

ircDDB installation on Standard Icom Gateways at the US-Trust network

The easiest way to install the ircDDB add on tool on a standard Icom system at the US-Trust-network is to use the install-script.

This script can be downloaded direct to the gateway:

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

After that run the script:

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

The script will request the password which is needed to access the ircDDB network. This password is provided by the development team on request.

After that everything runs automatically.

The necessary files are downloaded, installed and configured.

If no error occurs the installation will start automatically.

That's all! Welcome to the ircDDB Open DStar network!

If you have got error messages so far please let us know.

You will find a helpful debug script „**test-ircddb.sh**“ in the directory /opt/ircDDB/install .

This script copies all relevant configuration information and other settings to a textfile

```
/tmp/ircddb-instcheck.<callsign>
```

The textfile will be uploaded to our server if “curl” works, if not, please send this file to us or upload it to our server. It helps finding and fixing any issues.

(Your access password will be included, but we know it anyway, just remember this when sending the file to anybody else)

You may stop reading here. The following part will describe what modules are installed, what configuration files are modified. The goal is to make the installation more transparent and make it easier to maintain.

Description of the ircDDB components, their installation and configuration:

The ircDDB-Add-On-Tool for standard-Icom-Gateways at the US-Trust network is mainly based on 3 components:

- dstarmonitor (DSM),
- ircDDB application,
- a trigger function written in the postgres script language.

DStar-Monitor:

The grafics on our website show, that the communication from the Gateway to the ircDDB network is done by D-Star-Monitor, a component which is mandatory on the US-Trust-network. (For ircDDB-only-systems without any connection to any trust server system we provide a different client which does not need DSM).

DStarMonitor is installed at /opt/dstarmon.

By default DSM sends lastheard information to dstarusers.org.

This configuration can be extended to feed additional databases.

This may be Postgres or MySQL databases, dstarusers.org uses MySQL, the ircDDB-server uses Postgres.

DSM allows to provide data to 2 different tables, one if it is a LastHeard-table which shows callsign, area-repeater and timestamp of a heard station:

target_cs, last_mod_time and arearp_cs

The documentation from Pete Loveall shows a lot of more fields in the LastHeard-table but these are only filled if the LastXmt function is also used.

IrcDDB does not use these fields.

Some additional configuration lines have to be added to the dstarmonitor.properties to fill the LastHeard-table on the ircDDB Postgres server.

A detailed description can be found in the DSM-documentation provided by Pete Loveall, AE5PL.

The part of dstarmonitor.properties which is relevant for us on a Standard Icom-gateway at the US-trust looks like this:

```
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
```

This is the configuration of the LHDriver which is needed for the reporting to the DStaruser.org website:

LHDriver	defines the driver name from the java connector (in this case MySQL)
LHURI	defines the URL of the database which has to be used
LHParameters	defines the well known login-parameters to the database

Each database which should be used needs to be configured using this 3 configuration parameters.

The install-script will add these settings in the following 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;*****
#
```

You may recognize the 2nd driver with the index 1 at the end of the parameters. Pete's documentation explains how to use these indexes. No index means „0“. The install script stops inserting this lines if it discovers an existing LHDriver1. In this case you need to add it manually in the correct form.

Username and password are automatically inserted by the install-script.

- Username is the callsign of the gateway in lower case!!
- The password is provided by the admin team by email.

The install-script asks for the password when started.

The changes on the DSM properties are now complete.

Finally the postgres-connector (postgresql-8.4-701.jdbc3.jar) needs to be added to the class-path of the DSM startscript. This is done automatically if the dstarmonitor-startfile is found to be default. Otherwise it needs to be added manually.

The default monstart.sh looks like this:

```
$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 &
```

The install script changes it to this:

```
$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 &
```

The postgres-connector has been added to the end of the classpath:

:postgresql-8.4-701.jdbc3.jar

dstar_gw

The last change to existing software is the start of the ircDDB process from the script `/etc/init.d/dstar_gw`.

We decided to take that way instead of using an own startup script because this central script is used during remote updates and other processes.

ircDDB needs to be stopped before the postgres database can be closed and the `dstar_gw` script seems to be the best place where to do it whenever necessary.

The install-script creates copies before doing any changes on existing configuration files.

Now we finished changes to existing software. The following part describes the installation of additional tools.

ircDDB-Client

The ircDDB-client is the software which adds incoming routing data to the local database.

The install.script has to be run by root but the ircDDB software will be run by user „ircddb“ which is created automatically using the home directory „/opt/ircDDB“.

The script downloads additional software during first run from our webserver.

The downloaded tgz includes this files:

```
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
```

The java binaries are not included, they will be downloaded by the script „run.sh“ from the ircDDB server when started.

The binaries may be updated on this way from remote via a command on the IRC channel.

We will supply more information on that later.

„ircDDB.policy“ and „ircDDB.keystore“ make sure, that only updates from trusted sources are accepted.

run.sh is a script running an endless loop to provide the automatic update mechanism.

If the client is stopped using an IRC command the last stable version will be saved and in cases of issues it will be used later again.

The actual version of the software modules is shown in the „realname“ of the gateways and may be requested using the „/WHO *“ command on the IRC channel.

We will supply a more detailed documentation later.

start.sh is the start script which is called from /etc/init.d/dstar_gw . It can also be used for manual start.

dstar_gw was already mentioned before.

It will replace the default startup file /etc/init.d/dstar_gw.

It starts and stops all gateway processes and is also used by the gw_schedule-script for remote-administration.

ircDDB needs to be stopped before Postgres can be stopped, otherwise a restart may not run properly.

The install script will start the software automatically when the installation is complete.

The **ircDDB.properties** file looks like this:

```
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
```

Parameter of the form „<...>“ will be replaced by the install script.
Better do not touch it unless you know exactly what you do.

Manual entries cannot be updated later.

The 1st block shows the access parameters for the IRC-Server.
With this hostnames we are using a round robin system.
The access data for the IRC server and for the database server in the DSM properties are identical.

After that you will find 2 switches which allow to support different platforms:

`rptr_fix_tables=yes or no`

Setting "yes" will switch adding of entries in sync_gip and sync_mng on.
The default is "no".

Repeater at the US-Trust system need to set "rptr_fix_tables=yes" !!
Systems without sync_gip-table do not use this entry.
Icom- and OpenG2 are using sync_gip to reach other gateways.

`rptr_fix_unsync_gip=yes bzw. no`

This parameter is only used if "rptr_fix_tables" is set to „yes“.
If set to "yes" the software will also add entries to unsync_gip parallel to sync_gip .
There is no unsync_gip table on openg2 , so "rptr_fix_unsync_gip=no" has to be used there.
On standard Icom systems this line is not necessary, the default is "yes".

The last block shows the access information to the local database with the well known login parameters.

run.sh shows paths to software updates and the Java-Postgres-Connector, which may have to be adjusted on non standard systems.

Again parameters showing „<...>“ will be replaced by the install script.

Postgres-Trigger:

Finally another important tool needs to be installed, the postgres trigger.

This trigger function is implemented to the postgres database. It will be used on all INSERTs and UPDATEs from external sources, for example from the trust server, and replaces the supplied routing information by more actual data from the ircDDB database if available.

The trigger is installed automatically!

The install script may be started more than once and thus may be used for later updates too. Warnings from the database on existing tables, index and trigger may be ignored.

ircDDB is started automatically.

If all worked fine you will see the gateway call on our status-page <http://status.ircddb.net> .

In case of any issues please contact us!

You will find a helpful debug script „**test-ircddb.sh**“ in the directory /opt/ircDDB/install .

This script copies all relevant configuration information and other settings to a textfile /tmp/ircddb-instcheck.<callsign>

Please send this file to us or upload it to our server, it helps fixing any issues.

(Your access password will be included, but we know it anyway, just remember this when sending the file to anybody else)

Ideas for improvements, additional documentation, translations to other languages, ... it is all very welcome!!

The software is really Open Source, please find the download URLs on our website!

73

Hans, [DL5DI](#)
ircDDB developer team