

# **DStarMonitor User's Guide**

## **2.6b06**

DStarMonitor is Copyright © 2010 - Pete Loveall AE5PL [pete@ae5pl.net](mailto:pete@ae5pl.net)

Use of the software is acceptance of the agreement to not hold the author or anyone associated with the software liable for any damages that might occur from its use. The software may not be distributed in any form without prior written permission from the author. The software may only be used by amateur radio operators for amateur radio purposes and may not, in any way, be used to alter or inject false information into existing networks, databases, or other entities.

javAPRSSvr, DStarQuery, and any other referenced software or hardware is licensed separately.

APRS is a trademark of Bob Bruninga

Other trademarks included in the following text are recognized as belonging to the respective trademark holders.

# Table of Contents

<b>SECTION 1 - INTRODUCTION .....</b>	<b>4</b>
<b>SECTION 2 - PROGRAM REQUIREMENTS AND DESCRIPTION .....</b>	<b>4</b>
<b>SECTION 3 - CONFIGURATION PARAMETERS .....</b>	<b>5</b>
<b>GENERAL PARAMETERS .....</b>	<b>5</b>
<i>stderr=err.log</i> .....	5
<b>GATEWAY INTERFACE PARAMETERS .....</b>	<b>5</b>
<i>GWIntf=eth1</i> .....	5
<i>CtrlrIP=172.16.0.1</i> .....	5
<i>UDPPort=20000</i> .....	5
<b>LASTHEARD DATABASE PARAMETERS .....</b>	<b>6</b>
<i>LHDriver=</i> .....	6
<i>LHURI=</i> .....	6
<i>LHParameters=</i> .....	6
<b>LASTXMT DATABASE PARAMETERS .....</b>	<b>6</b>
<i>LXDriver=</i> .....	6
<i>LXURI=</i> .....	6
<i>LXParameters=</i> .....	6
<b>SERIAL PORT PARAMETERS .....</b>	<b>7</b>
<i>numCtrlrs=1</i> .....	7
<i>numRptrs=4</i> .....	7
<i>serialPort=127.0.0.1:24580</i> .....	7
<b>APRS OBJECT PARAMETERS .....</b>	<b>8</b>
<i>callsign=</i> .....	8
<i>repeaterIDx=</i> .....	8
<i>repeaterIDx;callsign=</i> .....	8
<i>RptrObjectInterval=20</i> .....	8
<b>JAVAPRSSRVR PARAMETERS .....</b>	<b>9</b>
<i>upstreamHubs=DPRSGW-1.dstarusers.org:14580;DPRSGW-2.dstarusers.org:14580</i> .....	9
<i>noEchoPorts=127.0.0.1:24578</i> .....	9
<i>statusPorts=</i> .....	9
<i>IGateIntf=</i> .....	9
<b>SECTION 4 - RECOMMENDED CONFIGURATIONS .....</b>	<b>10</b>
<b>SAMPLE DSTARMONITOR.PROPERTIES .....</b>	<b>10</b>
<b>TABLE FORMATS .....</b>	<b>11</b>
<i>PostgreSQL</i> .....	11
<i>mySQL</i> .....	11
<b>SECTION 5 - INSTALLATION INSTRUCTIONS .....</b>	<b>13</b>

## Section 1 - Introduction

DStarMonitor provides last heard and usage information from DStar controllers and gateways. DStarMonitor utilizes pcap network interface monitoring capabilities provided by libpcap.so (Linux) or winpcap.dll (Windows) to monitor gateway/repeater controller UDP packets to collect the data. DStarMonitor populates two SQL tables, LastHeard and LastXmt, with the data collected. DStarMonitor is written in Java to allow portability between operating systems and databases.

## Section 2 - Program Requirements and Description

DStarMonitor is designed to run on any OS with any recent Java Virtual Machine (1.5 or higher). The DStarMonitor requires root or Administrator permissions to open the gateway NIC using pcap.

DStarMonitor is comprised of a number of classes which Java looks at as objects. The main class is Main. This class is called at startup, sets parameters, and begins execution of the support threads.

DStarMonitor works in conjunction with jpcap to provide full active monitoring of the communications. DStarMonitor supports separate databases for each table allowing for a centralized LastHeard table while giving each sysop the ability to monitor usage on their individual gateways via the LastXmt table.

LastHeard is updated any time a station is heard via a repeater connected to the controller. This includes the internal "CALLSN S" (S identifier) controller identification.

LastXmt is updated differently depending on whether the transmission is voice (and low-speed data) or high-speed data. Voice transmissions consist of an RF header packet followed by individual voice/data 12 octet packets. DStarMonitor records the information in the RF header and then accumulates a byte count for that packet and the ensuing packets. When the end of transmission is received, DStarMonitor adds the record to LastXmt.

