

ShackNerd

The guides

Every guide from shacknerd.com, in one document.

Version 1.0.0, 2026-08-28. The site copy is always current.

Contents

Getting started	3
Operating	6
Transmitting	9
CW	11
FT8 and digital modes	14
Logging	18
Reverse Beacon Network spotting	21
Remote operation	22
Radio reports: help ShackNerd support your radio	25
Troubleshooting	26

Getting started

From download to seeing your waterfall takes about ten minutes. The longest part is telling the radio to allow network control, which you only do once.

Once connected, the interface is five workspaces with a tab bar, at the bottom on a phone: **Operate** for the cockpit, dial, waterfall and push to talk; **Digital** for the native FT8 and FT4 station; **CW** for the keyer, decoder and trainer; **Log** for contacts and who heard you; **Setup** for everything else. Nothing stops when you switch tabs.

1. Download and run

Grab the build for the machine that will sit near your radio from the downloads. The file name carries the version and date, so your downloads folder always says whether you have the latest. Any machine that stays on works: the shack PC, a laptop, or a Raspberry Pi.

Windows. Download and double-click the .exe; there is nothing to unzip. The build is code-signed through Microsoft's own signing service, but SmartScreen also keeps a reputation score per publisher, built from how many machines have run their software, and a new signature starts at zero. If Windows shows "unrecognized app", click **More info**, check that the publisher reads Nicholas Dube, and choose **Run anyway**. The screen goes away on its own as the score grows. ShackNerd appears as an icon in the notification area, not as a window.

macOS. Unpack the archive to get **ShackNerd.app**, drag it to your Applications folder, and double-click it. The app is signed and notarized by Apple, so there is nothing to approve. It runs as a background app, no Dock icon and no window, because a Dock icon on a program with no windows made macOS wrongly report it as not responding. The browser page is the whole interface; to stop the program, press **Quit ShackNerd** under Settings, Help and support. The bare command-line binary is still inside the app at `ShackNerd.app/Contents/MacOS/shacknerd` for anyone who wants the `-probe` and `-help` flags.

Linux and Raspberry Pi. Unpack and run:

```
tar xzf shacknerd-*-linux-amd64.tar.gz
./shacknerd
```

On a desktop Linux launched from the file manager you get a tray icon; from a terminal it logs to the terminal.

Now open the address it shows, `http://localhost:7300` on the same machine, or `http://<that machine's IP>:7300` from your phone.

2. Let the radio accept network control

One-time setup on the radio's front panel. On an IC-7300 MK2:

1. **MENU** → **SET** → **Network**
2. Set **Network Control** to **ON**

3. Set **Network User1 ID** and **Network User1 Password** to something you will remember. These are what ShackNerd logs in with.
4. Note the radio's **IP address** on the same screen.

The radio and the ShackNerd machine must be on the same network for this part. Ethernet on the radio beats WiFi adapters for reliability.

3. Connect

In the browser, open **Settings**:

1. Choose **Network** as the connection
2. Enter the radio's IP address, the User1 ID and password
3. Press **Save and connect**

Within a couple of seconds the frequency appears and the waterfall starts moving. That is the whole setup for receive.

USB instead? If your radio is connected by USB cable, choose **USB**, press **Find my radio**, and wait the minute the sweep takes. On every connect ShackNerd also asks the radio which CI-V address it answers from, so a config that once belonged to a different Icom cannot leave the new radio silently ignoring every command. The sweep it asks every serial port in every language a radio might speak and fills in the port, speed and settings by itself, for Icoms and for Kenwood and Yaesu radios alike. Then pick the radio's sound card in the audio selectors below, so the waterfall, digital modes and CW decoding have something to listen to. On an Icom the waterfall updates several times slower over USB than the network can deliver, because that is a limit of the radio's serial port.

4. Turn on audio

In **Settings**, pick a receive audio rate. 16 kHz is ample for voice and easy on the network; 48 kHz is the full quality the radio offers. Then open the **Audio** section and press start. Phones require you to tap once, because browsers do not let pages make sound on their own.

5. What to try first

- Drag the tuning dial under the waterfall, or flick it to spin across the band
- Tap a signal on the waterfall to jump to it
- Tap the frequency readout and type one, like 7.200
- Open **Presets and bands** and save the frequencies you live on

When you are ready for more:

- Transmitting, including voice from your phone
- Remote operation from anywhere, the safe way
- CW, logging and FT8

If it does not connect

The troubleshooting guide covers the handful of things that actually go wrong, in the order they usually go wrong.

Operating

Everything you use while actually working stations.

Everything here lives in the **Operate** workspace, the first tab; the digital modes, CW and the log each have their own.

