

Von: Hans-Jürgen Barthen DL5DI [dl5di@gmx.de]
Datum: Samstag, 29. Mai 2010 13:03
Betreff: ircDDB Installationsinfos

Vorabinfo – frühes Pre-Release

Die Installation dieses Pre-Releases erfolgt am einfachsten auf folgendem Weg:

PRE-PRE-PRE-Vorabversion !

Noch nicht Korrektur-gelesen!

Änderungen, Ergänzungen, Korrekturen etc. bitte zu mir!

dl5di@darz.de

Die Software auf den Gateways besteht im Wesentlichen aus 3 Komponenten, DSM, ircDDB und einer Trigger-Funktion, die in der Postgres-Datenbank des Systems installiert wird.

Hier eine Beschreibung, was das Installations-Script macht, damit man die Abläufe leichter nachvollziehen und evtl. kontrollieren und korrigieren kann.

(Quellen am Ende)

DStar-Monitor:

Wie aus den von Michael, DL1BFF erstellten Grafiken ersichtlich, basiert die Kommunikationsrichtung vom Gateway zum Datenbank-Netz im ersten Schritt auf dem DStar-Monitor, wie er im US-Trust-Netz Vorgabe in den Anschaltbedingungen ist.

Das muss für andere Systeme nicht so bleiben, sollte mit XMonitor oder ähnlichen Programmen, die am Datenstrom des Controllers oder äquivalenter Systeme lauschen und die gehörten LastHeard-Informationen an eine Datenbank weiterleiten können, auch gehen.

Diese Funktionalität kann auch später in bestehende Software integriert werden.

Es muss auf Dauer auch nicht zwangsläufig eine SQL-Kommunikation bleiben, wie DStarMonitor sie nutzt, wir können auch auf Server-Seite ein [abgestimmtes](#) Interface integrieren, was in die Datenbank schreibt.

Hier sind wir Vorschlägen gegenüber offen.

DSM hat für den Einstieg den großen Vorteil, dass das beim US-Trust-Team als vorhandenes, sogar vorgeschriebenes und unkritische Tool bekannt ist, was nicht in die Funktionalität des Systems eingreift.

DStarMonitor ist normalerweise unter /opt/dstarmon installiert.

Das ist nur wichtig, weil sich dstarmon und unser Tool den Postgres-Java-Connector teilen und der Pfad in dessen Config drinsteht, also ggf. angepasst werden müsste wenn man etwa grundsätzlich anders installieren will.

DSM kann problemlos erweitert werden und außer die Datenbank bei dstaruser.org auch weitere Datenbanken mit Daten beschicken.

Das können Postgres oder MySQL Datenbanken sein, dstarusers.org verwendet MySQL, der ircDDB-Server verwendet Postgres.

Dabei können 2 verschiedene Tabellen beschickt werden, eine LastHeard-Liste (Tabelle „lastheard“, wo nur Rufzeichen, Einstiegs-Repeater und Zeitstempel übertragen werden):

target_cs, last_mod_time und arearp_cs

In der von Pete Loveall beschriebenen LastHeard-Tabelle existieren erheblich mehr Felder, die aber nur gefüllt werden wenn man die LastXmt-Funktion parallel aktiviert.

Das nutzen wir für unser System nicht.

In den Properties für DSM müssen für unsere Anwendung entsprechende Datenbankeinstellungen hinzugefügt werden und zwar nur zum Schreiben der LastHeard-Table.

Wie das im Detail geht ist in der DSM-Dokumentation von Pete Loveall, AE5PL im Detail beschrieben.

Der für uns relevante Teil der Datei sieht bei einem Standard Icom-System am US-Trust folgendermaßen aus:

```
stderr=/var/log/dstarmon.log
GWIntf=eth1
CtrlrIP=172.16.0.1
LHDriver=com.mysql.jdbc.Driver
LHURI=jdbc:mysql://dsm.dstarusers.org:3306/DSTAR2
LHParameters=user;dstar;password;icom
```

Man erkennt den LHDriver für das Reporting an DStaruser.org und seine 3 Konfigurationszeilen

- LHDriver mit dem Treibernamen
- LHURI mit der Serveradresse
- LHParameters mit den Login-Parametern, wie man die allseits kennt

Jede mit Daten zu beschickende Datenbank wird durch 3 solcher Zeilen konfiguriert.

