Problems with WiRES-X 2.X

Category: WiRES-X
August 9, 2026
Note: Yaesu has now released version 2.01. Our recommendation remains unchanged. This version mostly addresses those few people who need the software to work with the FT-1.

Yaesu recently released version 2.x of the WiRES-X software. These are the primary things to know:

- It is the only version of WiRES-X that will work with the FT-1. (But hopefully you're using your FT-1 to do more than make a WiRES-X node!)
- IPv6 is now supported. Good news for people stuck with ISPs that only do IPv6.
- There is an IPv4 compatibility mode that can be used.
- Version 2.x IS NOT backward compatible. Version 2.x CANNOT connect to rooms running 1.5.x.
- Most Room operators are sticking with version 1.5.x because of bugs and other issues. This means that if you update to 2.x, you will NO LONGER be able to connect to almost all of the popular Rooms.
- Version 2.0 is buggy. It is a complex update to the software and it may take several releases before Yaesu gets it stable.

Note: I've been taking one for the team and have been running FusionTech on 2.01 for more than a month. These are the reasons why I don't like running a room on 2.01.

- This really is a problem! If you're running a room and need to reboot WiRES-X, you want that reboot to go as fast as possible. With a 1.5.7 reboot, nodes would rejoin the room immediately. This is because the room still exists in

the Yaesu list server. Nodes will automatically try to rejoin every 30 seconds. Thus within 30 seconds the nodes have rejoined. This is NOT TRUE with 2.0.1!!! If a 2.0.1 room is rebooted, it sends a command to the list server to DELETE THE ROOM'S ENTRY IN THE LIST SERVER. This means that a node has to propagate through the system. The result is that it can take more than 10 MINUTES for nodes to rejoin. Very bad if this happens during a net! (It has happened to me a number of times.)

- But it gets worse. I use the repeater that hosts the room to bridge to our YSF reflector. However the repeater can't rejoin the room until the Yaesu list server updates the local list of rooms. Depending upon where your room is in the list, this can take 10 to 15 minutes. So in the middle of the net, the YSF users are disconnected from WiRES-X for a very long time as is true for all users on the local repeater – which probably includes net control.
- What???? The node running on the same WiRES-X instance as the room cannot connect to that room right away? Nope. Even though both are running in the same software!
- Here's what's happening. 1.5.7 used a distributed model for providing rooms. When connecting to a room the traffic between your node and the room goes from you WiRES-X to the WiRES-X software used to host the room. Not so with 2.0.1. With 2.0.1 your WiRES-X connects to a Yaesu server in Japan. All of the rooms are hosted in Japan, not on the rooms WiRES-X software. This has advantages and disadvantages.
- Advantages: Port forwarding is not required since your WiRES-X software connects directly to a public server – like pulling up a web page. No port forwarding is needed. The efficiency of hosting on a server is much better than hosting on a consumer IP service. (Performance is mostly constant and not degraded by Netflix.) This also

integrates well when you don’t host a room and directly connect your computer to your PC via PDN. In this case Yaesu uses TCP to connect directly to the server in Japan. Japan then converts the TCP to UDP and sends the traffic to the host WiRES-X. Now both methods do the same thing.

- Disadvantages: Yaesu chose to use TCP to send audio. Nobody does this. When you listen to a podcast, music online, or watch Netflix, this is all done over UDP. Why? TCP requires a hand-shake between the server and the client. It also retries transmissions if one is lost. It guarantees that all packets are received in the correct order? Nice, eh? Until a packet or two get lost and a retry occurs. This can cause a delay of up to a second based on the ping times between the two computers. Thus the audio will have a noticeable stutter even though only 100 ms of data is lost. TCP is much more compute intensive than UDP. UDP does not have a hand-shake. Once the connection is made it just starts streaming packets to the client. The packets may arrive out of order if the Internet path between the two computers changes from one packet to the next. One would think this is terrible, but it isn’t. It means that UDP keeps on sending even if random packets get dropped. You’ll hardly notice a dropped packet (100 ms, or part of a word) versus loosing the time it takes for a retry – a second or more. You won’t notice a dropped packet on opening a web page, but you will certainly notice it when watching video or listening to audio.
- Why is Yaesu using TCP? Nobody else uses it for this purpose? I honestly don’t know. We know that UDP works fine because with 1.5.7 that’s what we’re using. Maybe they’re using TCP so that a port is open to connect to a web server? Or something? This makes no sense as they could just as easily have the UDP port open. That works.