Tuning

There are five ways to move, and they all respect the **step** you have selected in the row under the dial (10 Hz to 5 kHz):

- **The dial.** Drag the tick-mark tape under the waterfall. Each 16 pixels of drag is one step, so the feel does not change when you zoom the scope. A quick flick spins it with momentum like a weighted knob; a slow drag stops dead where you stop.
- **Tap the waterfall.** Jumps to the tapped signal, snapped to your step, so with 1 kHz selected you land on a clean kilohertz rather than 7.203.47.
- **Drag on the waterfall.** Tunes continuously at the same rate as the dial.
- **Scroll wheel** over the frequency readout or the dial tape: one notch is one step. Over the waterfall the wheel also tunes, at half rate, but only once the cursor has rested there for a moment, so scrolling the page past the scope never moves the radio. A toggle on the Display settings page turns waterfall wheel tuning off entirely.
- **Type it.** Tap the frequency readout, enter 14.074 or 7200 or a raw count in hertz, press Enter. The readout shows frequencies the way the radio's own display does: 7.233.20.

Bands and presets

Presets and bands holds both. The band buttons remember where you last were on each band, including mode, exactly like the radio's own band stacking. Presets are frequencies you name and keep: your nets, your calling frequencies, the local repeater input you monitor.

Modes and filters

The mode buttons live below the scope, with the three filter presets and a width slider that shows the passband drawn live on the spectrum edge. The **DATA** toggle selects the radio's data input for digital modes; see the digital modes guide.

Split, VFO B, RIT

The **VFO B, split and offsets** section has A and B selection, A/B exchange, A=B, SPLIT, and RIT/XIT with a clearable offset. Working DX split is first class: type VFO B's transmit frequency into the box beside SPLIT, or tap **+1k**, **+2k** or **+5k** to set the customary offsets above your receive frequency. The scope span buttons underneath resize the panadapter without touching the radio's front panel, and the toggles row includes the antenna tuner: inline, bypass, and a TUNE cycle that is gated behind transmit permission because tuning transmits.

The meters

Receiving, the S meter reads in segments like the radio's own, with a peak marker that hangs at the loudest recent reading. Transmitting, the same meter becomes power, with SWR and ALC beside it. It switches the moment you key, so the first syllable of a transmission is metered as transmit, not receive.

Display settings

The **Display** section sets the waterfall floor, range, smoothing and scroll rate. These, and your tuning step, are remembered per browser, so your phone and your desktop can each keep their own.

The band map

Turn on **DX cluster** in Settings and every station the cluster hears about is laid along the waterfall at its actual frequency: tap a callsign to tune straight to it. Stations already in your log are dimmed; a station that would be a new country or a new band is highlighted. Spots age out after fifteen minutes. Small teal ticks along the bottom edge are FT8 stations recently decoded, pinned at the radio frequency they were heard on. The feed uses your callsign as its login and the default VE7CC node unless you name another.

Instant replay

While audio is playing, two buttons under the audio controls replay the last ten or thirty seconds of what you just heard, for the callsign that went by too fast. Live audio pauses during the replay and resumes when it ends.

Parks on the Air

The **Parks on the Air** section on the Operate screen shows who is activating a park right now, straight from the POTA spot feed. Tap a row and two things happen: the radio tunes to the activator's frequency, and their callsign lands in the quick-log bar ready to log. Activators you have worked before sit dimmed. On the other chair, type your park reference and **Spot myself** announces you on the feed at your current frequency and mode.

Contest mode

Contest mode lives in the **CW** workspace now, beside the keyer it is usually worked with: a serial number that increments itself with every contact, a dupe check scoped to this contest and this band (working the same station again on a new band is the whole point), and a rate meter keeping the last-hour pace. The exchange is stored in its own fields, so **Export Cabrillo** in the Log workspace produces a submittable log with every QSO line filled in; only the header's contest name is yours to complete.

Sharing the radio at a club station

Give each browser an **Operator name** in Settings and a roster appears above the cockpit showing who is connected and who holds the transmit key. **Ask for the key** raises a polite notice the current holder sees; nothing is ever taken, because grabbing a transmitter mid-QSO is not politeness. Every contact logged carries the operator's name, so one shared log still says who worked whom. When someone else moves the shared dial, the roster says so for a moment, "Dave tuned to 14.074 MHz", so a retune mid-QSO reads as a housemate rather than a haunting.

Transmitting

ShackNerd treats transmit as the feature that deserves the most care, because an unattended transmitter is a licensing problem, not a software bug.

Enable it

Transmit ships **off**. In **Settings**, enable transmit and set your callsign, then save. If you started the daemon with `-no-tx`, nothing in the interface can turn it on, which is the mode for an installation you want provably receive-only.

Push to talk

The **PUSH TO TALK** key sits in the main operating screen. It is hold to transmit: releasing it, switching apps, or the page going hidden all drop the key immediately. Underneath, the daemon runs its own watchdog and drops the key on its own if your device goes silent, disconnects, or holds a transmission past five minutes. One client transmits at a time; a second device trying to key while you transmit is refused.

The key's label tells you which microphone is live:

- **ON AIR phone mic** means your device's microphone is going out over the air
- **ON AIR radio mic** means the radio keyed but the audio comes from its own hand mic, and the text under the key explains why

Voice from your phone needs two things

HTTPS. Browsers only grant microphone access to pages served over a secure connection. The one-click Tailscale setup in remote access gives you that; so does any certificate of your own. On plain `http://` the key still works, but voice has to come from the radio's own microphone.

Transmit audio on. In **Settings**, below the receive audio rate, turn on **Transmit audio**. It takes effect on save, no restart.

