

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:

IPKG

vi

GCC-Compiler

collectord Quelltext

Notwendige Bibliotheken

cmake

mySQL Connector C++

Leider sind meine Linux-Kenntnisse nicht allzu groß. Deswegen ist hier erst mein aktueller Stand der Kompilierung zu finden. Diese Anleitung soll auch für Anfänger funktionieren die nicht erst noch nach Grundlagen googlen müssen.

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 aber genauso funktionieren.

IPKG Installation

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

Dieser Anleitung bis zum editieren von /root/.profile folgen.

Statt den Pfad auszukommentieren sollte 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 Einfügeböschung 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 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 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](#)

Leider gibt es noch kein IPKG-Paket für die MySQL-Connector-C++ Bibliothek und diese muss auch kompiliert werden. Hier bitte Source Code auswählen und den Sourcecode herunterladen. Es ist auch möglich über den unteren Link den Sourcecode herunter zu laden ohne sich registrieren zu lassen.

[MySQL connector](#)

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.

Sourcecode auf die Diskstation kopieren

vermutlich gibt es auch eine einfacher Möglichkeit mit wget. Ich habe bisher den Quelltext über einen PC heruntergeladen und über Netzwerk auf die Diskstation in den public-Ordner kopiert. Wenn der nicht existiert muss erst ein Share public über DSM erstellt werden.

Da der collectord hinterher wohin /opt/bin gehört habe ich dort erst mal einen unterordner erstellt und dann die Quelltext von public in diesen Ornder kopiert.

```
cp /volume1/public/ /opt/bin/ems-collector-master
```

Diese Bibliothek ist im Moment mein aktuelles Problem für das ich bisher keine Lösung habe. Beim Quelltext sind Cmake Scripte vorhanden. Diese sind für das CrossCompile. Die Makefiles sind nicht vorhanden. Daher ist so kein einfaches kompilieren über make möglich.

Vermutlich könnte dass der entscheidende Link sein um die Bibliothek zu kompilieren:

```
ipkg install mysql5
... hier gibt es noch einige Warnungen, erst mal ignorieren
```

[Dynamische Bibliotheken kompilieren](#)

mySQL Connector Bibliotheken installieren

```
cd /volume1/@tmp wget
http://cdn.mysql.com/Downloads/Connector-C++/mysql-connector-c++-1.1.5.tar.gz
tar zxf mysql-connector-c++-1.1.5.tar.gz
```

```
cd mysql-connector-c++-1.1.5
mkdir build
cd build
cmake ..
```

- hier hakt es noch - undurchsichtige Fehlermeldungen

collectord bauen und installieren

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

aufräumen

```
rm -rf /volumel/@tmp/cmake-3.1* /volumel/@tmp/mysql-connector-*
Löschen des Git-Verzeichnisses „ems-collector“ ganz nach Gusto:
rm -rf /volumel/@tmp/ems-collector
```

Konfiguration collectord

identisch mit Raspberry

Autostart collectord

..... in Arbeit

From:
<https://emswiki.thefischer.net/> -

Permanent link:
<https://emswiki.thefischer.net/doku.php?id=wiki:ems:dscollectord&rev=1423771564>



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