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.

We’re Now FusionTech

Category: Fusion
August 9, 2026
MNWis has changed its name to FusionTech. Nothing else changes. The name was changed to better reflect the nature of the organization from Minnesota and Wisconsin midwest centric to global covering interests in technology and Fusion in particular.

The network is still reachable at Room and YSF 21493. The Fusion Technical Net is the same and the people running it are the same.

The FusionTech network is part of the Bakken Amateur Radio Society (BARS). BARS was formed as a club of amateurs who worked at Medtronic. Since that time the relationship with Medtronic

employees has become less connected and subsequently most members are not from Medtronic.

The founders of FusionTech are: Chuck, K0ORK; Pete, AD0MI; and Chris, K9EQ. The net started in 2014 and is the 2nd oldest net to be held using Fusion. (Milwaukee has the honor of being number one.)

Yaesu Repeater Lockup

Category: Fusion,Troubleshooting
August 9, 2026
It is well documented that the Yaesu DR-1X and DR-2X repeaters can lock up from time to time. When they lock up, they may:

- Not respond to any transmission or control function and
  - Transmit continuously or
  - Not transmit or
  - Transmit for the duration of the time-out timer, go off the air, key up and start all over again.
- The transmission may be digital “noise” or it may be a carrier

There is another form of lock-up in which an attached HRI-200 can be used to reset the repeater.

Why do they lock up?

We don’t know for sure why this happens as we don’t have access to the source code of the repeaters. But guesses can be made

from knowing what the problem is, how microprocessor and their software works, and the common mistakes programmers make.

The firmware consists of a number of modules that do specific tasks. Data is sent to a module, the module may call other modules, and it returns with possible data. The input data may contain values that are out of range for the module. For example one value might be mode where 0=FM, 1=DN, 2=VW, and 3=VW. But what if the module receives a 4? It should just stop and return to the calling module with an error indication. Most likely the solution will be to ignore the request and stop processing. That's what should happen. But a sloppy programmer may NOT VALIDATE INCOMING DATA and allow the '4' to be processed where the results may be unpredictable since no code was written to handle this case. There is good evidence that Yaesu software and firmware DOES NOT VALIDATE INCOMING DATA resulting in an UNHANDLED EXCEPTION.

There are other types of invalid data which should not be processed. The Fusion digital data might have uncorrected errors in it, thus it's impossible to know what the correct values are. A field of data that is sent may be longer than it should be. This problem is commonly known as a BUFFER OVERFLOW. There is some evidence to support that a buffer overflow is occurring. Since the receiving buffer must be limited in size, too much data may cause the firmware to overwrite values that are past the end of the buffer. These may be critical values that are needed for the correct operation of the repeater.

The repeaters use a real-time operating system. An RTOS is event driven and will perform a function when an event occurs such as a timeout, a signal received, a clock tick, a button push, etc. There may be cases where two events are related such that one must occur before the other. For example, one event may lock access to a hardware resource and the other event clears it. If

the order were reversed the hardware resource would never be accessible and the actions of the hardware may be unpredictable or stop altogether. This is known as a RACE CONDITION. It is very likely that this error exists in the firmware and that it may be combined with another firmware error. For example, a BUFFER OVERLOW may result in a RACE CONDITION which the engineers did not see.

As a result of the above firmware problems, it is not easily predictable what the microcontroller may do next. Some systems include a Watchdog Timer. This is a hardware timer that is set for a certain period of time, perhaps 10 seconds. The microcontroller's firmware periodically resets the WDT if it is running properly. If the WDT times out due to some sort of problem, the hardware resents the controller. We have some evidence that Yaesu has implemented a WDT in the DR-2X repeater. It is therefore apparent that the WDT has been turned off, set to a crazy long value, or that the modules that reset the timer are still running even though nothing else is.

A SOFT LOCKUP may occur if a RACE CONDITION removes access to the receiver and/or transmitter. While the repeater won't receive or transmit, the other functions are working. This means the repeater can be reset without needing to cycle the power. (There is no reset button like on your PC.)