That’s what we do with 1.5.7. TCP requires more server overhead, while UDP is very, very simple.

- Yaesu never needed users to open ports!!!! Here’s how that works. Let’s say A and B what to have a VoIP conversation using UDP. But A can’t talk to B because B doesn’t have any open ports. That where a public server, C, comes in. Both A and B connect to C and tell C who they are and what their IP address is. When A want’s to call B, A contacts C with his intentions. C responds with B’s IP address. Meanwhile the server lets B know that A is trying to contact him so B then sends a UDP packet to A. Meanwhile A is sending a UDP packet to B. These packet probably never get received but what they do is important. Sending a UDP packet causes your NAT router/firewall to open an incoming port on the same port you send the UDP packet from. Thus A and B have both opened a UDP port known only to A and B, and they can start sending packets directly to each other without involving C. This is how the older version of WiRES-X should have worked! This is what everybody else does. It’s not a secret.
- YSF works and an open port is not required. In many ways YSF is a lot nicer than WiRES-X. It only uses UDP. When a YSF user wants to connect to a reflector, they look in a Host file for the IP address and port number of the reflector. They send a UDP packet to that address, opening up an incoming port. The reflector then sends traffic to that port. YSF only requires an open UDP port if a reflector is being hosted and most people are not doing this. There are other advantages to YSF over WiRES-X which I will point out is a subsequent post.
- Single point of failure. Since ALL WiRES-X traffic is now going to through one service, failure of that service shuts down the ENTIRE WiRES-X network. This can happen when Yaesu does a reboot. No matter when they do a reboot

it will be prime time for someone on this earth. That wouldn't happen with 1.5.7. Only people that have just booted WiRES-X will notice that the list server is down. That's because everyone who has WiRES-X running already have a cache of all the nodes and rooms – they're just not getting any updates. But with 2.0.1 everything goes down. No longer can we recommend WiRES-X networking for anything related to emergency services. WiRES-X IS NOT RECOMMENDED FOR ANY EMERGENCY SERVICES ORGANIZATION. YSF is a much better choice as you have total flexibility in how the networking is done and where the server is.
- One last disadvantage is that it's impossible to connect to 1.5.7 rooms. A number of rooms that I normally connect to have been inaccessible since running 2.0.1. It is inconceivable to me why Yaesu chose to make 2.0.1 incompatible with prior versions of WiRES-X!

It is for these reasons that FusionTech recommends AGAINST updating to WiRES-X 2.x. Stay tuned to this website or the FusionTech net for any updates.

WXScheduler

Category: Automation,Projects,WiRES-X
August 9, 2026
Update: WXScheduler 2.9.9 is available. The 2.9.9 installer is here.

Here's what has changed with 2.9.9:

WXscheduler v2.9.9 updates: Several fixes related to missing

automation delays that are required for properly updating Wires-X Settings.

WXscheduler v2.9.8 updates: – Now restoring Wires-X app to the Desktop only when Wires-X is currently minimized and WXscheduler is about to perform automated Wires-X Settings updates. – If the Wires-X app was minimized at the start of automated Wires-X updates, then the Wires-X app will be minimized again after the automation actions have been completed. – Validation of 5 digit "Return to RoomID" having an even first digit, has been removed.

WXscheduler v2.9.7 provides many improvements. The key ones are: – Compatible with Wires-X versions 2.010 and v1.5xx – Faster automation of the scheduled Wires-X Settings updates that are independent of CPU speed and background load. – Replaced main window's "Scheduler list" with "View/Modify Schedule" button. – "Wait for Wires-X.exe to Restart" functionality that maintains a valid automation connection to Wires-X. Additional information about WXscheduler can be found in the installation folder: WXsched-info.txt WXsched-license.txt WXsched-Python.txt WXsched-updates.txt WXscheduler.pyw (source code) for those who still want to run it as Python program. All I ask is that you pass along any improvements to w9lbr@arrl.net for the benefit of the amateur radio community.

2.9.1 fixes problems with some Windows installations not working because there no longer is a "Programs (x86)" directory.

WXScheduler is a program that allows WiRES-X nodes to automatically switch to a Room or Node at a predetermined date and time. It's written in Python, but we now have a Windows installer which does not require the installation of Python.