Als XTrust-Benutzer würde ich diese Default-Reporting von DSM an dstarusers.org rausnehmen!

Einige X-Trust-Installationen sind bzw. waren mit DSM vorkonfiguriert, wo dieser Datenbankteil dann fehlte.

Wir erweitern diese Einstellungen mit dem Installations-Script auf folgende Form:

```
stderr=/var/log/dstar/dstarmon.log
GWIntf=eth1
CtrlrIP=172.16.0.1
LHDriver=com.mysql.jdbc.Driver
LHURI=jdbc:mysql://dsm.dstarusers.org:3306/DSTAR2
LHParameters=user;dstar;password;icom
#
LHDriver1=org.postgresql.Driver
LHURI1=jdbc:postgresql://group1-postgres.ircddb.mdx.de:5432/ircddb
LHParameters1=user;db0xyz;password;*****
#
```

Man erkennt den 2. Treiber mit dem Index 1 an den Parameternamen.

Laut Pete's Doku muss die Reihenfolge der Indizes stimmen und durchgehend sein, „ohne Index“ entspricht dabei dem Index „0“.

Sobald das Script feststellt, dass bereits etwas geändert wurde, LHDriver1 schon existiert o.ä., bricht es das Einfügen unserer Ergänzung ab.

In dem Fall muss man diesen Teil manuell einfügen und den Index anpassen.

Wer dort schon gewerkelt hat, weiß sicherlich was er dann zu tun hat.

Username und Passwort ersetzt das Installations-Script automatisch, Username ist das Call des Gateways in Kleinbuchstaben!!

Das Passwort vergeben wir beim Anlegen der Accounts im System und schicken es per EMail.

Das Installations-Script fragt das Passwort nach dem Start ab.

Anmerkung für nicht US-Trust-Systeme:

- *DSM ist meistens nicht installiert, muss also nachinstalliert oder durch etwas äquivalentes ersetzt werden. Wir liefern in Kürze etwas anderes.*
- *Die Konfiguration sollte am besten manuell wie oben angegeben vorgenommen werden.*
- *Wenn man die Konfiguration für dstarusers.org weglässt, bekommt unser Treiber den Index 0, (also keinen).*
- *Das sieht dann so aus:*

```
stderr=/var/Log/dstar/dstarmon.Log
GWIntf=eth1
CtrlrIP=172.16.0.1
#
LHDriver=org.postgresql.Driver
LHURI=jdbc:postgresql://group1-postgres.ircddb.mdx.de:5432/ircddb
LHParameters=user;db0xyz;password;*****
#
```

Damit sind die Einstellungen in den DSM-Properties abgeschlossen.

Anschließend muss noch der Postgres-Connector (postgresql-8.4-701.jdbc3.jar) in den Class-Path des DSM-Start-Scripts eingesetzt werden.

Auch das wird abgebrochen falls die Default-Konfiguration nicht erkannt wird oder er schon vorhanden ist und muss dann manuell gemacht werden.

Vor der Konfiguration (default) sieht die Datei monstart.sh so aus:

```
$javadir -cp DStarMonitor.jar:javAPRSSrvr.jar:jpcap-0.01.16/jars/net.sourceforge.jpcap-0.01.16.jar:mysql-connector-java-5.1.6-bin.jar Main &
```

Nach dem Durchlauf des Installations-Script sollte es so aussehen:

```
$javadir -cp DStarMonitor.jar:javAPRSSrvr.jar:jpcap-0.01.16/jars/net.sourceforge.jpcap-0.01.16.jar:mysql-connector-java-5.1.6-bin.jar:postgresql-8.4-701.jdbc3.jar Main &
```

Man erkennt, dass der Postgres-Connector am Classpath hinten angehängt wurde:

:postgresql-8.4-701.jdbc3.jar

Anmerkung für nicht US-Trust-Systeme:

Der APRS/DPRS-Server wird ebenfalls über DSM konfiguriert.

Es gibt die Möglichkeit, die nicht in der Config enthaltenen Defaultwerte durch eigene Vorgaben zu überschreiben.

Das ist teilweise nicht dokumentiert, ich kann gerne ein Muster mailen.

Mein DSM/javAPRS bei DBOMYK sendet die Daten z.B. nicht auf den US-APRS-Server, sondern lokal auf dl.aprs2.net.