Radio settings that matter

On the radio, **MENU** → **SET** → **Connectors** → **MOD Input**:

- **DATA OFF MOD** chooses where voice modulation comes from when the data mode is off. Set it to **LAN** to talk through ShackNerd. Note that while this is set, the radio's own hand mic is not the modulation source.
- **LAN MOD Level** sets how loud that input is. Around 50% is a sensible start; watch the ALC meter while speaking and adjust.

While keyed, a level bar under the key shows your microphone level, so "is my voice getting through" has an answer on screen: bar moving and power meter moving means you are modulating.

The safety net, spelled out

Three independent layers keep a transmission from running away: the browser releases the key when your finger leaves it or the page loses focus, the daemon drops it when keepalives stop arriving, and a five minute ceiling ends even a healthy transmission. Stopping a transmission is never queued behind other commands and is never refused.

Why your voice arrives a beat late

Transmit audio from the browser rides a small jitter buffer before it goes on the air. The buffer starts at a tenth of a second and only deepens on a connection that actually jitters, so a wired LAN stays quick while a phone on far WiFi trades a little latency for smooth audio. When you unkey, the transmitter holds on just long enough for your last words to finish playing out of the buffers, then drops; nothing you said is cut off, and the hold is capped so the key can never stick.

Slide to talk on touch screens

On a phone or tablet the key is a slider, not a button: slide the thumb most of the way across and hold to transmit, release anywhere to stop. A stray tap against a pocket or a palm does nothing, which is the point. The slide itself warms the microphone, so your first word goes out whole.

CW

A keyer, a decoder, and a passband scope that ties them together.

Before anything: two radio states that make CW look broken

The radio accepts a CW message and silently ignores it in two situations, so ShackNerd checks both and tells you instead of letting you wonder:

- **The mode must be CW or CW-R.** In any other mode the radio takes the command and does nothing. The panel offers a switch-to-CW button.
- **Break-in must be on.** With break-in off, the radio waits for a manual key instead of sending the message itself. The panel offers to turn it on.

Sending

Clicking a macro loads its text into the send box, where you can read what %C and %R will become and edit before anything transmits; the Send button is the only thing that keys the radio.

Type into the message box and send, or use the macros. Messages are limited to what the radio accepts, thirty characters, and validated before they key anything. **STOP** aborts a message in progress and is never refused, and the **WPM** slider beside it sets the radio's own keyer speed without a trip to the front panel. When the decoder is reading a station, its measured speed shows beside the Decoded heading, the number that decides whether to answer.

Macros cover a whole contact out of the box: CQ, answering, a report, exchange, handover and 73. Two placeholders make them reusable:

- %C becomes your callsign from Settings
- %R becomes whatever is in the **Call** field

Type the other station's call once and every macro fills it in. Edit the Info macro to carry your actual name and QTH.

The trainer

The CW panel ends with a trainer: copy, groups, QSO, text, pileup, send, stats, and a reference chart. Your progress follows you: every browser pointed at your station shares one record through the daemon, so the phone picks up at the level the desktop reached, and the backup file carries it too.

The **chart** tab is the whole code on one screen: letters, numbers, punctuation and the common prosigns, each tappable to hear it at your current speed, prosigns run together the way they sound on the air. It is the reference, not a drill; nothing there is scored.

Copy runs in sessions of twenty five characters with a report at the end. You start with K and M together, Koch's actual method, because telling two rhythms apart is the skill and a single character contains no listening at all. Each character plays, you press its key, and a miss replays the same character until you have it: the pattern shows after the miss, as feedback, never as the prompt.

Characters you miss come up more often on their own, the colored letter row shows where every character stands, and ninety percent in a session unlocks the next character. Past the fortieth, the ladder continues as speed: a **Squeeze spacing** button raises the Farnsworth rate one word per minute at a time, shrinking the thinking gap between letters until copy runs at full character speed.

Groups is the bridge from characters to real copying: groups of two from your learned set, growing to three and five as your accuracy holds.

QSO plays a whole exchange line by line, ragchew or contest style, with random callsigns and reports each time: copy the line, check, see the truth, next. This is the ritual you will hear on the air, so by the time your first real contact comes you have already copied a hundred of them. **Text** is the endurance half: short passages sentence by sentence, scored by words.

The send trainer reuses the material: switch it to phrases and it asks for real lines with your own callsign, CQ CQ DE you K, a report, a QTH, a 73, graded by the decoder like everything else.

Send is the half no practice oscillator can offer. With break-in off the paddle keys only the sidetone: nothing transmits. The trainer names a target, you send it, and the same decoder that reads the band prints what your fist actually said. The sidetone has to reach the decoder, and the route depends on how your radio is connected:

- **USB radio:** the setup row names anything in the way and offers one-tap fixes: switch to CW, break-in off, monitor on, and monitor level up, because a monitor at the factory's zero reads as ready and hears nothing.
- **Network radio:** the LAN audio stream cannot carry the sidetone; Icom offers no such routing. Plug in the USB cable as well, set the radio's USB AF Beep/Speech Output to ON (SET, Connectors), and choose the radio's USB Audio CODEC as the **CW practice input** in Settings. The network connection keeps running the station; the second cable exists only so the decoder can hear your paddle.