Description

WXScheduler is a program that allows your WiRES-X node to be connected to another Room or Node at a predetermined day of week, hour and minute in the specified timezone. If your browser balks downloading WXscheduler-v290-Install.exe and you cannot override the warning, then rename the xxxxx.crdownload file using File Explorer or CMD. NOTE: Recent Windows Security updates may rightly so quarantine WXscheduler-v290-Install.exe. Detecting it as a Trojan. Well that is correct only on the basis that it will install another EXE file You can override this by going into Windows Security –> Protection History and find the most recent “threat” and click the Restore link. Finally this especially occurs if you are using Windows’ OneDrive. Which I highly recommend that you do not use! WXscheduler is written in Python, converted into a Windows EXE, and delivered as an installable Windows application for Windows-7 or higher. Which means the User no longer has to install Python and library packages to run WXscheduler.pyw. Additional information about WXscheduler can be found in the installation folder: WXsched-info.txt WXsched-license.txt WXsched-Python.txt WXsched-updates.txt WXscheduler.pyw (source code) for those who still want to run it as Python program. All I ask is that you pass along any improvements to w9lbr@arrl.net for the benefit of the amateur radio community.

The 2.9.0 update changes:
-Address varying CPU speeds and their impact on the automation

-Fixes the problem when AccHistory.log contains garbage in a location string, causing a non recoverable exception. Now the corrupt string is displayed in place of the 6 character GridSquare string.

The old 2.6.1 installer is available here. The Github repository

is here.

If you experience problems, please use the Fusion Technical Net to report them (preferred), via Github issues, or email.

Revision History:

WXScheduler was written by Bill, W9LBR. It has been packaged as an .exe with a Windows installer by Chris K9EQ. In addition, the following changes have been made.

-2025-09-15 Address varying CPU speeds and their impact on the automation

Fixes the problem when AccHistory.log contains garbage in a location string, causing a non recoverable exception. Now the corrupt string is displayed in place of the 6 character GridSquare string.

– 2023-02-12 2.6.1a Clicking the ‘x’ button on the scheduler menu no longer causes the program to crash

– 2023-01-10 2.5.1 Removed radioID error messages from the log (there were too many of them and they didn’t add value)
Added an icon for the program

The installer creates a WXScheduler directory under the WIRESXA directory where the Yaesu software lives. It installs the following files:
– Readme.txt, this text
– WXScheduler.pyw, the Python source file
– WXScheduler.ico, the icon file
– WXScheduler.exe, the executable
– Uninstal.exe, the program to uninstall WXScheduler

The uninstaller removes these files and the directory. It does not remove the .json

configuration file one directory above.

The packaged version displays the GUI without a shell console. If you want the console as well,
rename the file to *.py instead of *.pyw. You'll then need to open up a shell (command prompt)
and launch with a 32-bit Python interpreter as in: "C:\python32 WXSheduler.py".
The source code is available at https://github.com/K9EQ/WXScheduler.
Also see HamOperator.com for more information about WXScheduler.

Note that the WXScheduler is either stored in the \OneDrive\Documents\WIRESXA folder, if it exists, or in the user's Documents\WIRESXA folder. So if you can't find it in your WIRESXA folder, you may have OneDrive so look there.

WXscheduler v2.5 updates:
– Schedule settings now require a Timezone parameter.
– Compensation for Daylight Savings Time is now automatic on a per timezone basis
– Load Python Timezone package using Windows Command Prompt:
pip install pytz
– WXscheduler.cfg is now more portable and user editable
– User's HOME path is now resolved at run time and no longer saved in WXscheduler.cfg
– JSON data fields are now saved in pretty mode
– When WXscheduler-v2.5 detects an existing WXscheduler.cfg without timezone data fields,
the user will be prompted to set their local timezone value and all existing events will
be updated with the local timezone value.
– WXscheduler's main window:

– Scheduled events are displayed in chronological order
– When the main window is moved, its new position is remembered. And all sub-windows
will be opened in the same position. This replaces PySimpleGUI's default of centering
window within the display.
– [Add Event] and [Delete Event] buttons have been replaced with the [Scheduler] button that
when clicked provides a new window with buttons: [New] [Delete] [Update] [Cancel]

WXscheduler v2.4 updates:
– Additional exception debug information
– Changed main window title from "Wires-X Scheduler" to "WXscheduler (v…)"

WXscheduler v2.3 update:
At startup determines whether Microsoft OneDrive is active or not and locates
the Documents/WIRESXA and user/Desktop folders accordingly.

WXscheduler v2.2 updates:
At startup verifies:
– Wires-X.exe is accessable at its standard location
– /Users/????/Documents/WIRESXA folder is accessable (where WXscheduler.cfg is stored)

WXscheduler v2.1 updates:
Better displays exception information
Shows expected WIRESXA pathname to last heard data file

WXscheduler v2.0 update adds:
Display Last Heard information that the Wires-X application updates once a minute.
If available, Lat/Lon displayed as 6-character Grid Square.

