

YSF Reflector Software

The Ham0perator YSF Reflector is currently supporting MNWis, US Kentucky, and America-Link. A test reflector runs at US K9EQ.

A project has been going on at the Ham0perator to improve the YSF Reflector software. Why do this?

The original software is very basic and offers very little capability and control. It also creates a HUGE problem when trying to bridge YSF hotspots to WiRES-X rooms because it just passes the data along with no filtering. That means whatever garbage gets sent to the reflector, it doesn't even need to be Fusion, that same data is sent to every connection including the bridged WiRES-X node. When the WiRES-X node gets this data it can misbehave or crash. Hence a project to fix these problems.

And while we're fixing problems, why not make it better by giving it more capabilities?

Current differences between the standard YSF Reflector software and the K9EQ version:

- Decodes all meta data including FICH and all data fields for every mode.
- Extensive logging allows selected logging of parameters needed to determine "what's going on?".
- Filters to drop packets if:
 - The FICH is invalid
 - The FICH does not pass sanity checks
 - Data Wide packets
 - Wires-X control packets
- Outputs a text file of connected nodes to simplify dashboards.
- Performance reporting

- Options can be changed in a *.ini file
- Ability to host multiple YSF Reflectors on a single server
- Internal documentation of the program (comments)
- Several builds are available:
 - Windows
 - Linux
 - Debian server (MNWis and US Kentucky are running in the “cloud”)

Future enhancements will include:

- Drop problematic WiRES-X packets with invalid room/node number in the CSD1 field
 - Future improvement: Edit the packets to remove the problematic information
- Ability to black list callsigns and IP addresses.
- Kerchunk filter to drop short key-ups
- Remote control and status reporting of the reflector via Fusion text messages
- Integrated dashboard removes need for php and greatly reduces CPU overhead and network bandwidth
- Programmed reporting of audio levels informing users if their mic level is too high or too low
- Integration with IMRS
- New, engineered and reviewed API to more tightly integrate hotspots with the reflector
- Ability to programmatically send messages
- Ability to programmatically edit data fields on the fly (i.e., change GPS data, call sign, etc.)
- Improve internal program documentation
- Complete program re-write in Python. This will provide the ultimate in portability

The reflector will not support non-Fusion communications such as DMR or D*. The reason for this is that non-Fusion systems do not include all of the data that Fusion provides. This would force the reflector to become the lowest common

denominator resulting in technology and feature restrictions.

The intent is for our version of YSF Reflector to be open sourced. It will be developed and enhanced with feedback from the Fusion community. By using good engineering practices, our hope is to provide a high quality service that equals or exceeds the existing quality of the Fusion/WiRES-X system. This is an enhancement to WiRES-X, not a replacement.

YSF WiRES-X Unintended Node Switching

A problem exists with YSF/MMDVM hotspots bridging to WiRES-X rooms. Here's what happens.

A user has a hotspot and accesses it using a Fusion HT. They use the WiRES-X control mode to change the hotspot to different YSF Reflectors. Let's say the user wants to access MNWis (Room #21,493):

- They enter the number of the YSF Reflector, "US MNWis 21493", which is 37,624. The hotspot switches to US MNWis 21493 which is bridged to WiRES-X Room # 21,493.
- The user leaves their radio in the WiRES-X control mode. They hear stations, but when they talk, nobody answers and nobody is talking.

Here's what happened: When they keyed up to transmit, their radio sent a command to switch to Room #37,624. The Wires-X node that was previously connected to #21,493 now connects to #37,624 which belongs to a very nice gentleman in France.

How does this happen?

The Fusion transmitted data includes callsign fields CSD1, CSD2, and others. We're interested in CSD1. This 10-byte field contains two items: The room number of the connection and; The TxID or DP-ID of the radio. A typical CSD1 looks like 21493F0yxh where 21493 is the room number and F0yxh is the DP-ID. The '21493' in this field will cause a Wires-X node to switch to that room.

When the radio IS NOT in Wires-X control mode, the CSD1 field will be: '*****F0yxh' and no switch will be made.

Bottom line:

DON'T USE WiRES-X control mode to talk through a hotspot or MMDVM!

Workarounds and fixes: (Updated 23-Aug-2019)

Node operators who are experiencing this problem should use the WiRES-X block feature to prevent the node from switching to the room that has the YSF reflector number.

For example: Assume that our node connects to MNWis, Room # 21,493. Our node also provides bridging to "YSF MNWis 21493" which is # 37,624.