Left-handed? Set Paddle Polarity to REVERSE in the radio's keyer menu before your muscle memory forms, not after.

The decoder

Morse heard on the receive audio appears as text in the panel. The decoder finds the tone itself, follows the sender's speed itself, and stays quiet on an empty band rather than inventing letters from noise. Strong, cleanly keyed signals read well; weak and fading ones degrade the way they do for every decoder, including the radio's own. Treat it as a crutch while your ears learn, and a safety net for a missed callsign.

The passband scope

Above the keyer, a second waterfall shows the few kilohertz inside your filter, fine enough to separate signals 100 Hz apart:

- **Tap a trace** and the radio retunes so that signal lands on your own sidetone pitch, read from the radio's CW pitch setting, so a 500 Hz operator and a 700 Hz operator each land where their ears expect.

- **The teal line** marks the trace the decoder is currently reading. If the decoded text looks like nonsense, the usual reason is visible: it is locked on the other signal. Tap the one you want.

Who heard your CQ

With Reverse Beacon Network spotting on, the panel beside the keyer shows who decoded your CQ around the world within seconds, with signal reports and the speed they measured. It lists CW reports only; the digital modes' reception reports live with the log. Twelve spots and no answers means the band is fine and everyone is lazy, which is useful to know.

The pileup simulator

The trainer's **pileup** tab is the contest drill: call CQ and up to six stations answer at once, each at its own pitch, speed and strength, some fading with QSB. Copy one call out of the tangle, type it, work them, and the next wave answers. Clean streaks deepen the pileup; misses thin it. In the app, callers are drawn from stations genuinely on the air when the DX cluster watch is running. And the trainer now works offline: visit it once and it loads with no connection, which is what long flights are for.

FT8 and digital modes

ShackNerd has FT8 built in: decoding, transmitting, the whole contact and the log entry, in the browser, with no other software. WSJT-X still plugs in through the rigctld bridge for anyone who wants its deeper decoder or the other modes, and both paths are described here.

Native FT8: one tap per contact

One-time setup: put your callsign and your grid square in **Settings**, and make sure transmit and transmit audio are enabled.

One radio setting matters and it is easy to get crossed: MENU → SET → Connectors → MOD Input → DATA MOD → LAN. Native FT8 sends its audio over the network, and when data mode is on the radio takes modulation from the DATA MOD input. If that input is set to USB, for WSJT-X, your FT8 transmissions key the radio and carry nothing: power on the meter, silence on the air, and no spots anywhere. Flip it to LAN for native FT8, back to USB when you use WSJT-X over the USB cable.

During your first transmission, watch the meter strip in the panel and drag the **TX drive** slider down until the ALC just stops moving. An overdriven digital signal splatters across the band and decodes worse, not louder; this is the same ritual as WSJT-X's power slider, the setting is remembered, and the ALC, COMP, SWR and drain current meters sit right beside the slider so the whole adjustment happens in one place.

Turning on **PSK Reporter** in Settings uploads what the monitor hears to pskreporter.info, which puts this station on the map as a receiver and is also the fastest way to verify your own transmissions: call CQ and check pskreporter.info for your callsign a few minutes later.

1. Tune to an FT8 frequency, or let the **tune to...** band menu do it: it jumps the radio to the customary frequency for the chosen protocol on any band, in data mode, ready to go. For FT4, flip the protocol toggle; the slots halve to 7.5 seconds and everything else works identically.
2. Open **FT8** and press **Monitor FT8**. The band appears fifteen seconds at a time: the strip shows this slot's stations laid out by frequency, the list keeps the history, and stations you have worked on this band and mode are dimmed, so an SSB ragchew from last year never hides a fresh FT8 band slot. An **Alerts** toggle in the Band activity header sounds a tone and flashes the browser tab when a decode is addressed to you, and **Clear** empties both panes by hand; a change of frequency clears them on its own, because every decode describes the frequency it was heard on.
3. **Tap a CQ**, in the strip or the list. That is the whole contact: the exchange runs itself, grid, reports, RR73, and lands in the log with both reports and their grid. The status bar narrates each step and **Halt** stops everything instantly. Tap a station you already worked on this band and mode and ShackNerd says so instead of running a duplicate; a second tap on the same decode works them anyway, for the sked that is the point. A station that stops answering is abandoned after five calls with a note that says who was given up on; the note clears itself, and Halt dismisses it early.
4. Or press **Call CQ** and work whoever answers, the same way.

The slot clock above the list fills as each cycle runs, and once a contact is active it turns amber during the slots that are yours to transmit in. Your own transmissions appear in the list marked TX.

The operating surface

The decode area is two panes, the layout every FT8 operator already knows: **Band activity** on the left with everything the band said, and **RX frequency** on the right holding only your own exchange. Every row carries a place: the country from the callsign prefix for DX, the licensed state from the FCC database for US stations, or the distance in miles when a grid was announced.