Wxscheduler v1.3 updates:
Last Heard information now correctly displays the newer Yaesu models
(i.e. FT5-D, FTM-200, and FTM-500)
Desktop\Wires-X_Last_Heard.html contains hypertext encoded callsigns
that perform a QRZ.com callsign lookup.

Prerequisites (versions below are what was tested on):

Windows PC (win7 and higher)
Wires-x App (Ver-1.550)
Python 3 (32-bit) because Wires-X App uses the win32 UI
site-packages: PySimpleGUI, pywinauto, pytz

Installing Python:

Recommend going to https://www.python.org/downloads/windows/
to find a Python release that matches Windows on your PC.

After installing Python, use a CMD window to load site-packages:
– pip install pywinauto PySimpleGui pytz

Installing WXscheduler.pyw:

Copy WXscheduler.pyw to your Desktop

Double click on the WXscheduler.pyw icon to launch the program.

Hint: To see any startup or run time issues:
– Open a CMD window
– Copy WXscheduler.pyw WXscheduler.py
– WXscheduler.py

Customizing WiRES-X

Category: WiRES-X
August 9, 2026
In this post we discuss how the operation of WiRES-X can be customized through the use of the Windows Registry.

Okay, here’s a bit of a WiRES-X secret. Many of the settings which cannot normally be changed, can be changed. These include the call sign, location, etc. In fact, it is possible to use an HRI-200 with one node number with a different node number. This is necessary should one be running a room and need to switch to a different server located elsewhere.

Here’s an example that we have used. Room #21493 is FusionTech. This HRI-200 belongs to Pete, AD0MI, now located in Idaho. However the server we’re using is located near San Diego and has a number of 21811. If you check, the callsign is not AD0MI as registered with Yaesu, but K9EQ. Being able to do this is a necessity for popular rooms that cannot tolerate a room being down because an HRI-200 is being sent in for repair.

This data is stored in the Windows Registry. In the Windows search bar, type in the text “regedit”. The Registry Editor will open. This is where Windows and applications store a lot of information such as window location and size, settings, preferences, file locations, etc. Be very careful when using regedit as any changes you make are immediately effective and there is no undo. One can really mess a computer up with regedit!

In Regedit, press CNTL-F or use Edit->Find. Enter the search term “YAESUMUSEN” and deselect everything except “Keys”. Wait as it searches. The registry is very, very large and Microsoft’s

search function is not very efficient.

Expand the WIRESXA folder. These folders contain all of the operating parameters of the WiRES-X software. Click on “USER_INF”.
 In the folder you will see the key “callsign” and the value of whatever callsign you’re using.

To change an item, stop the WiRES-X software. Double click on the key whose value you wish to change. Type in the new value then click “OK”. Restart the WiRES-X software. If you changed the callsign, you’ll see that the callsign has changed.

In this folder you will also find items like the HRI-200 serial number, room number, and node number. If you change this information to match a different HRI-200, the software will then take up the identity of that HRI-200. Thus you can swap server locations or HRI-200s without changing the room number. Note that the node and room numbers appear under other folders and these must be changed as well.

Not all of the values are obvious as to what they mean but probably contain values we wish we could change. The DATA_COM folder has some interesting values in it. It would be interesting to perform experiments just to see what all of these things do.

I should note that in our long use of this product, we have identified many strange bugs. One common one is that there is no outgoing traffic even though incoming traffic is fine and everything looks like it is working. Usually restarting the software solves the problem, but not always. In another case transmissions from a YSF bridge were blocked but everything else worked. Fixing this took three months. It was necessary to uninstall the software, delete any residual configuration files, AND delete all the data in the registry. It was then necessary to reinstall the software and repeat the authentication process and re-entering all of the parameters manually before it would work again. So if you have a problem you’ve never been able to

figure out……

Yaesu’s list server contains some of the information in the registry. In some cases it’s only updated when the software is registered (callsign, node number, etc.) and some of it is updated every time the software starts (city, state, location data, etc.). In this last category the workaround is to run a script that updates the registry just before WiRES-X is started.

If you find this stuff interesting, please stop by our FusionTech net which is Monday, 7:30 PM Central in the FusionTech Room / YSF 21493. We like talking about the technical nature of our toys. We can also be found on HamOperator.com.