The Evidence

The most difficult problems to fix are the ones that almost never happen. These lock-up problems are in the category of infrequent but very high risk (a drive to the mountain to reset the repeater. With remotely-controlled repeaters, a lock-up is an unacceptable risk. After all, other repeaters don't seem to have this problem.

(There is a repeater bug involving the CWID on FM, that is not being discussed here.)

It is very likely that these problems occur while a user is transmitting in the digital mode. We've not been able to observe a repeater suddenly locking up without some sort of input to key the transmitter. (However, since we haven't seen it doesn't mean it doesn't happen.)

My observations are that this is more likely to happen when a user is transmitting a WiRES-X command such as changing rooms.

It appears to be more likely if the user has a weak signal into the repeater.

It is more likely to happen on two meter repeaters than on 70 cm repeater.

Notice that each of these conditions have the word "likely". That's because there are unknown additional parameters which may be involved. For example, a weak digital signal will cause more digital errors. A particular error may be involved in creating the problem.

It may also be the result of a transmission starting at the exact time the microcontroller is performing a specific function, thus resulting in a RACE CONDITION. This problem is virtually impossible to observe.

It was previously mentioned that using a two meter repeater makes this problem more likely. The issue here is the amount of noise on the two meter band. All of our electronics and networks add noise to the environment. This noise level is much stronger on two meters than on UHF. There is also some evidence that the ethernet cables can deliver RF to the ethernet devices and these devices may act like a mixer. All of this causes more noise

making stations effectively noisier. The end result is more bit errors and buffer overflows.

What to do?

The correct answer to this section is to have Yaesu fix their bugs in the microcontroller – but don't hold your breath. This could be a very costly to find the cause since it is difficult to reproduce. Another problem is that there are no Fusion repeaters in Japan. (Only D* is allowed – that's along story.) Thus the Japanese engineers working on this project would have no way to experience the problem. (The repeaters can only be operated in a shielded room – thus no noise.) I've solved problems like this before and the first step is to make the problem happen more often while monitoring the CPU. Keep trying things until the probability of a lockup increases, then try doing more of that. The business model for this is that spending a bunch of money on this is not likely to sell more radios. I also believe that Yaesu farms out certain tasks, like firmware development, to outside suppliers. In any case there is no guarantee that the firmware engineers use radios outside of the lab.

If the issue is the result of a SOFT LOCKUP, sending a command to the repeater via an attached HRI-200 may resolve the issue. This may only require switching to the repeater page in the WiRES-X software an applying the repeater's settings with the same value.

If it is a HARD LOCKUP, the only solution is to reset the microcontroller. I.e., turn the power off then on again. Ouch!

- The simplest way to address this problem is to take a timer plug, set it to turn off during a time the repeater is not likely to be in use, then turn it on again as

quickly as possible. You may have to put up with a continuous carrier for a good part of 24 hours, but at least you won't have to drive to the hill.

- If there is Internet connectivity at the repeater site, you might be able to use a Wi-Fi enabled power switch to cycle power remotely. (I do this using my home automation.) Unfortunately this requires the action of a control operator.
- Since it is unlikely Yaesu will fix this problem, the ultimate solution is to have a circuit that will automatically reset a lockup condition. Thus we have the De-lockup Project (below).

De-lockup Project

Attached to this post is a project that will automatically reset a locked-up repeater. The concept was originated by Mike, W0IH. Mike informs me that he does not have the bandwidth to provide support for this project in the long term. It is likely we will be refining the project as time goes on. One thing I'd like to do is eliminate the need for a cheap SWR meter. You can obtain a zip file of the project from here.

The project uses an inexpensive Arduino Uno microcontroller board. In addition there is a relay board plus a few extra components. The circuit does not require you to even take the cover off of your repeater and therefore will maintain the warrantee. You can learn more about Arduino at The Arduino home page.

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/

FT70D High Power Problems

Category: Fusion,Troubleshooting
August 9, 2026
The other day I was working my Fusion repeater with my FT70D. Since I was "DX" I had the HT on high power. Although it looked like I was getting into the repeater just fine, I couldn't enter the WiRES-X control mode. However when I switched to low power I WAS able to control the repeater.

What gives? Why could I control the repeater (node) on high power, but not on low power?