Above the meters sits a waterfall of the whole FT8 window, each decoded station labeled at its audio offset and your own transmit lane drawn in red. **Tap an empty stretch to transmit there**; the **TX lane** field does the same by number, and auto answers stations on their own frequency the way WSJT-X does without hold.

The decoder itself runs the same deep techniques WSJT-X uses, fine synchronisation, ordered-statistics decoding and multi-pass subtraction, so a few extra weak decodes trickle in a few seconds after each slot's first batch. That trickle is normal and welcome.

Controls worth knowing

- **TX armed / TX held** sits beside Halt during an exchange. Held lets the transmission on the air finish and starts no more, with the exchange frozen where it stood; re-arm and it resumes. **Halt** stops immediately and abandons the contact.
- The **auto | even | odd** control picks your transmit slots for a CQ, in WSJT-X's TX even/1st sense, so coordinating with another operator is one word.
- The small field beside **Call CQ** takes a modifier: DX, POTA, NA, or any one to four letters, and the call goes out as CQ DX W2NRD FN43.
- The **free text box** transmits exactly what you type: thirteen characters of free text or any standard message. Mid-exchange it replaces the next automatic message once; from idle it repeats until halted.
- **RF power** and the radio's **monitor** live in the panel too, so the whole power ritual, coarse power, fine drive, ALC, happens on one screen.

Who heard you

With PSK Reporter enabled, ShackNerd also reads the map back: reception reports of your own signal appear in the **Log** workspace beside your contacts, receiver, grid, frequency and SNR, a few minutes behind the air. The clock check is automatic as well: everyone you decode should arrive about six tenths of a second into the slot, and if the whole band is consistently early or late the panel says your computer clock is off and by how much.

Two practical notes. The machine running ShackNerd needs an accurate clock, which is the same requirement WSJT-X has; if the DT column of received stations drifts past a second, turn on time synchronisation. And the radio is switched to USB with the data input automatically when a contact

starts.

The ShackNerd map

ShackNerd Reporter in Settings shares what your station decodes with the network's own map, the same spirit as PSK Reporter but drawn from ShackNerd stations only: shacknerd.com/map.html (<https://shacknerd.com/map.html>) shows every path heard in the last fifteen minutes, hover any path for the details, tap a band to filter, or type a callsign to see only who hears it. With the Reporter on, the digital screen also answers the question directly: **Heard by N ShackNerd stations** appears under your QSO panel while you transmit. A few hundred bytes a minute, your callsign and grid as the identity, and off by default like every outbound connection.

WSJT-X through the bridge

WSJT-X, JS8Call, fldigi and most loggers already know how to talk to a rigctl. ShackNerd serves that protocol, so they drive the radio through it with no extra cables and no virtual serial ports. WSJT-X's decoder also digs a little deeper than the built-in one on a truly marginal band, though the gap has narrowed: the built-in decoder now runs the same deep techniques, fine synchronization, ordered-statistics decoding and multi-pass subtraction of decoded signals.

In ShackNerd: open **Access and remote**, expand **Advanced**, set the **Digital mode software (rigctl)** port to 4532, and apply.

On the radio, for the usual USB audio path: **MENU** → **SET** → **Connectors** → **MOD Input** → **DATA MOD = USB**, and connect the radio's USB cable to the computer running WSJT-X.

File → Settings → **Radio** in WSJT-X:

Setting	Value
Rig	Hamlib NET rigctl
Network Server	127.0.0.1:4532
PTT Method	CAT
Mode	Data/Pkt
Split Operation	None

Audio tab: input and output are the radio's **USB Audio CODEC**.

PTT from WSJT-X goes through the same transmit lease and watchdog as every other way of keying, so a crashed program cannot leave the radio transmitting. The bridge listens on this machine only, because the protocol has no authentication and controls a transmitter.

The DATA toggle by hand

The **DATA** button beside the mode buttons switches the radio's data input directly, for anything not driven through the bridge.

Working a DXpedition (Fox/Hound)

Big DXpeditions run FT8 in Fox/Hound mode, and its etiquette differs from a normal contact. Turn on **Hound** next to the Halt button before you call: your first call rides above 1000 Hz where the fox listens, your rogered report automatically moves to the fox's own frequency, and when the fox sends RR73 the contact logs with no 73 from you, because a fox running a pileup needs the slot more than the goodbye. Turn Hound off again for ordinary contacts.

Logging

The log captures what the daemon already knows, so recording a contact is typing a callsign.

Logging a contact

A quick-log bar sits in the **Operate** and **CW** workspaces, so logging never means switching tabs: type the callsign, press enter, done. Frequency, band, mode and UTC time are taken from the radio at that moment, which is why it pays to log while you are still on the frequency. The full entry row with a name field lives in **Log**.

Reports default by mode: 599 for CW and RTTY, 59 for voice. A real report typed over them wins.

In the CW workspace, callsigns in the decoded text are tappable: one tap drops the call into the log bar. If a call is already in the log, the bar says so as you type it, including whether this band would be a new one.

The list

Contacts sit under the entry row with UTC time, call, frequency, mode and reports, fifty to a page, most recent first, with a search box that filters by callsign, name or QTH as you type and band and mode menus beside it. **edit** on any row opens the contact for correction, every field including the frequency, mode and time, which are exactly the ones most likely to be wrong when a contact was logged after the band change. Delete asks first; there is still no undo.