Dazu hängt man am Ende der dstarmonitor.properties folgenden Eintrag an:

upstreamHubs=dl.aprs2.net:14580

Musterconfigs, Anleitungen und Alternativlösungen für nicht Standard-Icom-Systeme willkommen!!

Das Hauptziel ist, die Software auf allen Systemen zum Laufen zu bekommen und so die Netzgrenzen zu überwinden!

Damit sind alle Eingriffe in bestehende Software und deren Konfiguration abgeschlossen, es folgen nur noch neue Installationen und Konfigurationen.

Anzumerken bleibt, dass das Script vor Änderungen Backups von existierenden Dateien anlegt, so dass man immer wieder leicht zurück kann.

ircDDB-Client

Nun zum ircDDB-Client, der Software, die sich an den IRC-Server hängt und die dort kommenden Daten in die lokale Datenbank einarbeitet.

Beim ersten Connect lädt die sich auch die Liste bereits gehörter Stationen vom Server runter, bei Verbindungsunterbrechungen beschränkt er sich auf die Daten, die seit dem letzten Login angefallenen sind, so dass lokal immer alle Daten aktuell sind.

Das Installations-Script läuft zwar mit root-Rechten, aber die eigentliche Software hinterher nicht!

Das Script legt zunächst einen Benutzer „ircddb“ an mit Home-Directory „/opt/ircDDB“. Es lädt dann die für die Erstinstallation notwendigen Dateien in Form eines tgz von einem Webserver und entpackt es.

Das tgz enthält folgende Dateien:

```
opt/  
opt/ircDDB/  
opt/ircDDB/ircDDB.policy  
opt/ircDDB/ircDDB.keystore  
opt/ircDDB/run.sh  
opt/ircDDB/start.sh  
opt/ircDDB/ircDDB.properties  
opt/ircDDB/install/  
opt/ircDDB/install/trigger.txt  
opt/ircDDB/install/postgresql-8.4-701.jdbc3.jar  
opt/ircDDB/install/dstar
```

Die eigentlichen Binaries sind nicht enthalten, sie werden beim ersten Start von run.sh aktuell vom Server nachgeladen und lassen sich auf Wunsch auf dem gleichen Weg automatisch updaten. Mehr Informationen dazu folgen an anderer Stelle.

„ircDDB.policy“ und „ircDDB.keystroke“ stellen sicher, dass beim Auto-Update niemand dem System etwas falsches unterschiebt.

Run.sh ist ein endlos laufendes Script, was den autoupdate-Mechanismus beinhaltet. Der Client kann per Kommando über IRC beendet werden, sichert dann die letzte lauffähige Version, sucht nach Updates, wenn verfügbar installiert er sie und startet dann wieder neu. *(Das kann man natürlich ändern, bzw. anpassen um ggf. eigene Clients aus anderer Quelle upzudaten)*

Im IRC-Kanal kann man die Versionsnummer der Module aller Gateways im „Realname“ sehen (mit „/WHO *“).

Wir werden eine Dokumentation zur Bedienung noch veröffentlichen.

Start.sh ist die Startdatei, sie wird aus /etc/init.d/dstar_gw aufgerufen und kann auch zum manuellen Start verwendet werden.

Dstar_gw startet und stoppt alle wichtigen Prozesse des Gateways und wird über das gw_schedule-Script auch zur Remote-Administration verwendet.

ircDDB muss beendet werden bevor die Postgres-Datenbank Script gestoppt, andernfalls bricht z.B. der Restart-Vorgang ab.

Das Installations-Script startet die Software beim Abschluss automatisch.

Die Config-Datei ircDDB.properties sieht wie folgt aus:

```
irc_server_name=<IRCSRV>
irc_server_port=<IRCPORT>
irc_channel=<IRCCHAN>

irc_password=<PASSWORD>
rptr_call=<USERNAME>

debug=0
rptr_fix_tables=yes

ext_app=de.mdx.ircDDBApps.RptrApp

jdbc_class=org.postgresql.Driver
jdbc_url=jdbc:postgresql://127.0.0.1:5432/dstar_global
jdbc_username=dstar
jdbc_password=icom
```

Die Parameter der Form „<...>“ werden vom Installations-Script durch die erforderlichen Werte ersetzt.

Bitte hier nichts ohne Grund manuell ändern!
Manuelle Einträge können später nicht upgedatet werden.