Back at the house where my repeater is located, the problem still existed. I could use the WiRES-X control mode on low power but not on high power. Since I'm within 10 meters of the repeater, signal strength couldn't be an issue.

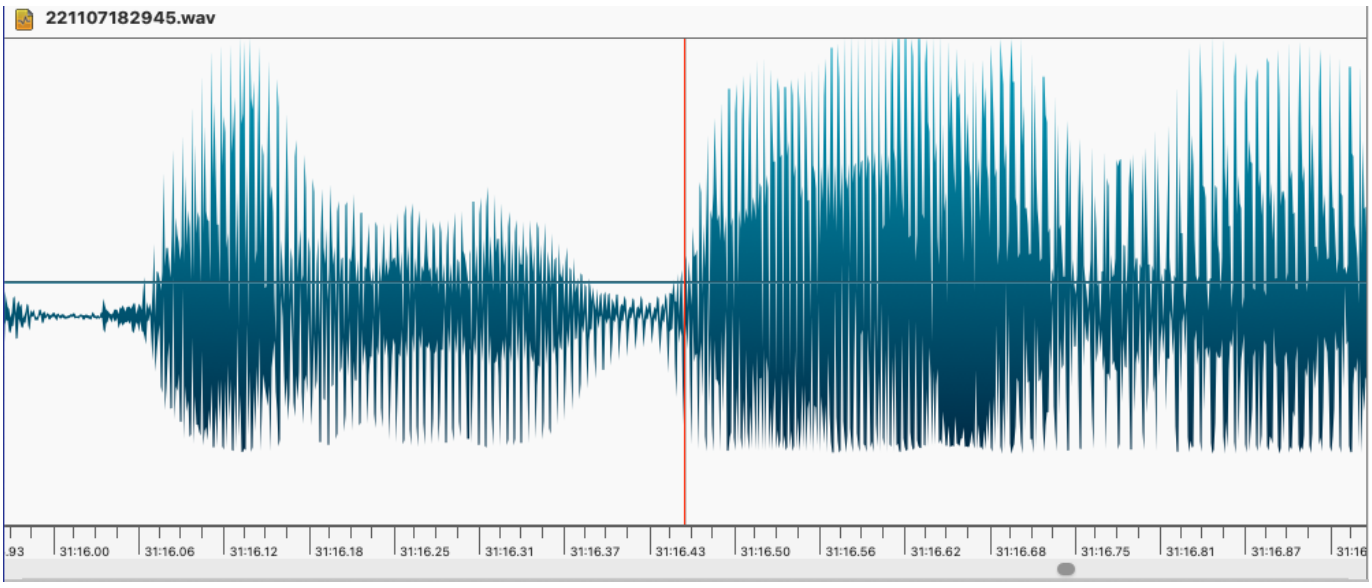
Looking at the output spectrum of the FT70 revealed the problem. On high power there were some spurs that shouldn't have been there. The spurs were much weaker on low power. Other radios didn't exhibit the spurs. Click on the images to compare and see the details.