Solving WiRES-X ISP Connection Problems

Category: WiRES-X
August 9, 2026
If you want to set up an HRI-200 on Starlink, AT&T, or use you phone as a hotspot, there’s a problem. You need to open incoming ports and that can’t be done in many cases. The answer is to use a VPN to tunnel past your ISP to the raw Internet. One solution for some carriers is to use the PDN mode which does not require incoming ports to be open. But that may still not work.

The other problem is we’re running out of IPv4 space. New ISPs, such as Starlink, don’t have a lot of IPv4 addresses. IPv6 solves this problem plus many others. Thus it’s no problem for each ISP subscriber to have thousands of unique IPv6 addresses.

Unfortunately the WiRES-X software DOES NOT support IPv4. At all. IPv6 will become more and more dominant and this will just become a bigger problem.

If you set up a PDN (no HRI-200), a commercial VPN will work well. With a VPN, your IPv4 traffic will be sent in an IPv6 tunnel. At both ends, your node and the Internet, IPv4 addresses will be used. This works because PDN does not require any incoming ports to be open. It may be that Starlink provides this service automatically. Don’t know.

If you use an HRI-200 then YOU NEED to open an incoming port. Most every commercial VPN does not permit assigning specific ports for incoming traffic. This is because they will share a single IPv4 address between dozens of subscribers. A couple of VPN providers do permit this. I have used Golden Frog’s VyprVPN in the past for this purpose. VyprVPN is a “high end” VPN service and will cost you quite a bit more than free or low-cost VPNs ($60/yr). You’ll need to turn off the NAT firewall so make sure your exposed computer is well protected! This will give you complete, unfiltered access to the Internet – something you can’t get with your ISP!

See https://www.vyprvpn.com/

Node Cycling In and Out

Category: WiRES-X
August 9, 2026
This post addresses a WiRES-X node that constantly leaves and rejoins a room.

This type of problem is almost always the result of a networking issue. When certain networking problem occur the HRI-200 software will wack out (a technical term). Once the software has been wacked out, it will continue to cycle in and out of whatever room it is connected to. Restarting the WiRES-X software usually solves the problem.

Router problems:
1. Turn off UPnP. It should never be used!!!!!!
2. Turn off all quality of service features like those to improve Xbox gaming or VoIP functions. (Yes I know we're using VoIP so you'd think it would help.)
3. In fact turn off everything you don't have to have.
4. Some routers are total crap and need to be rebooted every few weeks.
5. You're not using Wi-Fi are you? Wired is always better than wireless – we should know! Drop-outs in Wi-Fi can cause this issue.
6. Did I mention, turn off everything on the router you don't need?

ISP problems:
1. Something as simple as the ISP dropping your connection for a few minutes in the middle of the night when they're doing maintenance.
2. The ISP just doing a bad job of getting packets to you during prime evening Netflix viewing.
3. The ISP switching your IP address. (I think CenturyLink just does this for the fun of it.)

Computer:
1. Make sure Win10 doesn't put the USB interface to the HRI-200 to sleep. It really, really likes to do this. Just going to the obvious place in the control panel to turn this feature off doesn't really turn it off. See the documents on running Win 7/10 24x7x365 remotely in the Fusion Help section at HamOperator.com.
2. Make sure the computer isn't crap. Cheap computers may not have quite the reliable communication interfaces that we need.

3. Banish all RFI and ground loops. Make sure RF is not getting into the USB interface to the HRI-200. This can cause message errors and really screw things up.
4. Make sure you computer actually has enough power to run the HRI-200 reliably. Get one of those USB power thingies and make sure you have a solid 5VDC under load.

Over the air:
1. Bad data on the WiRES-X network can mess things up. This happens mostly with hotspots that are bridged into the WiRES-X room. On MNWis and a few other networks we've banished this problem by banishing FCS and running YSF server software that banishes bad data from hotspots. So if you have this problem in one room, say AmericaLink and not others, there may be nothing you can do.
2. Yaesu radios that are running really old firmware or have not had the firmware updated correctly (I.e., "I did the main CPU but I'll get around to the DSP later.")

Side note:
The port check should not be relied upon to definitely prove it's working or not working. The test is not exactly the same as actually communicating with the Yaesu list servers and the room server. So it can lie to you with false positives and false negatives. This is because the port check uses the Yaesu servers in Japan to perform the test, not the room server you're trying to connect to. Routing issues can cause one to work whilst the other doesn't.

You may also experience a similar problem if the Yaesu list servers are not able to keep up with the demand or if maintenance is being done. (When they do maintenance in the wee hours of the morning, that's right in the middle of the day for us.)