What the log tells the bands

Every FT8 and FT4 decode is checked against the log. Stations you have worked are dimmed, so a screen full of CQs reads as "who is new". A station that would be a first, a country never worked, a US state never worked, or either on a band where you have not worked it yet, gets a NEW DX, NEW STATE or "new on 20m" tag beside its state or country. The last one means you have worked that place before, just never on this band: a fresh band slot toward the per-band awards.

The callbook knows who you are working

With your QRZ account entered under **Logbook services** in Settings, typing a callsign in the quick-log bar shows the operator's name, state and grid while they are still on frequency, and the name lands in the log with the contact. A QRZ XML subscription unlocks the full records; without one the lookups are thinner but still confirm the call.

Confirmations, without the ritual

Enter your Logbook of The World username and password in the same section and ShackNerd pulls your confirmations down once a day, marking each confirmed contact with a check in the LoTW column. **Sync LoTW** in the Log workspace does it on demand. Uploading goes the other way through TQSL, the ARRL's signing program, installed on the shack machine with your callsign certificate.

Enter the station location name exactly as TQSL shows it under **Logbook services** and every new contact is signed and uploaded by itself within a minute of being logged; the arrow in the QSL column marks the ones that have gone. **Upload to LoTW** sends anything waiting right now. Imported contacts are not re-uploaded; LoTW already has them.

Every contact, everywhere

With a QRZ Logbook API key (a separate credential from the login above: Logbook page, settings gear, Logbook API) or an eQSL account entered, every new contact is pushed to those services by itself within a minute of being logged. The QSL column in the log shows each service's verdict per contact: dim while waiting, bright when delivered, red with the reason on hover when a push failed. Imported contacts are deliberately not pushed; the services already have them.

Your whole history, in one place

Pull history from LoTW and QRZ in the Log workspace fetches every contact those services hold, deduplicates them against each other and against this log, and merges: contacts new to ShackNerd are added with the marks the services earned, contacts already here are enriched from QRZ's richer records (a missing name or report, blanks only, never overwritten) and lit as confirmed where LoTW says so. Anything one service has and the other lacks flows out through the ordinary pushes. The same reconcile runs nightly afterwards, so the three logs never drift, and the Backup file in Settings becomes a complete archive of every contact you have ever made.

The scoreboard

Statistics, beside the log, keeps score: contacts, LoTW confirmations, DX countries, grid squares, best distance, and the Worked All States card, where filled means worked and bright means LoTW-confirmed. The band buttons above the card switch it to a single band, which is the per-band WAS chase in one glance, and the band table counts DX countries per band, the Challenge-style ladder. Below it, the DXCC list names every country worked, confirmed ones bright. The empty cells are what the NEW STATE and new DX tags on the digital screen are pointing at.

Bringing a log in

Import ADIF in the Log workspace merges an existing log file in. Duplicates, the same call, band and mode within the same half hour, are skipped, so importing the same file twice is harmless and so is importing a file that overlaps what ShackNerd already logged. Confirmations and states travel with the file: twenty years of confirmed contacts arrive confirmed, not amnesiac. Imported contacts are normally marked as already uploaded so the QSL services are not spammed with duplicates; tick **never uploaded** for a paper log typed in, and they push like new contacts.

Backup

Settings has a Backup section: one JSON file holds the connection settings, the whole log, presets, macros and band memory. Restore merges rather than replaces, using the same duplicate rules as

the ADIF import. The CW trainer's progress lives in each browser and is not included.

Getting your log out

Export ADIF downloads the whole log as a standard ADIF file, every field the log holds, which imports into Logbook of The World via TQSL, QRZ Logbook, and every logging program. Bands are derived from frequency and modes are mapped to ADIF's vocabulary, so LSB and USB export as SSB the way importers expect. **Export Cabrillo** beside it writes the contest rows as a submittable Cabrillo file.

ShackNerd deliberately does not try to be logging software. It captures the moment and hands it to whatever you already use.

Reverse Beacon Network spotting

The Reverse Beacon Network is a worldwide set of receivers that decode CW around the clock and publish every station they hear. ShackNerd can watch that feed for your callsign.

Turning it on

In **Settings**, make sure your callsign is set, then turn on **Spotting**. It is off by default because it makes an outbound connection to the network's public feed, and ShackNerd does not talk to third parties without being told to.

Using it

Call CQ on CW. Within seconds, **Where you were heard** fills with the receivers that decoded you: their identity, the exact frequency they heard you on, your signal to noise ratio at their antenna, and your sending speed as they measured it.

What it tells you:

- **Spots from three continents** and no answers: the band is open, your signal is fine, keep calling.
- **No spots at all** on an active band: check the antenna, the tuner and the power before blaming propagation.
- **Strong reports in one direction only**: that is your antenna's pattern, drawn for you by the ionosphere.

The panel beside the keyer keeps the thirty most recent CW spots. Spots are of your transmitted signal, so this only works when you actually transmit, and only on CW, since that is what the network's receivers decode.