Die Daten im ersten Block sind für den Zugang zum IRC-Server.
Eingesetzt werden hier die Daten unseres Round-Robin-Systems.
Die Zugangsdaten zum IRC-Server sind die gleichen wie die zur Datenbank in den DSM-Properties (s.o.).

Dann kommen 2 Schalter, die Unterschiede zwischen den Systemen ermöglichen:

`rptr_fix_tables=yes` bzw. `no`

Bei "yes" werden eventuell fehlende Zeilen in `sync_gip` und `sync_mng` angelegt.
Die Voreinstellung ist "no".
Bei Repeatern am US-Trsut ist "rptr_fix_tables=yes" einzutragen!

Wer auf einem anderen System keine `sync_gip`-Tabelle hat, lässt den Eintrag weg.
Icom- und OpenG2 nutzen wir die `sync_gip` zum Erreichen der anderen Gateways.

`rptr_fix_unsync_gip=yes` bzw. `no`

Dieser Parameter wird nur dann benötigt, wenn die "rptr_fix_tables"-Funktion aktiviert ist.
Beim Wert "yes" wird zusätzlich zu einer neu angelegten Zeile in `sync_gip` auch eine Zeile in `unsync_gip` eingetragen.
Bei `openg2` gibt es diese Tabelle nicht, deswegen muss dort "rptr_fix_unsync_gip=no" gesetzt werden.
Bei Standard-Icom-Systemen kann die Zeile entfallen, die Voreinstellung ist "yes".

Im unteren Block kommen dann noch die Zugangsdaten zur lokalen Datenbank mit den allseits bekannten Login-Daten.

In der run.sh findet man Pfade zu Software-Updates und zum Java-Postgres-Connector, die ggf. bei nicht-Standard-Systemen angepasst werden müssten.

Auch hier werden „<...>“-Werte vom Installations-Script passend ersetzt.

Postgres-Trigger:

Zum Schluss dann noch was ganz Wichtiges, die Postgres-Trigger-Funktion.

Diese Funktion hängt sich direkt in Postgres rein, fängt INSERTs und UPDATEs aus anderen Quellen an, z.B. vom Trust-Server, und ersetzt die darin enthaltenen Routing-Infos ggf. durch aktuellere Daten aus unserer Datenbanktabelle.

Installiert wird [dieser](#) Trigger automatisch!

Falls irgendwelche Probleme oder fragen auftreten, bitte melden!

Es wäre nett, wenn jemand, der ein abweichendes System benutzt die Anleitung ergänzen könnte. Bitte Anmerkungen, Ergänzungen, andere Pfade, Screen-Shots etc. zu mir!

Das Ganze ist nicht als Konkurrenz zu neuen Routing-Konzepten, neuen Trust-Ideen etc.. gedacht, sondern als eine gemeinsame, Netz-übergreifende Plattform, die die Kommunikation aller Benutzer untereinander wieder ermöglichen soll.

So wie es bei der VUS-Tagung in Baunatal als gemeinsames Ziel gesetzt wurde.

Deshalb bitte bei Verbesserungsvorschlägen und anderen Ideen immer beachten, ob sich das auch ins US-Trust-Netz, was nun mal schon aufgrund der Anzahl dort angeschlossener Systeme der Standard ist, problemlos integrieren lässt.

Wir denken, dass die Anpassung an andere Systeme sicherlich Dank Open Source dieses Projektes leichter sein wird.

Ideen, Verbesserungsvorschläge, Hilfe bei der Doku und Installations-Script-Erstellung, alles willkommen!!

Die Software ist Open Source, ohne wenn und aber!

ircDDB ist ein allgemeines Projekt von Michael im Internet (<http://github.com/dl1bff/ircDDB>).

Die Quellcodes werden noch komplettiert und dokumentiert und kommen dann alle ins Web.

Aktuell ist das Installations-Script hier zu beziehen:

<http://download.prgm.org/dl5di-soft/ircddb/ircDDB-install-Icom.sh>

Das Installationspaket liegt hier:

<http://download.prgm.org/dl5di-soft/ircddb/install-ircddb-for-icom.tgz>

Das wird umziehen sobald es fertig ist!!

73

Hans-Jürgen, [DL5DI](#)

SysOp @[DA5UDI](#), [DB0LJ](#), [DB0MYK](#)

<http://www.prgm.org/>

<http://afuforum.prgm.org>

Tel Shack: 02652 938377

Skype: hjbarthen