High-speed data is passed in self-contained UDP packets. In other words, each UDP packet contains an RF header and a complete Ethernet packet in the data. DStarMonitor updates LastXmt with that information upon receipt of each UDP packet.

DStarMonitor now contains a module that can send APRS object packets to APRS-IS. This can be used to identify repeaters and their locations to the APRS network. This module uses javAPRSSrvr which must be in the classpath at startup. javAPRSSrvr is licensed separately.

DStarMonitor can parse to TCP ports the serial data information sent by the Icom radios. This function can be used to gate D-PRS to APRS-IS using javAPRSSrvr as well as monitoring the serial data stream for other applications. This function is read-only.

Currently, DStarMonitor has the ability to start individual instances of javAPRSSrvr and DStarQuery (licensed separately) for each voice repeater. Installation requires javAPRSSrvr.jar and DStarQuery.jar to reside in the DStarMonitor directory for the respective applications to start. DStarQuery must have dstarqueryx.properties files in the directory as well (x = a, b, and c for each DV repeater A, B, and C, respectively).

## Section 3 - Configuration Parameters

The configuration parameters reside in a configuration file which, by default, is called `dstarmonitor.properties`

The parameters are CASE SENSITIVE. Defaults are shown.

NOTE: UNLESS YOU REQUIRE A SETTING OTHER THAN THE DEFAULT, DO NOT INCLUDE ANY PARAMETERS WITH DEFAULT SETTINGS.

**List parameters** may be defined on the line or may be defined in a text file. If defined on the line, each entry is separated by a semicolon. If defined in a file, each entry is put on a separate line. Do not put blank lines in the file. The file must have the extension `.lst` For instance, this might be the definition for `LHParameters`:

```
LHParameters=user;postgres;prepareThreshold;1
```

Or you could have the following 4 lines in `LHParameters.lst`:

```
User  
Postgres  
prepareThreshold  
1
```

You would then put the following line in your configuration file:

```
LHParameters= LHParameters.lst
```

### ***General Parameters***

#### **`stderr=err.log`**

This is used to send error messages to a file instead of to the console.

### ***Gateway Interface Parameters***

#### **`GWIntf=eth1`**

This tells DStarMonitor which Ethernet interface is connected to the controller.

If this is set to nothing (`GWIntf=` ) then no database updates will be performed and the database settings will be ignored.

#### **`CtrlrIP=172.16.0.1`**

This is the controller's IP address.

#### **`UDPPort=20000`**

This is the UDP port number used by the controller and gateway.

`UDPPort` is the port on both the controller and the gateway for data transfer.

### ***LastHeard Database Parameters***

These parameters can also have a number appended, starting with 1. This allows multiple databases to be updated. E.g. LHDriver1=xxxx (No number equals zero and should always be in the dstarmonitor.properties file).

#### **LHDriver=**

This is the JDBC driver name as defined by the JDBC driver vendor.

#### **LHURI=**

This is the path the JDBC driver uses to access the LastHeard table.

#### **LHParameters=**

(List)This is the parameter (property) pairs used by the JDBC driver. For instance, if the JDBC driver vendor states that user=xyz is a parameter to use for defining the user name to xyz, you would use LHParameters=user;xyz This list is semicolon delimited.

### ***LastXmt Database Parameters***

If LXDriver is not defined then LastXmt will not be connected to or updated.

These parameters can also have a number appended, starting with 1. This allows multiple databases to be updated. E.g. LXDriver1=xxxx If the LX... parameters match the LH... parameters, a single connection will be used (recommended for timeout reasons).

#### **LXDriver=**

This is the JDBC driver name as defined by the JDBC driver vendor.

#### **LXURI=**

This is the path the JDBC driver uses to access the LastXmt table.

#### **LXParameters=**

(List)This is the parameter (property) pairs used by the JDBC driver. For instance, if the JDBC driver vendor states that user=xyz is a parameter to use for defining the user name to xyz, you would use LHParameters=user;xyz This list is semicolon delimited.

## ***Serial Port Parameters***

### **numCtrls=1**

This is the number of repeater controllers talking to the gateway (include all repeater controllers connected via 10 GHz links). **Repeater controllers must be numbered sequentially from 1** (Repeater ID in the Icom RP2C configuration).

### **numRptrs=4**

This is the total number of repeaters each controller can have attached.

**This number should be left at the default of 4 so all DV repeater serial data streams are parsed.**

### **serialPort=127.0.0.1:24580**

This is the TCP/IP port number to start listening for connections.

