

# Collectord für Synology Diskstation

## Generelles

Für meine Diskstation möchte ich den collectord kompilieren. Für die aktuellen Diskstation-Modelle wurden verschiedenen Prozessoren verwendet. Daher muss für jeden Prozessor der collector einmalig kompiliert werden.

Hier erst mal meine ersten Versuche den collectrod auf einer Diskstation zu kompilieren.

## Voraussetzungen zum kompilieren

### Hardware:

Eine Diskstation

### Software:

SSH-Client für den Computer

Zur Kompilierung ist eine SSH/Telnet Terminal Verbindung notwendig. Diese Terminal Verbindung muss erst auf der Diskstation über DSM aktiviert freigeschaltet werden. Da die SSH-Verbindung verschlüsselt ist sollte diese Vorgezogen werden. Bei Telnet wird das Passwort unverschlüsselt übertragen und sollte nur im eigenen Netz ohne Internet-Verbindung verwendet werden.

Für die SSH-Verbindung nehme ich TeraTerm. Auf meinen Diskstation läuft DSM 5.1. Das Vorgehen sollte auf älteren Versionen so, oder ähnlich funktionieren.

## IPKG Installation

Für die Installation gibt es von Synology ein [IPKG-Wiki](#).

Dieser Anleitung folgen.

Bei mir hat das auskommentieren der Pfadangaben in **/root/.profile** nicht geholfen. Statt die beiden Zeilen für den Pfad mit **#** auszukommentieren muss dieser nur noch erweitert werden. Am Ende noch **:/opt/bin:/opt/sbin** anhängen. Diese Änderung ist notwendig damit IPKG, und die andere Software, auch ohne den absoluten Pfad gestartet werden kann.

## Editieren mit vi

zum editieren einfach nur vi mit dem Dateinamen angeben.

```
vi /root/.profile
```

mit den Pfeiltasten zur Einfügeposition gehen und **i** drücken um in den Einfügemodus zu gelangen. Jetzt kann der PATH erweitert werden.

Nach der Erweiterung keine Eingabetaste drücken.

Zurück in den Befehlsmodus von vi kommt man mit der **Escape**-Taste.

Um die Änderungen zu speichern und vi ohne Nachfrage zu verlassen folgendes eingeben: `:wq!`

Bei Eingabefehlern einfach mit Escape zurück in den Befehlsmodus und **:quit!** eingeben und noch mal von vorne starten.

## Paketliste aktualisieren

```
ipkg update
```

## GCC und Bibliotheken installieren

```
ipkg install gcc
ipkg install git
ipkg install optware-devel
ipkg install boost-iostreams
ipkg install boost-dev
ipkg install boost-thread
ipkg install boost-system
ipkg install boost-date-time
ipkg install ncurses ncurses-dev
Aus https://code.google.com/p/google-authenticator/issues/detail?id=364: rm
/opt/arm-none-linux-gnueabi/lib/libdl.so
ln -s /lib/libdl.so.2 /opt/arm-none-linux-gnueabi/lib/libdl.so
```

## Quelltexte

Ich habe ein Fork von Danny-Baumanns <https://github.com/maniac103/Quelltext> erstellt. Vermutlich müssen kleine Änderungen am Import des MySQL-Connector gemacht werden. Falls es läuft kann das vielleicht von Danny in seinen Master-Branch eingefügt werden.

[collectord](#)

## CMake aus Quellcode installieren

Auf [CMake Download-Seite](#) könnte es eine neuere Version von Cmake geben. Dann einfach die Versionsnummern in den folgenden Befehlen anpassen:

```
cd /volume1/@tmp
wget http://www.cmake.org/files/v3.1/cmake-3.1.0-1-src.tar.bz2
tar jxf cmake-3.1.0-1-src.tar.bz2
./cmake-3.1.0-1.sh prep
./cmake-3.1.0-1.sh conf
./cmake-3.1.0-1.sh build
```

```
cd cmake-3.1.0/.build/  
bin/cpack -G STGZ  
Erstellt selbstentpackendes Archiv cmake-3.1.0-Linux-<ARCH>.sh  
./cmake-3.1.0-Linux-*.sh --prefix=/opt
```

- Lizenztext-Anzeige mit „q“ abbrechen und mit „y“ quittieren
- Do you want to include the subdirectory cmake-3.1.0-Linux-armv5tel?

Saying no will install in: „/opt“ [Yn]:n

Die Installation von cmake ist nun abgeschlossen.

## MariaDB Client Library for C Bibliotheken kompilieren

Leider habe ich keine Möglichkeit gefunden den die Datei über wget auf die Diskstation zu bekommen. Um an die **MariaDB Client Library for C** muss man sich erst auf <https://mariadb-com> registrieren. Jetzt kann man über >Produkt >Connector & Plugins den Sourcecode herunterladen. Diese am besten im public-share auf der Diskstation ablegen. Wenn dieser nicht vorhanden ist dann über DSM der Diskstation erstellen.

```
cp /volume1/public/mariadb_client-2.0.0-src.tar.gz /volume1/@tmp/ cd  
/volume1/@tmp  
wget  
http://cdn.mysql.com/Downloads/Connector-C++/mysql-connector-c++-1.1.5.tar.gz  
tar xzf mariadb_client-2.0.0-src.tar.gz  
cd mariadb_client-2.0.0-src  
mkdir build  
cd build  
cmake ..  
make all
```

## collectord bauen und installieren

```
cd /volume1/@tmp  
git clone https://github.com/ingof/ems-collector.git  
cd ems-collector/collector  
make  
cp -a collectord /opt/bin  
cd /root
```

Hier gibt es jetzt allerdings die letzte Hürde. Der GCC für die Diskstations ist zu alt und kennt die Option **-std=c++0x** nicht. Der GCC-Compiler unterstützt das erst ab Version 4.7.

Mir ist bisher keine Möglichkeit bekannt dieses Problem zu umgehen.

## aufräumen

```
rm -rf /volume1/@tmp/cmake-3.1* /volume1/@tmp/mysql-connector-*
```

Löschen des Git-Verzeichnisses „ems-collector“ ganz nach Gusto:  
`rm -rf /volume1/@tmp/ems-collector`

## Konfiguration collectord

identisch mit Raspberry

## Autostart collectord

..... in Arbeit .....

From:

<http://emswiki.thefischer.net/> -

Permanent link:

<http://emswiki.thefischer.net/doku.php?id=wiki:ems:dscollectord&rev=1424035102>



Last update: **2015/12/30 21:00**