Repeater Lock-Ups

Category: Hotspots,Repeaters,WiRES-X
August 9, 2026

Many people have reported Fusion repeater lock-ups that required the repeater to be shut down (power removed) and rebooted. I've had repeater lock happen as well with an HRI-200 directly connected to the DR2X. In this particular case I believe the event was triggered by sending a number of commands from the radio (such as the GM mode) which resulted in a race condition within the firmware.

I have previously observed that invalid data (I.e., with a lot of bit errors or badly formed packets from a hotspot) can also cause the lock-up. My assumption is that this is a problem of not validating inputs to a function prior to executing the function. The invalid data is not handled correctly and causes one of the lock-up faults I mention below to occur.

I have observed that data is more likely to be invalid if there is a lot of noise on the signal. Also a major source of problems are badly behaving hot spots that inject all sorts of crap into the data stream. There's no control over these devices and it just gets worse as there is more work being done converting from one thing to another thing, etc.

Communications software is EXTREMELY HARD to write and much of what I see out there is poorly documented (which leads to bad code) and poorly tested.

Reflectors such as YSF or FCS are pretty dumb when they relay signals. They look for an incoming packet from one of the connected hotspots and just send that same packet out to all the other hotspots (including bridges to WiRES-X). Garbage in, garbage out. To address this I have modified the YSF reflector

software to be very aggressive at dropping packets with CRC errors, invalid data in fields, and most importantly invalid FICH headers. The FICH header is extremely important for everything to work right. (You can look up FICH headers in the System Fusion documentation published by Yaesu.) This modified software is running on MNWis, Princeton Kentucky, MVARC (Idaho) and America Link. It has dramatically reduced the number of problems with WiRES-X problems.

So what is a “lock-up”. A lock-up occurs when the processor looses track of what it was doing and either halts or ends up executing invalid code. In either case it is no longer processing the code it was intended to run. Events that cause the processor to, using a technical term, go out to out to lunch can result of errors in the software that cause it to happen, corrupted memory, invalid states in a state machine, incorrectly handled race conditions, or a really nasty thing called priority inversion (where a high priority task which is pre-empting use of the CPU is waiting for a low priority task to complete, which can’t complete because, well, it’s not high priority.)

How to solve the problem?
First of all, the repeater firmware should never crash. Since it is very hard (but not impossible) to write code with no bugs, most continuously operating real-time systems will incorporate a “watch dog timer”. It is the purpose of this timer to detect when the “wheels have fallen off the processor” and perform a reset.

Since we can’t modify the repeater’s software, there are some actions we can take.

First would be to prevent bad data from getting into the system. A careful examination of the logs possibly combined with audio recordings could be helpful.

A quick fix might be to install a timer on the power supply such that the repeater's power is removed for a short time once or several times a day.

I have a technical concept for a generic lock-up detector that would reboot the repeater in the event of a lock-up without the need to make any modifications. Not sure if I'll work on this because I don't know how much interest there would be. This solution would also work with non-Yaesu repeaters.

One further note. I made an observation that this problem was more likely to occur on 2 meters than on 440. Why? My theory is that there is a lot more interaction with the environment on 2 meters. Computer networks, small power supplies, computers, almost everything digital generally creates a lot more noise on 2 meters than it does on 440 MHz. I observed a situation where mixing products were occurring on a 2 meter repeater in a noisy environment. This repeater would lock up regularly. My theory was that the mixing products contained a lot of noise along with the C4FM signal causing a high number of bit errors and thus crashing the repeater. This was observed on a DR1X.

If you are struggling with this problem, I suggest you contact Juan at Yaesu Customer Service with your details. Provide Juan with as much information as you can.

Running two HRI-200's on one

IP Address

Category: WiRES-X
August 9, 2026
I use VyprVPN from GoldenFrog. It is necessary to turn of NAT for your account. That means that each VPN connection is a raw connection straight to the Internet. YOU HAVE NO PROTECTION OTHER THAN WHAT YOU PROVIDE!!! (The same company will rent you a VPN that you can host on your own cloud provider where you can control port forwarding – but I suspect this is well beyond most folks.)

I've been using this solution for over a year. If you've ever seen me demonstrate WiRES-X, I'm either using AT&T/cell or the local Wifi along with the VPN to get access to the incoming ports.

Good news! My fully patched and maintained Win7 boxes have not yet been hacked. One of which has been online via VPN for over a year.

