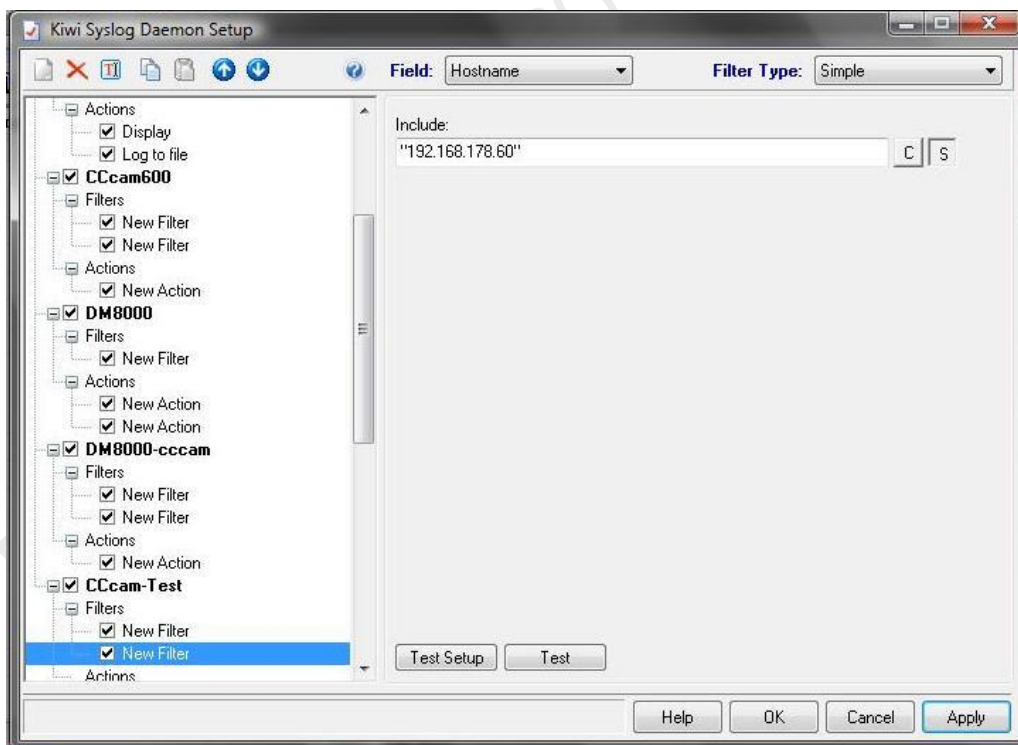


Als erstes wählen wir im **Field** den Punkt **Message text** tipp dort CCcam ein. Bitte das ganze in Anführungsstrichen, wie es im Bild zu sehen ist.

