21.10.2021

XLX Multiprotocol Reflector Gateway for D-Star

Installation Package for Linux

Document version 1.0.5

1. Copyright:

The XLX Multiprotocol Gateway Reflector Server is part of the software system for the D-Star Network.
The sources are published under GPL Licenses.

2. Usage:

The packages which are described in this document are designed to install server software which is used for the D-Star network infrastructure.
It requires a 24/7 internet connection which can **support 20 voice streams** or more to connect repeaters and hotspot dongles!!

- The server requires a **fix IP-address** !
- The public IP address should have a DNS record which must be published in the common host files.

If you want to run this software please make sure that you can provide this service **free of charge,** like the developer team provides the software and the network infrastructure free of charge!

3. Requirements:

The software packages for Linux are tested on Debian7 (Wheezy) 32 and 64bit or newer.
Raspbian will work but is not recommended.
Please use the stable version listed above, we cannot support others.

4. Supported Protocols:

Since version 2.5.x, the XLX supports the follow protocols.

- **In D-Star**, Icom-G3Terminal, DExtra, DPLus and DCS
- **In DMR**, DMRPlus (dongle) and DMRMmdvm
- **In C4FM**, YSF, Wires-X and IRMS
- XLX Interlink protocol

Debian 7 (Wheezy) 32 and 64bit:

```
// after a clean installation of debian make sure to run update and upgrade
# apt-get update
# apt-get upgrade
```

```
// install git
# apt-get install git git-core
```

```
// install webserver with PHP5 support
# apt-get install apache2 php5
```

```
//install g++ compiler
# apt-get install build-essential
# apt-get install g++-4.7 (skip this step on Debian 8.x)
```

```
//download and compile the XLX sources
# git clone https://github.com/LX3JL/xlxd.git
# cd xlxd/src/
# make clean
# make
# make install
```

```
// copy startup script "xlxd" to /etc/init.d
# cp ~/xlxd/scripts/xlxd /etc/init.d/xlxd
```

```
// adapt the default startup parameters to your needs
# pico /etc/init.d/xlxd
```

// the ip address to use must be the address bind to your nic followed by the ip address of your AMBE server! If your AMBE server is running on the same machine as your XLX server, use 127.0.0.1. If you are not running any AMBE server also use 127.0.0.1 as a dummy ip.

//XLX v2.x.x needs a dmrid.dat to run, it can be retrieved from the XLXAPI server. The best way to keep it up to date is to add a cronjob for this task.

```
# crontab -e
30 2 */2 * * wget -O /xlxd/dmrid.dat http://xlxapi.rlx.lu/api/exportdmr.php
```

```
// Last step is to declare the service for automatic startup and shutdown
# update-rc.d xlxd defaults
```

```
// start or stop the service with:
# service xlxd start
# service xlxd stop
```

```
// copy dashboard to /var/www
# cp -r ~/xlxd/dashboard /var/www/db
```

```
// give the dashboard read access to the server log file

# chmod +r /var/log/messages

// reboot server to see if the auto-start is working

# reboot
```

Firewall settings:

XLX Server requires the following ports to be open and forwarded properly for in- and outgoing network traffic:

TCP port 80	(http) optional	TCP port 443 (https)
TCP port 8080	(RepNet) optional	
UDP port 10001	(json interface XLX Core)	
UDP port 10002	(XLX interlink)	
TCP port 22	(ssh) alternative	TCP port 10022
UDP port 42000	(YSF protocol)	
UDP port 30001	(DExtra protocol)	
UPD port 20001	(DPlus protocol)	
UDP port 30051	(DCS protocol)	
UDP port 62030	(MMDVM protocol)	
UDP port 8880	(DMR+ protocol)	
UDP port 10100	(AMBE controller port)	
UDP port 10101-10199	(AMBE transcoding port)	
UDP port 12345 – 12346	(Icom Terminal presence and request port)	
UDP port 40000	(Icom Terminal dv port)	
UDP port 21110	(Yaesu IRMS protocol)	

Dashboard configuration file:

The configuration file of the dashboard is called config.inc.php and can be found in /var/www/html/pgs (depending of your apache configuration).

//open the configuration file and adapt the parameters to your need.

# pico /var/www/html/pgs/config.inc.php

All possible parameters are self-explained.
When you are satisfied with your settings, don’t forget to turn on the calling home feature.

```
$CallingHome['Active']    = true;                    // xlx phone home, true or false
```

The hash file to access the xlxapi server and the way of working:

After activating the calling home feature, the dashboard creates a file called "callinghome.php" in the /tmp folder of your Linux.
This file contains the timestamp and the hash file that you need to access the xlxapi server database.
The first time, we need to use this folder because it's the only folder on a Linux system where everybody has full write access.
The problem with the /tmp folder is that Linux cleans up this folder on every server reboot and you are losing the callinghome.php with your hash file!
If the dashboard doesn't find the callinghome.php, a new one will be created automatically, but of course the hash which is random will be a new one.
So this one does not match the hash you had initially and your XLX has no access to the database any more.
You have two possibilities.

1.You backup your working hash, located in /tmp and put it back to /tmp after rebooting.

2.You are creating a folder called for example callhome, give full write access to this folder and change the callinghome.php path in the config.inc.php to the new location.

Definitely the best way, and of course you can set this up before activating the calling home feature.

```
# cd /
# mkdir callhome
# chmod 777 callhome
# pico /var/www/html/pgs/config.inc.php
```

```
//change the calling home location from /tmp/callinghome.php
to /callhome/callinghome.php
```

When done, the dashboard will create a new callinghome.php with a new hash file into this folder.
The reflector status will stay red, until I update the new hash file manually in the database.

Known issue:

From time to time, it happens that the hash file became blank?!
I'm not 100% sure about the cause, but one fact makes me believe that it has something to do with the growing /var/log/messages.
The behavior only happens after a long running period without rebooting the server.
After purging the messages logfile (# echo"" > /var/log/messages), the callinghome.php stays proper again !??

<u>XLX api server and useful links:</u>

The xlxapi server is located at xlxapi.rlx.lu
Do **not** change this link in the config.inc.php!

The XLX api server is providing a few services to the sysop.

- xlx interlink visualization page, http://xlxapi.rlx.lu/admin
- xlx host file generator, http://xlxapi.rlx.lu/api.php?do=GetReflectorHostname
- xlx reflector list in xml format, http://xlxapi.rlx.lu/api.php?do=GetReflectorList
- xlx dmr master list http://xlxapi.rlx.lu/api.php?do=GetXLXDMRMaster
- xlx ysf master list http://xlxapi.rlx.lu/api.php?do=GetXLXYSFMaster