

ircDDB Installation

Das ircDDB-Add-On-Tool wird am einfachsten mit dem Installations-Script installiert.
Dieses Script kann direkt auf das Gateway geladen werden.

```
cd /tmp
wget http://download.ircddb.net/ircddb-icom-dsm/install/ircDDB-install-Icom.sh
```

Anschließend das Script starten

```
/bin/sh ./ircDDB-install-Icom.sh
```

Das Script fragt dann das Passwort des ircDDB-Accounts ab.
Anschließend läuft alles Notwendige vollautomatisch ab.
Das Software-Archiv wird vom Server geladen, installiert und konfiguriert.
Nach kurzer Zeit ist die Installation abgeschlossen und die Software wird automatisch gestartet.

Traten keine Fehler auf ist die Installation an dieser Stelle bereits erfolgreich abgeschlossen!

Die Software wird automatisch gestartet und das Gateway-Call erscheint auf der Status-Webseite
<http://status.ircddb.net> .

Es gibt speziell für Sysops eine Statusseite, die mehr Details zeigt:

<http://status.ircddb.net/cgi-bin/ircddb-gwst>

Die linke "LED" zeigt den Status der IRC-Verbindung, rot oder grün.
Wenn das Call gar nicht in der Liste auftaucht, hat noch keine Verbindung bestanden.

Die rechte "LED" zeigt den Status der DSM-Verbindung, rot, grün, gelb oder grau.
Eigenbau-Systeme benötigen kein DSM, deshalb muss das System zunächst feststellen, was im Einsatz ist. Die LED startet beim ersten IRC-Connect mit dem DSM-Status "gelb" / unbekannt.
Wird ein DSM-Connect erkannt wird sie grün, es wird dann ein Flag in der Datenbank gesetzt, dass dies ein DSM-basierendes System ist.
Kommen Daten rein ohne DSM-Connect wird sie grau (ungenutzt), in dem Fall wird intern das Flag gesetzt, dass dies kein DSM-basierendes System ist.

Falls irgendwelche Probleme oder Fragen auftreten, bitte melden!

Im Verzeichnis `/opt/ircDDB/install` findet man das Script „`test-ircddb.sh`“.
Dieses Script schreibt alle Konfigurationswerte und andere wichtige Einstellungen in ein Textfile
`/tmp/ircddb-instcheck.<callsign>` und lädt es auf unseren Server.
Hierzu wird „curl“ verwendet.
Sollte der Upload nicht klappen, bitte mir dieses File mailen (dl5di@dar.c.de), es hilft sehr bei der Fehlersuche!

Der nachfolgende Teil wendet sich an diejenigen, die etwas genauer wissen wollen, was installiert wird.

Detaillierte Installations- und Konfigurationsanleitung

Das ircDDB-Add-On-Tool für Standard-Icom-Gateways am US-Trust 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.

DStar-Monitor:

Wie aus den Grafiken der Dokumentation ersichtlich ist, basiert die Kommunikationsrichtung vom Gateway zum ircDDB-Netz auf dem DStar-Monitor, wie er im US-Trust-Netz Vorgabe in den Anschaltbedingungen ist.

(Für ircDDB-only-Systeme, also Systeme ohne Anbindung an einen Trust-Server, ist eine Version verfügbar, die keine DSM-Installation benötigt).

DStarMonitor ist unter /opt/dstarmon installiert.

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 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.

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.net:5432/ircddb
LHParameters1=user;db0xyz;password;*****;loginTimeout;10;autoReconnect>true
#
```

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 etwas verändert 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.

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

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.

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  
opt/ircDDB/install/test-ircddb.sh
```

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 automatischen Update-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.
(Wer das nicht möchte, kann es natürlich ändern)

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

Wir werden eine Dokumentation zur Bedienung noch separat 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=net.ircddb.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-Trust ist unbedingt "rptr_fix_tables=yes" einzutragen!

Wer auf einem anderen System keine sync_gip-Tabelle hat, lässt den Eintrag weg.

Icom- und OpenG2 nutzen 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 ab, z.B. vom Trust-Server, und ersetzt die darin enthaltenen Routing-Infos ggf. durch aktuellere Daten aus unserer Datenbanktabelle.

Installiert wird [dieser Trigger automatisch!](#)

Das Installations-Skript kann für Updates beliebig oft gestartet werden, bereits ersetzte Platzhalter werden dabei nicht nochmal ersetzt.

Bei weiteren Durchläufen sind Warnmeldungen der Datenbank bzgl. bereits existierender Tabellen, Indizes und Trigger normal und können ignoriert werden.

Wenn die Software läuft erscheint das Gateway-Call automatisch in der Übersicht auf unserer Webseite www.ircddb.net

Es gibt speziell für Sysops eine Statusseite, die mehr Details zeigt:

<http://status.ircddb.net/cgi-bin/ircddb-gwst>

Die linke "LED" zeigt den Status der IRC-Verbindung, rot oder grün.

Wenn das Call gar nicht in der Liste auftaucht, hat noch keine Verbindung bestanden.

Die rechte "LED" zeigt den Status der DSM-Verbindung, rot, grün, gelb oder grau.

Eigenbau-Systeme benötigen kein DSM, deshalb muss das System zunächst feststellen, was im Einsatz ist. Die LED startet beim ersten IRC-Connect mit dem DSM-Status "gelb" / unbekannt.

Wird ein DSM-Connect erkannt wird sie grün, es wird dann ein Flag in der Datenbank gesetzt, dass dies ein DSM-basierendes System ist.

Kommen Daten rein ohne DSM-Connect wird sie grau (ungenutzt), in dem Fall wird intern das Flag gesetzt, dass dies kein DSM-basierendes System ist.

Falls irgendwelche Probleme oder Fragen auftreten, bitte melden!

Im Verzeichnis `/opt/ircDDB/install` findet man das Script „`test-ircddb.sh`“.

Dieses Script schreibt alle Konfigurationswerte und andere wichtige Einstellungen in ein Textfile `/tmp/ircddb-instcheck.<callsign>` und lädt es auf unseren Server.

Hierzu wird „curl“ verwendet.

Sollte der Upload nicht klappen, bitte mir dieses File mailen (dl5di@darz.de), es hilft sehr bei der Fehlersuche!

Die Software ist Open Source, ohne wenn und aber!

Das Installations-Script ist hier zu beziehen:

<http://download.ircddb.net/ircddb-icom-dsm/install/ircDDB-install-icom.sh>

Das Installationspaket für eine manuelle Installation liegt hier (wird vom Script automatisch geladen):

<http://download.ircddb.net/ircddb-icom-dsm/install/install2-ircddb-for-icom.tgz>

73

Hans-Jürgen Barthen, DL5DI

2010-07-20