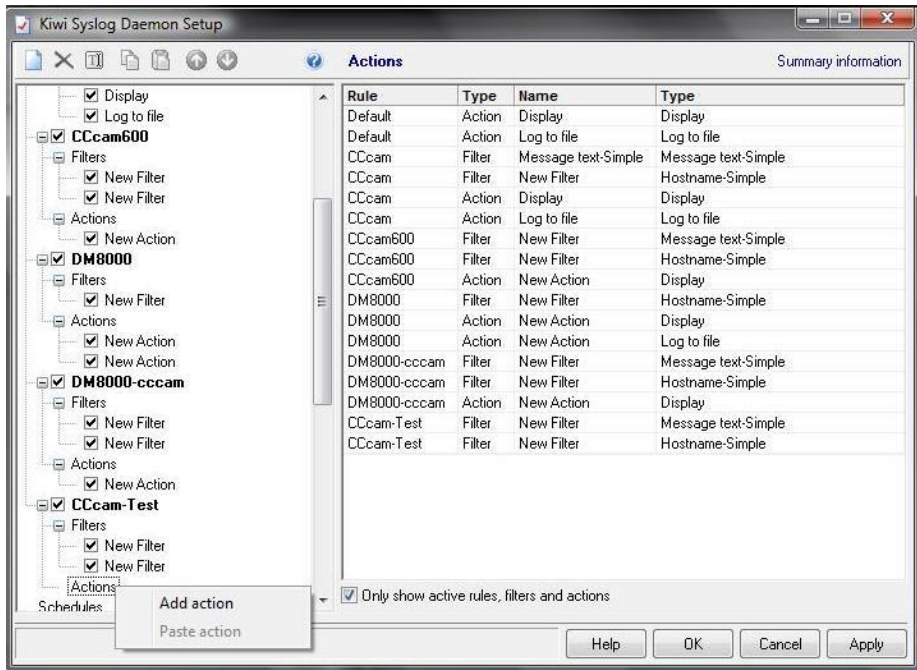




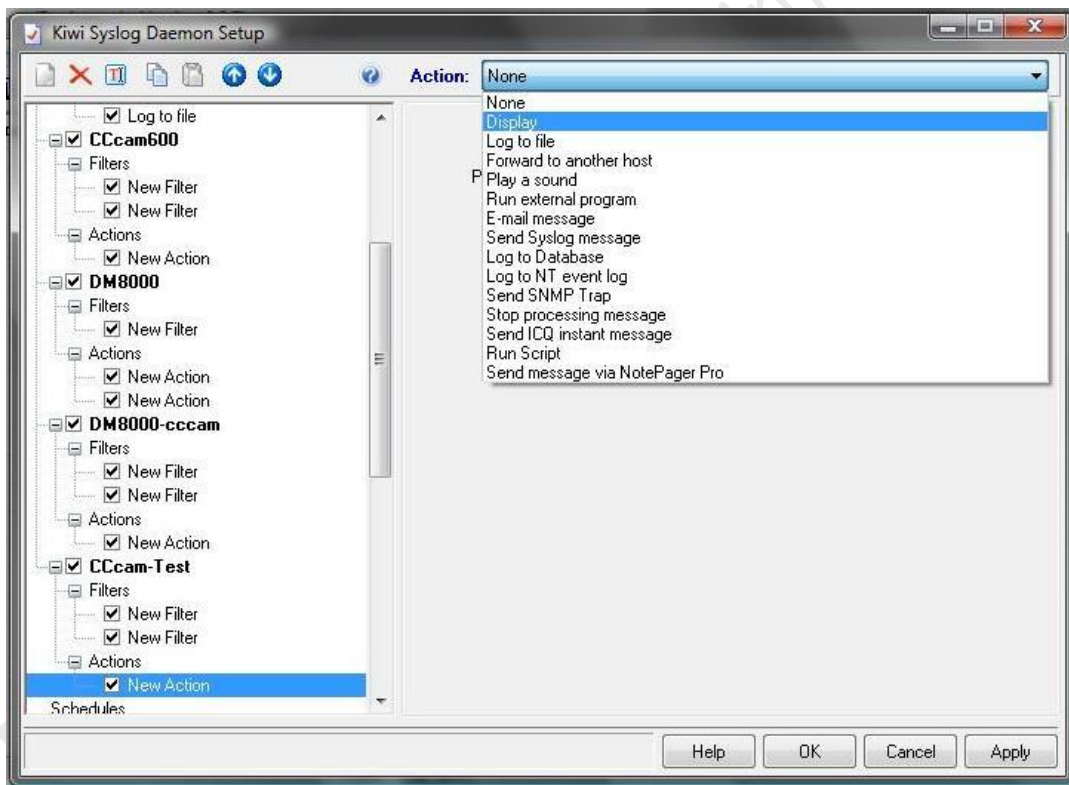
Jetzt kommt noch ein weiterer Filter hinzu und zwar wählen wir Hostname aus. Dort geben wir jetzt die IP unserer Dream ein. Wieder an die Anführungsstriche denken.



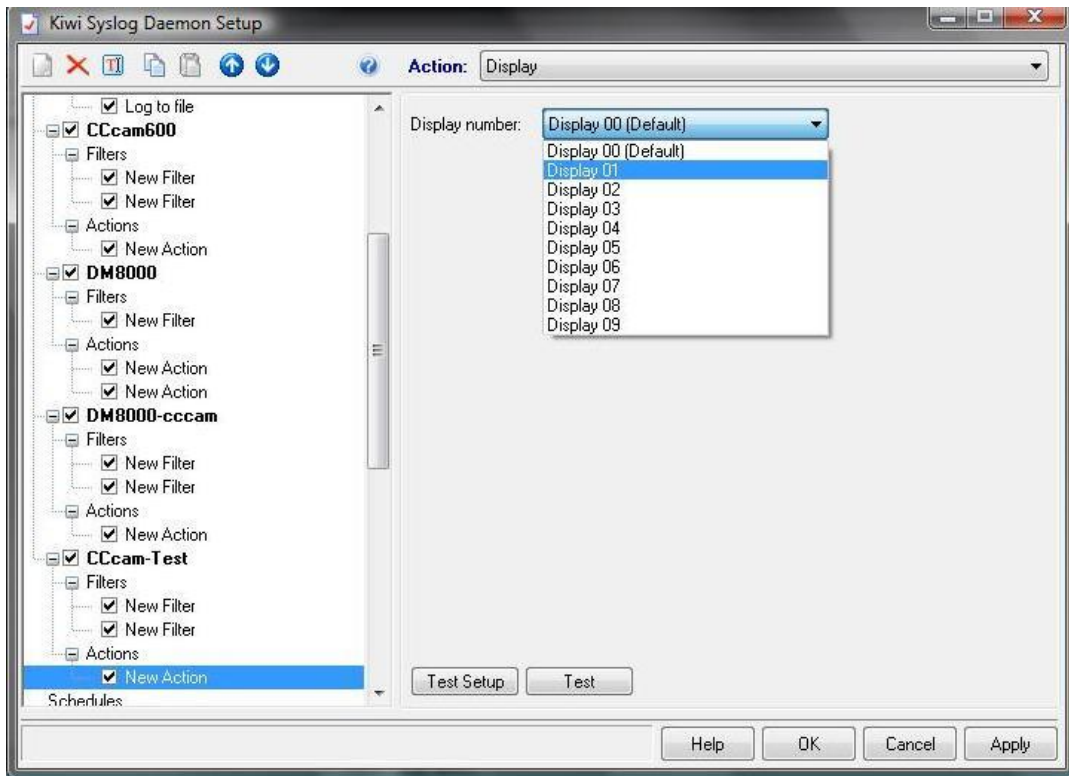
Als letztes legen wir noch fest, wo die Sachen ausgegeben werden sollen. Dazu fügt eine neue Aktion hinzu.