Most VPNs won't work because they don't give users their own IP. The VPN host shares IPs between multiple users which means they can support more users with fewer IPs (and better obscures your traffic). I'm only aware of two VPN providers that allow you to specify incoming ports.

The previous post of using the Ham's IP space is a good one. I plan on giving that a try when I have time. But VyprVPN is very much plug-and-play.

Note: You will need to PAY for VyprVPN (~$70/yr). The free service does not allow you to disable NAT.

What Does the HRI-200 Actually Do?

Category: WiRES-X
August 9, 2026
What’s the difference between using an HRI-200 and directly connecting to the radio?

1. It is the only option for connecting a repeater to WiRES-X.
2. It uses UDP communications sent directly to the Room. So network communications are very efficient.
PDN uses TCP instead of UDP. (Why use UDP? UDP is simple. If anything gets lost, it is dropped. Good for real-time communications. If a packet is missed with TCP, it will keep retrying up to a limit which is not good for real-time. VoIP, VPN’s, Netflix, etc. all use UDP.)

The PDN TCP packets (non-HRI-200) are sent to a server in Japan. This server converts the packets to UDP and then sends them to the selected room in the same way the HRI-200 does. So by using the HRI-200 you bypass a drip to Japan and a potentially problematic conversion from TCP to UDP and back to TCP for the return trip to your station.

Yaesu uses the TCP approach to get around forwarding of ports. The WiRES-X PDN software establishes a TCP connection with the Yaesu server. It’s through that connection stations can contact you even though you don’t have an open port. With the HRI-200, stations just get your IP address and send you packets directly. This is why PDN cannot efficiently host a room – all room traffic would have to go to Japan for the conversion process.

In a previous post (2016) I described what the HRI-200 does. Why it is the way it is is somewhat historical from the history of WiRES, WiRES-II, and now WiRES-X. Obviously an Ethernet port on the back with an embedded processor to run things would be a better way to go. I can only guess that the ROI is simply not there for Yaesu. They do, after all, have to make a profit.

One further note. It's probably a good thing non-Yaesu equipment doesn't connect to WiRES-X. The biggest problem in bridging hotspots (YSF, FCS, etc.) to WiRES-X is that the hotspots are bad neighbors. They generate bad packets, do crazy things, and can really trash a network. As a room operator we have no control over who tries to connect, so at least the Yaesu equipment has gone through compatibility testing. Believe me, very little testing is done on hotspot software.

MNWis (21493) bridges to YSF (21493) via a YSF Reflector where I have extensively modified the code to drop bad packets, control information, and all manner of things that shouldn't see there way into the WiRES-X node. I had to do this because we were constantly having hotspots connect and mess everything up. It got very, very tiring working with the (mostly new) hotspot owners to fix their problems. I mean the conversations were endless and constant! So I just dropped anything that looked bad in the YSF Reflector software and the problems went away as well as a few hotspots who could no longer connect or talk to people.

System engineering is critical. We need Yaesu to play the roll of system engineering to keep all the Fusion products working with each other. It's so nice that all the Fusion equipment basically works the same way and does the same thing all of the time!

BTW, you can bring your questions/comments to the MNWis Fusion Technical Net held Monday's at 7:30 PM Central in the above

mentioned locations.

WiRES-X Online/Offline

Category: WiRES-X
August 9, 2026
When a WiRES-X node keeps going online then offline, this type of problem is almost always the result of a networking issue. A networking problem will occur and will wack out the HRI-200 software. Once the software has been wacked out, it will continue to cycle in and out of whatever room it is connected to. Restarting the WiRES-X software usually solves the problem.

Router problems:
1. Turn off UPnP. It should never be used!!!!!!
2. Turn off all quality of service features like those to improve Xbox gaming or VoIP functions. (Yes I know we're using VoIP so you'd think it would help.)
3. In fact turn off everything you don't have to have.
4. Some routers are total crap and need to be rebooted every few weeks.
5. You're not using Wi-Fi are you? Wired is always better than wireless – we should know! Drop-outs in Wi-Fi can cause this issue.

ISP problems:
1. Something as simple as the ISP dropping your connection for a few minutes in the middle of the night when they're doing maintenance.
2. The ISP just doing a bad job of getting packets to you during prime evening Netflix viewing.