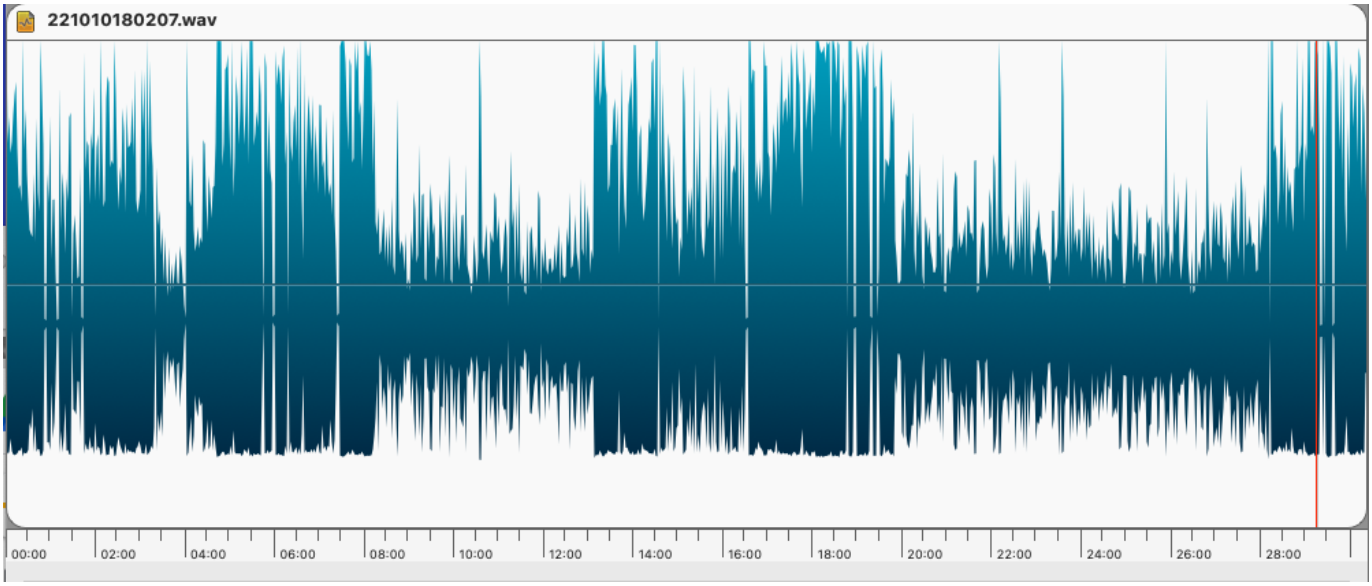
FT70D spectrum on low power

FT70D on High Power

C4FM Audio Quality

Category: Fusion
August 9, 2026
Photos show the waveform capture from an FT5 during a MNWis net. Note that they are not symmetrical around zero. The 2nd image has more detail and you can see that the waveforms are not symmetrical. The clipping sound is heard when the audio level hits the limits even though it is still not at 100%. Odd.





Waveform of MNWis net

IMRS LAN Firmware Update

Category: IMRS
August 9, 2026
I don’t know why Yaesu doesn’t make this public, but they don’t. You can download version 1.44 of the IMRS LAN firmware. This solves some problems with LAN connections constantly dropping.

Listen to 10 C4FM with an HT

Category: Fusion
August 9, 2026
During the Monday Night Fusion Technical Net, a comment was made that it was possible to use an FT5D to receive C4FM on 29.250.

This started a conversation about other HT’s as well. Thanks to W4IOD, AD0MI, N0PXT, and N0OQL, we now have the definitive answer for most of the Yaesu Fusion radios.

Background

Using 29.250 for C4FM has become popular as 10 meter conditions have improved significantly. The Yaesu FT-991(A) is the only radio that is capable of Tx and Rx on 10 meters as well as operating in Group Mode (GM). When the band is open, you may see or hear C4FM traffic on this frequency.

One way to check 10 meter band conditions is to tune AM down to the CB frequencies near 27.195 MHz. If you hear a lot of racket, then 10 is probably open.

The ‘A’ band must be used since the ‘B’ band on all radios cannot tune 10 meters.

Analysis

Note that none of these radios can transmit on 29.250. In addition if 29.250 works, then six meters will work as well.

The following radios cannot receive 29.250 MHz since they don’t tune down that low, bummer:

Mobile Radios: FTM-100, FTM-200, FTM-300, FTM-400, FTM-3200, FTM-3207, FTM-7250

HT’s: FT-70D

The following HT’s tune 29.250 and DECODE C4FM (show callsign and switch between DN/VW)* but do not provide audio:

FT-1D (US version), FT-2D, and maybe FT-3D

*Must be set to Rx DN or VW.

The following HT’s will receive C4FM on 29.250 but cannot engage GM. GM is handy for seeing the callsigns of stations that are beaconing.

FT-1D (Japan version) Interesting! Maybe the JP versions of other HT’s will do this as well.

The FT1D and FT2D can enter GM using the following process:

1. Enter 29.250 on the A side.
2. Enable both A and B if both sides are not enabled.
3. Switch to the B side.
4. Press the GM key.

The following HT’s can receive 29.250 and can enter GM. Note that they display the GM information but do not beacon:

FT-5D

Bottom line

The FT-5D has more C4FM capability since it allows reception wherever it tunes while other radios do not.