Remote operation

Operating from anywhere, without ever exposing the radio to the internet.

First: a real address at home

Before any remote-access setup, one button fixes the daily annoyance of certificates on your own network. In Settings under **A name for your own network**, **Get my shacknerd.com address** gives this daemon a permanent address built from your callsign, like `w2nrd.s.shacknerd.com`, claimed automatically from the callsign in your settings, first come first served. It resolves to this machine's LAN address and comes with a real Let's Encrypt certificate that renews itself. Bookmark it on every phone and laptop in the house: green padlock, working microphone, nothing to install anywhere, and no warnings ever. (If your callsign is already claimed, you still get a permanent hash-based name; the callsign is the convenience, the hash is the guarantee.)

Your radio traffic never leaves your network; only the name and this machine's private addresses are published, the same trade Plex users have accepted for a decade. Like any name, it needs DNS: with your internet down, the plain `http://<this machine's IP>:7300` shown in the enrollment note keeps working, because the radio itself never depends on the internet. Needs an access password set first, like everything reachable by name. The security page explains why ShackNerd refuses the port-forwarding shortcut; this page is the path that works.

The short version

Your radio talks only to ShackNerd on your home network. You reach ShackNerd over an encrypted tunnel, protected by a real password and HTTPS. Tailscale provides the tunnel free for personal use, with no router configuration at all.

Step by step

1. **Set an access password.** In **Settings** → **Access password**, set one. Every remote feature refuses to work without it, on purpose. Once set, every device signs in, and repeated wrong guesses lock the login out for a few minutes.
2. **Install Tailscale** on the ShackNerd machine and on your phone, from tailscale.com (<https://tailscale.com>), signed into the same account. In the Tailscale admin console under DNS, enable **HTTPS certificates**.
3. **One click in ShackNerd.** Open **Access and remote**. It shows what Tailscale is doing on the machine and, once it is running, a **Set up remote access** button. That button obtains a real HTTPS certificate for the machine's tailnet name, stores it, and switches ShackNerd over to it without a restart.
4. **Operate.** From anywhere your phone has signal, open the address the panel shows, something like `https://shack-pc.your-tailnet.ts.net:7300`, sign in, and everything works: waterfall, audio, and because it is HTTPS, your microphone for transmit.

Why HTTPS matters here

Browsers only allow microphone access on secure pages, so voice transmit from your phone requires it. The Tailscale certificate is a real one, so there are no browser warnings to click through.

Certificates of your own

Under **Advanced** in the same panel you can point ShackNerd at any certificate and key files, choose the listen address, and set the hostname it answers to. Requests arriving under any other name are refused, which is part of what keeps a random website from reaching your daemon through your browser.

If a saved setting locks you out

Command line flags override saved settings for one run: `shacknerd -listen :7300` starts with the default listener no matter what was saved, so there is always a way back in from the machine itself.

Share your station, view-only

Share your station in Settings makes a link anyone can open: they see the live waterfall, hear the audio and read the decodes, and can touch nothing. The restriction is enforced by the daemon, not the page, so the link is safe to hand out: an elmer watching your first contact, your club during a special event, or a stream overlay. **New link** revokes the old one everywhere at once; so does turning sharing off.

Sharing beyond your LAN

The link decides what a viewer can do; your network decides who can reach it at all. In order of effort:

Same network: nothing to do. Anyone in the house or at the club opens the link and it works. This covers most club and family cases.

A friend elsewhere, with Tailscale: share the machine. Tailscale can share a single machine with someone else's Tailscale account, free on personal plans:

1. In the Tailscale admin console (<https://login.tailscale.com/admin/machines>), find the ShackNerd machine, open its menu, choose **Share**, and send the invite to your friend's email.
2. They install Tailscale, sign in with any account, and accept.
3. Send them the spectate link with your tailnet address in it, the same `https://shack-pc.your-tailnet.ts.net:7300/spectate?key=...` form the panel shows you.

They can now watch from anywhere. The share is to the machine, not your whole tailnet, and you can revoke it in the same console.

Anyone at all, no software: Tailscale Funnel. Funnel publishes the daemon to the public internet over HTTPS at your machine's tailnet name, so a viewer needs nothing but the link:

1. Enable Funnel for your tailnet when asked (the command walks you through it), then on the ShackNerd machine: `tailscale funnel 7300`

2. Your station is now reachable at the address Funnel prints. Send the spectate link built on that address.

3. `tailscale funnel off` ends it.

Being publicly reachable is exactly what the access password and the view-only link are for: a stranger with the spectate link gets the read-only station, and everything else asks for your password. Funnel's bandwidth suits a handful of viewers, not a crowd. Turn it off when the show is over; standing exposure you are not using is never worth keeping.

Port forwarding: still no. The security page explains why, and nothing about sharing changes it.

Streaming to Twitch or YouTube does not need any of this: OBS runs on your own machine, captures the spectate page from your LAN, and your stream platform does the distribution. The spectate page is designed to be a clean capture source, no controls, no settings, just the station.

Radio reports: help ShackNerd support your radio

ShackNerd is fully tested against exactly one radio, the IC-7300 MK2 its author owns. Support for every other model gets built from radio reports sent in by people who own them.