Wählt bei **Action** den Punkt **Display** aus. Jetzt habt ihr welche zur Auswahl, nehmt am besten das Display01.







Jetzt noch unten rechts einmal auf Apply und dann OK.

Als letztes nehmen wir uns die CCcam.cfg vor und aktivieren dort den Debug. Dazu fügen wir folgendes in die CCcam.cfg ein → DEBUG : yes

Schiebt die Config wieder auf die Box und startet eure CCcam neu.

**Achtung:** Bei Boxen mit wenig Speicher könnte es zu Problemen kommen. Dort wäre es gut, wenn ihr den Syslog nach dem Überprüfen wieder deaktiviert.

Bei manchen kann es vorkommen, dass der Port 514 am PC freigegeben werden muss.

Jetzt müsste der Trace/Log bei euch schön durchlaufen, wenn ihr Display01 auswählt.

Kiwi Syslog Daemon (Registered - Version 8.3.7)				
File Edit View Help				
Display 01				
Date	Time	Priority	Hostname	Message
02-22-2009	21:45:13	Daemon.Debug	192.168.178.55	CCcam: client [REDACTED] ecm request for handler 0x227 0x1722(0x0) sid 0x298 failed
02-22-2009	21:45:13	Daemon.Debug	192.168.178.55	CCcam: newcamd ecm [REDACTED]
02-22-2009	21:45:13	Daemon.Debug	192.168.178.55	CCcam: newcamd ecm -> [REDACTED]
02-22-2009	21:45:13	Daemon.Debug	192.168.178.55	CCcam: client [REDACTED] ecm request for handler 0x6cd5 0x1702(0x0) sid 0x3d ok
02-22-2009	21:45:13	Daemon.Debug	192.168.178.55	CCcam: local ecm <- card /dev/sci1 ok
02-22-2009	21:45:13	Daemon.Debug	192.168.178.55	CCcam: client [REDACTED] ecm request for handler 0x1f5 0x1722(0x0) sid 0x298 failed
02-22-2009	21:45:13	Daemon.Debug	192.168.178.55	CCcam: newcamd ecm [REDACTED]
02-22-2009	21:45:13	Daemon.Debug	192.168.178.55	CCcam: newcamd ecm -> [REDACTED]
02-22-2009	21:45:12	Daemon.Debug	192.168.178.55	CCcam: local ecm -> card /dev/sci1 0x1702(0x000) sid 0x03d
02-22-2009	21:45:12	Daemon.Debug	192.168.178.55	CCcam: client [REDACTED] ecm request for handler 0x6cd5 0x1702(0x0) sid 0x43 ok
02-22-2009	21:45:12	Daemon.Debug	192.168.178.55	CCcam: local ecm <- card /dev/sci1 ok
02-22-2009	21:45:11	Daemon.Debug	192.168.178.55	CCcam: local ecm -> card /dev/sci1 0x1702(0x000) sid 0x043
02-22-2009	21:45:11	Daemon.Debug	192.168.178.55	CCcam: client [REDACTED] ecm request for handler 0x6cd5 0x1702(0x0) sid 0x4461 ok
02-22-2009	21:45:11	Daemon.Debug	192.168.178.55	CCcam: local ecm <- card /dev/sci1 cw's from cache
02-22-2009	21:45:11	Daemon.Debug	192.168.178.55	CCcam: local ecm -> card /dev/sci1 0x1702(0x000) sid 0x4461
02-22-2009	21:45:11	Daemon.Debug	192.168.178.55	CCcam: client [REDACTED] ecm request for handler 0x6cd5 0x1702(0x0) sid 0x4461 ok
02-22-2009	21:45:11	Daemon.Debug	192.168.178.55	CCcam: local ecm <- card /dev/sci1 ok
02-22-2009	21:45:10	Daemon.Debug	192.168.178.55	CCcam: local ecm -> card /dev/sci1 0x1702(0x000) sid 0x4461
02-22-2009	21:45:10	Daemon.Debug	192.168.178.55	CCcam: client [REDACTED] ecm request for handler 0x6cd5 0x1702(0x0) sid 0x298 failed
02-22-2009	21:45:10	Daemon.Debug	192.168.178.55	CCcam: local ecm <- card /dev/sci1 error
02-22-2009	21:45:10	Daemon.Debug	192.168.178.55	CCcam: local ecm -> card /dev/sci1 0x1702(0x000) sid 0x298
100% 15K MPH				21:45 02-22-2009

Viel Spaß wünscht euch [CUUIRA](#) !

Wenn ihr Fragen habt oder was nicht eindeutig sein soll, findet ihr mich im CW-SHARE-Board unter <http://cw-share.com> .

Thx an devnull und cwshare fürs drüberlesen und korrigieren!