- In the View->Node-Info(N) window press "Add".
- In the "Input ID" dialog box enter the YSF reflector number. In our example this is "37624".
- Press "OK" then "Close".
- When a station using the YSF bridge still has WiRES-X control mode enabled accesses the node, the command to switch rooms will be rejected by the WiRES-X software. The software will indicate a rejected attempt to switch rooms.

As an alternative, set the node so that connection changes are not allowed.

- File->Settings->Call Settings->Uncheck "Round Room Connection".
- Un check "Accept calls while in Round Room QSO.
- Check "Return to Room"
- Fill in the WiRES-X room number. Example: 21493.

YSF server side solutions:

I maintain a version of YSF Reflector that has filters to prevent hotspots causing problems with WiRES-X nodes. My first step is to drop all packets that contain an incorrect WiRES-X room number. Eventually I plan to replace the first 5 bytes of CSD1 with '*****'. This last step is difficult to do because it requires the reflector to decode the data, modify it, then recompute the CRC. This also involves working with the interleaving and forward error correction. It is obviously easier to just drop the packets, but that may confuse people on the YSF side since nobody on the WiRES-X side will hear them.

If you are interested in running the enhanced YSF Reflector software that will fix this problem at some point, please contact me privately.

73,

Chris, K9EQ

WiRES-X Automation

Yaesu does not provide a mechanism that allows the WiRES-X software to be controlled by another program, i.e., having another program switch to a certain Room when a net starts.

Windows does, however, permit another program to send events to a program. Each window, menu item, and dialog in a Windows program has a unique identifier. It is possible to use these identifiers to send "message" to the WiRES-X software.

The WiRES-X Automation Project's purpose is to bring together people who are interested in developing this technology and sharing their results.

To get things going, here are two mechanisms for automating WiRES-X:

1. AutoIT: <http://www.autoitscript.com>
2. Python – an excellent programming language found at python.org

Dave, N9TOW, provided the following information:

Packages I have installed on my WiresX system.

```
C:\Users\WiresX>pip list
```

```
comtypes (1.1.3)
```

```
pip (9.0.1)
```

```
pywin32 (220)
```

```
pywinauto (0.6.3)
```

```
setuptools (28.8.0)
```

```
six (1.10.0)
```

```
https://github.com/pywinauto/pywinauto
```

To install

```
pip install -U pywinauto
```

Script that executes changing channels on WiresX app

```
import time
```

```
from pywinauto import Application
```

```
app = Application().connect(path="C:\Program Files  
(x86)\YAESUMUSEN\WIRES-X\wires-X.exe")
```

```
app.WiresX.menu_select("Connect(C)->Connect To(T)")
```

```
time.sleep(1.5)
app.InputID.Edit.set_edit_text("21493")
time.sleep(.5)
app.InputID.OK.click()
time.sleep(4)
dialogs = app.windows()
##app.Dialog.CloseButton.click()
```

Listen to MNWis Online

You can now listen to MNWIS online. Just follow this link with your favorite browser:

[MNWIS Online](#)

As an alternative, you can listen to MNWis using the Broadcastify app on your smart device. Download Broadcastify and search for "MNWIS".

[Apple Download](#)

[Android Download](#)

MNWis Available on Hotspots

The MNWIS Room is on FCS003-23.

We are also experimenting with a YSF reflector. Currently it

is only running during the net. It is located at "US MNWis 21493". Give it a try

MNWis has been available via hotspots since 31 July, 2016

Yaesu Published Webinars

Webinar links are to YouTube

[DG-ID and DP-ID Webinar 9-Sep-2018](#)

[Overview of Fusion II 9-Sep-2018](#) (YouTube)

[Overview and setup of the DR2X \(DP-ID\) 9-Oct-2018](#)

[Overview and setup of IMRS \(DG-ID\)](#)

[WiRES-X: Everything you wanted to know part 1 27-Nov-2018](#)

Mobile WiRES-X Nodes

FT2D, FTM-100, and FTM-400 Can Now Be Used As Mobile Node

Yaesu has now released software and firmware that enables the FT2D, FTM-100 and FTM-400 to be used as a WiRES-X mobile node. You'll need to update the radio's firmware and install WiRES-X software version 1.510 from HRI-200 and radio downloads at Yaesu.com. Instructions are available at the download site.

A document listing the current firmware and software for Fusion radios is on the [Fusion Help page here](#).

Stop by for the MNWis Monday night Fusion Technical Net for more information.