3. The ISP switching your IP address. (I think CenturyLink just does this for the fun of it.)

Computer:
1. Make sure Win10 doesn't put the USB interface to the HRI-200 to sleep. It really, really likes to do this. Just going to the obvious place in the control panel to turn this feature off doesn't really turn it off. See the documents on running Win 7/10 24x7x365 remotely in the Fusion Help section at HamOperator.com.
2. Make sure the computer isn't crap. Cheap computers may not have quite the reliable communication interfaces that we need.
3. Banish all RFI and ground loops. Make sure RF is not getting into the USB interface to the HRI-200. This can cause message errors and really screw things up.
4. Make sure you computer actually has enough power to run the HRI-200 reliably. Get one of those USB power thingies and make sure you have a solid 5VDC under load.

Over the air:
1. Bad data on the WiRES-X network can mess things up. This happens mostly with hotspots that are bridged into the WiRES-X room. On MNWis and a few other networks we've banished this problem by banishing FCS and running YSF server software that banishes bad data from hotspots. So if you have this problem in one room, say AmericaLink and not others, there may be nothing you can do.
2. Yaesu radios that are running really old firmware or have not had the firmware updated correctly (I.e., "I did the main CPU but I'll get around to the DSP later.")

Side note:
The port check should not be relied upon to definitely prove it's working or not working. The test is not exactly the same as actually communicating with the Yaesu list servers and the room

server. So it can lie to you with false positives and false negatives.

You may also experience a similar problem if the Yaesu list servers are not able to keep up with the demand or if maintenance is being done. (When they do maintenance in the wee hours of the morning, that’s right in the middle of the day for us.)

And keep in mind the MNWis Fusion Technical Net held in MNWis WiRES-X and YSF # 21,493 Monday nights at 7:30 PM Central.

WiRES-X and USB Problems

Category: WiRES-X
August 9, 2026
If you’re having problems with USB problems when using WiRES-X with or without other programs, read on.

The WiRES-X software is pretty good at identifying the HRI-200 vs. connecting directly to the radio. Windows, however, is very bad at managing USB ports. The WiRES-X software makes two USB connections: One as a virtual COM port and the other as a sound device. Windows is also very bad at managing sound devices.

Virtual COM ports can be identified by the port number (a real pain as it is always changing – see below) or by identifying the hardware via the manufacturer and product number that is burned into the USB device. Generally software that does not provide an option to specify the COM port (as with the HRI-200) is using the hardware identification to establish the association.

When Windows sees a USB device it will enumerate it and create a driver for it. If you move it to another port it will, for reasons we can only guess at, create another driver. It will keep making new drivers as long as you have USB ports and USB devices left. Why, we can only guess. Periodically it may be necessary to go through a clean out all of these “drivers” that Windows keeps making and saving until the end of time. You can do this from the command line or from within Device Manager.

I have found WiRES-X to play nicely with other devices. For example I have no problem running hotspots, Ham Operator Deluxe, WSJT-X, etc. on the same computer that is running WiRES-X.

You may find it beneficial to use a USB 2.0 vs. 3.0 port. Why? USB limits the number of connections that can made per USB controller. Keep in mind that several of the ports on you computer may use the same controller. Also, each USB device may use more than one connection. As I recall sound devices generally use two connections: One for the audio and one for control. A USB 2.0 controller supports 255 connections. How do I know? I actually ran up against this limit using a couple of 16 port USB hubs. USB 3.0, while faster, does not permit as many connections. I don’t remember the exact number as it has been some time, but the number SIX sticks in my brain. As you might imagine, it is very easy to burn up six connections. Also keep in mind that the USB controller may be servicing hardware that is internal to the computer. It all depends on how the computer was designed. Top end computers will generally have a higher controller to port ratio. Bottom line: you may want to use USB 2.0 if you have been using USB 3.0.

Perhaps try something like this:
1. Blow away the WiRES-X and any conflicting software. Make sure everything is unplugged from USB.
2. Blow away all of the USB drivers. Just uninstall them. Don’t

worry, the files to recompile a driver are still on the computer. [Device Manager with admin. Enable option to view all devices. Delete drivers under COM ports. Delete drivers under sound devices (except for hardware built into your computer.)]
3. Install the WiRES-X software. Allow it to associate with the USB devices it needs by plugging in the HRI-200, etc. If you're not using an HRI-200, you may find it beneficial to use an external USB audio device for WiRES-X. If you can use USB 2.0 ports.
4. With WiRES-X happy running, install the other software. Plug in the necessary USB devices so it can similarly make the association. At this point everybody should be happy with both programs running just fine.
5. Write down which device was plugged into which USB port. In the future, always use the same USB port for the devices. We don't want Windows creating more drivers.

Good luck solving the USB problems!