This can include a NIC interface. 127.0.0.1:24580 is recommended so only local applications can access the serial data. If numCtrls=1, numRptrs=4, and serialPort=127.0.0.1:24580, then DStarMonitor will listen for TCP connections on 127.0.0.1 ports 24580, 24581, 24582, and 24583. These correspond to the repeaters connected to port 1 through 4 on the controller.

## **APRS Object Parameters**

### **callsign=**

This is the generic callsign of the gateway. Do not include the ID character or any extra spaces  
This field is mandatory.

### **repeaterIDx=**

This defines the object for each repeater.

The x is replaced with the individual repeater's ID. Append a D to the ID if it is a data repeater. For instance, the A data repeater would be repeaterIDAD. This field is mandatory for every repeater.

The format after the = is latitude;longitude;range;comments

Latitude is decimal degrees, South is negative.

Longitude is decimal degrees, West is negative.

Range is in statute miles.

Comments should be kept short. Usually use the format frequency,offset,DV.

### **repeaterIDx;callsign=**

This defines the object for each repeater on remote controllers.

This is the same as repeaterIDx with the addition of being able to define the callsign used by the remote controller. This is necessary as remote controllers must have a different callsign from the gateway and controller #1 callsign.

### **RptrObjectInterval=20**

The number of minutes between beaconing the APRS repeater objects to the Internet.

This is defaulted to 20 minutes which should be sufficient for most installations. If you are gating these to the local APRS RF frequency, you may want to increase the frequency to every 10 minutes.

Zero disables the objects (blocking them from being seen on jFindu, for instance). If beaconing, either leave this at 20 minutes or set it to 10 minutes for local gating.

## ***javAPRSSrvr Parameters***

These are the only parameters recognized in `dstarmonitor.properties` for the internal `javAPRSSrvr` instance.

**upstreamHubs=DPRSGW-1.dstarusers.org:14580;DPRSGW-2.dstarusers.org:14580**

These are servers which you want to both send and receive data from through this connection.

Only one server will be connected at a time. The default is recommend for all D-STAR gateways to minimize connection loads to specific servers.

**noEchoPorts=127.0.0.1:24578**

Bidirectional

This defines ports which do standard authentication and can receive data from verified clients. Data from the connected client is not echoed. This port sends all APRS packets to the connected client once the client is logged in.

**statusPorts=**

This creates a listener port to send HTML (default) and text status pages.

This is for web browser access to display configuration and status for the internal instance of `javAPRSSrvr`.

**IGateIntf=**

If set to `DPlus` (case-sensitive) and `DPLUSIntf.class` files are in the `DStarMonitor` directory, this will enable a bidirectional `IGate` using the D-Plus 2.2e capability to send text files to a specific repeater. This is a messaging-only interface as only gated packets are sent to RF. This allows bidirectional communication with an APRS client connected to a D-STAR radio via D-PRS Interface or other bidirectional D-PRS translator.

## Section 4 - Recommended Configurations

### ***Sample dstarmonitor.properties***

#Note that default values are not duplicated here

#stderr points to where any error messages will be logged

stderr=/var/log/dstarmon.log

#GWIntf is the NIC card name that talks to the repeater controller

GWIntf=eth0

#CtrlrIP is the IP address of the repeater controller NOT the gateway

CtrlrIP=172.16.0.1

#The LH parameters set the database parameters for LastHeard

#These settings are correct to talk to the d-starusers.org database

LHDriver=com.mysql.jdbc.Driver

LHURI=jdbc:mysql://d-starusers.org:3306/dstar

LHParameters=user;dstar;password;icom

#The following parameters are for APRS reporting

#Replace URCALL with the repeater's callsign (does not include ID letter)

callsign=URCALL

#Create a repeaterIDx for each repeater (append a D for data repeaters)

#Format for repeaterID is lat(-S);lon(-W);range in miles;comments

#comments should be short for display on APRS radios and cannot contain semicolons

repeaterIDA=32.9628;-96.4427;1;1292.700Mhz,-20Mhz,DV

repeaterIDAD=32.9628;-96.4427;1;1252.700Mhz,DD

repeaterIDB=32.9628;-96.4427;1;440.60Mhz,+5 Mhz,DV

## **Table formats**

XmtType in LastHeard is 'V' for voice/low-speed data and 'D' for high-speed data.

XmtType in LastXmt AND iXmtType in LastHeard is 0x20 for voice/low-speed data and 0x40 for high-speed data.