What a report is

A read-only snapshot of how your radio describes itself:

- What the radio said during the network handshake, including its name, its CI-V address and its audio capabilities
- The scope geometry actually observed, which is what a display profile needs
- A sweep of read-only queries and the radio's raw answers, including the refusals, which are just as informative. Icoms are swept in CI-V; Kenwood and Yaesu radios are swept in their own CAT dialect, so a report from any of the three makes is equally useful

It changes nothing on the radio and never keys the transmitter. Every probe is a read. It contains no passwords and no network addresses.

Making one

In **Settings**, under **Help and support**, press **Download a radio report**. It takes about half a minute and saves a small JSON file. Email that file to hello@shacknerd.com with a line about which radio it is.

If ShackNerd cannot even connect to your radio

Press the button anyway. The report degrades instead of failing: it records whatever the radio did say during the attempt and exactly where things stopped, and that partial report is the most valuable kind, because the radios that fail to connect are the ones with the most to teach.

The command line equivalent is `shacknerd -radio-report report.json`.

Troubleshooting

The things that actually go wrong, in roughly the order people hit them.

It will not start, or the browser cannot reach it

Something else is using port 7300. ShackNerd notices and quietly takes the next free port; the window and the browser it opens show the address it actually got. (Frig, Hyper-V's reserved ranges and friends are the usual squatters.) Only if you need a specific port does `shacknerd -listen :7301` matter.

You started ShackNerd twice. The newer launch politely replaces the older one within a couple of seconds. That is on purpose: the usual reason for a second launch is a fresh download, and the newest build should win.

The tray icon is there but you closed the browser tab. Right-click the icon and choose **Open ShackNerd**. Closing tabs never stops the daemon; quitting from the tray does, and disconnects from the radio cleanly.

The frequency stays 0.000.00 on a USB radio

The port opened but the radio is not answering CI-V. ShackNerd probes the common CI-V addresses on every serial connect and adopts whichever answers, so a wrong address heals itself; if the display still stays dark, the usual causes are the CI-V baud rate in the radio's menu not matching the one chosen here, or a radio whose USB is a hub showing two COM ports where only one carries CI-V; try the other. "Serial port busy" in the log means another program holds the port.

It cannot connect to the radio

Work down this list; it is ordered by likelihood:

1. **Network Control is off on the radio.** MENU → SET → Network → Network Control → ON.
2. **Wrong ID or password.** These are the radio's Network User1 credentials, not anything else. Watch for a trailing space when typing on a phone.
3. **Another program holds the radio.** The radio serves one client. Close RS-BA1, wfview or another ShackNerd, or power-cycle the radio to clear a stuck session.
4. **Wrong network.** The daemon machine and radio must reach each other. `shacknerd -lan-scan auto` searches your networks for radios.
5. **A VPN is grabbing the traffic.** ShackNerd binds to the interface that is actually on the radio's subnet, but exotic VPN setups can still win. If the log shows the connection using a VPN address, that is the problem.

Audio

No receive audio on the phone until you tap. That is the browser, not ShackNerd: pages may not make sound without a gesture. Tap start once.

Audio is delayed or drops. Try 16 kHz in Settings; it is a third of the bandwidth of 48 kHz. Buffering trades delay for smoothness and is set beside the audio controls.

Transmit

The key says ON AIR radio mic, not phone mic. Two causes, and the text under the key names which: the page is not on HTTPS, so the browser will not grant microphone access, or Transmit audio is off in Settings.

Keyed, mic level moving, but no power out. The radio is not using the network as its modulation source. MENU → SET → Connectors → MOD Input → DATA OFF MOD → LAN, and check LAN MOD Level is not at zero.

CW messages do nothing. The radio must be in CW mode with break-in on; the panel checks both and offers the fix.

A logbook service stopped getting your contacts

Settings keeps an **Integrations** strip: the last word from every connected service, LoTW, QRZ, eQSL, PSK Reporter and the ShackNerd Reporter, green when the most recent attempt worked and red with the error under your pointer when it did not. A wrong password or an expired API key shows up there the minute it starts failing, not months later as a hole in your log.

Staying current

ShackNerd checks for a new build once a day, and **Check for updates** under Settings, Help and support asks right now. When a newer build exists, a banner appears in the interface with an **Update now** button. One click: the daemon downloads the release itself, verifies it against the published checksum, swaps itself out and restarts; the page reconnects to the new build a few seconds later, still signed in. The download link stays beside it for anyone who prefers their own hands, and the version and date are in every file name either way.

The interface says connected but nothing updates

A network that silently died. The page notices within ten seconds and reconnects on its own; if you are impatient, reload. If it recurs on WiFi, the radio end is losing packets and a wired connection to the radio fixes more than anything else.

Still stuck

hello@shacknerd.com, and attach the file that **Download diagnostics** in Settings produces: one zip with the version, the daemon's view of the radio, a redacted copy of the settings and the log tail, which turns most email exchanges into one. The raw log is shacknerd.log beside the settings database (the previous run is kept as shacknerd.log.1), or on screen if you ran it from a terminal.