## **PostgreSQL**

JDBC Drivers can be obtained from <http://jdbc.postgresql.org/>

```
CREATE TABLE LastHeard (  
  ReportTime timestamp NOT NULL,  
  StationCall char(8) NOT NULL PRIMARY KEY,  
  RepeaterCall char(8) NOT NULL,  
  XmtType char(1) NOT NULL  
  ,iXmtType smallint NULL,  
  Flag1 smallint NULL,  
  Flag2 smallint NULL,  
  Flag3 smallint NULL,  
  DestRptr char(8) NULL,  
  SrcRptr char(8) NULL,  
  DestStn char(8) NULL,  
  SrcStn char(8) NULL,  
  SrcStnExt char(4) NULL,  
  Length int NULL);  
CREATE INDEX RepeaterCall ON LastHeard (RepeaterCall);
```

```
CREATE TABLE LastXmt (  
  StartTime timestamp NOT NULL,  
  Duration int NOT NULL,  
  XmtType smallint NOT NULL,  
  Flag1 smallint NOT NULL,  
  Flag2 smallint NOT NULL,  
  Flag3 smallint NOT NULL,  
  DestRptr char(8) NOT NULL,  
  SrcRptr char(8) NOT NULL,  
  DestStn char(8) NOT NULL,  
  SrcStn char(8) NOT NULL,  
  SrcStnExt char(4) NOT NULL,  
  Length int NOT NULL);  
CREATE INDEX StartTime ON LastXmt (StartTime);  
CREATE INDEX SrcStn ON LastXmt (SrcStn);  
CREATE INDEX DestStn ON LastXmt (DestStn);
```

## **mySQL**

JDBC drivers can be downloaded from <http://dev.mysql.com/downloads/connector/j>

```
CREATE TABLE `LastHeard` (  
  `ReportTime` timestamp NOT NULL default '0000-00-00 00:00:00',  
  `StationCall` char(8) NOT NULL default "",  
  `RepeaterCall` char(8) NOT NULL default "",  
  `XmtType` char(1) NOT NULL default ""  
  ,`iXmtType` tinyint(4) NULL default NULL,
```

```

`Flag1` tinyint(4) NULL default NULL,
`Flag2` tinyint(4) NULL default NULL,
`Flag3` tinyint(4) NULL default NULL,
`DestRptr` char(8) NULL default NULL,
`SrcRptr` char(8) NULL default NULL,
`DestStn` char(8) NULL default NULL,
`SrcStn` char(8) NULL default NULL,
`SrcStnExt` char(4) NULL default NULL,
`Length` int(11) NULL default NULL,
PRIMARY KEY (`StationCall`),
KEY `RepeaterCall` (`RepeaterCall`)
) ENGINE=MyISAM DEFAULT CHARSET=ascii

```

```

CREATE TABLE `LastXmt` (
`StartTime` timestamp NOT NULL default '0000-00-00 00:00:00',
`Duration` int(11) NOT NULL default '0',
`XmtType` tinyint(4) NOT NULL default '0',
`Flag1` tinyint(4) NOT NULL default '0',
`Flag2` tinyint(4) NOT NULL default '0',
`Flag3` tinyint(4) NOT NULL default '0',
`DestRptr` char(8) NOT NULL default "",
`SrcRptr` char(8) NOT NULL default "",
`DestStn` char(8) NOT NULL default "",
`SrcStn` char(8) NOT NULL default "",
`SrcStnExt` char(4) NOT NULL default "",
`Length` int(11) NOT NULL default '0',
KEY `StartTime` (`StartTime`),
KEY `SrcStn` (`SrcStn`),
KEY `DestStn` (`DestStn`)
) ENGINE=MyISAM DEFAULT CHARSET=ascii

```

The columns of LastHeard in blue are optional. However if specified, they must all be defined.

## Section 5 - Installation Instructions

All necessary files to communicate with DStarUsers.org and to gate D-PRS to APRS-IS are included in the distribution package using the G2 Java installation. To install, run the supplied script which will create a dsm service.

A command line similar to below can be found in the monstart.sh script. If you need to add or change drivers to the class path, stop the dsm service, modify the classpath in the script, and then start the dsm service.

```
$javadocir -cp DStarMonitor.jar:javAPRSSrvr.jar:jpcap-0.01.16/jars/net.sourceforge.jpcap-0.01.16.jar:mysql-connector-java-5.1.7-bin.jar Main &
```

If DStarQuery.jar and dstarqueryx.properties (x=a, b, and c) are found in the DStarMonitor directory, DStarMonitor will also start instances of DStarQuery just the same as javAPRSSrvr for D-PRS. It will properly configure the SerialPPort for each instance.

If javAPRSSrvr.jar does not exist, javAPRSSrvr instances and DStarQuery instances will not be started and javAPRSSrvr.jar should not be included in the classpath.